TOSHIBA

8 Bit Microcontroller
TLCS-870/C Series

TMP86FP24

TOSHIBA CORPORATION

Revision History

Date	Revision	
2007/8/24	1	First Release
2008/8/29	2	Contents Revised

Caution in Setting the UART Noise Rejection Time

When UART is used, settings of RXDNC are limited depending on the transfer clock specified by BRG. The combination "O" is available but please do not select the combination "–".

The transfer clock generated by timer/counter interrupt is calculated by the following equation :

Transfer clock [Hz] = Timer/counter source clock [Hz] ÷ TTREG set value

BRG setting	Transfer clock [Hz]	RXDNC setting			
		00 (No noise rejection)	01 (Reject pulses shorter than 31/fc[s] as noise)	10 (Reject pulses shorter than 63/fc[s] as noise)	11 (Reject pulses shorter than 127/fc[s] as noise)
000	fc/13	O	O	O	–
110 (When the transfer clock generated by timer/counter interrupt is the same as the right side column)	fc/8	O	–	–	–
	fc/16	O	O	–	–
	fc/32	O	O	O	–
The setting except the above		O	O	O	O

CMOS 8-Bit Microcontroller

TMP86FP24FG

The TMP86FP24 is the high-speed, high-performance and low-power consumption 8-bit microcomputer, including ROM, RAM, LCD driver, multi-function timer/counter, serial interface (UART, HSIO), a 10-bit AD converter and two clock generators on chip. The TMP86FP24 has a 2 K bytes BOOT ROM (masked ROM) for programming to flash memory.

Product No.	Flash Memory	BOOT ROM	RAM	Package	Emulation Chip
TMP86FP24FG	48 K × 8 bits	2 K × 8 bits	2 K × 8 bits	LQFP80-P-1212-0.50A	TMP86C948XB

Feautures

♦ 8-bit single chip microcomputer TLCS-870/C series

♦ Instruction execution time: 0.25 µs (at 16 MHz)
  122 µs (at 32.768 kHz)

♦ 132 types and 731 basic instructions

♦ 19 interrupt sources (External: 5, Internal: 14)

♦ Input/output ports (54 pins)
  (Out of which 16 pins are also used as SEG pins)

LQFP80-P-1212-0.50A

TMP86FP24FG

♦ 16-bit timer counter: 2 ch

  • Timer, event counter, pulse width measurement, external trigger timer, window, PPG output modes

♦ 8-bit timer counter: 2 ch

  • Timer, event counter, PWM output, programmable divider output, capture modes

♦ Time base timer

♦ Divider output function

♦ Watchdog timer

  • Interrupt source/internal reset generate (Programmable)

- ♦ Serial interface
  - • UART/SIO: 1ch
  - • SIO: 1ch

- ♦ ROM corrective function
  - • Four register bank
  - • 1 byte or 2 bytes replace mode
  - • Address replace mode

- ♦ 10-bit successive approximation type AD converter
  - • Analog input: 8 ch

- ♦ Five key-on wakeup pins

- ♦ LCD driver/controller
  - • Built-in voltage booster for LCD driver
  - • With display memory (12 bytes)
  - • LCD direct drive capability (Max 24 seg × 4 com)
  - • 1/4, 1/3, 1/2 duties or static drive are programmably selectable

- ♦ Dual clock operation
  - • Single/dual clock mode

- ♦ Nine power saving operating modes
  - • STOP mode: Oscillation stops. Battery/capacitor backup.
                        Port output hold/high impedance.
  - • SLOW1, 2 mode: Low power consumption operation using low-frequency clock (32.768 kHz).
  - • IDLE0 mode: CPU stops, and peripherals operate using high-frequency clock of time-base-timer. Release by falling edge of TBTCR<TBTCK> setting.
  - • 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.
  - • SLEEP0 mode: CPU stops, and peripherals operate using low-frequency clock of time-base-timer. Release by falling edge of TBTCR<TBTCK> setting.
  - • SLEEP1 mode: CPU stops, and peripherals operate using low-frequency clock.
                        Release by interrupts.
  - • SLEEP2 mode: CPU stops, and peripherals operate using high and low frequency clock.
                        Release by interrupts.

- ♦ Wide operating voltage: 1.8 to 3.6V at 8 MHz/32.768 kHz
                            2.7 to 3.6V at 16 MHz/32.768 kHz

Pin Assignments (Top view)

LQFP80-P-1212-0.50A

Top pins (left to right): COM0, COM1, COM2, COM3, SEG0, SEG1, SEG2, SEG3, SEG4, SEG5, SEG6, SEG7, P97 (SEG8), P96 (SEG9), P95 (SEG10), P94 (SEG11), P93 (SEG12), P92 (SEG13), P91 (SEG14), P90 (SEG15)

Top pin numbers: 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41

Left side pins:
- V3 — 61
- V2 — 62
- V1 — 63
- C1 — 64
- C0 — 65
- $\overline{\text{WAKE}}$ — 66
- (SO2) P10 — 67
- (SI2) P11 — 68
- ($\overline{\text{SCK2}}$) P12 — 69
- ($\overline{\text{PWM5}}$ / PDO5 /TC5) P13 — 70
- (INT3/TC3) P14 — 71
- (TC1) P15 — 72
- P30 — 73
- P31 — 74
- P32 — 75
- P33 — 76
- P34 — 77
- P35 — 78
- P36 — 79
- P37 — 80

Right side pins:
- 40 — P47 (SEG16)
- 39 — P46 (SEG17)
- 38 — P45 (SEG18)
- 37 — P44 (SEG19)
- 36 — P43 (SEG20)
- 35 — P42 (SEG21)
- 34 — P41 (SEG22)
- 33 — P40 (SEG23/STOP4)
- 32 — P53
- 31 — P52
- 30 — P51 ($\overline{\text{DVO}}$)
- 29 — P50 ($\overline{\text{PPG}}$)
- 28 — P67 (AIN7/STOP3)
- 27 — P66 (AIN6/STOP2)
- 26 — P65 (AIN5/STOP1)
- 25 — P64 (AIN4/STOP0)
- 24 — P63 (AIN3)
- 23 — P62 (AIN2)
- 22 — P61 (AIN1)
- 21 — P60 (AIN0)

Bottom pin numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Bottom pins (left to right): VSS, XIN, XOUT, TEST, VDD, (XTIN) P21, (XTOUT) P22, $\overline{\text{RESET}}$, ($\overline{\text{STOP}}$/INT5) P20, (BOOT) P23, ($\overline{\text{INT0}}$) P00, (INT1) P01, (INT2) P02, (TC2) P03, P04, (RXD/SI1) P05, (TXD/SO1) P06, ($\overline{\text{SCK1}}$) P07, AVDD, VAREF

Block Diagram

Pin Funtions

The TMP86FP24 has MCU mode and serial PROM mode.

(1) MCU mode

Make sure to fix the TEST pin to low level.

(2) Serial PROM mode

In the serial PROM mode, programming to flash memory is available by executing BOOT ROM. For details, refer to 2.16 "Serial PROM Mode".

Pin Functions (1/2)

Pin Name	Input/Output	Functions	
P07 ($\overline{SCK1}$)	I/O (I/O)	8-bit input/output port with latch. When used as a serial interface output or UART output, respective output latch (P0DR) should be set to "1". When used as an input port, an serial interface input, UART input, timer counter input or an external interrupt input, respective output control (P0OUTCR) should be cleared to "0" after setting P0DR to "1".	Serial clock input/output 1
P06 (TXD, SO1)	I/O (Output)		UART data output, serial data output 1
P05 (RXD, SI1)	I/O (Input)		UART data input, serial data input 1
P04	I/O		
P03 (TC2)	I/O (Input)		Timer counter 2 input
P02 (INT2)	I/O (Input)		External interrupt 2 input
P01 (INT1)	I/O (Input)		External interrupt 1 input
P00 ($\overline{INT0}$)	I/O (Input)		External interrupt 0 input
P15 (TC1)	I/O (Input)	6-bit input/output port with latch. When used as a timer/counter output or serial interface output, respective output latch (P1DR) should be set to "1". When used as an input port, a timer counter input, an external interrupt input or serial interface input, respective output control (P1OUTCR) should be cleared to "0" after setting P1DR to "1".	Timer counter 1 input
P14 (TC3, INT3)	I/O (Input)		Timer counter 3 input, External interrupt 3 input
P13 ($\overline{PWM5}$, $\overline{PDO5}$, TC5)	I/O (I/O)		PWM5 output, PDO5 output, Timer/counter 5 input
P12 ($\overline{SCK2}$)	I/O (I/O)		Serial clock input/output 2
P11 (SI2)	I/O (Input)		Serial data input 2
P10 (SO2)	I/O (Output)		Serial data output 2
P23	I/O	4-bit input/output port with latch. When used as an input port or an external interrupt input, respective output control (P2OUTCR) should be cleared to "0" after setting output latch (P2DR) to "1".	
P22 (XTOUT)	I/O (Output)		Resonator connecting pins (32.768 kHz)
P21 (XTIN)	I/O (Input)		For inputting external clock, XTIN is used and XTOUT is opened.
P20 ($\overline{INT5}$, $\overline{STOP}$)	I/O (Input)		External interrupt input 5 or STOP mode release signal input
P37 to P30	I/O	8-bit input/output port with latch (N-ch high current output). When used as an input port, respective output control (P3OUTCR) should be cleared to "0" after setting output latch (P3DR) to "1".	
P47 (SEG16) to P41 (SEG22)	I/O (Output)	7-bit input/output port with latch. When used as an input port, respective output latch (P4DR) should be set to "1" after LCD output control (P4LCR) is cleared to "0".	LCD segment output
P40 (SEG23, STOP4)	I/O (I/O)	1-bit input/output port with latch. When used as an input port, the output latch (P4DR) should be set to "1" after the LCD output control (P4LCR) is cleared to "0". When used as a LCD output, the P4LCR should be set to "1" after the STOPCR<STOP4EN> should be cleared to "0". When used as a key-on wakeup input, the STOPCR<STOP4EN> should be set to "1".	LCD segment output STOP mode release input
P53	I/O	4-bit input/output port with latch (N-ch high current output). When used as an input port, respective output control (P5OUTCR) should be cleared to "0" after setting output latch (P5DR) to "1". When used as a PPG output or divider output, respective P5DR should be set to "1".	
P52	I/O		
P51 ($\overline{DVO}$)	I/O (Output)		Divider output
P50 ($\overline{PPG}$)	I/O (Output)		PPG output

Pin Functions (1/2)

Pin Name	Input/Output	Functions		
P67 (AIN7, STOP3)	I/O (Input)	8-bit programmable input/output port (Tri-state). Each bit of this port can be individually configured as an input or an output under software control. When used as an input port, respective input/output control (P6CR1) should be cleared to "0" after setting input control (P6CR2) to "1". When used as an analog input or key-on wakeup input, respective P6CR1 should be cleared to "0" after clearing P6CR2 to "0". When used as a key-on wakeup input, STOPCR<STOPiEN> should be set to "1". (i = 0 to 3)	STOP 3 input	AD converter analog inputs
P66 (AIN6, STOP2)	I/O (Input)		STOP 2 input	
P65 (AIN5, STOP1)	I/O (Input)		STOP 1 input	
P64 (AIN4, STOP0)	I/O (Input)		STOP 0 input	
P63 (AIN3)	I/O (Input)			
P62 (AIN2)	I/O (Input)			
P61 (AIN1)	I/O (Input)			
P60 (AIN0)	I/O (Input)			
P97 (SEG8) to P90 (SEG15)	I/O (Output)	8-bit input/output port with latch. When used as an input port, respective output latch (P9DR) should be set to "1" after LCD output control (P9LCR) is cleared to "0".	LCD segment output	
SEG7 to SEG0	Output	LCD segment outputs		
COM3 to COM0		LCD common outputs		
V3 to V1	LCD voltage booster pin	LCD voltage booster pin. Capacitors are required between C0 and C1 pin and V1/V2/V3 pin and GND.		
C1 to C0				
$\overline{\text{WAKE}}$	Output	STOP mode monitor output. During CPU operation (including IDLE0/1/2, SLEEP0/1/2, warm-up period), it becomes "L" level state. In RESET and STOP mode, it becomes the high-impedance state.		
XIN, XOUT	Input output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.		
$\overline{\text{RESET}}$	Input	Reset signal input		
TEST	Input	Test pin for out-going test. Be fixed to low.		
VDD, VSS	Power supply	Power supply for operation		
VAREF		Analog reference voltage for AD conversion		
AVDD		AD circuit power supply		

Operational Description

1. CPU Core Functions

The CPU core consists of a CPU, a system clock controller, and an interrupt controller.

This section provides a description of the CPU core, the program memory, the data memory, the external memory interface, and the reset circuit.

1.1 Memory Address Map

The TMP86FP24 memory consists of 5 blocks: FLASH memory, BOOT ROM, RAM, DBR (Data buffer register) and SFR (Special function register). They are all mapped in 64-Kbyte address space. Figure 1.1.1 shows the TMP86FP24 memory address map. The general-purpose registers are not assigned to the RAM address space.

ROM: FLASH memory includes:
Vector table
BOOT ROM: FLASH writing program
RAM: Random access memory includes:
Data memory
Stack
SFR: Special function register includes:
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Interrupt control registers
Program status word
DBR: Data buffer register includes:
Peripheral control registers
Peripheral status registers
LCD display memory

Figure 1.1.1 Memory Address Maps

1.2 Program Memory (ROM)

The TMP86FP24 has a 48 K × 8 bits (Address 4000H to FFFFH) of program memory (FLASH).

1.3   Data Memory (RAM)

The TMP86FP24 has 2048 bytes of internal RAM. The first 192 bytes (0040H to 00FFH) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

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

```
Example:   Clears RAM to "00H".
                LD      HL, 0040H            ;   Start address setup.
                LD      A, H                 ;   Initial value (00H) setup.
                LD      BC, 07FFH
SRAMCLR:        LD      (HL), A
                INC     HL
                DEC     BC
                JRS     F, SRAMCLR
```

1.4 System Clock Controller

The system clock controller consists of a clock generator, a timing generator, and a standby controller.
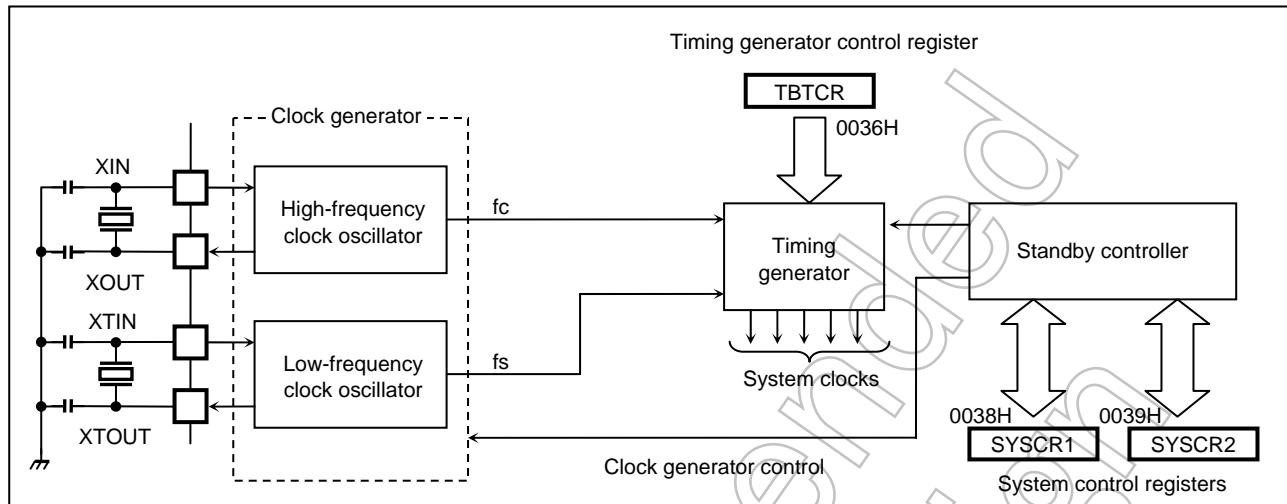
Figure 1.4.1 System Clock Control

1.4.1 Clock Generator

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

The high-frequency (fc) and low-frequency (fs) clocks can easily be 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 XIN/XTIN pin with XOUT/XTOUT pin not connected.
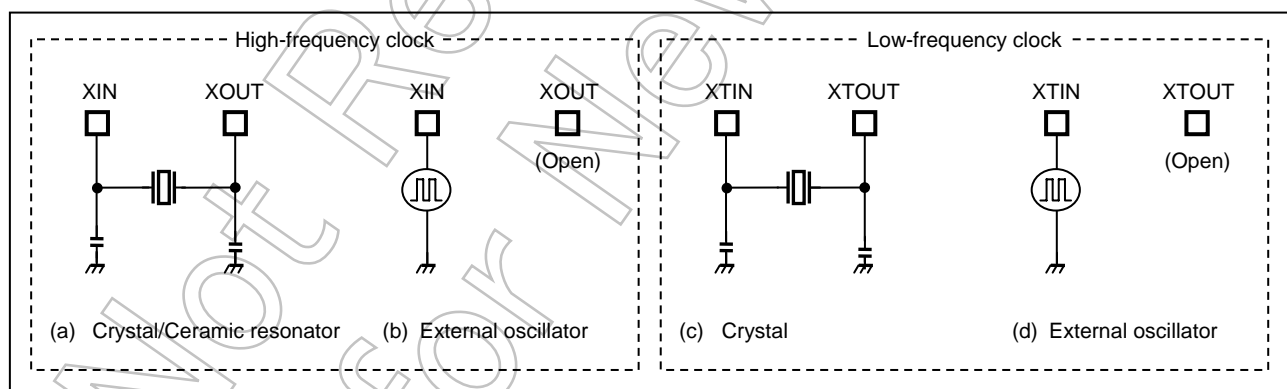
Figure 1.4.2 Examples of Resonator Connection

Note: The function to monitor the basic clock directly at external is not provided for hardware, however, with disabling all interrupts and watchdog timers, the oscillation frequency can be adjusted by monitoring the pulse which the fixed frequency is outputted to the port by the program.

The system to require the adjustment of the oscillation frequency should create the program for the adjustment in advance.

1.4.2   Timing Generator

The timing generator generates the various system clocks supplied to the CPU core and peripheral hardware from the basic clock (fc or fs). The timing generator provides the following functions.

a. Generation of main system clock

b. Generation of divider output ($\overline{\text{DVO}}$) pulses

c. Generation of source clocks for time base timer

d. Generation of source clocks for watchdog timer

e. Generation of internal source clocks for timer/counters and serial interface

f. Generation of warm-up clocks for releasing STOP mode

(1) Configuration of timing generator

The timing generator consists of a 2-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

An input clock to the 7th stage of the divider depends on the operating mode, TBTCR<DV7CK>, that is shown in Figure 1.4.4. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to "0".
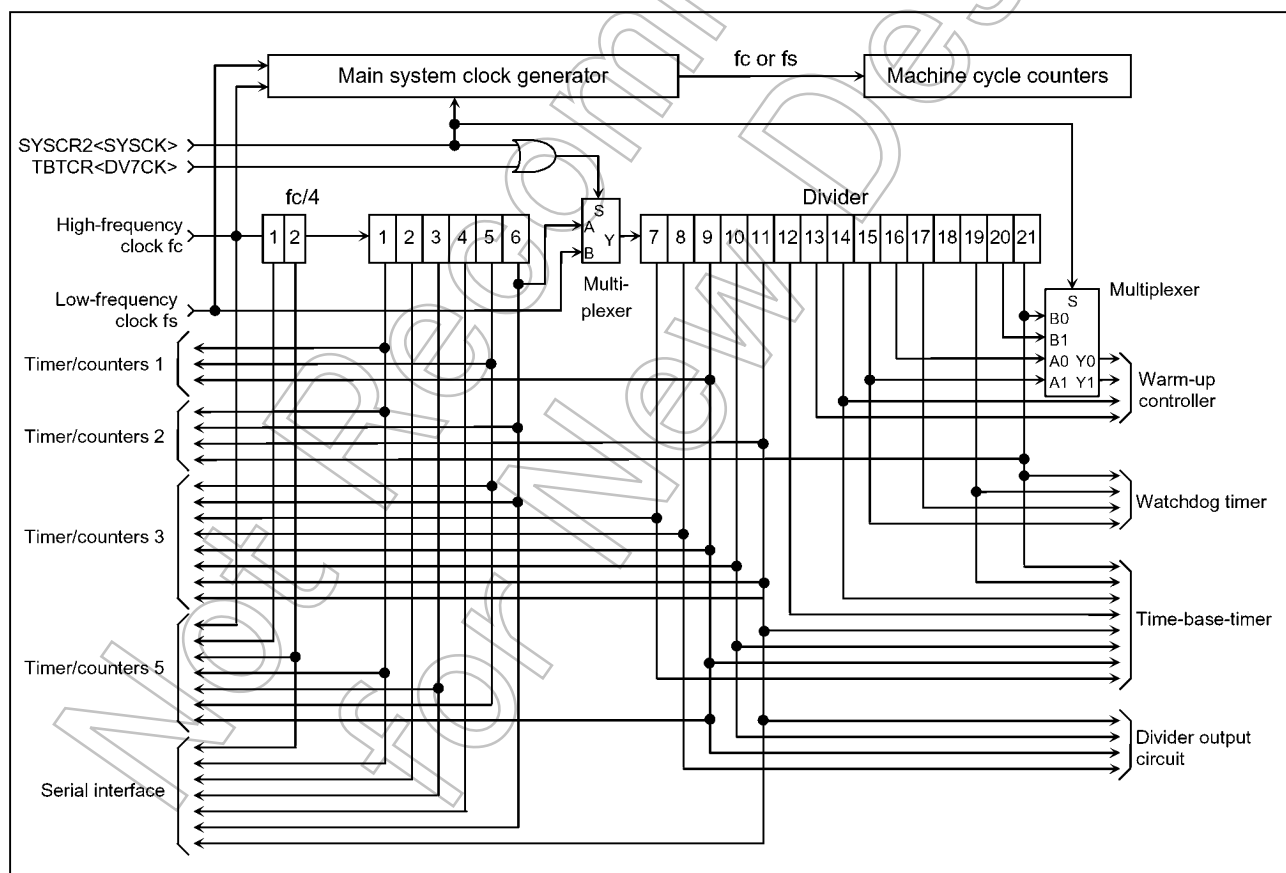
Figure 1.4.3 Configuration of Timing Generator

| TBTCR
(0036H) | 7
(DVOEN) | 6 | 5
(DVOCK) | 4
DV7CK | 3
(TBTEN) | 2 | 1
(TBTCK) | 0 | (Initial value: 0000 0000) |

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

Note 1: In single clock mode, do not set DV7CK to "1".

Note 2: Do not set "1" on DV7CK while the low-frequency clock is not operated stably.

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

Note 4: In SLOW1/2 and SLEEP1/2 modes, the DV7CK setting is ineffective, and fs is input to the 7th stage of the divider.

Note 5: When STOP mode is entered from NORMAL 1/2 mode, the DV7CK setting is ineffective during the warm-up period after release of STOP mode, and the 6th stage of the divider is input to the 7th stage during this period.

Figure 1.4.4 Timing Generator Control Register

(2) Machine cycle

Instruction execution and peripheral hardware operation are synchronized with the main system clock.

The minimum instruction execution unit is called an "machine cycle". There are a total of 10 different types of instructions for the TLCS-870/C series: Ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10-machine cycles for execution.

A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.
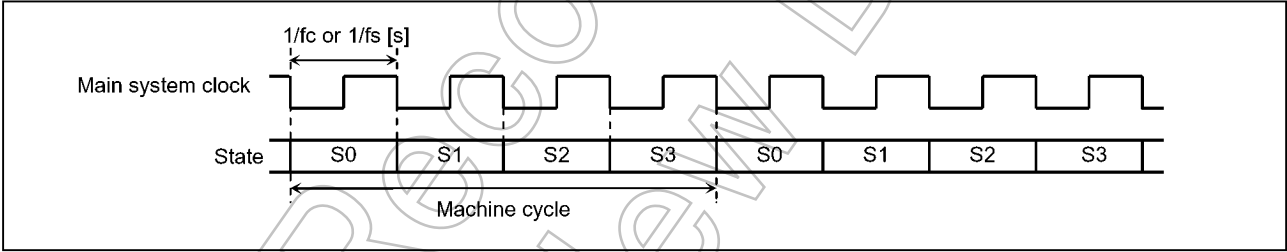
Figure 1.4.5 Machine Cycle

1.4.3 Operation Mode Control Circuit

The operation mode control circuit 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 and SYSCR2).

Figure 1.4.6 shows the operating mode transition diagram and Figure 1.4.7 shows the system control registers.

(1) Single-clock mode

Only the oscillation circuit for the high-frequency clock is used, and P21 (XTIN) and P22 (XTOUT) pins are used as input/output ports. The main-system clock is obtained from the high-frequency clock. In the single-clock mode, the machine cycle time is 4/fc [s].

a. NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock.

The TMP86FP24 is placed in this mode after reset.

b. 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 SYSCR2<IDLE>, and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When the IMF (Interrupt master enable flag) is "1" (Interrupt enable), the execution will resume with the acceptance of the interrupt, and the operation will return to normal after the interrupt service is completed. When the IMF is "0" (Interrupt disable), the execution will resume with the instruction which follows the IDLE1 mode start instruction.

c. IDLE0 mode

In this mode, all the circuit, except oscillator and the Time-base-timer, stops operation.

This mode is enabled by setting "1" on bit TGHALT on the system control register 2 (SYSCR2).

When IDLE0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

When returned from IDLE0 mode, the CPU restarts operating, entering NORMAL1 mode back again. IDLE0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = "1", EF7 (TBT interrupt individual enable flag) = "1", and TBTCR<TBTEN> = "1", interrupt processing is performed. When IDLE0 mode is entered while TBTCR<TBTEN> = "1", the INTTBT interrupt latch is set after returning to NORMAL1 mode.

(2) Dual-clock mode

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

The TLCS-870/C is placed in the single-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on at the start of a program.

a. NORMAL2 mode

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

b. SLOW2 mode

In this mode, the CPU core operates with the low-frequency clock, while both the high-frequency clock and the low-frequency clock are operated. On-chip peripherals are triggered by the low-frequency clock. As the SYSCK on SYSCR2 becomes "0", the hardware changes into NORMAL2 mode. As the XEN on SYSCR2 becomes "0", the hardware changes into SLOW1 mode. Do not clear XTEN to "0" during SLOW2 mode.

c. SLOW1 mode

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

Switching back and forth between SLOW1 and SLOW2 modes are performed by XEN bit on the system control register 2 (SYSCR2). In SLOW1 and SLEEP mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

d. IDLE2 mode

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

e. SLEEP1 mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted; however, on-chip peripherals remain active (Operate using the low-frequency clock). Starting and releasing of SLEEP mode are the same as for IDLE1 mode, except that operation returns to SLOW mode. In SLOW and SLEEP mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

f. SLEEP2 mode

The SLEEP2 mode is the IDLE mode corresponding to the SLOW2 mode. The status under the SLEEP2 mode is same as that under the SLEEP1 mode, except for the oscillation circuit of the high-frequency clock.

g. SLEEP0 mode

In this mode, all the circuit, except oscillator and the Time-base-timer, stops operation.

This mode is enabled by setting "1" on bit TGHALT on the system control register 2 (SYSCR2).

When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

When returned from SLEEP0 mode, the CPU restarts operating, entering SLOW1 mode back again. SLEEP0 mode is entered and returned regardless of how TBTCR<TBTEN> is set. When IMF = "1", EF7 (TBT interrupt individual enable flag) = "1", and TBTCR<TBTEN> = "1", interrupt processing is performed. When SLEEP0 mode is entered while TBTCR<TBTEN> = "1", the INTTBT interrupt latch is set after returning to SLOW1 mode.

(3) STOP 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 a lowest power consumption during STOP mode.

STOP mode is started by the system control register 1 (SYSCR1), and STOP mode is released by a inputting (Either level-sensitive or edge-sensitive can be programmably selected) to the $\overline{STOP}$ pin or key-on wakeup pin input which is enabled by STOPCR. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.

Note 1: When the IDLE0/1/2 and SLEEP0/1/2 modes are started with the EEPCR<ATPWDW> = "0", the CPU wait period for stabilizing of the power supply of flash control circuit is executed after being released from these mode.

Note 2: When the STOP mode is started with the EEPCR<MNPWDW> = "1", the CPU wait period for stablizing of the power supply of flash control circuit is executed after the STOP warm-up time.

Note 1: NORMAL1 and NORMAL2 modes are generically called NORMAL; SLOW1 and SLOW2 are called SLOW;
IDLE0, IDLE1 and IDLE2 are called IDLE; SLEEP0, SLEEP1 and SLEEP2 are called SLEEP.

Note 2: The mode is released by falling edge of TBTCR<TBTCK> setting.

Operating Mode		Oscillator		CPU Core	TBT	Other Peripherals	Machine Cycle Time
		High frequency	Low frequency				
Single clock	RESET	Oscillation	Stop	Reset	Reset	Reset	4/fc [s]
	NORMAL1			Operate	Operate	Operate	
	IDLE1			Halt			
	IDLE0					Halt	–
	STOP	Stop			Halt		
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with high frequency	Operate	Operate	4/fc [s]
	IDLE2			Halt			
	SLOW2			Operate with low frequency			4/fs [s]
	SLEEP2			Halt			
	SLOW1	Stop		Operate with low frequency			
	SLEEP1			Halt			
	SLEEP0					Halt	–
	STOP		Stop		Halt		

Figure 1.4.6 Operating Mode Transition Diagram

System Control Register 1

SYSCR1	7	6	5	4	3	2	1	0	
(0038H)	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 pin (P20)	0: Edge-sensitive release 1: Level-sensitive release				
RETM	Operating mode after STOP mode	0: Return to NORMAL1/2 mode 1: Return to SLOW1 mode				
OUTEN	Port output during STOP mode	0: High impedance 1: Output kept				
WUT	Warm-up time at releasing STOP mode (Note 8)		Return to NORMAL mode	Return to SLOW mode		
		00	$3 \times 2^{16}/fc + (2^{10}/fc)$	$3 \times 2^{13}/fs + (2^3/fs)$		
		01	$2^{16}/fc + (2^{10}/fc)$	$2^{13}/fs + (2^3/fs)$		
		10	$3 \times 2^{14}/fc + (2^{10}/fc)$	$3 \times 2^6/fs + (2^3/fs)$		
		11	$2^{14}/fc + (2^{10}/fc)$	$2^6/fs + (2^3/fs)$		

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

Note 2: When STOP mode is released with $\overline{RESET}$ pin input, a return is made to NORMAL1 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 as undefined data when a read instruction is executed.

Note 5: As the hardware becomes STOP mode under OUTEN = "0", input value is fixed to "0"; therefore it may cause interrupt request on account of falling edge.

Note 6: When the Key-on wakeup input (STOP0 to STOP4) is used, RELM should be set to "1".

Note 7: Port P20 is used as $\overline{STOP}$ pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z mode.

Note 8: When the STOP mode is started with the EEPCR<MNPWDW> = "1", the CPU wait period for stabilizing of the power supply of flash control circuit is executed after the STOP warm-up time. (The CPU wait period for FLASH is shown in parentheses)
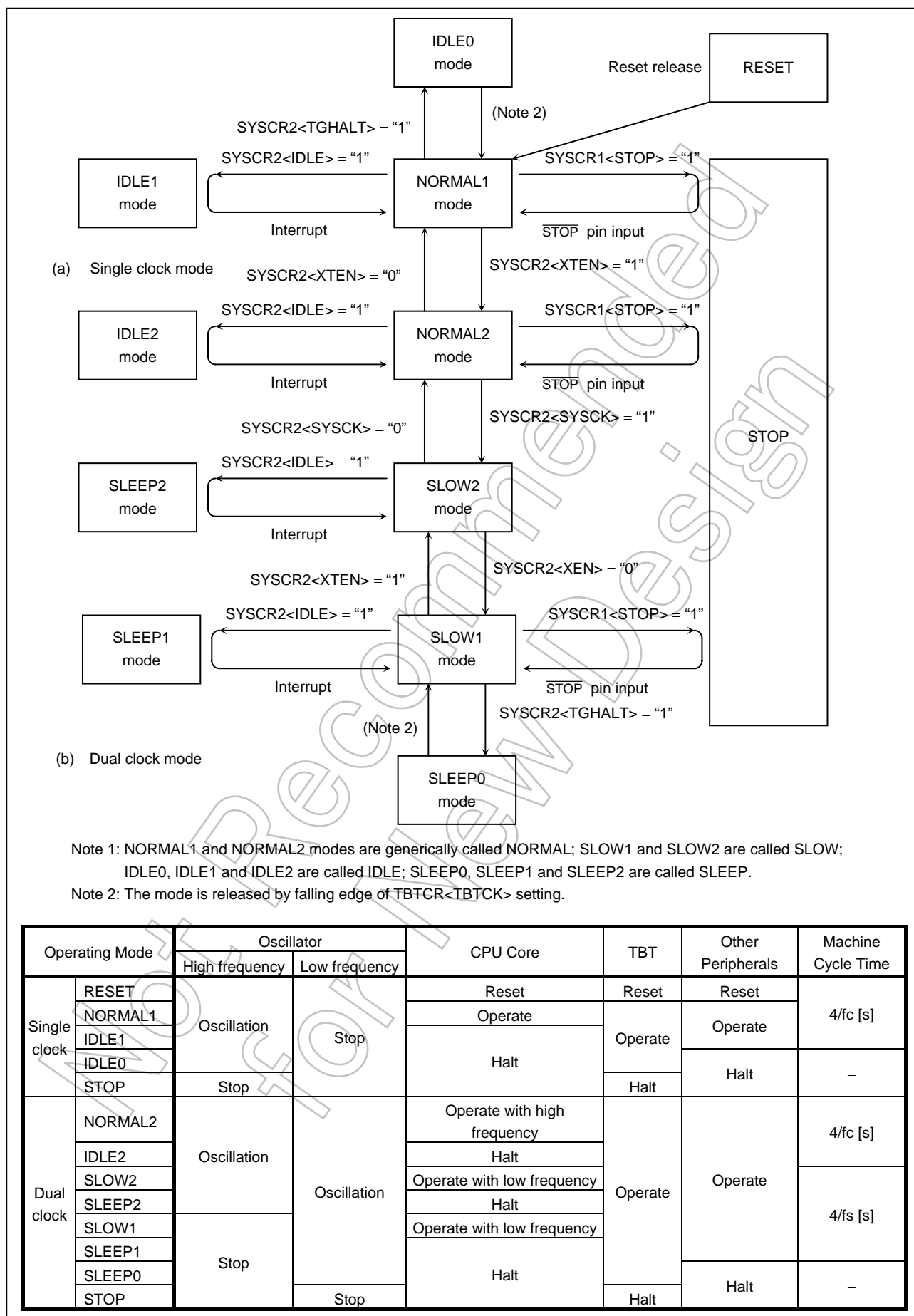
System Control Register 2

SYSCR2	7	6	5	4	3	2	1	0	
(0039H)	XEN	XTEN	SYSCK	IDLE		TGHALT			(Initial value: 1000 *0**)

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	CPU and watchdog timer control (IDLE1/2, SLEEP1/2 mode)	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (Start IDLE1/2, SLEEP1/2 mode)	
TGHALT	TG control (IDLE0, SLEEP0 mode)	0: Feeding clock to all peripherals from TG 1: Stop feeding clock to peripherals except TBT from TG. (Start IDLE0, SLEEP0 mode)	

Note 1: A reset is applied if both XEN and XTEN are cleared to "0", XEN is cleared to "0" when SYSCK = "0", or XTEN is cleared to "0" when SYSCK = "1".

Note 2: *: Don't care, TG: Timing generator

Note 3: Bits 3, 1 and 0 in SYSCR2 are always read as undefined value.

Note 4: Do not set IDLE and TGHALT to "1" simultaneously.

Note 5: Because returning from IDLE0/SLEEP0 to NORMAL1/SLOW1 is executed by the asynchronous internal clock, the period of IDLE0/SLEEP0 mode might be shorter than the period setting by TBTCR<TBTCK>.

Note 6: When IDLE1/2 or SLEEP1/2 mode is released, IDLE is automatically cleared to "0".

Note 7: When IDLE0 or SLEEP0 mode is released, TGHALT is automatically cleared to "0".

Note 8: Before setting TGHALT to "1", be sure to stop peripherals. If peripherals are not stopped, the interrupt latch of peripherals may be set after IDLE0 or SLEEP0 mode is released.

Figure 1.4.7 System Control Registers

1.4.4   Operating Mode Control

(1) STOP mode

STOP mode is controlled by the system control register 1, the $\overline{\text{STOP}}$ pin input and key-on wakeup input (STOP0 to STOP4) which is controlled by the STOP mode release control register (STOPCR).

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 SYSCR1<STOP> to "1". During STOP mode, the following status is maintained.

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

b. The data memory, registers, the program status word and port output latches are all held in the status in effect before STOP mode was entered.

c. The prescaler and the divider of the timing generator are cleared to "0".

d. The program counter holds the address 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started STOP mode.

STOP mode includes a level-sensitive mode and an edge-sensitive mode, either of which can be selected with the SYSCR1<RELM>. Do not use any STOPx (x: 0 to 4) pin input for releasing STOP mode in edge-sensitive mode.

When the STOP mode is started with the EEPCR<MNPWDW> = "1", the CPU wait for stabilizing of the power supply of flash control circuit is executed after the STOP warm-up time.

Note 1:   The STOP mode can be released by either the STOP or key-on wakeup pin (STOP0 to STOP4). However, because the $\overline{\text{STOP}}$ pin is different from the key-on wakeup and can not inhibit the release input, the $\overline{\text{STOP}}$ pin must be used for releasing STOP mode.

Note 2:   During STOP period (from start of STOP mode to end of warm up), due to changes in the external interrupt pin signal, interrupt latches may be set to "1" and interrupts may be accepted immediately after STOP mode is released. Before starting STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

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

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high or setting the STOPx (x: 0 to 4) pin input which is enabled by STOPCR. This mode is used for capacitor backup when the main power supply is cut off and long term battery backup.

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 two methods can be used for confirmation.

a. Testing a port P20.

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

Example 1: Starting STOP mode from NORMAL mode by testing a port P20.

	LD	(SYSCR1), 01010000B	;	Sets up the level-sensitive release mode.
SSTOPH:	TEST	(P2PRD). 0	;	Wait until the $\overline{STOP}$ pin input goes low level.
	JRS	F, SSTOPH		
	SET	(SYSCR1).7	;	Starts STOP mode.

Example 2: Starting STOP mode from NORMAL mode with an INT5 interrupt.

PINT5:	TEST	(P2PRD). 0	;	To reject noise, STOP mode does not start if port P20 is at high.
	JRS	F, SINT5		
	LD	(SYSCR1), 01010000B	;	Sets up the level-sensitive release mode.
	SET	(SYSCR1). 7	;	Starts STOP mode.
SINT5:	RETI			
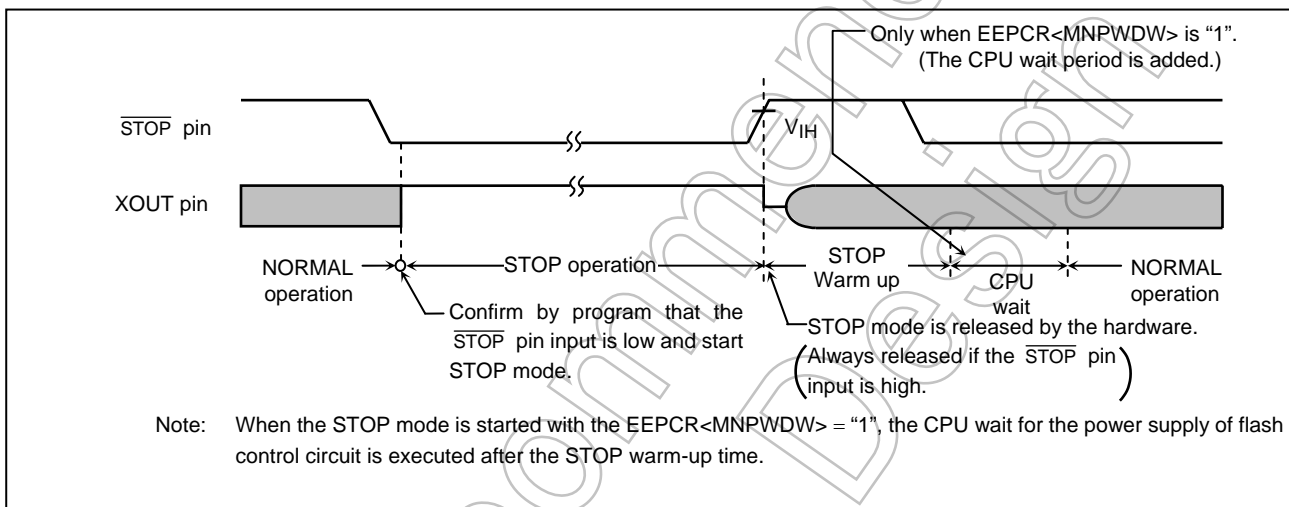
Figure 1.4.8 Level-sensitive Release Mode

Note 1: Even if the $\overline{STOP}$ pin input is low after warm-up start, the STOP mode is not restarted.

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

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

In this mode, STOP mode is released by a rising edge of the $\overline{STOP}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{STOP}$ pin. In the edge-sensitive release mode, STOP mode is started even when the $\overline{STOP}$ pin input is high level. Do not use any STOPx (x: 0 to 4) pin input for releasing STOP mode in edge-sensitive release mode.

Example: Starting STOP mode from NORMAL mode.

| | LD | (SYSCR1), 10010000B | ; | Starts after specified to the edge-sensitive release mode. |
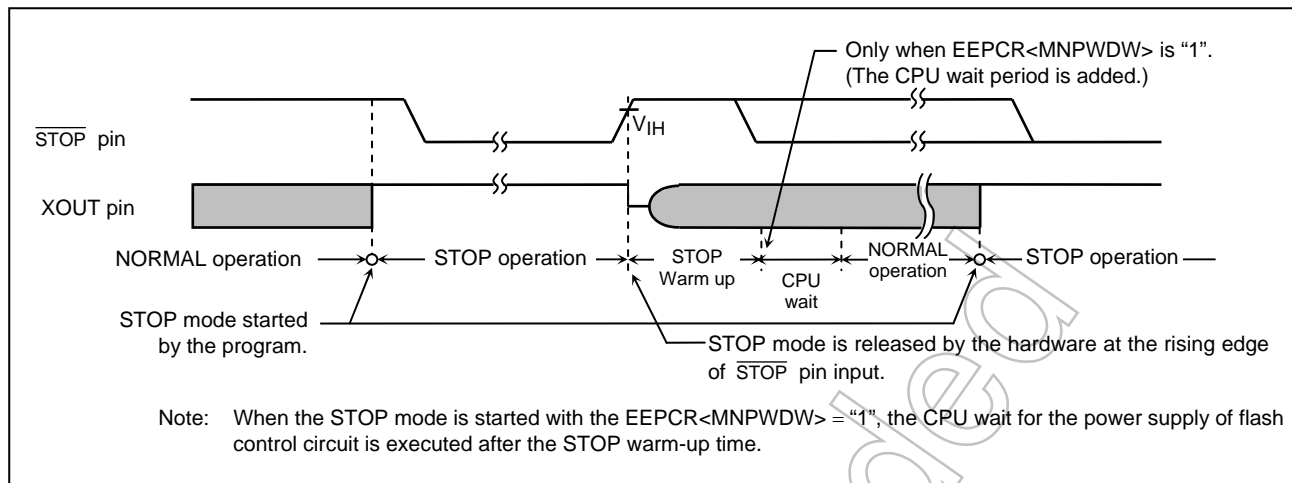
Figure 1.4.9 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

a.   In the dual-clock mode, when returning to NORMAL2, both the high-frequency and low-frequency clock oscillators are turned on; when returning to SLOW1 mode, only the low-frequency clock oscillator is turned on. In the single-clock mode, only the high-frequency clock oscillator is turned on.

b.   A STOP warm-up period is inserted to allow oscillation time to stabilize. During STOP warm up, all internal operations remain halted. Four different STOP warm-up times can be selected with the SYSCR1<WUT> in accordance with the resonator characteristics.

c.   When the EEPCR<MNPWDW> is "1", the CPU wait period is inserted to stabilize the power supply of flash control circuit. During CPU wait, though CPU operations remain halted, the peripheral function operation is resumed, and the counting of the timing generator is restarted. After the CPU wait is finished, normal operation resumes with the instruction following the STOP mode start instruction.

d.   When the EEPCR<MNPWDW> is "0", normal operation resumes with the instruction following the STOP mode start instruction after the STOP warm up.

Note 1: When the STOP mode is released, the start is made after the prescaler and the divider of the timing generator are cleared to "0".

Note 2: STOP mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the normal reset operation.

Note 3: 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 "H" level, 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 pace 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).

Table 1.4.1 Warm-up Time Example (at fc = 16.0 MHz, fs = 32.768 kHz)

| WUT | Warm-up Time [ms] (Note 2) | |
	Return to NORMAL Mode	Return to SLOW Mode
00	12.288 + (0.064)	750 + (0.244)
01	4.096 + (0.064)	250 + (0.244)
10	3.072 + (0.064)	5.85 + (0.244)
11	1.024 + (0.064)	1.95 + (0.244)

Note 1: The warm-up time is obtained by dividing the basic clock by the divider: Therefore, the warm-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 warm-up time must be considered an approximate value.

Note 2: The CPU wait period for FLASH is shown in parentheses.

(a) STOP mode start (Example: Start with SET (SYSCR1). 7 instruction located at address a)
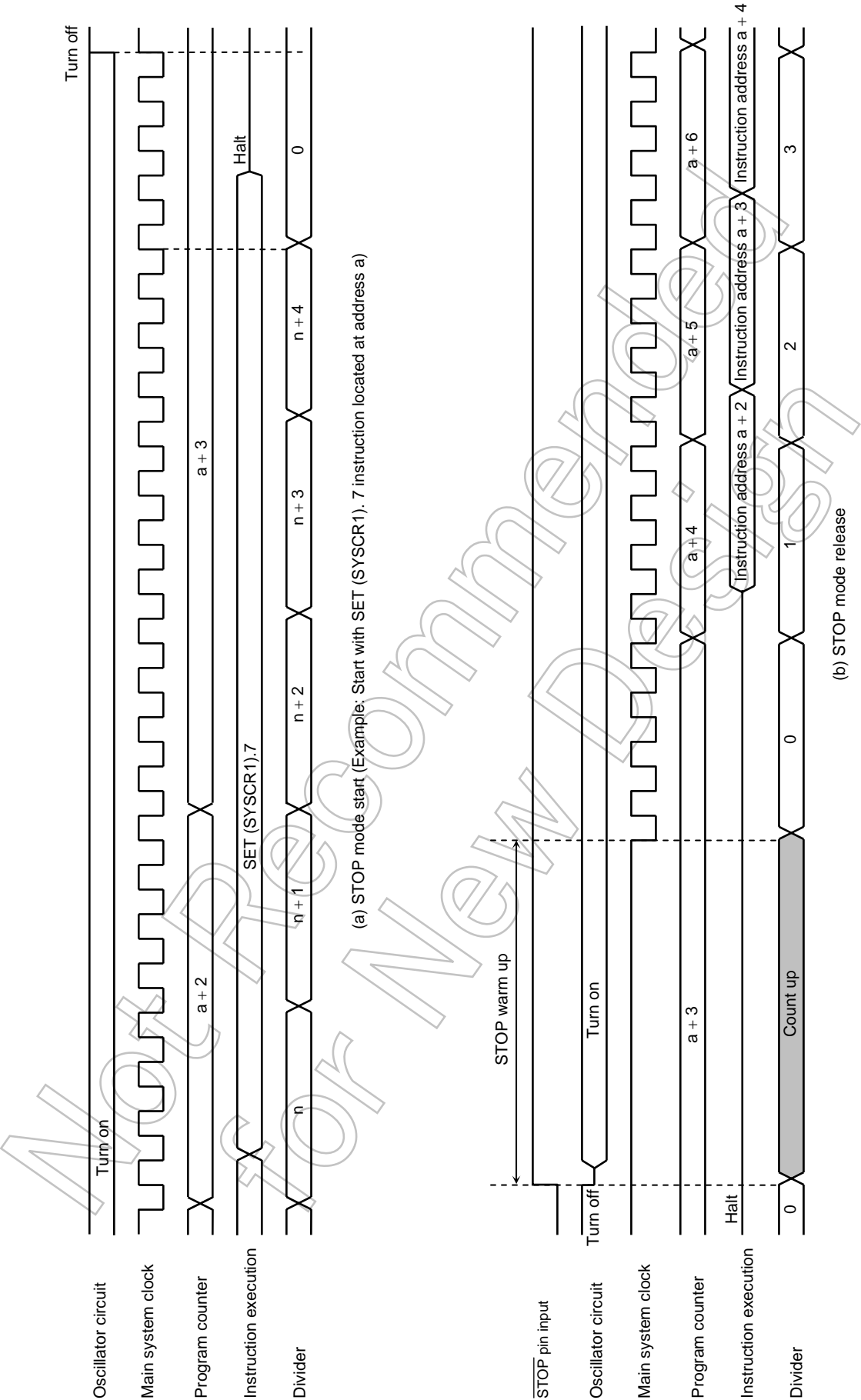
(b) STOP mode release

Figure 1.4.10 STOP Mode Start/Release (when EEPCR<MNPWDW> = "0")
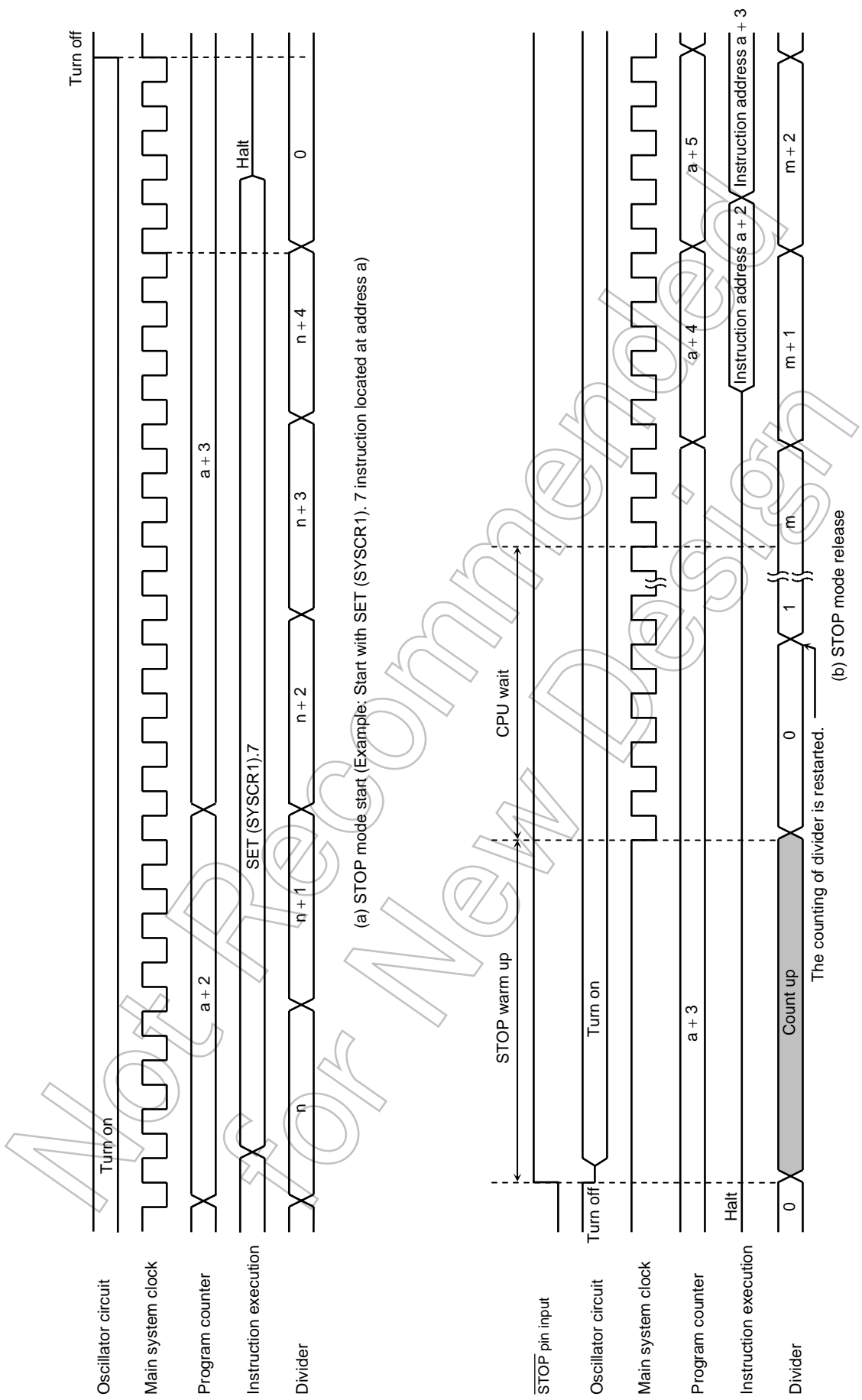
Figure 1.4.11 STOP Mode Start/Release (when EEPCR<MNPWDW> = "1")

(2) IDLE1/2 mode, SLEEP1/2 mode

IDLE1/2 and SLEEP1/2 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following status is maintained during these modes.

a. Operation of the CPU and watchdog timer (WDT) is halted. On-chip peripherals continue to operate.

b. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before these modes were entered.

c. The program counter holds the address 2 ahead of the instruction which starts these modes.

Note 1: EEPCR<ATPWDW> is a bit1 in EEPCR, which is a control bit of the power supply circuit for flash.
Note 2: During CPU wait, though CPU operations remain halted, the peripheral function operation is resumed. Therefore in this time, though the interrupt latch might be set, interrupt operation is not executed until the CPU wait is finished.
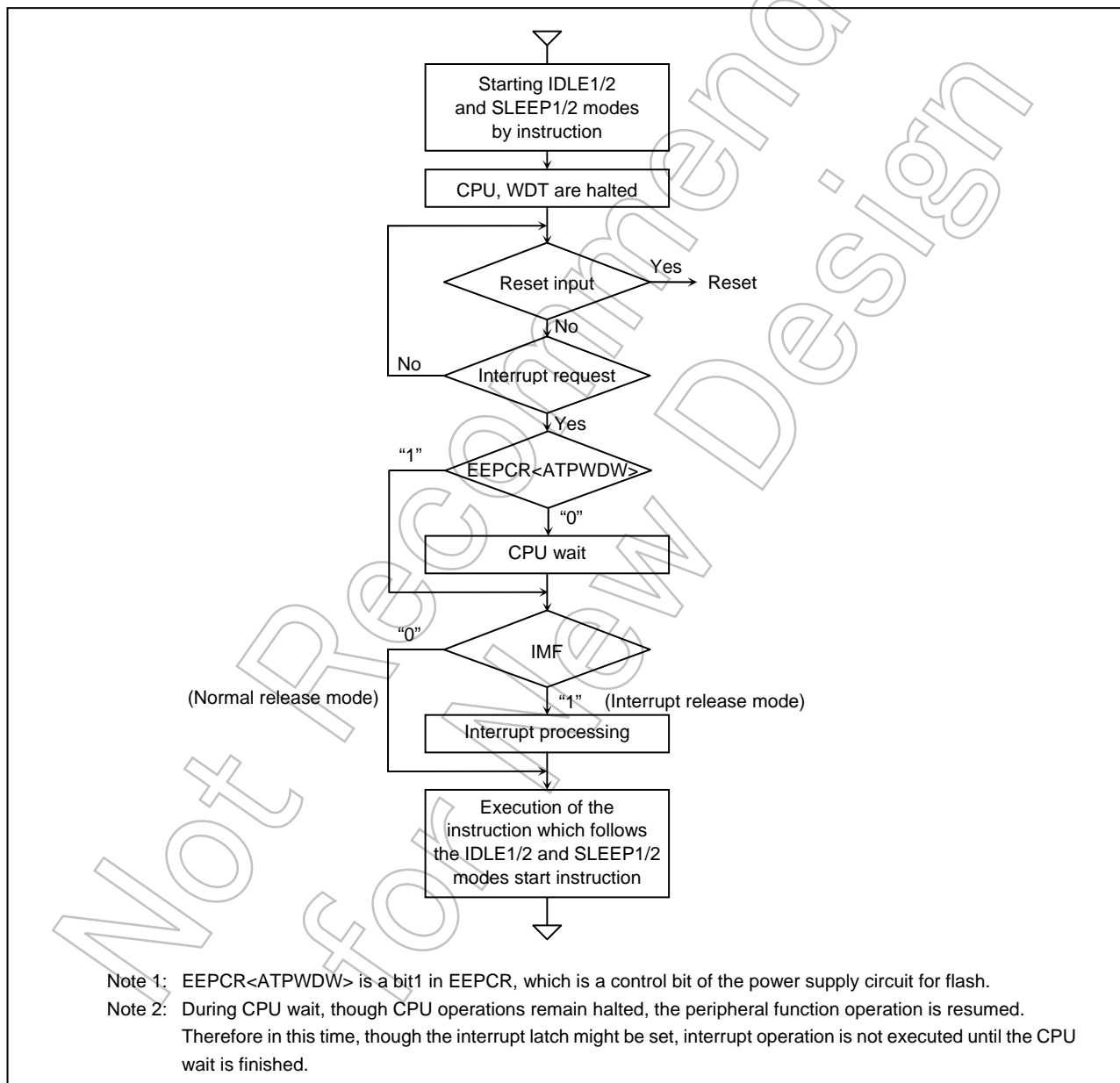
Figure 1.4.12 IDLE1/2, SLEEP1/2 Modes

- Start the IDLE1/2 and SLEEP1/2 modes

    When IDLE1/2 and SLEEP1/2 modes start, set SYSCR2<IDLE> to "1".

- Release the IDLE1/2 and SLEEP1/2 modes

    IDLE1/2 and SLEEP1/2 modes include a normal release mode and an interrupt release mode. These modes are selected by interrupt master enable flag (IMF).

    After releasing IDLE1/2 and SLEEP1/2 modes, the SYSCR2<IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding IDLE1/2 and SLEEP1/2 modes.

    When the IDLE1/2 and SLEEP1/2 modes are started with the EEPCR<ATPWDW> = "0", the CPU wait period for stabilizing of the power supply of flash control circuit is added before the operation mode is returned to the preceding modes. The CPU wait time of IDLE1/2 is 2^{10}/fc [s] and that of SLEEP1/2 mode is 2^3/fs [s].

    IDLE1/2 and SLEEP1/2 modes can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

    Note: During CPU wait, though CPU operations remain halted, but the peripheral function operation is resumed. Therefore in this time, though the interrupt latch might be set, interrupt operation is not executed until the CPU wait is finished.

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

    IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled by the individual interrupt enable flag (EF). After the interrupt is generated, the program operation is resumed from the instruction following the IDLE1/2 and SLEEP1/2 modes start instruction. Normally, the interrupt latches (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

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

    IDLE1/2 and SLEEP1/2 modes are released by any interrupt source enabled with the individual interrupt enable flag (EF). After the interrupt is processed, the program operation is resumed from the instruction following the instruction, which starts IDLE1/2 and SLEEP1/2 modes.

    Note: When a watchdog timer interrupts is generated immediately before IDLE1/2 and SLEEP1/2 modes are started, the watchdog timer interrupt will be processed but IDLE1/2 and SLEEP1/2 modes will not be started.

Figure 1.4.13 IDLE1/2, SLEEP1/2 Modes Start/Release

(3) IDLE0, SLEEP0 modes (IDLE0, SLEEP0)

IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR). The following status is maintained during IDLE0 and SLEEP0 modes.

a. Timing generator stops feeding clock to peripherals except TBT.

b. The data memory, CPU registers, program status word and port output latches are all held in the status in effect before IDLE0 and SLEEP0 modes were entered.

c. The program counter holds the address 2 ahead of the instruction which starts IDLE0 and SLEEP0 modes.

Note: Before starting IDLE0 or SLEEP0 mode, be sure to stop (Disable) periperals.

Note 1: EEPCR<ATPWDW> is a bit1 in EEPCR, which is a control bit of the power supply circuit for flash.

Note 2: During CPU wait, though CPU operations remain halted, but the peripheral function operation is resumed. Therefore in this time, though the interrupt latch might be set, interrupt operation is not executed until the CPU wait is finished.

Figure 1.4.14 IDLE0, SLEEP0 Modes

- Start the IDLE0 and SLEEP0 modes

    Stop (Disable) peripherals such as a timer counter.

    When IDLE0 and SLEEP0 modes start, set SYSCR2<TGHALT> to "1".

- Release the IDLE0 and SLEEP modes

    IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode.

    These modes are selected by interrupt master flag (IMF), individual interrupt enable flag (EF7) for INTTBT and TBTCR<TBTEN>.

    After releasing IDLE0 and SLEEP0 modes, the SYSCR2<TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding IDLE0 and SLEEP0 modes. Before starting the IDLE0 or SLEEP0 mode, when the TBTCR<TBTEN> is set to "1", INTTBT interrupt latch is set to "1".

    When the IDLE0 and SLEEP0 modes are started with the EEPCR<ATPWDW> = "0", the CPU wait period for stabilizing of the power supply of flash control circuit is added before the operation mode is returned to the preceding modes. The CPU wait time of IDLE0 is $2^{10}/fc$ [s] and that of SLEEP0 mode is $2^3/fs$ [s].

    IDLE0 and SLEEP0 modes can also be released by inputting low level on the $\overline{\text{RESET}}$ pin. After releasing reset, the operation mode is started from NORMAL1 mode.

    Note 1: IDLE0 and SLEEP0 modes start/release without reference to TBTCR<TBTEN> setting.

    Note 2: During CPU wait, though CPU operations remain halted, but the peripheral function operation is resumed. Therefore in this time, though the interrupt latch might be set, interrupt operation is not executed until the CPU wait is finished.

a. Normal release mode (IMF · EF7 · TBTCR<TBTEN> = "0")

    IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK>. After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction.

b. Interrupt release mode (IMF · EF7 · TBTCR<TBTEN> = "1")

    IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> and INTTBT interrupt processing is started.

    Note 1: Because returning from IDLE0, SLEEP0 to NORMAL1, SLOW1 is executed by the asynchronous internal clock, the period of IDLE0, SLEEP0 mode might be the shorter than the period setting by TBTCR<TBTCK>.

    Note 2: When a watchdog timer interrupt is generated immediately before IDLE0/ SLEEP0 mode is started, the watchdog timer interrupt will be processed but IDLE0/SLEEP0 mode will not be started.

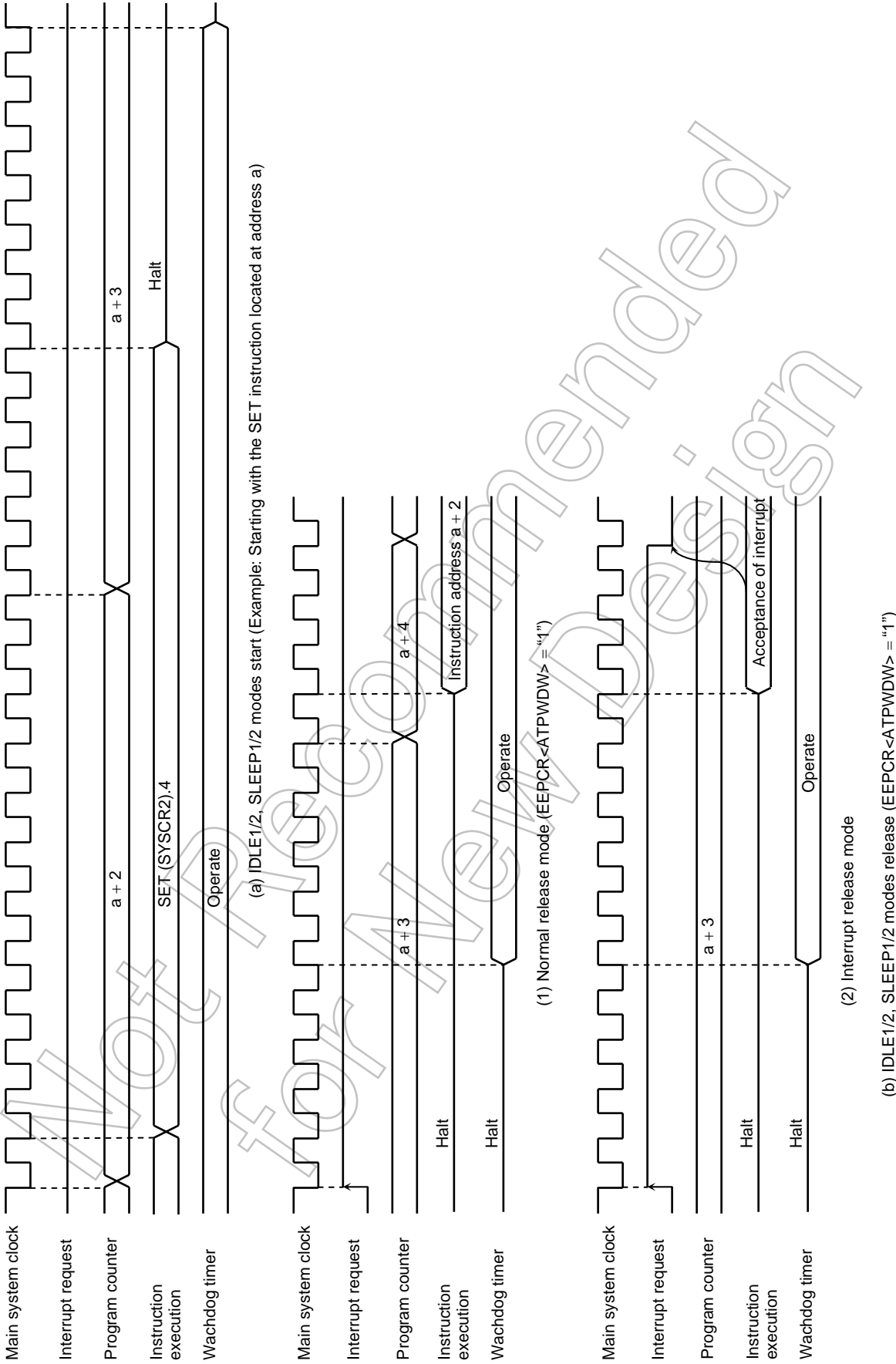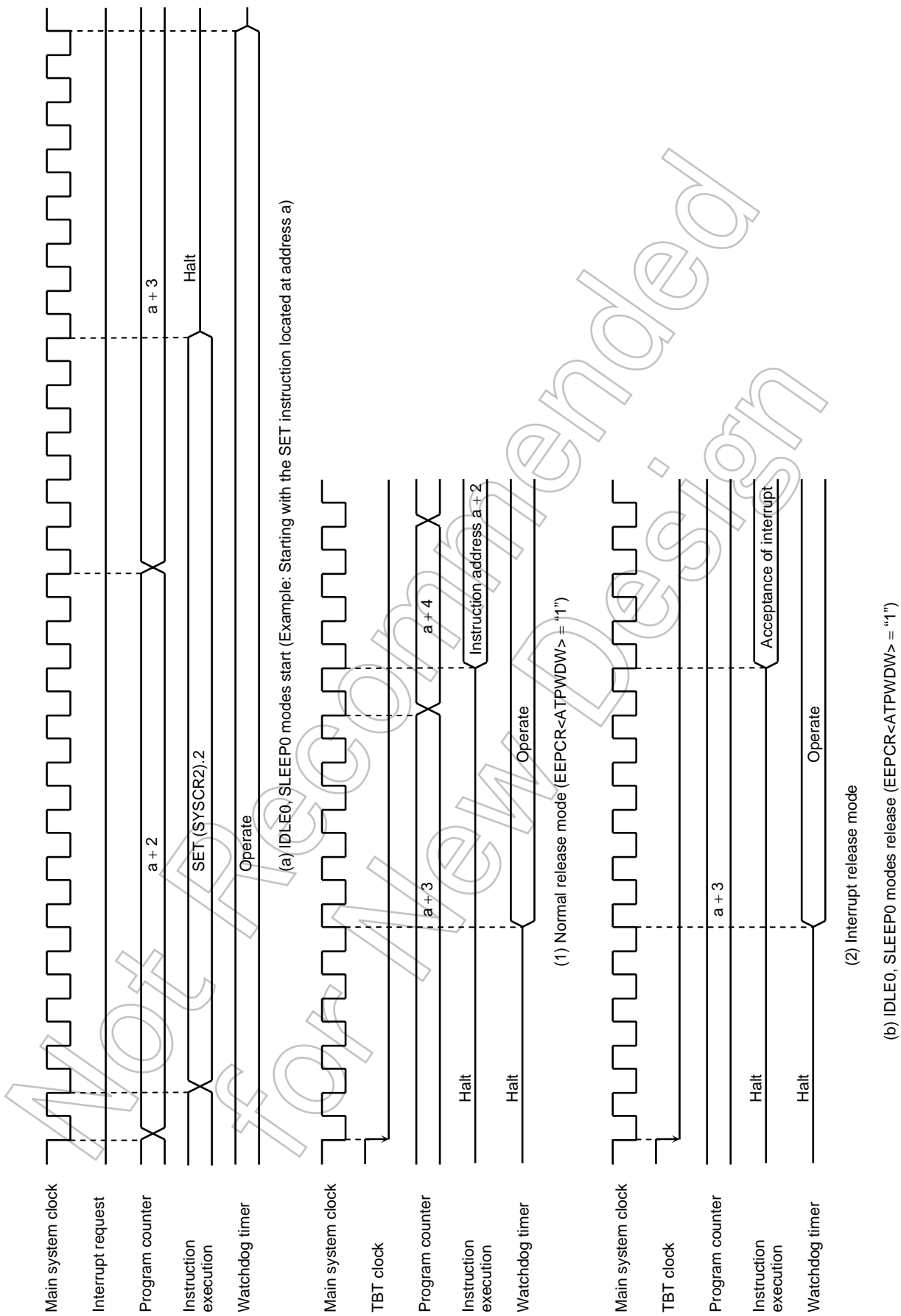Figure 1.4.15 IDLE0, SLEEP0 Modes Start/Release

(4) SLOW mode

SLOW mode is controlled by the system control register 2 (SYSCR2).

The following is the methods to switch the mode with the warm-up counter (TC2).

a. Switching from NORMAL2 mode to SLOW1 mode

First, set SYSCR2<SYSCK> to switch the main system clock to the low-frequency clock for SLOW2 mode.

Next, clear SYSCR2<XEN> to turn off high-frequency oscillation.

Note: The high-frequency clock oscillation can be continued to return quickly to NORMAL2 mode. But starting STOP mode while SLOW mode, the high-frequency oscillation must be stopped.

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

Example 1: Switching from NORMAL2 mode to SLOW1 mode.

```
        SET     (SYSCR2). 5      ;   SYSCR2<SYSCK> ← 1
                                     (Switches the main system clock to the
                                     low-frequency clock for SLOW2.)
        CLR     (SYSCR2). 7      ;   SYSCR2<XEN> ← 0
                                     (Turns off high-frequency oscillation.)
```

Example 2: Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
        SET     (SYSCR2). 6      ;   SYSCR2<XTEN> ← 1
        LD      (TC2CR), 14H     ;   Sets mode for TC2
        LDW     (TC2DRL), 8000H  ;   Sets warm-up time
                                     (Depend on oscillator accompanied.)
        DI                       ;   IMF ← 0
        SET     (EIRE). 4        ;   Enables INTTC2
        EI                       ;   IMF ← 1
        SET     (TC2CR). 5       ;   Starts TC2

PINTTC2:    CLR     (TC2CR). 5       ;   Stops TC2
        SET     (SYSCR2). 5      ;   SYSCR2<SYSCK> ← 1
                                     (Switches the main system clock to the
                                     low-frequency clock.)
        CLR     (SYSCR2). 7      ;   SYSCR2<XEN> ← 0
                                     (Turns off high-frequency oscillation.)
        RETI

VINTTC2:    DW      PINTTC2          ;   INTTC2 vector table
```

b. Switching from SLOW1 mode to NORMAL2 mode

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

Note 1: After SYSCK is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.

High-frequency clock

Low-frequency clock
Main system clock

SYSCK
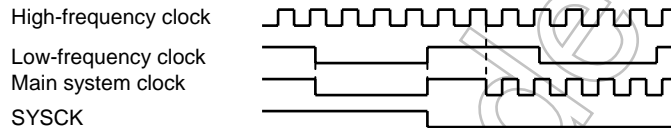
Note 2: SLOW mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the TMP86FP24 is placed in NORMAL1 mode.

Example: Switching from the SLOW1 mode to the NORMAL2 mode.
(fc = 16 MHz, warm-up time is = 4.0 ms.)

```
                SET     (SYSCR2). 7         ;   SYSCR2<XEN> ← 1
                                                (Starts high-frequency oscillation.)
                LD      (TC2CR), 10H        ;   Sets mode for TC2
                                                (Timer mode, fc for source.)
                LD      (TC2DRH), 0F8H      ;   Sets warm-up time
                                                (Depend on oscillator accompanied.)
                DI                          ;   IMF ← 0
                SET     (EIRE). 4           ;   Enables INTTC2
                EI                          ;   IMF ← 1
                SET     (TC2CR). 5          ;   Starts TC2

PINTTC2:        CLR     (TC2CR). 5          ;   Stops TC2
                CLR     (SYSCR2). 5         ;   SYSCR2<SYSCK> ← 0
                                                (Switches the main system clock to the
                                                high-frequency clock.)

                RETI

VINTTC2:        DW      PINTTC2             ;   INTTC2 vector table
```
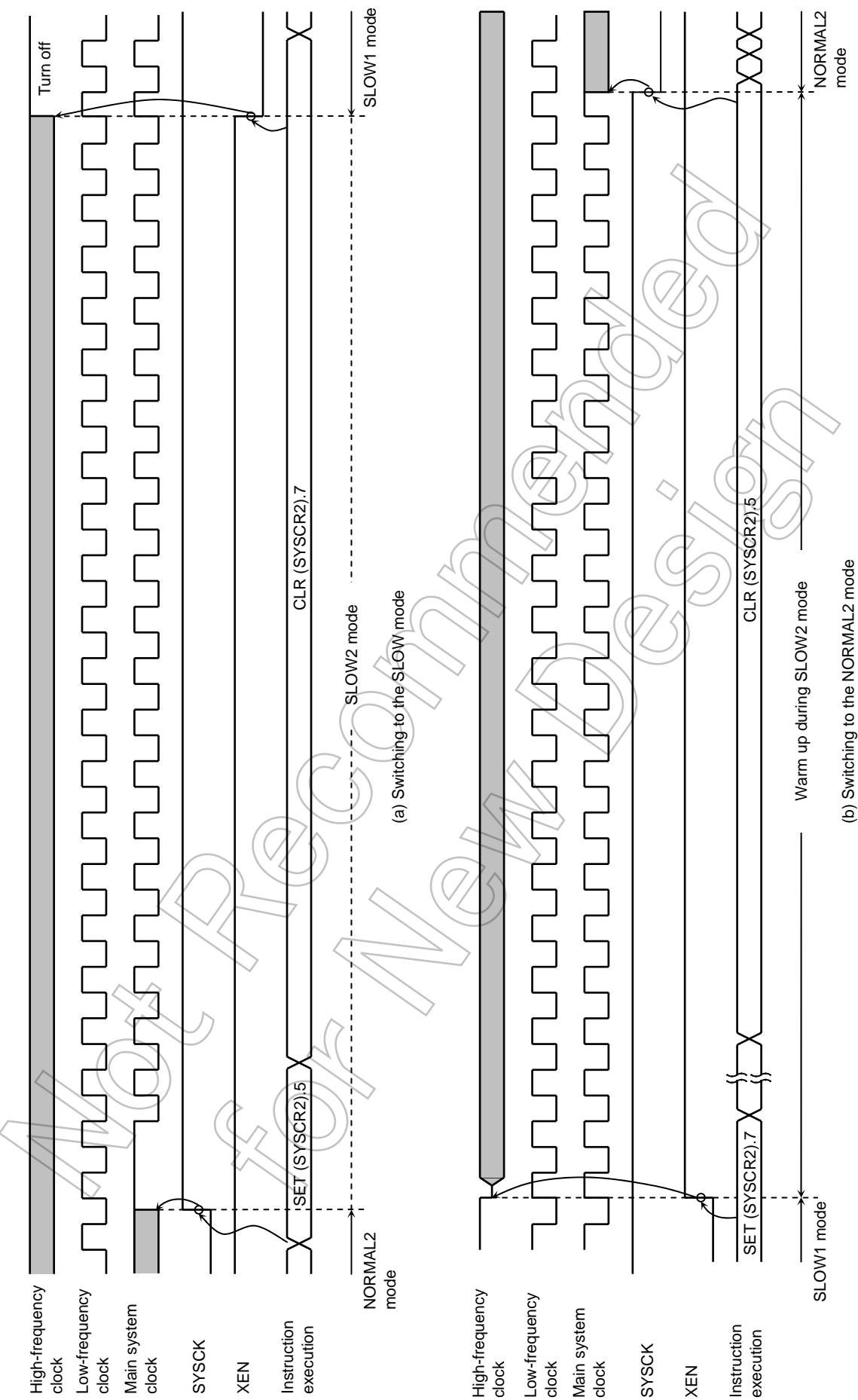
Figure 1.4.16 Switching between the NORMAL2 and SLOW Modes

1.5   Interrupt Control Circuit

The TMP86FP24 has a total (Reset is excluded) of 19 interrupt source: 5 externals and 14 internals. 4 of the internal sources are non-maskable interrupts, and the rest of them are maskable interrupts.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and independent vectors. The interrupt latch is set to "1" by the generation of its interrupt request which requests the CPU to accept its interrupts. Interrupts are enabled or disabled by software using the interrupt master enable flag (IMF) and interrupt enable flag (EF). If more than one interrupts are generated simultaneously, interrupts are accepted in order which is dominated by hardware. However, there are no prioritized interrupt factors among non-maskable interrupts.

Table 1.5.1 Interrupt Sources

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority	
Internal/External	(Reset)	Non maskable	–	FFFEH	High	1
Internal	INTSWI (Software interrupt)	Non maskable	–	FFFCH		2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non maskable	–	FFFCH		2
Internal	INTATRAP (Address trap interrupt)	Non maskable	IL_2	FFFAH		2
Internal	INTWDT (Watchdog timer interrupt)	Non maskable	IL_3	FFF8H		2
External	$\overline{INT0}$ (External interrupt 0)	$IMF \cdot EF_4 = 1$	IL_4	FFF6H		5
Internal	INTTC1 (TC1 interrupt)	$IMF \cdot EF_5 = 1$	IL_5	FFF4H		6
External	INT1 (External interrupt 1)	$IMF \cdot EF_6 = 1$	IL_6	FFF2H		7
Internal	INTTBT (Time-base-timer interrupt)	$IMF \cdot EF_7 = 1$	IL_7	FFF0H		8
External	INT2 (External interrupt 2)	$IMF \cdot EF_8 = 1$	IL_8	FFEEH		9
Internal	INTTC3 (TC3 interrupt)	$IMF \cdot EF_9 = 1$	IL_9	FFECH		10
Internal	INTSIO1 (Serial interface 1 interrupt)	$IMF \cdot EF_{10} = 1$	IL_{10}	FFEAH		11
Internal	INTSIO2 (Serial interface 2 interrupt)	$IMF \cdot EF_{11} = 1$	IL_{11}	FFE8H		12
Internal	INTTC5 (TC5 interrupt)	$IMF \cdot EF_{12} = 1$	IL_{12}	FFE6H		13
External	INT3 (External interrupt 3)	$IMF \cdot EF_{13} = 1$	IL_{13}	FFE4H		14
Internal	INTADC (AD converter interrupt)	$IMF \cdot EF_{14} = 1$	IL_{14}	FFE2H		15
	Reserved	$IMF \cdot EF_{15} = 1$	IL_{15}	FFE0H		16
	Reserved	$IMF \cdot EF_{16} = 1$	IL_{16}	FFBEH		17
	Reserved	$IMF \cdot EF_{17} = 1$	IL_{17}	FFBCH		18
Internal	INTRXD (UART received interrupt)	$IMF \cdot EF_{18} = 1$	IL_{18}	FFBAH		19
Internal	INTTXD (UART transmitted interrupt)	$IMF \cdot EF_{19} = 1$	IL_{19}	FFB8H		20
Internal	INTTC2 (TC2 interrupt)	$IMF \cdot EF_{20} = 1$	IL_{20}	FFB6H		21
External	$\overline{INT5}$ (External interrupt 5)	$IMF \cdot EF_{21} = 1$	IL_{21}	FFB4H		22
	Reserved	$IMF \cdot EF_{22} = 1$	IL_{22}	FFB2H		23
	Reserved	$IMF \cdot EF_{23} = 1$	IL_{23}	FFB0H	Low	24

Note 1: To use the watchdog timer interrupt (INTWDT), clear WDTCR1<WDTOUT> to "0" (It is set for the "reset request" after reset is released). For details, see 2.4 "Watchdog Timer".

Note 2: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to "0" (It is set for the "reset request" after reset is released). For details, see 2.4.5 "Address Trap".
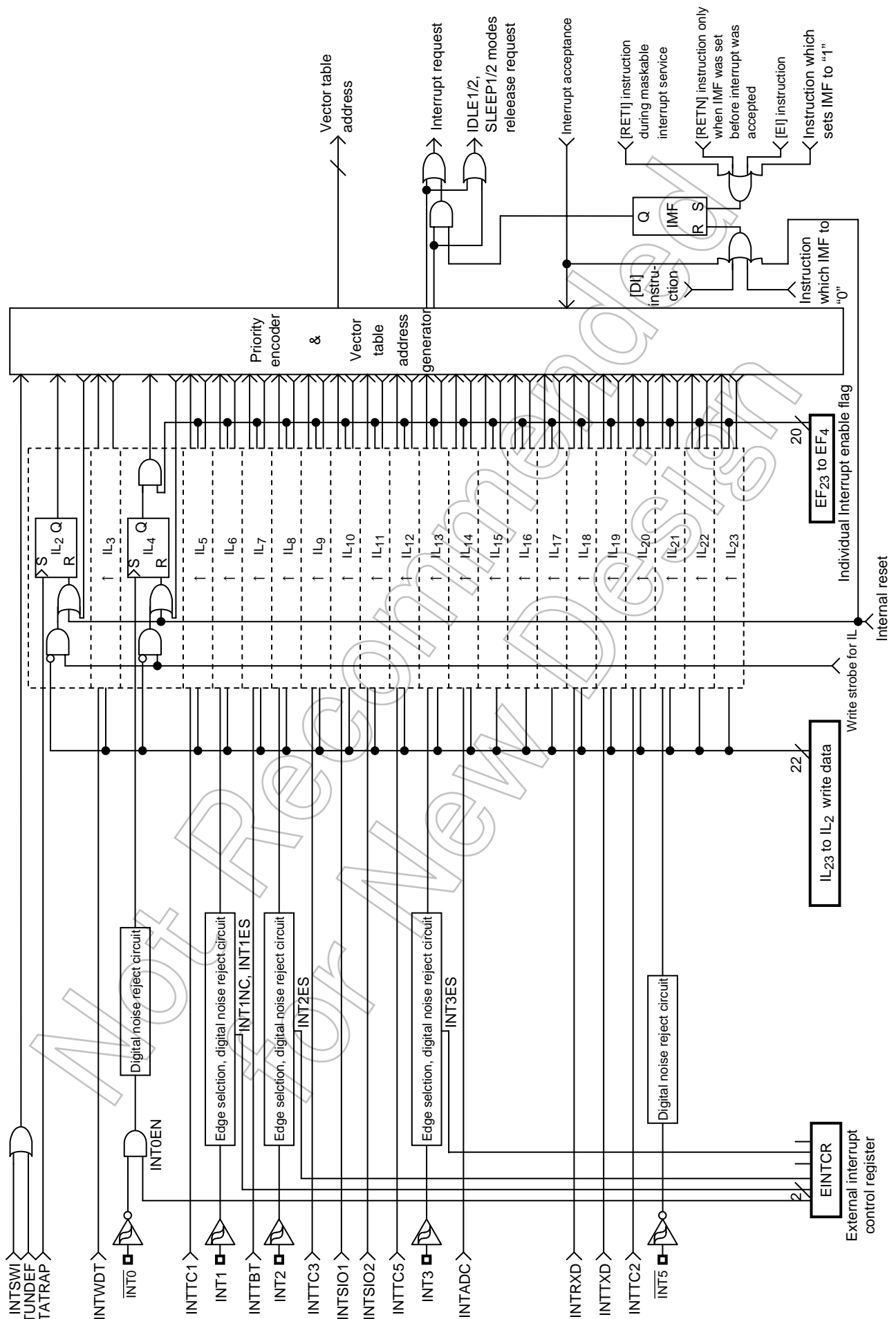
Figure 1.5.1 Interrupt Controller Block Diagram

(1) Interrupt latches (IL$_{24}$ to IL$_2$)

An interrupt latch is provided for each interrupt source, except for a software interrupt. When interrupt request is generated, the latch is set to "1", and the CPU is requested to accept the interrupt if its interrupt is enabled. All interrupt latches are initialized to "0" during reset.

The interrupt latches are located on address 002EH, 003CH and 003DH in SFR area. Except for IL$_3$ and IL$_2$, each latch can be cleared to "0" individually by instruction. (However, the read-modify-write instructions such as bit manipulation or operation instructions cannot be used. Interrupt request would be cleared inadequately if interrupt is requested while such instructions are executed.) Thus interrupt request can be canceled/initialized by software.

Interrupt latches are not set to "1" by an instruction. Since interrupt latches can be read, the status for interrupt requests can be monitored by software.

Note: When manipulating IL, clear IMF (to disable interrupts) beforehand.

```
Example 1: Clears interrupt latches.
           DI                                    ;  IMF ← 0
           LD        (ILE), 11110011B            ;  IL19, IL18 ← 0
           LDW       (ILL), 1110100000111111B    ;  IL12, IL10 to IL6 ← 0
           EI                                    ;  IMF ← 1
Example 2: Reads interrupt latches.
           LD        WA, (ILL)                   ;  W ← ILH, A ← ILL
Example 3: Tests an interrupt latches.
           TEST      (IL).7                      ;  IL7 = 1 then jump.
           JR        F, SSET
```

(2) Interrupt enable register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (Software interrupt, undefined instruction interrupt, address trap interrupt and watchdog timer interrupt). Non-maskable interrupt is accepted regardless of the contents of the EIR.

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located on address 002CH, 003AH and 003BH in SFR area, and they can be read and written by an instructions (including read-modify-write instructions such as bit manipulation or operation instructions).

a.   Interrupt master enable flag (IMF)

The interrupt enable register (IMF) enables and disables the acceptance of the whole maskable interrupt. While IMF = "0", all maskable interrupts are not accepted regardless of the status on each individual interrupt enable flag (EF). By setting IMF to "1", the interrupt becomes acceptable if the individuals are enabled. When an interrupt is accepted, IMF is cleared to "0" after the latest status on IMF is stacked. Thus the maskable interrupts which follow are disabled. By executing return interrupt instruction [RETI/RETN], the stacked data, which was the status before interrupt acceptance, is loaded on IMF again.

The IMF is located on bit0 in EIRL (Address: 003AH in SFR), and can be read and written by an instruction. The IMF is normally set and cleared by [EI] and [DI] instruction respectively. During reset, the IMF is initialized to "0", and maskable interrupts are not accepted until it is set to "1".

b. Individual interrupt enable flags (EF_{23} to EF_4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of its interrupt, and setting the bit to "0" disables acceptance. The individual interrupt enable flags (EF_{23} to EF_4) are located on EIRE, EIRL to EIRH (Address: 002CH, 003AH to 003BH in SFR), and can be read and written by an instruction. During reset, all the individual interrupt enable flags (EF_{23} to EF_4) are initialized to "0" and all maskable interrupts are not accepted until they are set to "1".

Note: Before manipulating EF, be sure to clear IMF (Interrupt disabled). Then set IMF newly again after operating on the interrupt enables flag (EF). Normally, IMF is clear to "0" automatically on service routine. When IMF is set to "1" for using a multiple interrupt on service routine, be sure to process as is the case with EF.

```
Example 1:  Enables interrupts individually and sets IMF.
            DI                                            ;   IMF ← "0"
            LD        (EIRE), 00001100B                   ;   EF19, EF18 ← "1"
            LDW       (EIRL), 0110100010100000B           ;   EF14, EF13, EF11, EF7, EF5 ← "1"
            ┊                                                 Note: IMF is not set.
            EI                                            ;   IMF ← "1"
Example 2:  C compiler description example.
            unsigned int _io (3AH) EIRL;                  ;   /* 3AH shows EIRL address */
            _DI ( );
            EIRL = 10100000B;
            ┊
            _EI ( );
```

Interrupt Latches

ILH, ILL (003CH, 003DH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IL_{15}	IL_{14}	IL_{13}	IL_{12}	IL_{11}	IL_{10}	IL_9	IL_8	IL_7	IL_6	IL_5	IL_4	IL_3	IL_2		

ILH (003DH)　　　　　　　　　　ILL (003CH)

(Initial value: 00000000 000000**)

ILE (002EH)	23	22	21	20	19	18	17	16
	IL_{23}	IL_{22}	IL_{21}	IL_{20}	IL_{19}	IL_{18}	IL_{17}	IL_{16}

ILE (002EH)

(Initial value: 00000000)

IL_{23} to IL_2	Interrupt Latches	at RD 0: No interrupt request 1: Interrupt request	at WR Clears the interrupt request (Note 1) (Interrupt latch is not set.)	R/W

Note 1: IL_2 and IL_3 are prohibited from clearing.

Note 2: When manipulating IL, clear IMF (to disable interrupts) beforehand.

Note 3: Do not clear IL with read-modify-write instructions such as bit operations.

Interrupt Enable Registers

EIRH, EIRL (003AH, 003BH)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	EF_7	EF_6	EF_5	EF_4				IMF

EIRH (003BH)

(Initial value: 00000000 00000***0)

EIRE (002CH)	23	22	21	20	19	18	17	16
	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}

EIRE (002CH)

(Initial value: 00000000)

EF_{23} to EF_4	Individual-interrupt enable flag (Specified for each bit)	0: Disable the acceptance of each maskable interrupt. 1: Enable the acceptance of each maskable interrupt.	R/W
IMF	Interrupt master enable flag	0: Disable the acceptance of all maskable interrupts. 1: Enable the acceptance of all maskable interrupts.	

Note 1: *: Don't care

Note 2: When manipulating EF, clear IMF (to disable interrupts) beforehand.

Note 3: Do not set IMF to "1" simultaneously with EF15 to EF4.

Figure 1.5.2 Interrupt Latch (IL), Interrupt Enable Registers (EIR)

1.5.1 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to "0" by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (4 μs at 8.0 MHz) after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts). Figure 1.5.3 shows the timing chart of interrupt acceptance processing.

（1）Interrupt acceptance processing is packaged as follows.

1. The interrupt master enable flag (IMF) is cleared to "0" in order to disable the acceptance of any following interrupt.

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

3. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.

4. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.

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

Note: When the contents of PSW are saved on the stack, the contents of IMF are also saved.

Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instructrion is stored

Note 2: On condition that interrupt is enabled, it takes 38/fc [s] or 38/fs [s] at maximum (If the interrupt latch is set at the first machine cycle on 10-cycle instruction) to start interrupt acceptance processing since its interrupt latch is set.

Figure 1.5.3 Timing Chart of Interrupt Acceptance/Return Interrupt Instruction

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

Vector table address   Entry address

FFF0H   03H   } Vector   ⇒   D203H   0FH   → Interrupt service
FFF1H   D2H   D204H   06H   program
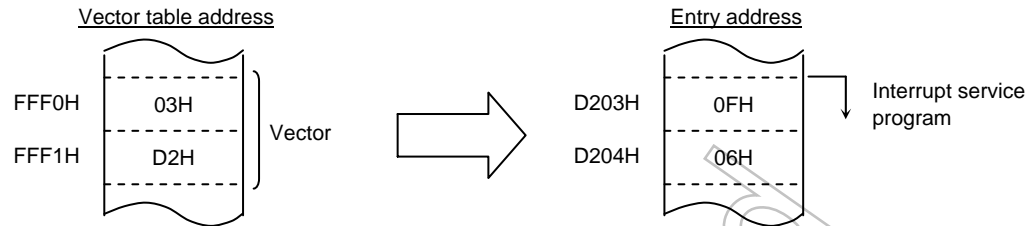
A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable
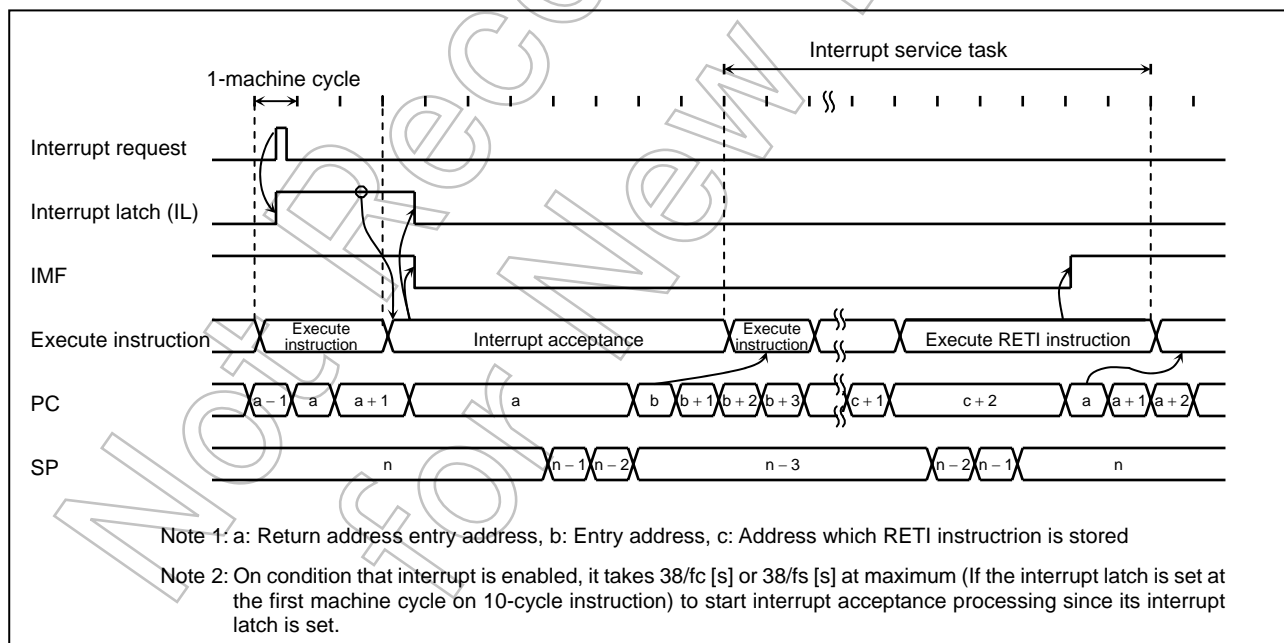interrupt higher than the level of current servicing interrupt is requested.
In order to utilize nested interrupt service, the IMF is set to "1" in the interrupt service
program. In this case, acceptable interrupt sources are selectively enabled by the
individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose
interrupt is currently serviced, before setting IMF to "1". As for non-maskable
interrupt, keep interrupt service shorter compared with length between interrupt
requests; otherwise the status cannot be recovered as non-maskable interrupt would
simply nested.

(2) Saving/restoring general -purpose registers

During interrupt acceptance processing, the program counter (PC) and the program
status word (PSW, includes IMF) are automatically saved on the stack, but the
accumulator and others are not. These registers are saved by software if necessary.
When multiple interrupt services are nested, it is also necessary to avoid using the
same data memory area for saving registers. The following methods are used to
save/restore the general-purpose registers.

a.   Using PUSH and POP instructions

To save only a specific register, PUSH and POP instructions are available.

Example: Save/store register using PUSH and POP instructions.
PINTxx: PUSH   WA   ;   Save WA register.
               (Interrupt processing)
               POP   WA   ;   Restore WA register.
               RETI   ;   RETURN

| | At acceptance of an interrupt | At execution of PUSH instruction | At execution of POP instruction | At execution of an RETI instruction | Address (Example) |

At acceptance of an interrupt ⇨ At execution of PUSH instruction ⇨ At execution of POP instruction ⇨ At execution of an RETI instruction
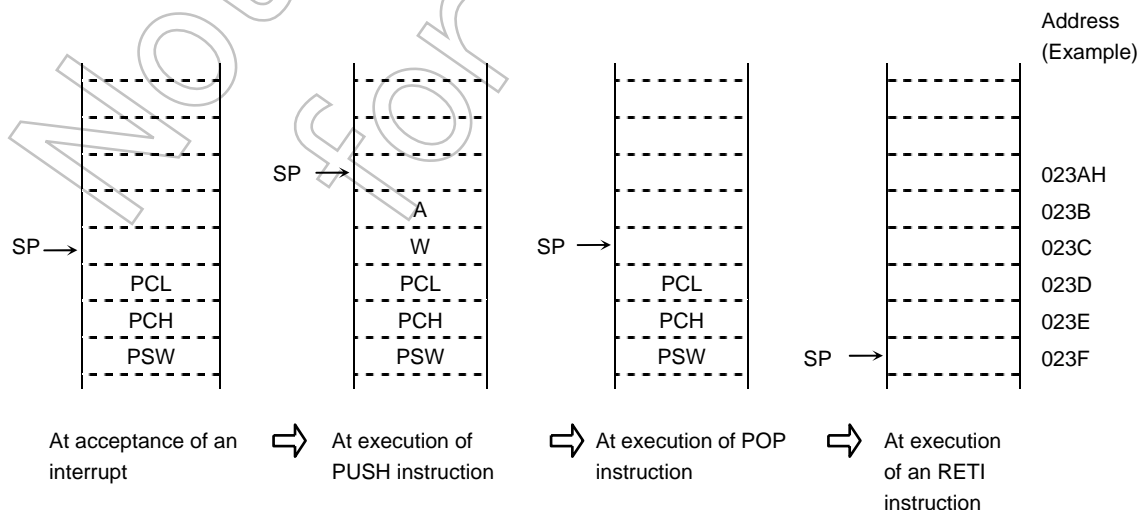
b.   Using data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

```
Example:  Save/store register using data transfer instructions.
PINTxx:          LD        (GSAVA), A            ;    Save A register.
                 (Interrupt processing)
                 LD        A, (GSAVA)            ;    Restore A register.
                 RETI                           ;    RETURN
```


Saving/restoring general-purpose registers using PUSH/POP instruction
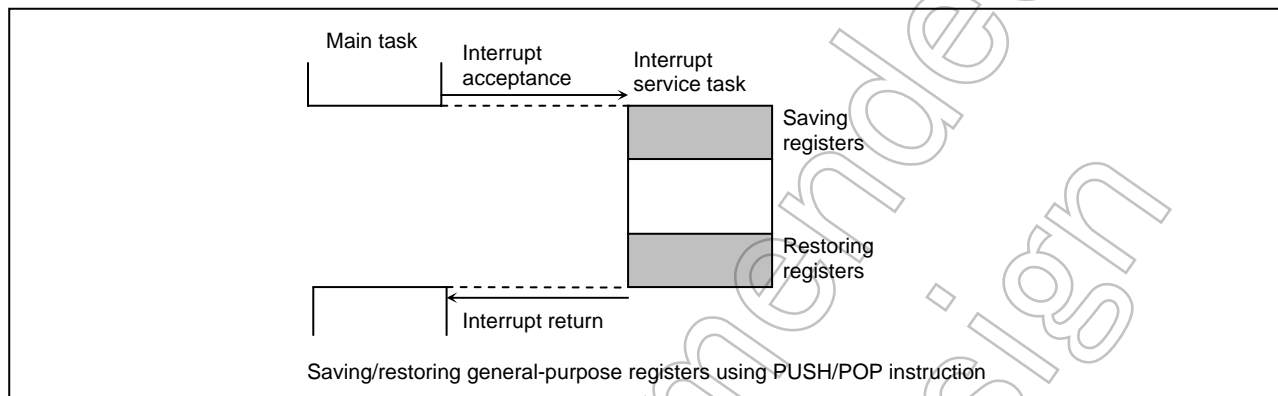
Figure 1.5.4 Saving/Restoring General-purpose Registers under Interrupt Processing

(3) Interrupt return

Interrupt return instructions [RETI]/[RETN] perform as follows.

[RETI]/[RETN] Interrupt Return
1.   Program counter (PC) and program status word (PSW, includes IMF) are restored from the stack.
2.   Stack pointer (SP) is incremented by 3.

As for address trap interrupt (INTARTAP), it is required to alter stacked data for program counter (PC) to restarting address, during interrupt service program. Otherwise returning interrupt causes INTATRAP again. When interrupt acceptance processing has completed, stacked data for PCL and PCH are located on address (SP + 1) and (SP + 2) respectively.

Note: If [RETN] is executed with the above data unaltered, the program returns to the address trap area and INTATRAP occurs again.

Example 1: Returning from address trap interrupt (INTATRAP) service program.

```
PINTxx:    POP      WA                    ;   Recover SP by 2.
           LD       WA, Return Address    ;
           PUSH     WA                    ;   Alter stacked data.
           (Interrupt processing)        ;   RETURN
           RETN
```

Example 2: Restarting without returning interrupt. (In this case, PSW (includes IMF) before interrupt acceptance is discarded.)

```
PINTxx:    INC      SP                    ;   Recover SP by 3.
           INC      SP
           INC      SP
           (Interrupt processing)
           LD       EIRL, data            ;   Set IMF to "1" or clear it to "0".
           JP       Restart Address       ;   Jump into restarting address.
```

Note: It is recommended that stack pointer be return to rate before INTATRAP (Increment 3 times), if return interrupt instruction [RETN] is not utilized during interrupt service program under INTATRAP (Such as Example 2).

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

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

1.5.2 Software Interrupt (INTSW)

Executing the [SWI] instruction generates a software interrupt and immediately starts interrupt processing (INTSW is highest prioritized interrupt).

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

(1) Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH 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 FFH to unused areas of the program memory. Address-trap reset is generated in case that an instruction is fetched from RAM or SFR areas.

(2) Debugging

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

1.5.3 Undefined Instruction Interrupt (INTUNDEF)

Taking code which is not defined as authorized instruction for instruction causes INTUNDEF. INTUNDEF is generated when the CPU fetches such a code and tries to execute it. INTUNDEF is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTUNDEF interrupt process starts, soon after it is requested.

Note: The undefined instruction interrupt (INTUNDEF) forces CPU to jump into vector address, as software interrupt (SWI) does.

1.5.4 Address Trap Interrupt (INTATRAP)

Fetching instruction from unauthorized area for instructions (Address trapped area) causes reset output or address trap interrupt (INTATRAP). INTATRAP is accepted even if non-maskable interrupt is in process. Contemporary process is broken and INTATRAP interrupt process starts, soon after it is requested.

Note: The operating mode under address trapped, whether to be reset output or interrupt processing, is selected on watchdog timer control register (WDTCR).

1.5.5 External Interrupts

The TMP86FP24 has five external interrupt inputs. These inputs are equipped with digital noise reject circuits (Pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT3. $\overline{INT0}$ /P00 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 reject control and $\overline{INT0}$ /P00 pin function selection are performed by the external interrupt control register (EINTCR).

Table 1.5.2 External Interrupts

Source	Pin	Secondary Function Pin	Enable Conditions	Edge	Digital Noise Reject
INT0	$\overline{INT0}$	P00	IMF = 1, EF$_4$ = 1, INT0EN = 1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT1	INT1	P01	IMF•EF$_6$ = 1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT2	INT2	P02	IMF•EF$_8$ = 1		Pulses of less than 7/fc [s] are eliminated as noise. Pulses of 25/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT3	INT3	P14/TC3	IMF•EF$_{13}$ = 1		
INT5	$\overline{INT5}$	P20/$\overline{STOP}$	IMF•EF$_{21}$ = 1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.

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

(1) INT1 pin   55/fc [s] (INT1NC = 1), 199/fc [s] (INT1NC = 0)

(2) INT2, INT3 pin   31/fc [s]

Note 2: Even if the falling edge of $\overline{INT0}$ pin input is detected at INT0EN = 0, the interrupt latch IL$_4$ is not set.

Note 3: When data changed and did a change of I/O when used external interrupt ports as a normal ports, interrupt request signal occurs incorrectly. Handling of prohibition of interrupt enable register (EIR) is necessary.

Note 4: The maximum time from modifying INT1NC until a noise reject time is changed is 2^6/fc

External Interrupt Control Register

EINTCR	7	6	5	4	3	2	1	0	
(0037H)	INT1NC	INT0EN			INT3ES	INT2ES	INT1ES		(Initial value: 00** 000*)

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	
INT0EN	P00/$\overline{INT0}$ pin configuration	0: P00 input/output port 1: $\overline{INT0}$ pin (Port P00 should be set to an input mode)	R/W
INT3ES INT2ES INT1ES	INT3 to INT1 edge select	0: Rising edge 1: Falling edge	

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

Note 2: When the system clock frequency is switched between high and low or when the external interrupt control register (EINTCR) is overwritten, the noise canceller may not operate normally. It is recommended that external interrupts are disabled using the interrupt enable register (EIR).

Figure 1.5.5 External Interrupt Control Register

1.6 Reset Circuit

The TMP86FP24 has four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset.

Since the reset circuit has an 11-stage counter for generation of flash reset, which is the reset counter for stabilizing of the power supply for flash, the reset period is 2^{10}/fc [s] (64 μs at 16.0 MHz).

Because the malfunction reset circuit such as watchdog timer reset, address trap reset and system clock reset is not initialized when power is turned on, the reset operation occur for the maximum 24/fc [s] (1.5 μs at 16.0 MHz).

Therefore, the maximum reset period is 24/fc [s] + 2^{10}/fc [s] (65.5 μs at 16.0 MHz).

Table 1.6.1 shows on-chip hardware initialization by reset action.

Table 1.6.1 Initializing Internal Status by Reset Action

On-chip Hardware		Initial Value	On-chip Hardware	Initial Value
Program counter	(PC)	(FFFEH)	Prescaler and Divider of timing generator	0
Stack pointer	(SP)	Not initialized		
General-purpose registers (W, A, B, C, D, E, H, L, IX, IY)		Not initialized		
Jump status flag	(JF)	Not initialized	Watchdog timer	Enable
Zero flag	(ZF)	Not initialized	Output latches of I/O ports	Refer to I/O port circuitry
Carry flag	(CF)	Not initialized		
Half carry flag	(HF)	Not initialized		
Sign flag	(SF)	Not initialized		
Overflow flag	(VF)	Not initialized		
Interrupt master enable flag	(IMF)	0		
Interrupt individual enable flags	(EF)	0	Control registers	Refer to each of control register
Interrupt latches	(IL)	0		
			RAM	Not initialized

1.6.1 External Reset Input

The $\overline{\text{RESET}}$ pin contains a schmitt trigger (Hysteresis) with an internal pull-up resistor. When the $\overline{\text{RESET}}$ pin is held at "L" level 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 2^{10}/fc (65.5 μs at 16 MHz) period passes after the $\overline{\text{RESET}}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFEH to FFFFH.
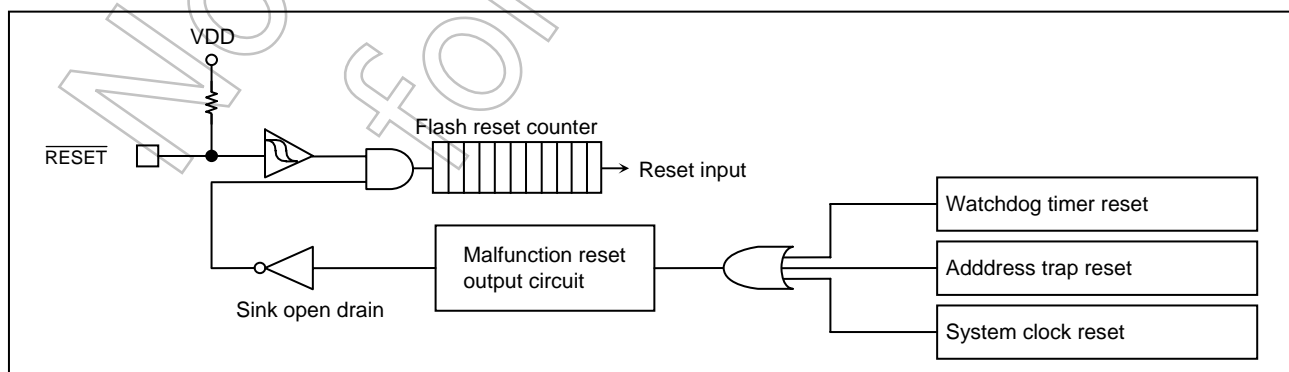
Figure 1.6.1 Reset Circuit

1.6.2 Address-trap-reset

If the CPU should start looping for some cause such as noise and an attempt be made to fetch an instruction from the on-chip RAM (when WDTCR1<ATAS> is set to "1") or the SFR area, address-trap-reset and the flash reset will be generated. The reset time is maximum $24/fc$ [s] $+ 2^{10}/fc$ [s] (65.5 μs at 16.0 MHz).
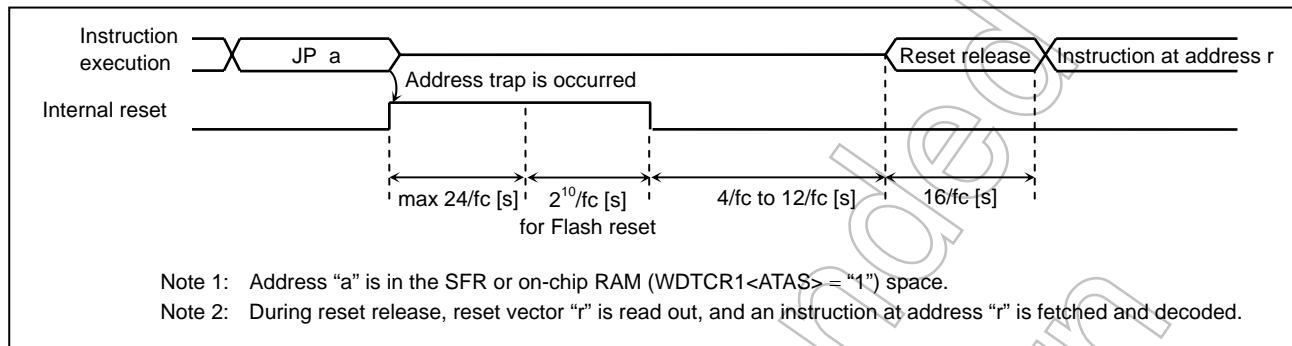
Figure 1.6.2 Address-trap-reset

Note: The operating mode under address trapped is alternative of reset or interrupt. Address trap or no address trap can be selected by WDTCR1<ATAS> for the internal RAM.

1.6.3 Watchdog Timer Reset

Refer to Section 2.4 "Watchdog Timer".

1.6.4 System-clock-reset

If the condition as follows is detected, the system clock reset occurs automatically to prevent dead lock of the CPU (The oscillation is continued without stopping).

- In case of clearing SYSCR2<XEN> and SYSCR2<XTEN> simultaneously to "0".
- In case of clearing SYSCR2<XEN> to "0", when the SYSCR2<SYSCK> is "0".
- In case of clearing SYSCR2<XTEN> to "0", when the SYSCR2<SYSCK> is "1".

When the system clock reset is generated, the flash reset is also generated. Therefore, the maximum reset period is $24/fc$ [s] $+ 2^{10}/fc$ [s] (65.5 μs at 16.0 MHz).

2. On-chip Peripherals Functions

2.1 Special Function Register (SFR)

The TMP86FP24 adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR). The SFR is mapped on address 0000H to 003FH, DBR is mapped on address 1F80H to 1FFFH.

Figure 2.1.1 to Figure 2.1.2 indicate the special function register (SFR) and data buffer register (DBR) for TMP86FP24.

Address	Read	Write	Address	Read	Write
0000H	P0DR (P0 port output latch)		0020H	TC1DRAL (Timer register 1A)	
01	P1DR (P1 port output latch)		21	TC1DRAH (Timer register 1A)	
02	P2DR (P2 port output latch)		22	TC1DRBL (Timer register 1B)	
03	P3DR (P3 port output latch)		23	TC1DRBH (Timer register 1B)	
04	P4DR (P4 port output latch)		24	TC2DRL (Timer register 2)	
05	P5DR (P5 port output latch)		25	TC2DRH (Timer register 2)	
06	P6DR (P6 port output latch)		26	ADCDR2 (AD result register 2)	−
07	Reserved		27	ADCDR1 (AD result register 1)	−
08	Reserved		28	P6CR2 (P6 port input control)	
09	P9DR (P9 port output latch)		29	TC3SEL (Timer counter 3 input control)	
0A	P0OUTCR (P0 port output control)		2A	P3OUTCR (P3 port output control)	
0B	P1OUTCR (P1 port output control)		2B	P4LCR (P4 segment output control)	
0C	P6CR1 (P6 port input/output control)		2C	EIRE (Interrupt enable register)	
0D	P5OUTCR (P5 port output control)		2D	Reserved	
0E	ADCCR1 (AD control register 1)		2E	ILE (Interrupt latch)	
0F	ADCCR2 (AD control register 2)		2F	Reserved	
10	TC3DRA (Timer register 3A)		30	Reserved	
11	TC3DRB (Timer register 3B)	−	31	Reserved	
12	TC3CR (Timer counter 3 control)		32	Reserved	
13	TC2CR (Timer counter 2 control)		33	Reserved	
14	TC5CR (Timer counter 5 control)		34	−	WDTCR1 (Watchdog timer control)
15	TC5DR (Timer register 5)		35	−	WDTCR2 (Watchdog timer control)
16	SIO1CR1 (SIO1 control 1)		36	TBTCR (TBT/TG/DVO control)	
17	SIO1CR2 (SIO1 control 2)		37	EINTCR (External interrupt control)	
18	SIO1SR (SIO1 status)	−	38	SYSCR1 (System control 1)	
19	SIO1BUF (SIO1 data buffer)		39	SYSCR2 (System control 2)	
1A	SIO2CR1 (SIO2 control 1)		3A	EIRL (Interrupt enable register)	
1B	SIO2CR2 (SIO2 control 2)		3B	EIRH (Interrupt enable register)	
1C	SIO2SR (SIO2 status)	−	3C	ILL (Interrupt latch)	
1D	SIO2BUF (SIO2 data buffer)		3D	ILH (Interrupt latch)	
1E	P4PDCR (P4 port pull-down control)		3E	Reserved	
1F	TC1CR (Timer counter 1 control)		3F	PSW (Program status word)	

Note 1: Do not access reserved areas by the program.
Note 2: −: Cannot be accessed.
Note 3: Write-only registers and interrupt latches cannot use the read-modify-write instructions (Bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

Figure 2.1.1 The Special Function Register (SFR) for TMP86FP24 (1/2)

Address

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1F80H		SEG1				SEG0		
81		SEG3				SEG2		
82		SEG5				SEG4		
83		SEG7				SEG6		
84		SEG9				SEG8		
85		SEG11				SEG10		
86		SEG13				SEG12		
87		SEG15				SEG14		
88		SEG17				SEG16		
89		SEG19				SEG18		
8A		SEG21				SEG20		
8B		SEG23				SEG22		

Address

	Read	Write
1FC0H	ROMCCR (ROM correction control register)	
C1	RCAD0L (ROM correction BANK 0 address low)	
C2	RCAD0H (ROM correction BANK 0 address high)	
C3	RCDT0L (ROM correction BANK 0 data low)	
C4	RCDT0H (ROM correction BANK 0 data high)	
C5	RCAD1L (ROM correction BANK 1 address low)	
C6	RCAD1H (ROM correction BANK 1 address high)	
C7	RCDT1L (ROM correction BANK 1 data low)	
C8	RCDT1H (ROM correction BANK 1 data high)	
C9	RCAD2L (ROM correction BANK 2 address low)	
CA	RCAD2H (ROM correction BANK 2 address high)	
CB	RCDT2L (ROM correction BANK 2 data low)	
CC	RCDT2H (ROM correction BANK 2 data high)	
CD	RCAD3L (ROM correction BANK 3 address low)	
CE	RCAD3H (ROM correction BANK 3 address high)	
CF	RCDT3L (ROM correction BANK 3 data low)	
D0	RCDT3H (ROM correction BANK 3 data high)	
D1	Reserved	
⋮		
DC	Reserved	
DD	UARTSR (UART status)	UARTCR1 (UART control 1)
DE	−	UARTCR2 (UART control 2)
DF	RDBUF (UART received data buffer)	TDBUF (UART transmit data buffer)
E0	EEPCR (FLASH control)	
E1	EEPSR (FLASH status)	−
E2	EEPEVA (FLASH write emulation time control)	
E3	LCDCR (LCD control)	
E4	P2OUTCR (P2 port output control)	
E5	Reserved	
E6	Reserved	
E7	Reserved	
E8	P9PDCR (P9 port pull down control)	
E9	P9LCR (P9 segment output control)	
EA	Reserved	
EB	Reserved	
EC	Reserved	
ED	P0PRD (P0 terminal input)	−
EE	P1PRD (P1 terminal input)	−
EF	P2PRD (P2 terminal input)	−
F0	P3PRD (P3 terminal input)	−
F1	P4PRD (P4 terminal input)	−
F2	P5PRD (P5 terminal input)	−
F3	P9PRD (P9 terminal input)	−
F4	Reserved	
F5	Reserved	
F6	Reserved	
F7	Reserved	
F8	Reserved	
F9	Reserved	
FA	Reserved	
FB	Reserved	
FC	Reserved	
FD	Reserved	
FE	STOPCR (Key-on wakeup control)	
FF	Reserved	

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

Note 2: −: Cannot be accessed.

Note 3: Write-only registers and interrupt latches cannot use the read-modify-write instructions (Bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

Figure 2.1.2 The Special Function Register (SFR) for TMP86FP24 (2/2)

2.2   I/O Ports

The TMP86FP24 has 8 parallel input/output ports (54 pins) as follows.

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	External interrupt input, serial interface input/output, UART input/output and timer/counter input .
Port P1	6-bit I/O port	External interrupt input, serial interface input/output and timer/counter input/output.
Port P2	4-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	8-bit I/O port	
Port P4	8-bit I/O port	Segment output and STOP mode release signal input.
Port P5	4-bit I/O port	Divider output and timer/counter output.
Port P6	8-bit I/O port	Analog input and STOP mode release signal input.
Port P9	8-bit I/O port	Segment output.

Each output port contains a latch, which holds the output data. All input ports do not have latches, so the external input data should be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 2.2.1 shows input/output timing examples.

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

Output data changes in the S2 state of the write cycle during execution of the instruction which writes to an I/O port.
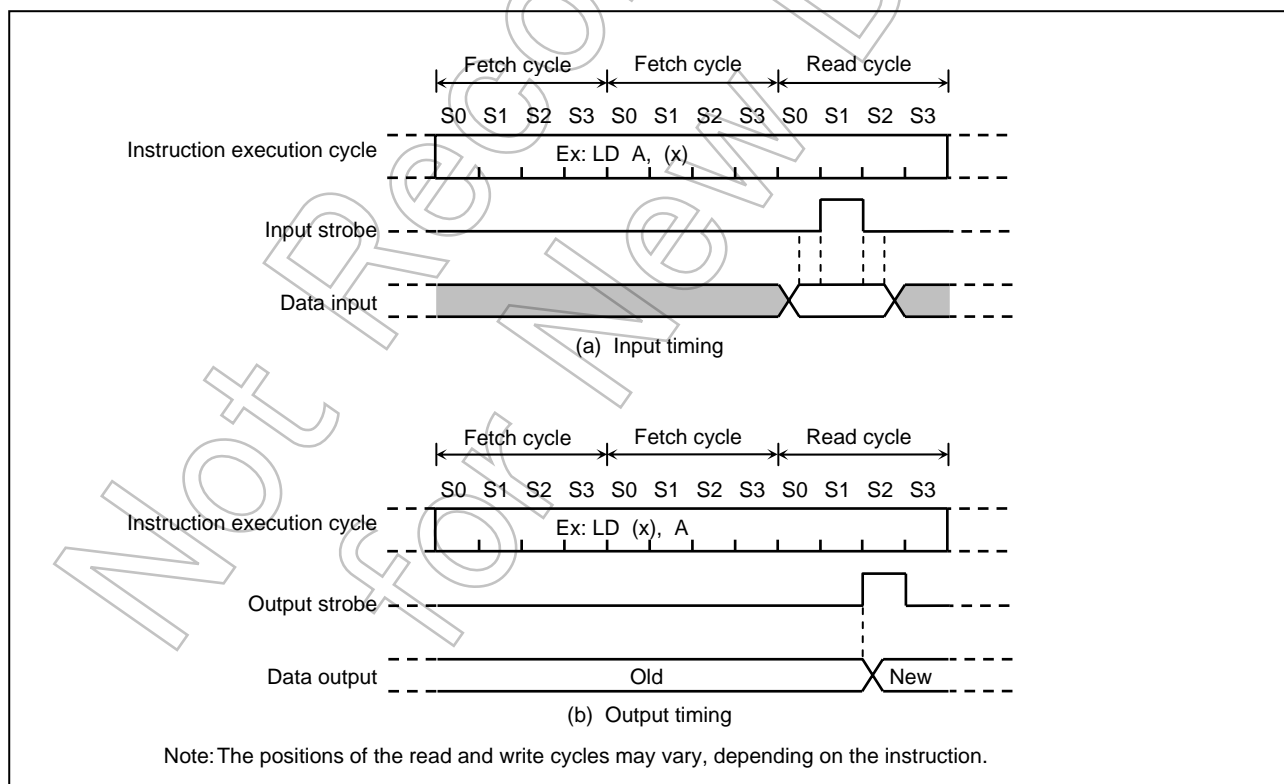
Figure 2.2.1 Input/Output Timing (Example)

2.2.1   Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port which is also used as an external interrupt input, serial interface input/output, timer/counter input and UART input/output. It can be selected whether output circuit of P0 port is CMOS output or a sink open drain individually, by setting the output circuit control (P0OUTCR). When a corresponding bit of P0OUTCR is cleared to "0", the output circuit is selected to a sink open drain and when a corresponding bit of P0OUTCR is set to "1", the output circuit is selected to a CMOS output.

When used as an input port or a secondary function input (External interrupt input, serial interface input, timer/counter input or UART input), the respective output latch (P0DR) should be set to "1" and its corresponding P0OUTCR bit should be cleared to "0".

When used as a secondary function output (Serial interface output or UART output), the respective P0DR should be set to "1".

During reset, the P0DR is initialized to "1" and P0OUTCR is initialized to "0".

P0 port output latch (P0DR) and P0 port terminal input (P0PRD) are located on their respective address. When read the output latch data, the P0DR should be read and when read the terminal input data, the P0PRD register should be read.
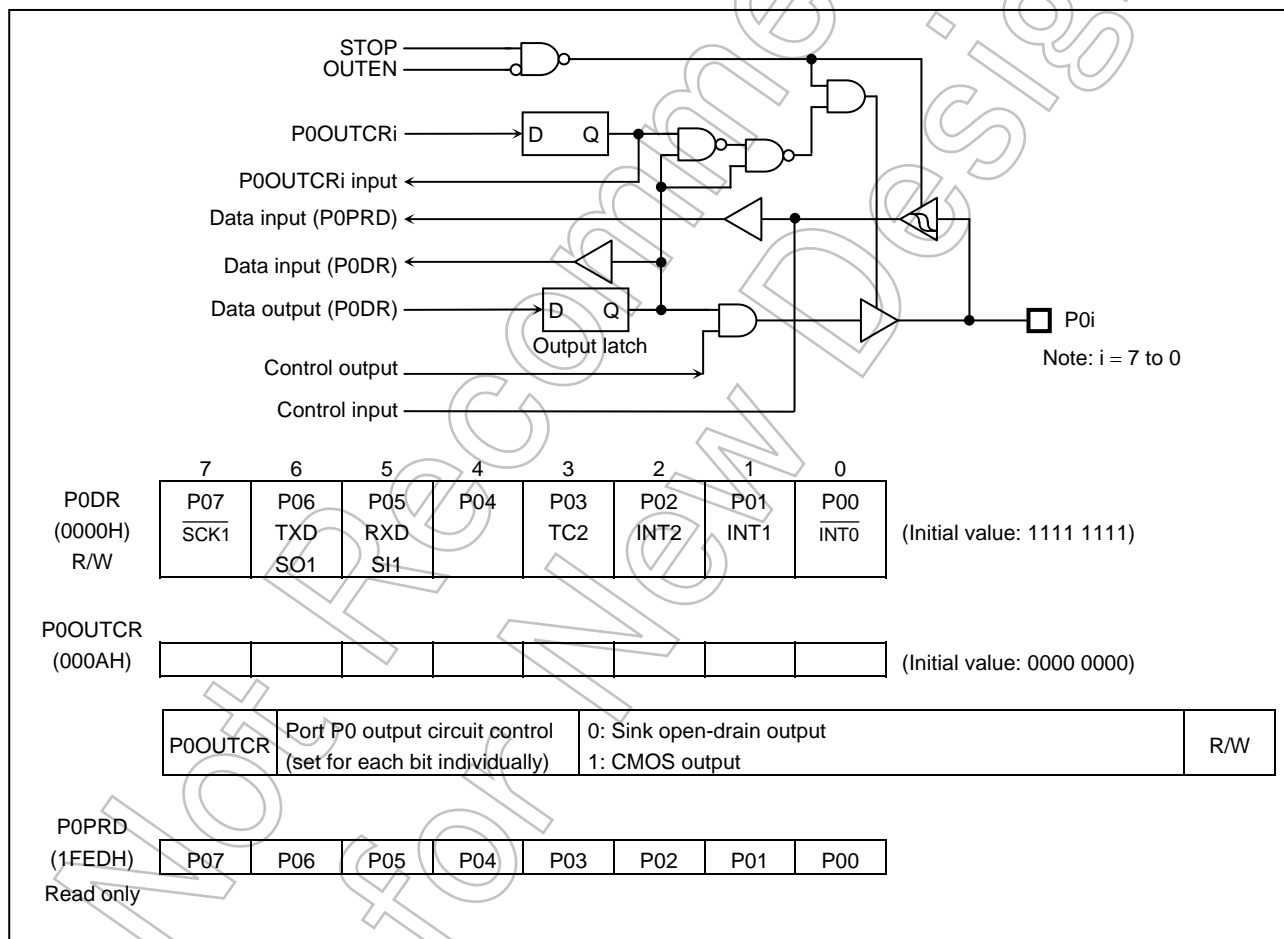
Figure 2.2.2 Port 0

2.2.2   Port P1 (P15 to P10)

Port P1 is a 6-bit input/output port which is also used as an external interrupt input, serial interface input/output and timer/counter input/output. It can be selected whether output circuit of P1 port is CMOS output or a sink open drain individually, by setting the output circuit control (P1OUTCR). When a corresponding bit of P1OUTCR is cleared to "0", the output circuit is selected to a sink open drain and when a corresponding bit of P1OUTCR is set to "1", the output circuit is selected to a CMOS output.

When used as an input port or a secondary function input (External interrupt input, serial interface input, timer/counter input), the respective output latch (P1DR) should be set to "1" and its corresponding P1OUTCR bit should be cleared to "0".

When used as a secondary function output (Serial interface output or timer/counter output), the respective P1DR should be set to "1".

During reset, the P1DR is initialized to "1" and P1OUTCR is initialized to "0".

P1 port output latch (P1DR) and P1 port terminal input (P1PRD) are located on their respective address. When read the output latch data, the P1DR should be read and when read the terminal input data, the P1PRD register should be read.

If a read instruction is executed for P1DR, P1OUTCR and P1PRD, read data of bits 7 and 6 are unstable.
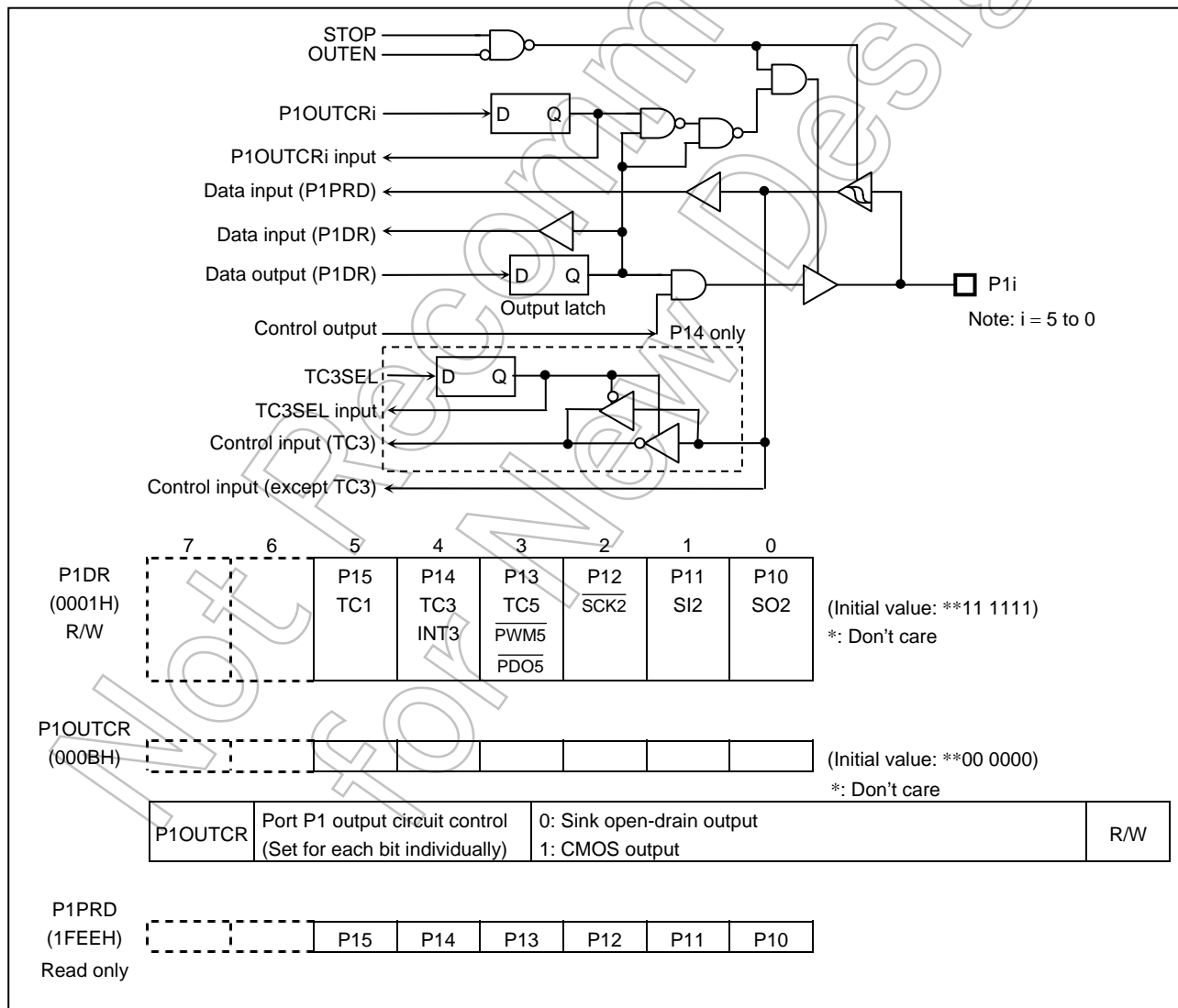
Figure 2.2.3 Port 1

The TC3 input can have its input waveform phase-inverted by using the TC3SEL register. For details, refer to 2.8 "8-Bit Timer/Counter 3". If a read instruction is executed for TC3SEL, read data of bits 7 to 1 are unstable.

TC3SEL
(0029H)

| | TC3INV | (Initial value: **** ***0) |

*: Don't care

| TC3INV | TC3 input control | 0: Normal input
1: Inverted input | R/W |
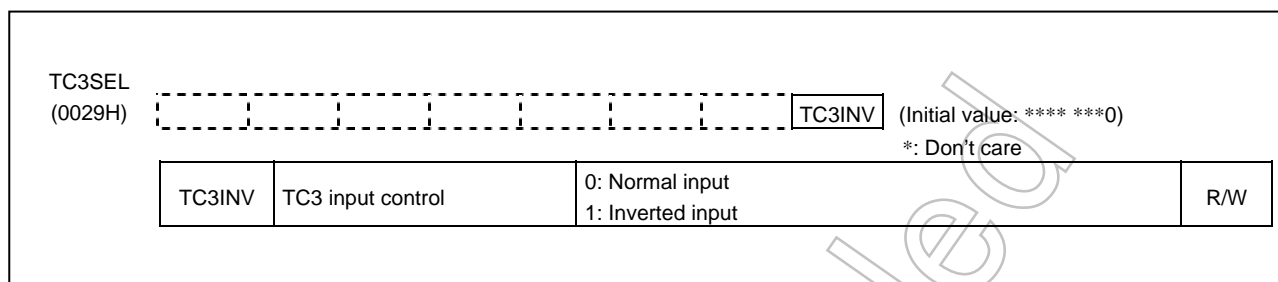
Figure 2.2.4 TC3 Input Control

2.2.3   Port P2 (P23 to P20)

Port P2 is a 4-bit input/output port. It is also used as an external interrupt, a STOP mode release signal input, and low-frequency crystal oscillator connection pins. It can be selected whether output circuit of P2 port is CMOS (P21 and P22 have a pull-up resistor) output or a sink open drain individually, by setting the output circuit control (P2OUTCR). When a corresponding bit of P2OUTCR is cleared to "0", the output circuit is selected to a sink open drain and when a corresponding bit of P2OUTCR is set to "1", the output circuit is selected to a CMOS output. (In case of P21 and P22, the pull-up resistor is connected.)

When used as an input port or an external interrupt input, the respective output latch (P2DR) should be set to "1".

During reset, the P2DR is initialized to "1" and P2OUTCR is initialized to "0".

A low-frequency crystal oscillator (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 it is used as an output port, the interrupt latch is set on the falling edge of the output pulse.

P2 port output latch (P2DR) and P2 port terminal input (P2PRD) are located on their respective address. When read the output latch data, the P2DR should be read and when read the terminal input data, the P2PRD register should be read.

If a read instruction is executed for port P2DR, P2OUTCR and P2PRD, read data of bits 7 to 4 are unstable.


Note: When XTEN<SYSCR2> is set to "1", P21 and P22 become a high impedance state.
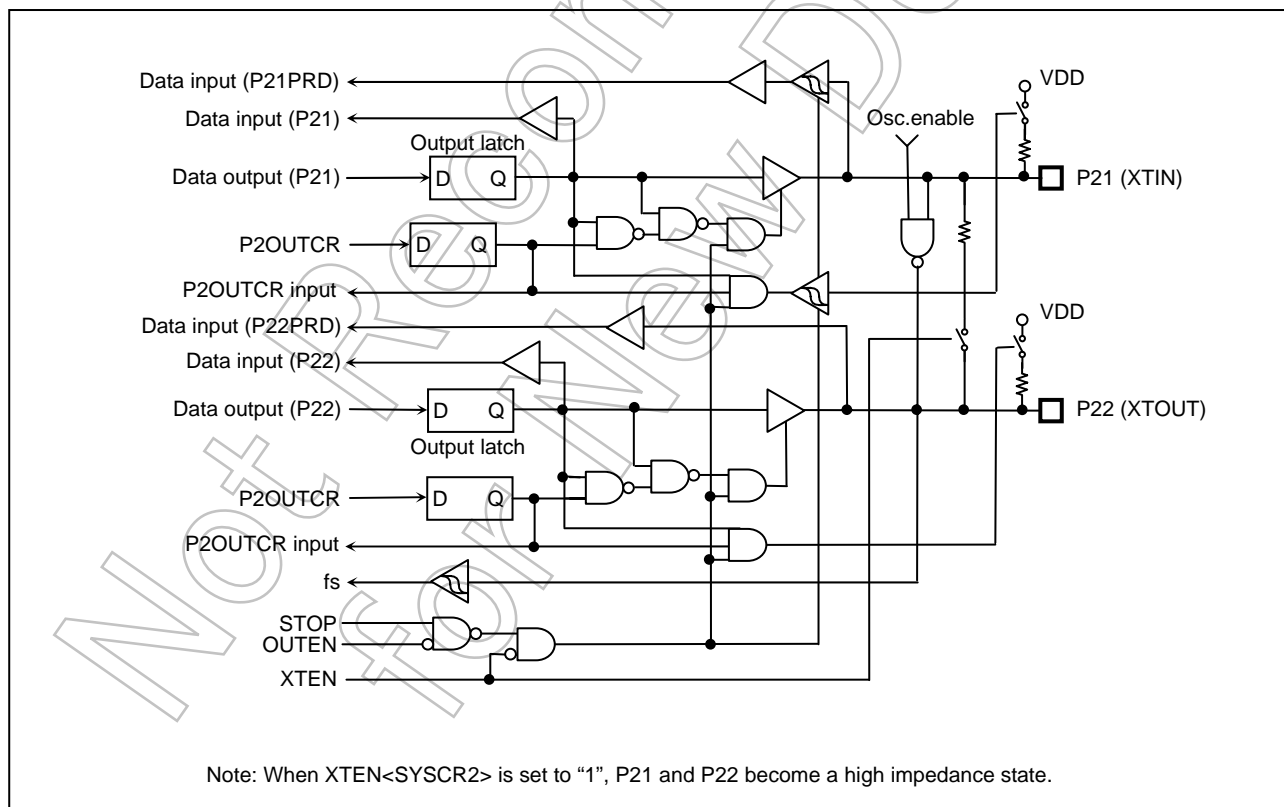
Figure 2.2.5 Port 2 (P21 and P22)
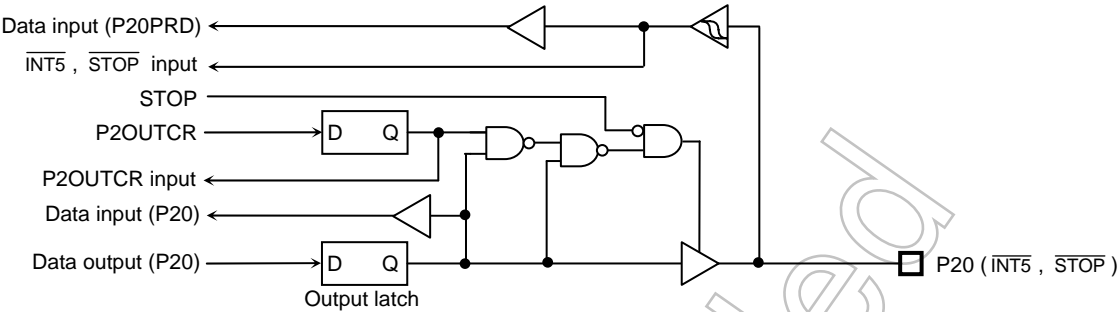
Note: Port P20 is used as $\overline{\text{STOP}}$ pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z state.

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W					P23	P22 XTOUT	P21 XTIN	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	(Initial value: **** 1111) *: Don't care

P2OUTCR (1FE4H)									(Initial value: **** 0000) *: Don't care
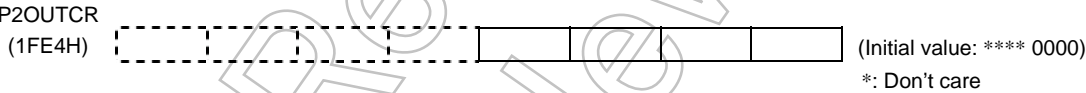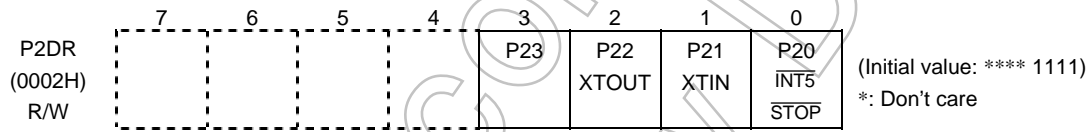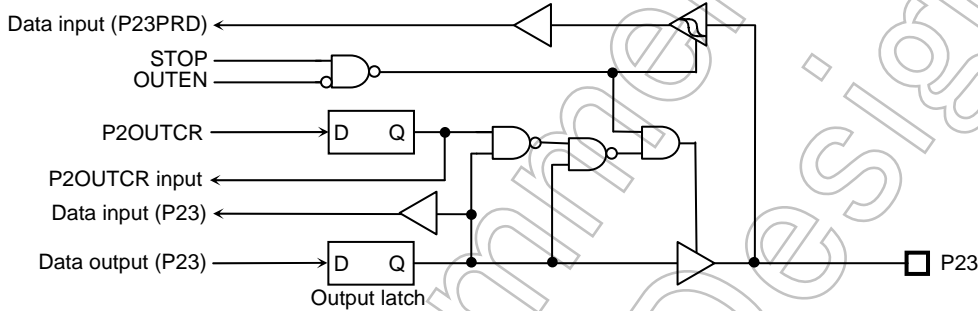
P2OUTCR	Port P2 output circuit control (Set for each bit individually)	0: Sink open-drain output 1: CMOS output (P20, P23 ports) CMOS output with pull-up resistor (P21, P22 ports)	R/W

P2PRD (1FEFH) Read only					P23	P22	P21	P20

Figure 2.2.6 Port 2 (P20 and P23)

2.2.4   Port P3 (P37 to P30)

Port P3 is an 8-bit input/output port. It can be selected whether output circuit of P3 port is CMOS output or a sink open drain individually, by setting P3OUTCR. (N-ch high current output) When a corresponding bit of P3OUTCR is cleared to "0", the output circuit is selected to a sink open drain and when a corresponding bit of P3OUTCR is set to "1", the output circuit is selected to a CMOS output.

When used as an input port, the respective output latch (P3DR) should be set to "1" and its corresponding P3OUTCR bit should be cleared to "0".

During reset, the P3DR is initialized to "1", and the P3OUTCR is initialized to "0".

P3 port output latch (P3DR) and P3 port terminal input (P3PRD) are located on their respective address. When read the output latch data, the P3DR should be read and when read the terminal input data, the P3PRD register should be read.
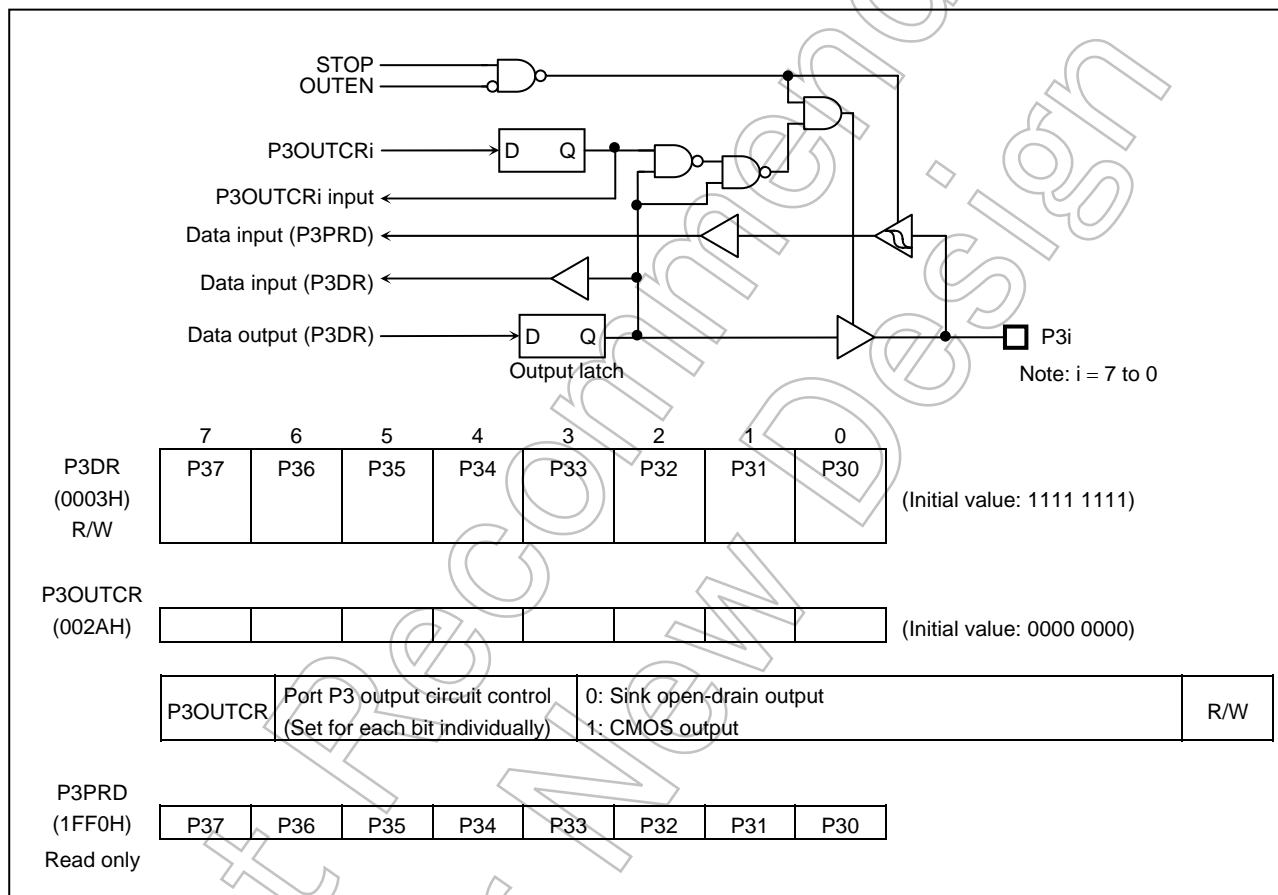
Figure 2.2.7 Port 3

2.2.5   Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port which is also used as a segment pins of LCD or key-on wakeup input. P4 port has pull-down resistors that programming control is possible. Pull-down control is specified by control register (P4PDCR). To connect pull-down resistor, P4PDCR should be set to "1".

When used as an input port, the respective output latch (P4DR) should be set to "1" and the respective P4LCR bit should be cleared to "0".

When used as an output port, the respective P4LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P4LCR should be set to "1".

When P40 is used as a key-on wakeup port, STOP4EN<STOPCR> should be set to "1".

During reset, the P4DR is initialized to "1" and the P4LCR and the P4PDCR are initialized to "1".

P4 port output latch (P4DR) and P4 port terminal input (P4PRD) are located on their respective address. When read the output latch data, the P4DR should be read and when read the terminal input data, the P4PRD register should be read.

Table 2.2.1 and Table 2.2.2 show a P4 state.

Table 2.2.1   P40 State

STOP4EN	P4LCR	EDSP	P4PDCR	P4DR	P4PRD read	Output	Remark
0	0	*	0	0	Terminal input	Low	I/O
0	0	*	0	1	Terminal input	High-Z	I/O
0	0	*	1	0	Terminal input	Low	I/O (Pull down)
0	0	*	1	1	Terminal input	Low	I/O (Pull down)
0	1	0	*	*	"0"	Low	LCD Blanking
0	1	1	*	*	"0"	Segment	LCD
1	*	*	*	*	Terminal input	High-Z	Key-on wakeup

Note 1:   *: Don't care

Note 2:   STOP4EN is bit3 in STOPCR.

Note 3:   EDSP is bit7 in LCDCR.

Table 2.2.2   P47 to P41 State

P4LCR	EDSP	P4PDCR	P4DR	P4PRD read	Output	Remark
0	*	0	0	Terminal input	Low	I/O
0	*	0	1	Terminal input	High-Z	I/O
0	*	1	0	Terminal input	Low	I/O (Pull down)
0	*	1	1	Terminal input	Low	I/O (Pull down)
1	0	*	*	"0"	Low	LCD Blanking
1	1	*	*	"0"	Segment	LCD

Note 1:   *: Don't care

Note 2:   EDSP is bit7 in LCDCR.

P4DR (0004H) R/W

7	6	5	4	3	2	1	0
P47 SEG16	P46 SEG17	P45 SEG18	P44 SEG19	P43 SEG20	P42 SEG21	P41 SEG22	P40 SEG23 STOP4

(Initial value: 1111 1111)

P4LCR (002BH)

(Initial value: 0000 0000)

P4LCR	Port P4/segment output control (Set for each bit individually)	0: P4 input/output port 1: LCD segment output	R/W

P4PDCR (001EH)

(Initial value: 0000 0000)

P4PDCR	Port P4 pull-down control (Set for each bit individually)	0: Pull-down disable 1: Pull-down enable	R/W

P4PRD (1FF1H) Read only

P47	P46	P45	P44	P43	P42	P41	P40
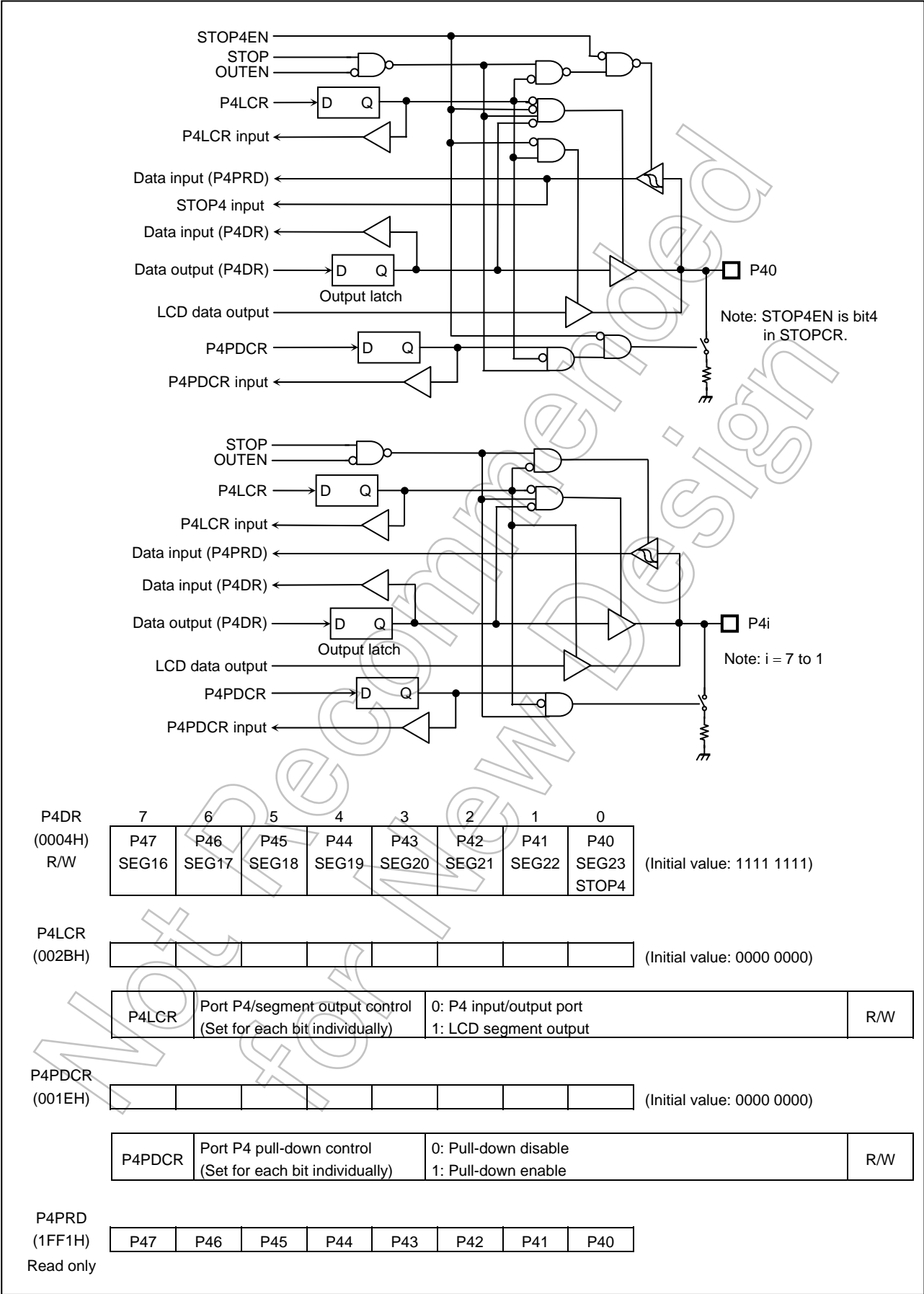
Figure 2.2.8 Port 4

2.2.6   Port P5 (P53 to P50)

Port P5 is an 4-bit input/output port which is also used as a timer/counter output and divider output. (N-ch high current output) It can be selected whether output circuit of P5 port is CMOS output or a sink open drain individually, by setting the output circuit control (P5OUTCR). When a corresponding bit of P5OUTCR is cleared to "0", the output circuit is selected to a sink open drain and when a corresponding bit of P5OUTCR is set to "1", the output circuit is selected to a CMOS output.

When used as an input port, the respective output latch (P5DR) should be set to "1" and its corresponding P5OUTCR bit should be cleared to "0".

When used as a secondary function output (Timer/counter output or divider output), the respective P5DR should be set to "1".

During reset, the P5DR is initialized to "1" and P5OUTCR is initialized to "0".

P5 port output latch (P5DR) and P5 port terminal input (P5PRD) are located on their respective address. When read the output latch data, the P5DR should be read and when read the terminal input data, the P5PRD register should be read.

If a read instruction is executed for P5DR, P5OUTCR and P5PRD, read data of bits 7 to 4 are unstable.

	7	6	5	4	3	2	1	0	
P5DR (0005H) R/W					P53	P52	P51 / $\overline{DVO}$	P50 / $\overline{PPG}$	(Initial value: **** 1111) *: Don't care

Figure 2.2.9 Port 5

P5OUTCR
(000DH)

| | | | | | | (Initial value: **** 0000) |

*: Don't care

| P5OUTCR | Port P5 output circuit control (Set for each bit individually) | 0: Sink open-drain output 1: CMOS output | R/W |

P5PRD
(1FF2H)
Read only

| | | | | P53 | P52 | P51 | P50 |
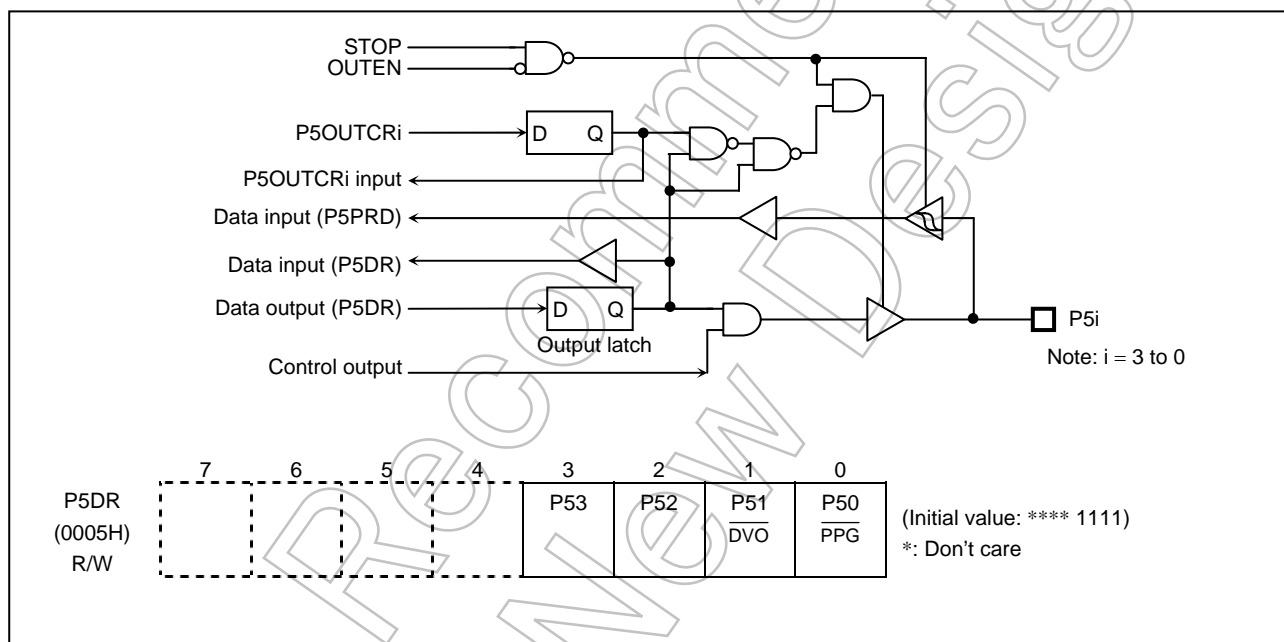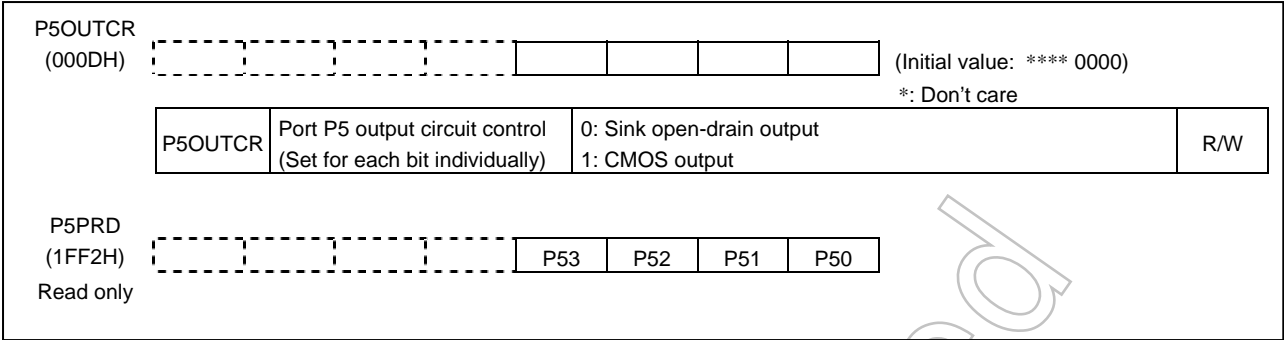
Figure 2.2.10 P5OUTCR and P5PRD

2.2.7　Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or an output in one-bit unit. Port P6 is also used as an analog input and key-on wakeup input. Input/output mode is specified by the P6 control register (P6CR1). P6 port input is controlled by the input control register (P6CR2).

When used as an output port, respective P6CR1 should be set to "1".

When used as an input port, respective P6CR1 should be cleared to "0" and respective P6CR2 should be set to "1".

When used as an analog input, respective P6CR2 should be cleared to "0" after respective P6CR1 is cleared to "0".

When used as a key-on wakeup input, respective STOPkEN<STOPCR> should be set to "1" (k = 3 to 0).

During reset, the P6CR1 and P6DR are initialized to "0", and the P6CR2 is initialized to "1". Table 2.2.3 and Table 2.2.4 show a P6 state.

Table 2.2.3　P63 to P60 State

P6CR1	P6CR2	P6DR	P6DR read	Output	Remark
0	0	*	"0"	High-Z	–
0	1	*	Terminal input	High-Z	Input mode
1	*	0	"0" (Output latch)	Low	Output mode
1	*	1	"1" (Output latch)	High	Output mode

*: Don't care

Table 2.2.4　P67 to P64 State

STOPkEN	P6CR1	P6CR2	P6DR	P6DR read	Output	Remark
0	0	0	*	"0"	High-Z	–
0	0	1	*	Terminal input	High-Z	Input mode
0	1	*	0	"0" (Output latch)	Low	Output mode
0	1	*	1	"1" (Output latch)	High	Output mode
1	*	*	*	Terminal input	High-Z	Key-on wakeup

Note 1:　*: Don't care

Note 2:　STOPkEN is bit7 to bit4 in STOPCR.
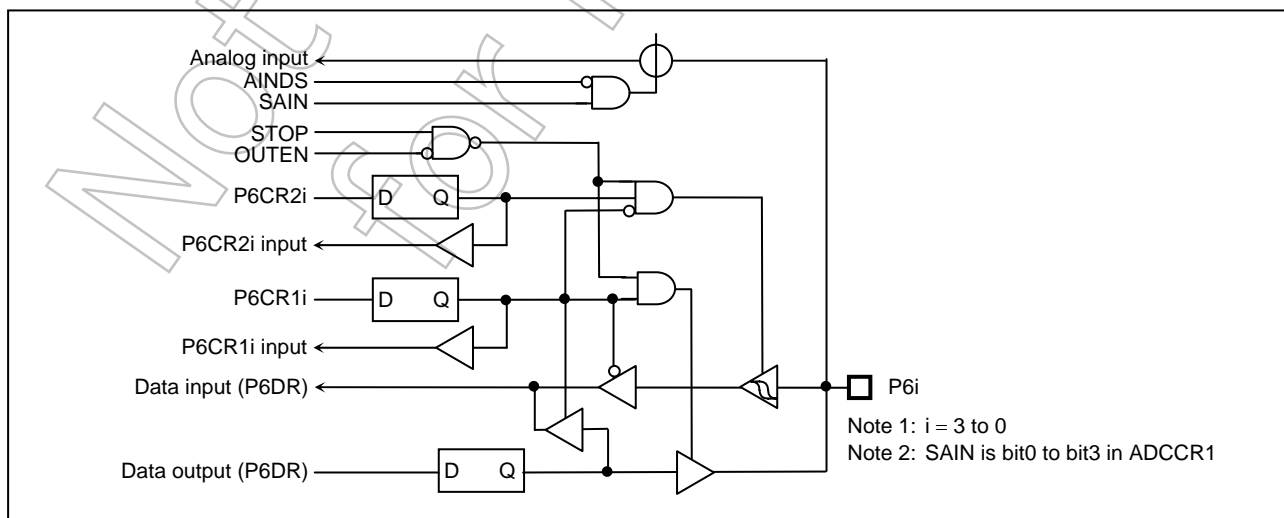
Figure 2.2.11　Port 6 (P63 to P60)
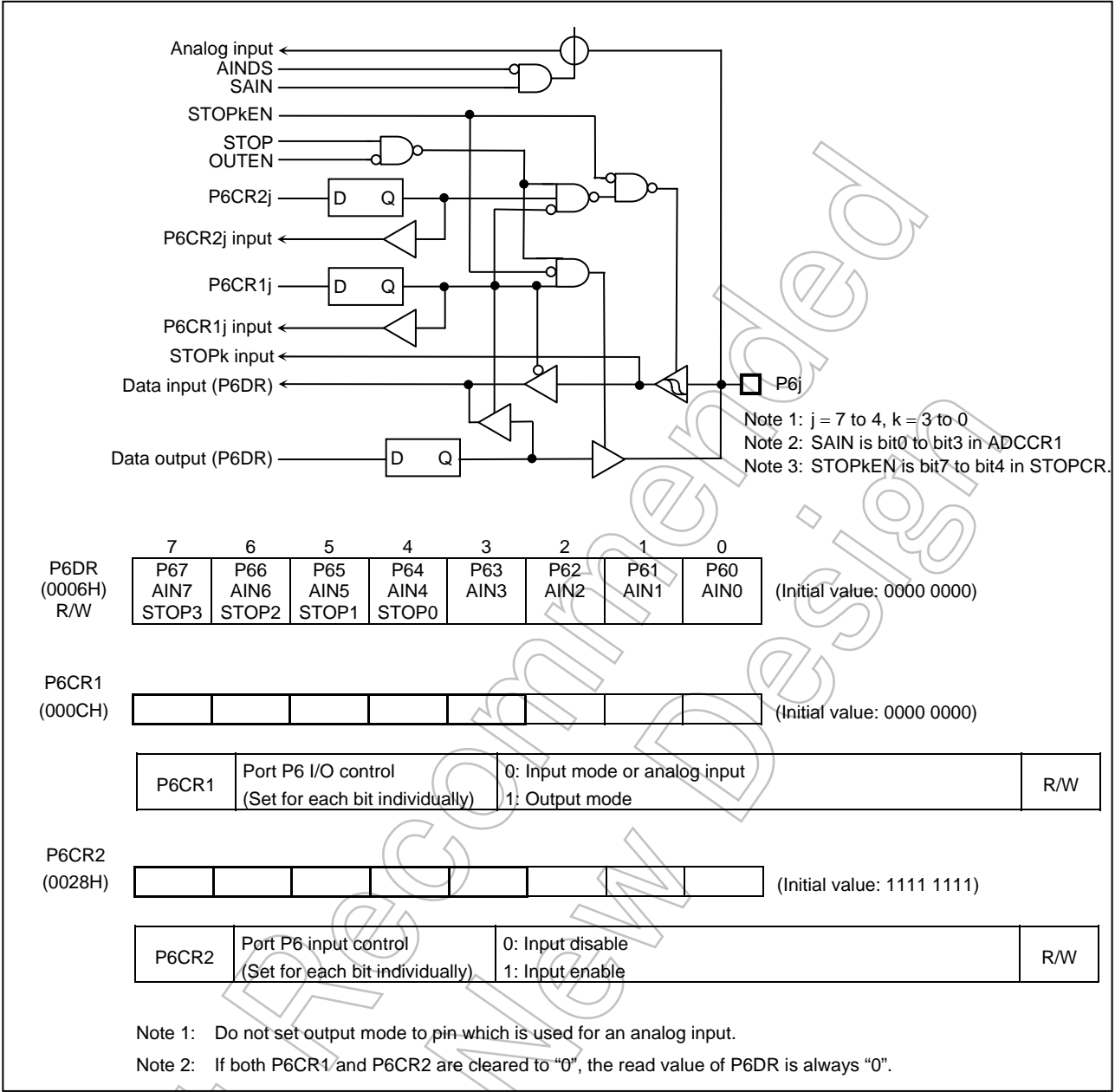
Figure 2.2.12 Port 6 (P67 to P64), P6DR, P6CR1 and P6CR2

2.2.8 Port P9 (P97 to P90)

Port P9 is an 8-bit input/output port which is also used as a segment pins of LCD. P9 port has pull-down resistors that programming control is possible. Pull-down control is specified by control register (P9PDCR). To connect pull-down resistor, P9PDCR should be set to "1".

When used as an input port, the respective output latch (P9DR) should be set to "1" and the respective P9LCR bit should be cleared to "0".

When used as an output port, the respective P9LCR bit should be cleared to "0".

When used as a segment pins of LCD, the respective bit of P9LCR should be set to "1".

During reset, the P9DR is initialized to "1" and the P9LCR and the P9PDCR are initialized to "0".

P9 port output latch (P9DR) and P9 port terminal input (P9PRD) are located on their respective address. When read the output latch data, the P9DR should be read and when read the terminal input data, the P9PRD register should be read.

Table 2.2.5 shows a P9 state.

Table 2.2.5 P97 to P90 State

P9LCR	EDSP	P9PDCR	P9DR	P9PRD read	Output	Remark
0	*	0	0	Terminal input	Low	I/O
0	*	0	1	Terminal input	High-Z	I/O
0	*	1	0	Terminal input	Low	I/O (Pull down)
0	*	1	1	Terminal input	Low	I/O (Pull down)
1	0	*	*	"0"	Low	LCD Blanking
1	1	*	*	"0"	Segment	LCD

Note 1:   *: Don't care

Note 2:   EDSP is bit7 in LCDCR.

Figure 2.2.13 Port 9

P9DR	7	6	5	4	3	2	1	0	
(0009H)	P97	P96	P95	P94	P93	P92	P91	P90	(Initial value: 1111 1111)
R/W	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	

P9LCR
(1FE9H)   (Initial value: 0000 0000)

P9LCR	Port P9/segment output control (Set for each bit individually)	0: P9 input/output port 1: LCD segment output	R/W

P9PDCR
(1FE8H)   (Initial value: 0000 0000)

P9PDCR	Port P9 pull-down control (Set for each bit individually)	0: Pull-down disable 1: Pull-down enable	R/W

P9PRD	P97	P96	P95	P94	P93	P92	P91	P90
(1FF3H) Read only								

Figure 2.2.14 P9DR, P9LCR, P9PDCR and P9PRD

2.2.9 $\overline{\text{WAKE}}$ Pin

The TMP86FP24 has the function of outputting a monitor signal to the outside upon release of STOP mode by an external interrupt signal input. This pin is assigned as the dedicated output pin and it has N-ch open-drain form. This function enables the real time notification of the start/release of STOP mode timing to the outside. Therefore, it is effective for the system which requires the stop control against a peripheral device connected to the microcontroller.
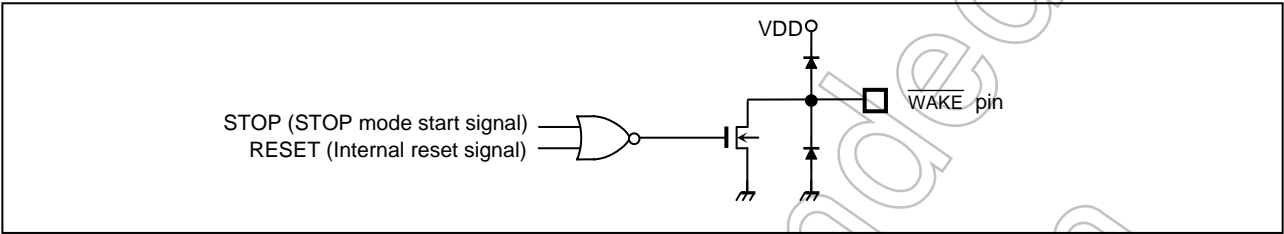
Figure 2.2.15 $\overline{\text{WAKE}}$ Pin

While the microcontroller is operating, "L" level is output from the $\overline{\text{WAKE}}$ pin. When STOP mode is initiated and the operation of CPU is stopped, the $\overline{\text{WAKE}}$ pin becomes the high impedance state. When STOP mode is released by an external interrupt signal input, the $\overline{\text{WAKE}}$ pin becomes "L" level. Therefore, during warm-up, the $\overline{\text{WAKE}}$ pin is "L" level. During reset, the $\overline{\text{WAKE}}$ pin is in the high impedance state. Table 2.2.6 shows the $\overline{\text{WAKE}}$ pin state and Figure 2.2.16 shows the $\overline{\text{WAKE}}$ pin output timing.

Table 2.2.6 $\overline{\text{WAKE}}$ Pin State

State	$\overline{\text{WAKE}}$ pin output
During reset	High impedance
During operation (except in STOP mode)	"L"
During STOP mode	High impedance
During warm-up	"L"
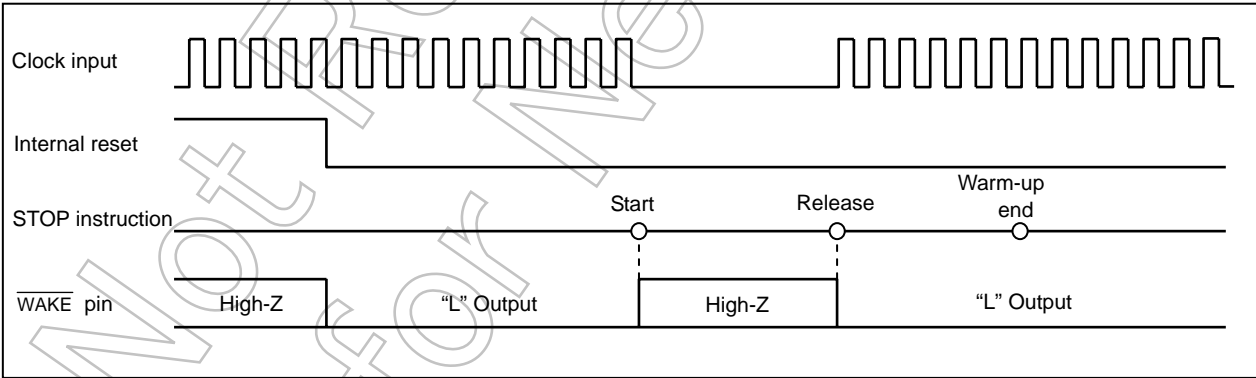
Figure 2.2.16 $\overline{\text{WAKE}}$ Pin Output Timing

2.3   Time-base-timer (TBT)

The time-base-timer generates time base for key scanning, dynamic displaying, etc. It also provides a time base timer interrupt (INTTBT).

An INTTBT is generated on the first falling edge of source clock (the divider output of the timing generator) after the time base timer has been enabled. The divider is not cleared by the program; therefore, only the first interrupt may be generated ahead of the set interrupt period (Figure 2.3.1 (b)).

The interrupt frequency (TBTCK) must be selected with the time base timer disabled (the interrupt frequency must not be changed with the disable from the enable state). Both frequency selection and enabling can be performed simultaneously.
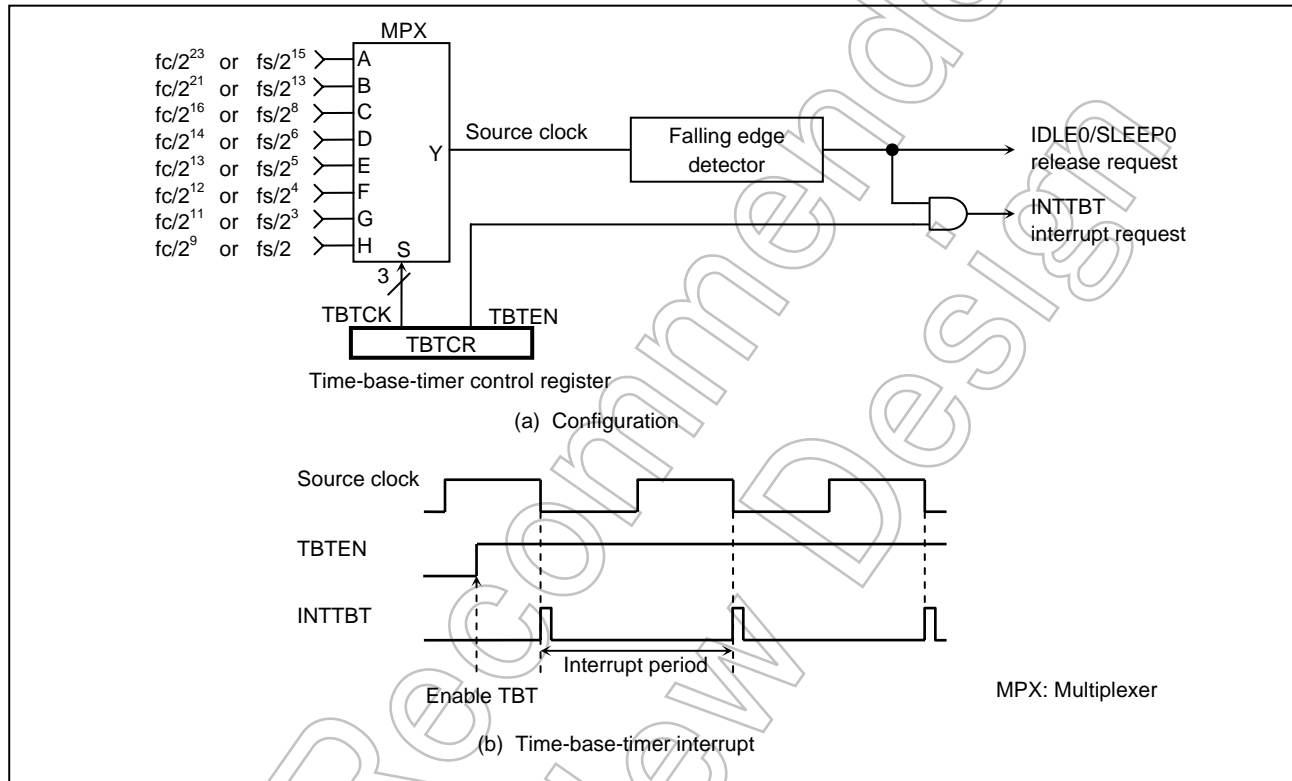
Figure 2.3.1 Time-base-timer

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

```
LD        (TBTCR), 00000010B      ;   TBTCK ← 010
LD        (TBTCR), 00001010B      ;   TBTEN ← 1
DI                                ;   IMF ← 0
SET       (EIRL), 6
```

TBTCR	7	6	5	4	3	2	1	0	
(0036H)	(DVOEN)	(DVOCK)		(DV7CK)	TBTEN		TBTCK		(Initial value: 0000 0000)

TBTEN	Time-base-timer enable/disable	0: Disable 1: Enable					
TBTCK	Time-base-timer interrupt frequency select [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode		R/W
			DV7CK = 0	DV7CK = 1			
		000	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$		
		001	$fc/2^{21}$	$fs/2^{13}$	$fs/2^{13}$		
		010	$fc/2^{16}$	$fs/2^{8}$	–		
		011	$fc/2^{14}$	$fs/2^{6}$	–		
		100	$fc/2^{13}$	$fs/2^{5}$	–		
		101	$fc/2^{12}$	$fs/2^{4}$	–		
		110	$fc/2^{11}$	$fs/2^{3}$	–		
		111	$fc/2^{9}$	$fs/2$	–		

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

Figure 2.3.2 Time-base-timer Control Register

Table 2.3.1 Time-base-timer Interrupt Frequency (Example: fc = 16 MHz, fs = 32.768 kHz)

TBTCK	Time-base-timer Interrupt Frequency [Hz]		
	NORMAL1/2, IDLE1/2 Modes		SLOW, SLEEP Modes
	DV7CK = 0	DV7CK = 1	
000	1.91	1	1
001	7.63	4	4
010	244.14	128	–
011	976.56	512	–
100	1953.13	1024	–
101	3906.25	2048	–
110	7812.5	4096	–
111	31250	16384	–

2.4   Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to rapidly detect the CPU malfunctions such as endless looping caused by noise or the like, or deadlock and resume the CPU to the normal state.

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

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

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

2.4.1   Watchdog Timer Configuration
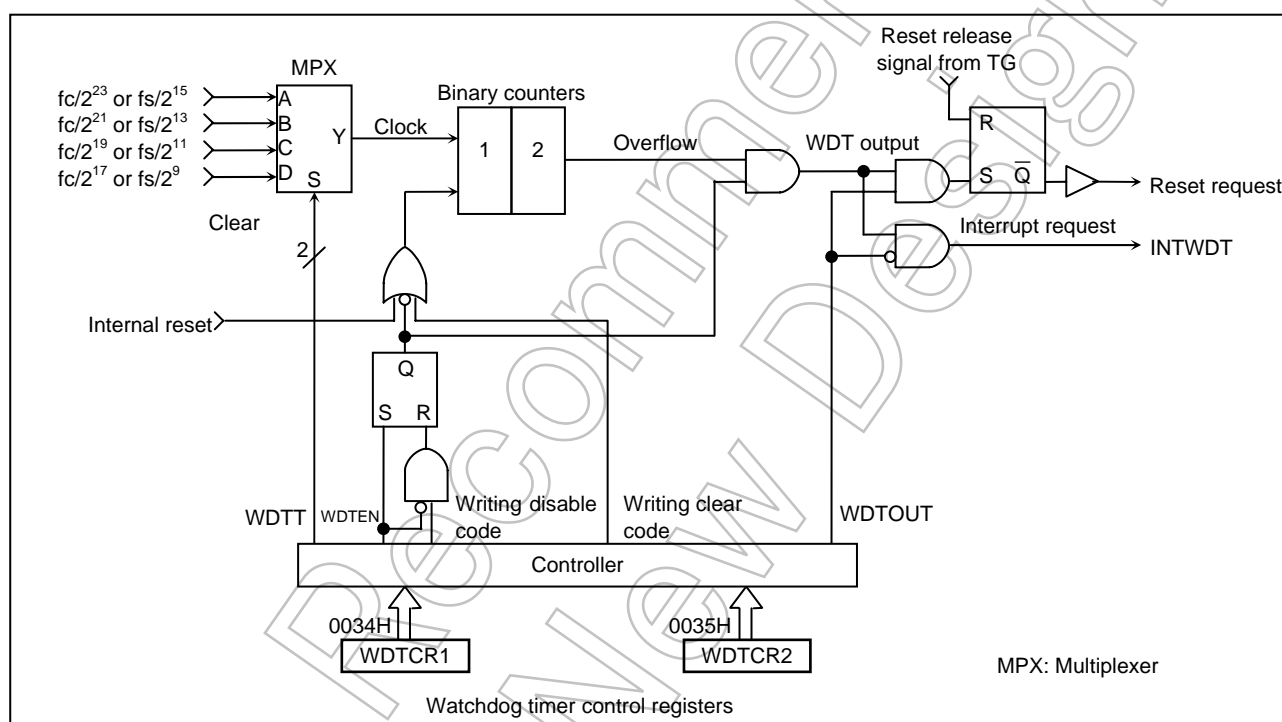
Figure 2.4.1 Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2.4.2 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected as follows.

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

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

If the CPU malfunctions such as endless looping or deadlock occur for any cause, the watchdog timer output will become active at the rising of an overflow from the binary counters unless the binary counters are cleared. At this time, when WDTCR1<WDTOUT> = "1", a reset is generated and the internal hardware is reseted. When WDTCR1<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 the STOP/IDLE mode is released.

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

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

LD	(WDTCR2), 4EH	;	Clears the binary counters.
LD	(WDTCR1), 00001101B	;	WDTT ← 10, WDTOUT ← 1
LD	(WDTCR2), 4EH	;	Clears the binary counters.
			(Always clear immediately before and after changing WDTT.)
LD	(WDTCR2), 4EH	;	Clears the binary counters.
LD	(WDTCR2), 4EH	;	Clears the binary counters.

Within 3/4 of WDT detection time

Within 3/4 of WDT detection time

Watchdog Timer Register 1

WDTCR1 (0034H)

7	6	5	4	3	2	1	0	
		(ATAS)	(ATOUT)	WDTEN		WDTT	WDTOUT	(Initial value: **11 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 [s]		NORMAL1/2 mode		SLOW mode		
			DV7CK = 0	DV7CK = 1			
		00	2^{25}/fc	2^{17}/fs	2^{17}/fs		
		01	2^{23}/fc	2^{15}/fs	2^{15}/fs		
		10	2^{21}/fc	2^{13}/fs	2^{13}/fs		
		11	2^{19}/fc	2^{11}/fs	2^{11}/fs		
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Reset request					

Note 1: WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".

Note 2: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz], *: Don't care

Note 3: WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions.

Note 4: The watchdog timer must be disabled or the counter must be cleared immediately before entering to the STOP mode. When the counter is cleared, the counter must be cleared again immediately after releasing the STOP mode.

Note 5: To disable the watchdog timer, always write "4EH" (Clear code) to WDTCR2 for clearing the binary counter before writing "0" to WDTEN, and then write "B1H" (Disable code) to WDTCR2.

Also, immediately before these procedure, disable the interrupt mater flag (IMF) by DI instruction.

Watchdog Timer Register 2

WDTCR2 (0035H)

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) $D2_H$: Enable assigning address trap area Others: Invalid	Write only

Note 1: The disable code is invalid unless written when WDTCR1<WDTEN> = 0.

Note 2: *: Don't care

Note 3: The binary counter of the watchdog timer must not be cleared by the interrupt task.

Note 4: Write clear code 4EH within 3/4 of the time set in WDTCR1<WDTT>.

Figure 2.4.2 Watchdog Timer Control Registers

(2) Watchdog timer enable

The watchdog timer is enabled by setting WDTCR1<WDTEN> to "1". WDTCR1<WDTEN> is initialized to "1" during reset, so the watchdog timer operates immediately after reset is released.

(3) Watchdog timer disable

To disable the watchdog time, write "4EH" (Clear code) to WDTCR2 for clearing the binary counter before writing "0" to WDTCR1<WDTEN>, and then write "B1H" (Disable code) to WDTCR2. The watchdog timer is not disabled if this procedure is reversed and the disable code is written to WDTCR2 before WDTCR1<WDTEN> is cleared to "0". Also, immediately before these procedure, disable the interrupt master flag (IMF) by DI instruction. During disabling the watchdog timer, the binary counters are cleared to "0".

Example: Disables watchdog timer.

DI		;	IMF ← 0
LD	(WDTCR2), 4EH	;	Clear the binary counter.
LDW	(WDTCR1), 0B101H	;	WDTEN ← 0, WDTCR2 ← Disable code.

Table 2.4.1 Watchdog Timer Detection Time (Example: fc = 16 MHz, fs = 32.768 kHz)

WDTT	Watchdog Timer Detection Time [s]		
	NORMAL1/2 Mode		SLOW Mode
	DV7CK = 0	DV7CK = 1	
00	2.097	4	4
01	524.288 m	1	1
10	131.072 m	250 m	250 m
11	32.768 m	62.5 m	62.5 m

2.4.3 Watchdog Timer Interrupt (INTWDT)

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

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

Example: Watchdog timer interrupt setting up.

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

2.4.4 Watchdog Timer Reset

If the watchdog timer reset request occur, a reset is generated and the internal hardware is reseted. When the Watchdog timer reset is generated, the EEPROM reset is also generated. Therefore, the maximum reset period is $24/fc$ [s] $+ 2^{10}/fc$ [s] (65.5 μs at 16.0 MHz).

Note: The high-frequency clock oscillator also immediately turns on when a watchdog timer reset is generated in SLOW mode. In this case, the reset time may include a certain amount of error if there is any fluctuation of the oscillation frequency at starting the high-frequency clock oscillation. Therefore, the reset time must be considered an approximated value.
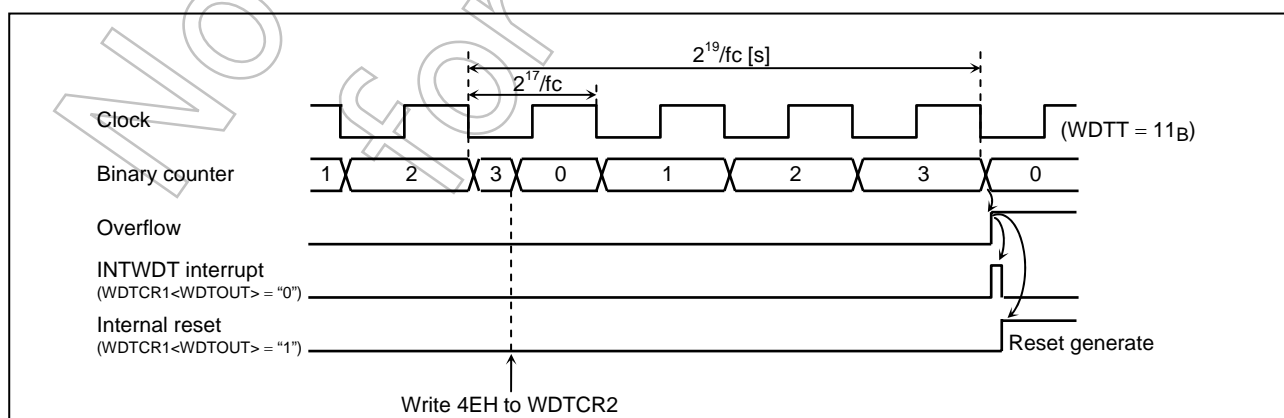

Figure 2.4.3 Watchdog Timer Interrupt/Reset

2.4.5   Address Trap

The watchdog timer control register 1, 2 shares its addresses with the control registers in case of address trap. These control registers for address trap are shown on Figure 2.4.4.

Watchdog Timer Control Register 1

WDTCR1	7	6	5	4	3	2	1	0	
(0034H)	–	–	ATAS	ATOUT	(WDTEN)	(WDTT)		(WDTOUT)	(Initial value: **11 1001)

ATAS	Selection of address trap in internal RAM	0: No address trap
		1: Address trap
		    (After setting ATAS to "1", it is necessary to write the control
		    code D2H to WDTCR2)
			Write only

ATOUT	Selection of operation at address trap	0: Interrupt request 1: Reset request	

Watchdog Timer Control Register 2

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

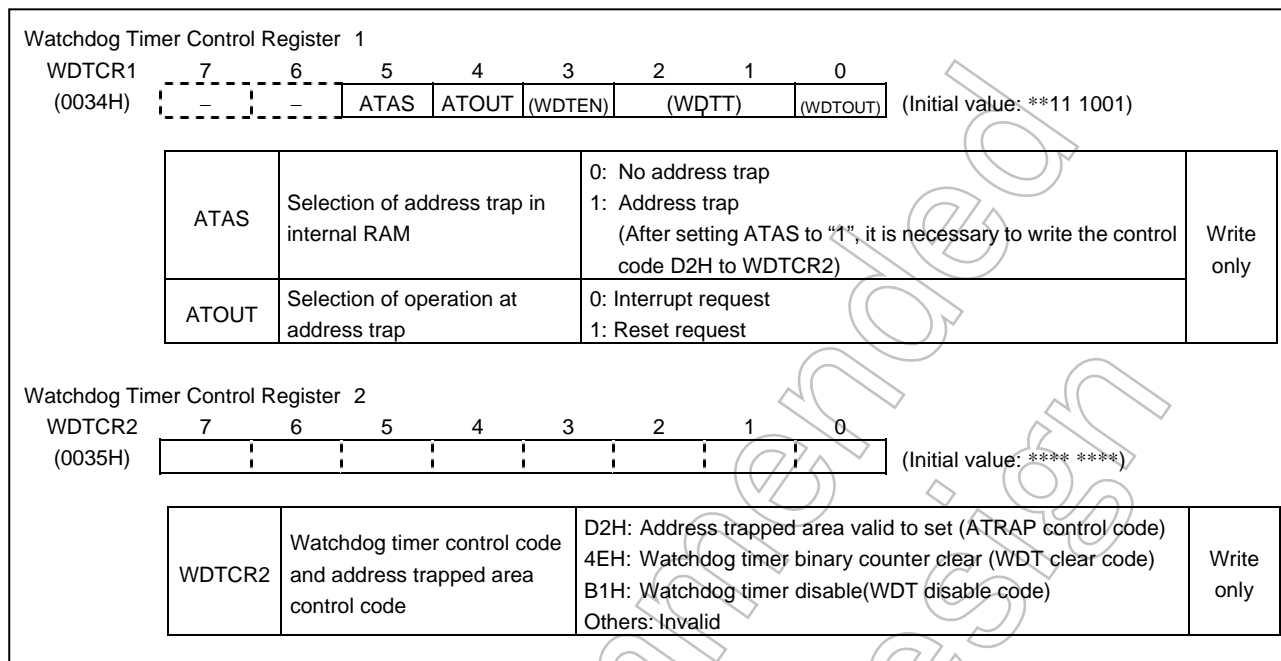|			
WDTCR2	Watchdog timer control code and address trapped area control code	D2H: Address trapped area valid to set (ATRAP control code) 4EH: Watchdog timer binary counter clear (WDT clear code) B1H: Watchdog timer disable(WDT disable code) Others: Invalid	Write only

Figure 2.4.4 Watchdog Timer Control Registers

(1) Selection of address trap in internal RAM (ATAS)

Using WDTCR1<ATAS>, address trap or no address trap can be selected for the internal RAM area. To execute an instruction in the internal RAM area, set "0" in WDTCR1<ATAS>. Setting in WDTCR1<ATAS> becomes valid after control code D2H is written in WDTCR2. Executing an instruction in the SFR/DBR area generates an address trap unconditionally regardless of the setting in WDTCR1<ATAS>.

(2) Selection of operation at address trap (ATOUT)

As the operation at address trap either interrupt request or reset request can be selected by WDTCR1<ATOUT>.

2.5　Divider Output (DVO)

Approximately 50% duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from pin P51 ($\overline{\text{DVO}}$). The P51 output latch should be set to "1".

Note:　Selection of divider output frequency must be made while divider output is disabled.
　　　　Also, in other words, when changing the state of the divider output frequency from enabled to disable, do not change the setting of the divider output frequency.

TBTCR (0036H)	7	6	5	4	3	2	1	0	
	DVOEN	DVOCK		(DV7CK)	(TBTEN)		(TBTCK)		(Initial value: 0000 0000)

DVOEN	Divider output enable/disable	0: Disable 1: Enable				
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]		NORMAL1/2 Mode		SLOW, SLEEP Mode	R/W
			DV7CK = 0	DV7CK = 1		
		00	$fc/2^{13}$	$fs/2^5$	$fs/2^5$	
		01	$fc/2^{12}$	$fs/2^4$	$fs/2^4$	
		10	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		11	$fc/2^{10}$	$fs/2^2$	$fs/2^2$	

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

Figure 2.5.1　Divider Output Control Register

Example:　1.95 kHz pulse output (at fc = 16.0 MHz).
```
SET     (P5DR).1              ;   P51 output latch ← "1"
LD      (TBTCR), 00000000B    ;   DVOCK ← "00"
LD      (TBTCR), 10000000B    ;   DVOEN ← "1"
```

Table 2.5.1　Divider Output Frequency (Example: at fc = 16.0 MHz, fs = 32.768 kHz)

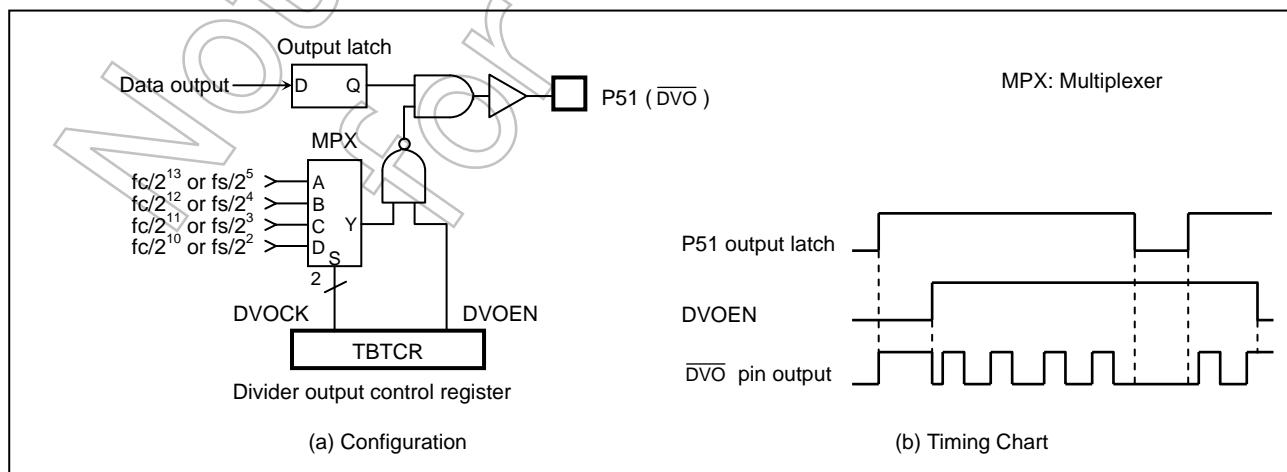|	Divider Output Frequency [Hz]		
DVOCK	NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode
	DV7CK = 0	DV7CK = 1	
00	1.953 k	1.024 k	1.024 k
01	3.906 k	2.048 k	2.048 k
10	7.813 k	4.096 k	4.096 k
11	15.625 k	8.192 k	8.192 k

Figure 2.5.2　Divider Output

2.6   16-Bit Timer/Counter 1
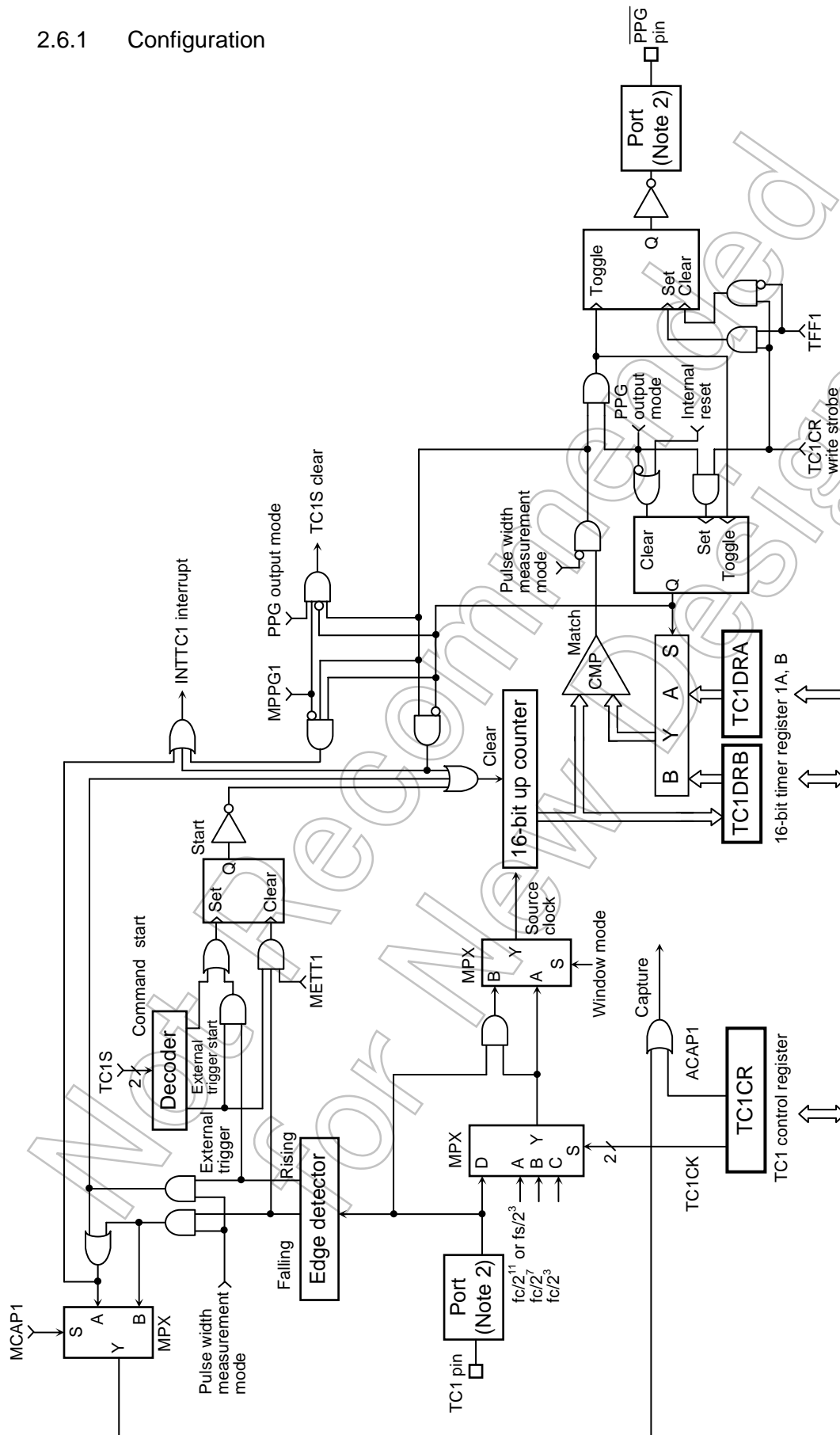
2.6.1   Configuration

Figure 2.6.1 Timer/Counter 1 (TC1)

Note 1:   MPX: Multiplexer
           CMP: Comparator

Note 2:   When control input/output is used, I/O port setting should be set correctly.
           For details, refer to 2.2 "I/O Ports".

2.6.2 Control

The timer/counter 1 is controlled by a timer/counter 1 control register (TC1CR) and two 16-bit timer registers (TC1DRA and TC1DRB).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0021,0020H) R/W				TC1DRAH (0021H)								TC1DRAL (0020H)				

(Initial value: 1111 1111 1111 1111)

	TC1DRBH (0023H)								TC1DRBL (0022H)							
TC1DRB (0023,0022H) R/W																

(Initial value: 1111 1111 1111 1111)

Note: TC1DRB should not be written except PPG mode.

	7	6	5	4	3	2	1	0
TC1CR (001FH)	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M	

(Initial value: 0000 0000)

TC1M	TC1 operating mode select	00: Timer/external trigger timer/event counter mode 01: Window mode 10: Pulse width measurement mode 11: PPG (Programmable pulse generate) output mode	

TC1CK	TC1 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2, SLEEP1/2 mode	
			DV7CK=0	DV7CK=1		
		00	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		01	$fc/2^7$	$fc/2^7$	–	
		10	$fc/2^3$	$fc/2^3$	–	
		11	External clock (TC1 pin input)			

			Timer	Extend	Event	Window	Pulse	PPG	
TC1S	TC1 start control	00: Stop and counter clear	○	○	○	○	○	○	R/W
		01: Command start	○	×	×	×	×	○	
		10: External trigger start at the rising edge	×	○	○	○	○	○	
		11: External trigger start at the falling edge	×	○	○	○	○	○	

ACAP1	Auto capture control	0: Auto-capture disable	1: Auto-capture enable
MCAP	Pulse width measurement mode control	0: Double edge capture	1: Single edge capture
METT1	External trigger timer mode control	0: Trigger start	1: Trigger start and stop
MPPG1	PPG output control	0: Continuous pulse generation	1: One shot
TFF1	Time F/F1 control	0: Clear	1: Set

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

Note 2: The timer register consists of two shift registers. A value set in the timer register is put in effect at the rising edge of the first source clock pulse that occurs after the upper data (TC1DRAH and TC1DRBH) are written. Therefore, the lower byte must be written before the upper byte (It is recommended that a 16-bit access instruction be used in writing). Writing only the lower data (TC1DRAL and TC1DRBL) does not put the setting of the timer register in effect.

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

Note 4: Auto capture can be used in only timer, event counter, and window modes.

Note 5: Values to be loaded to timer registers must satisfy the following condition.
TC1DRA > TC1DRB > 1 (PPG output mode), TC1DRA > 1 (others)

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

Note 7: Writing to the TC1DRB is not possible unless TC1 is set to the PPG output mode.

Note 8: On entering STOP mode, the TC1 start control (TC1S) is cleared to "00" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC1S again.

Note 9: Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition.

Note 10: Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Figure 2.6.2 Timer Registers and TC1 Control Register

2.6.3　Function

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

(1) Timer mode

In this mode, counting up is performed using the internal clock. The contents of TC1DRA are compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0". Counting up resumes after the counter is cleared. The current contents of up counter can be transferred to TC1DRB by setting TC1CR<ACAP1> to "1" (Auto-capture function). Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.

Table 2.6.1 Source Clock (Internal clock) for Timer/Counter 1 (Example: at fc = 16 MHz, fs = 32.768kHz)

TC1CK	NORMAL1/2, IDLE1/2 Modes				SLOW1/2, SLEEP1/2 Modes	
	DV7CK = 0		DV7CK = 1		Resolution [μs]	Maximum Time Setting [s]
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]		
00	128	8.39	244.14	16.0	244.14	16.0
01	8.0	0.524	8.0	0.524	–	–
10	0.5	32.77 m	0.5	32.77 m	–	–

Example 1: Sets the timer mode with source clock fc/2^{11} [Hz] and generates an interrupt 1 second later (at fc = 16 MHz, DV7CK = 0).

```
          LDW     (TC1DRA), 1E84H      ;    Sets the timer register.
                                            (1 s ÷ 2^11/fc = 1E84H)
          DI                                IMF = "0"
          SET     (EIRL). 5                 Enable INTTC1.
          EI                                IMF = "1"
          LD      (TC1CR), 00000000B        TFF1 ← "0", TC1CK ← "00", TC1M ← "00"
          LD      (TC1CR), 00010000B        Starts TC1.
Example 2:  Auto capture.
          LD      (TC1CR), 01010000B        ACAP1 ← "1" (Capture)
          LD      WA, (TC1DRB)              Reads the capture value.
```

Note : Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.
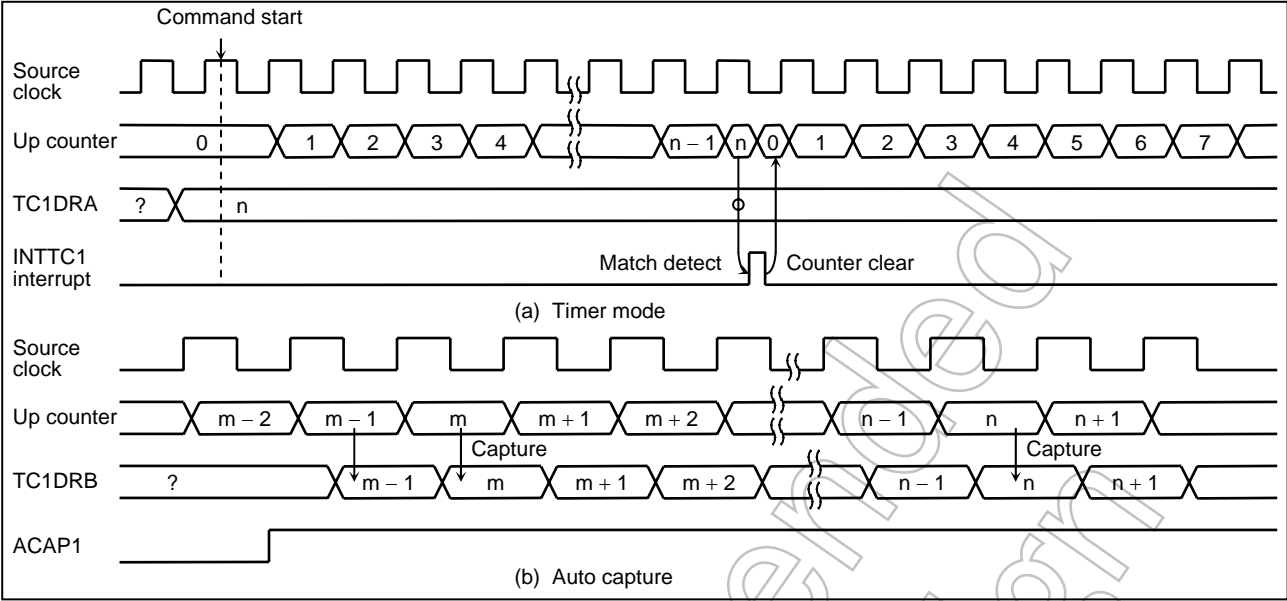
Figure 2.6.3 Timer Mode Timing Chart

(2) External trigger timer mode

In this mode, counting up is started by an external trigger. This trigger is the edge of the TC1 pin input. Either the rising or falling edge can be selected with TC1S. Source clock is an internal clock. The contents of TC1DRA is compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0" and halted. The counter is restarted by the selected edge of the TC1 pin input.

When TC1CR<METT1> is "1", inputting the edge to the reverse direction of the trigger edge to start counting clears the counter, and the counter is stopped. Inputting a constant pulse width can generate interrupts. When TC1CR<METT1> is "0", the reverse directive edge input is ignored. The TC1 pin input edge before a match detection is also ignored.

The TC1 pin input has the noise rejection; therefore, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [s] or more is required for edge detection in NORMAL1/2 or IDLE1/2 mode. The noise rejection circuit is turned off in SLOW1/2 and SLEEP1/2 modes. But, a pulse width of one machine cycle or more is required.

```
Example 1:  Detects rising edge in TC1 pin input and generates an interrupt 100 μs later
            (at fc = 16 MHz, DV7CK = 0).
            DI                                     ;   IMF = "0"
            LDW      (TC1DRA), 00C8H               ;   100 μs ÷ 2³/fc = C8H
            SET      (EIRL). 5                     ;   INTTC1 interrupt enable.
            EI                                     ;   IMF = "1"
            LD       (TC1CR), 00001000B            ;   TFF1 = "0", TC1CK = "10", TC1M = "00"
            LD       (TC1CR), 00101000B            ;   TC1 external trigger start, METT1 = "0"
Example 2:  Generates an interrupt, inputting "L" level pulse (Pulse width: 4 ms or more) to the TC1 pin
            (at fc = 16 MHz).
            DI                                     ;   IMF = "0"
            LDW      (TC1DRA), 1F40H               ;   4 ms ÷ 2³/fc = 1F40H
            SET      (EIRL). 5                     ;   INTTC1 interrupt enable.
            EI                                     ;   IMF = "1"
            LD       (TC1CR), 01001000B            ;   TFF1 = "0", TC1CK = "10", TC1M = "00"
            LD       (TC1CR), 01111000B            ;   TC1 external trigger start, METT1 = 1
```
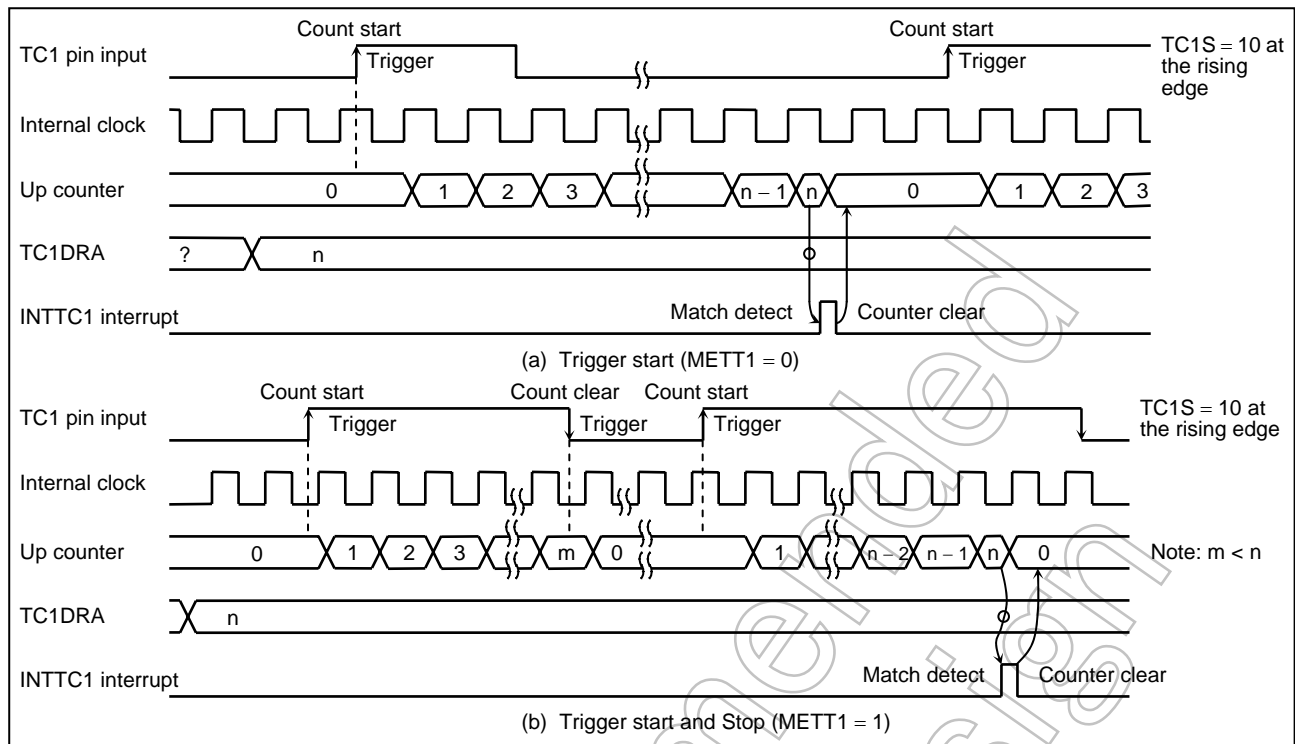
Figure 2.6.4 External Trigger Timer Mode Timing Chart

(3) Event counter mode

In this mode, events are counted at the edge of the TC1 pin input (Either the rising or falling edge can be selected with the external trigger TC1CR1<TC1S>). The contents of TC1DRA are compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. After the counter is cleared, the up counter starts counting by TC1 input edge. Match detect is executed on other edge of count up. A match can not be detected and INTTC1 is not generated when the pulse is still in same state. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

Setting TC1CR<ACAP1> to "1" transfers the current contents of up counter to TC1DRB (Auto-capture function). Use the auto-capture function in the operative condition of TC1. A captured value may not be fixed if it's read after the execution of the timer stop or auto-capture disable. Read the capture value in a capture enabled condition. Since the up-counter value is captured into TC1DRB by the source clock of up-counter after setting TC1CR<ACAP1> to "1". Therefore, to read the captured value, wait at least one cycle of the internal source clock before reading TC1DRB for the first time.
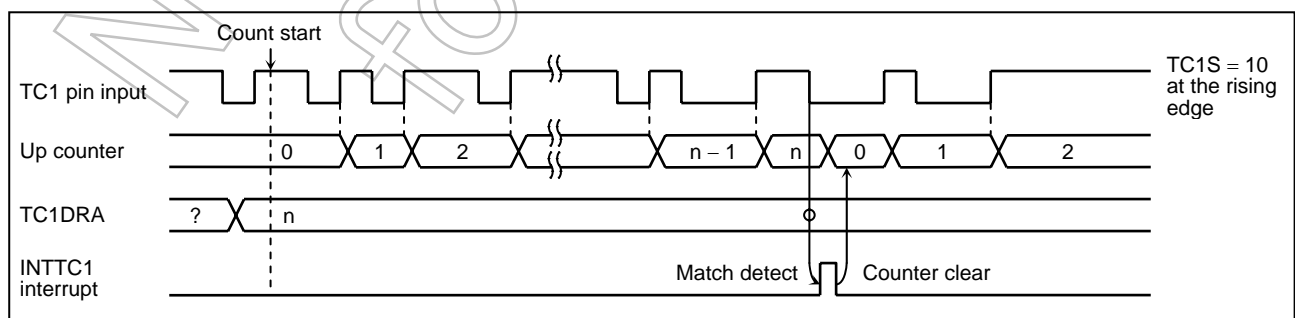

Figure 2.6.5 Event Counter Mode Timing Chart

Table 2.6.2 Timer/Counter 1 External Clock Source

	Minimum Input Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Modes	SLOW1/2, SLEEP1/2 Modes
"H" Width	2^3/fc	2^3/fs
"L" Width	2^3/fc	2^3/fs

(4) Window mode

In this mode, counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC1 pin input (Window pulse) and an internal clock. The contents of TC1DRA are compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. It is possible to select either positive logic or negative logic for the TC1 pin input (by using the TC1 start control TC1CR<TC1S>).

The maximum frequency that can be applied to the pin must be such that the related count can be analyzed by program. To put another way, the frequency of the applied pulse must be sufficiently low, compared with that of the internally set source clock.
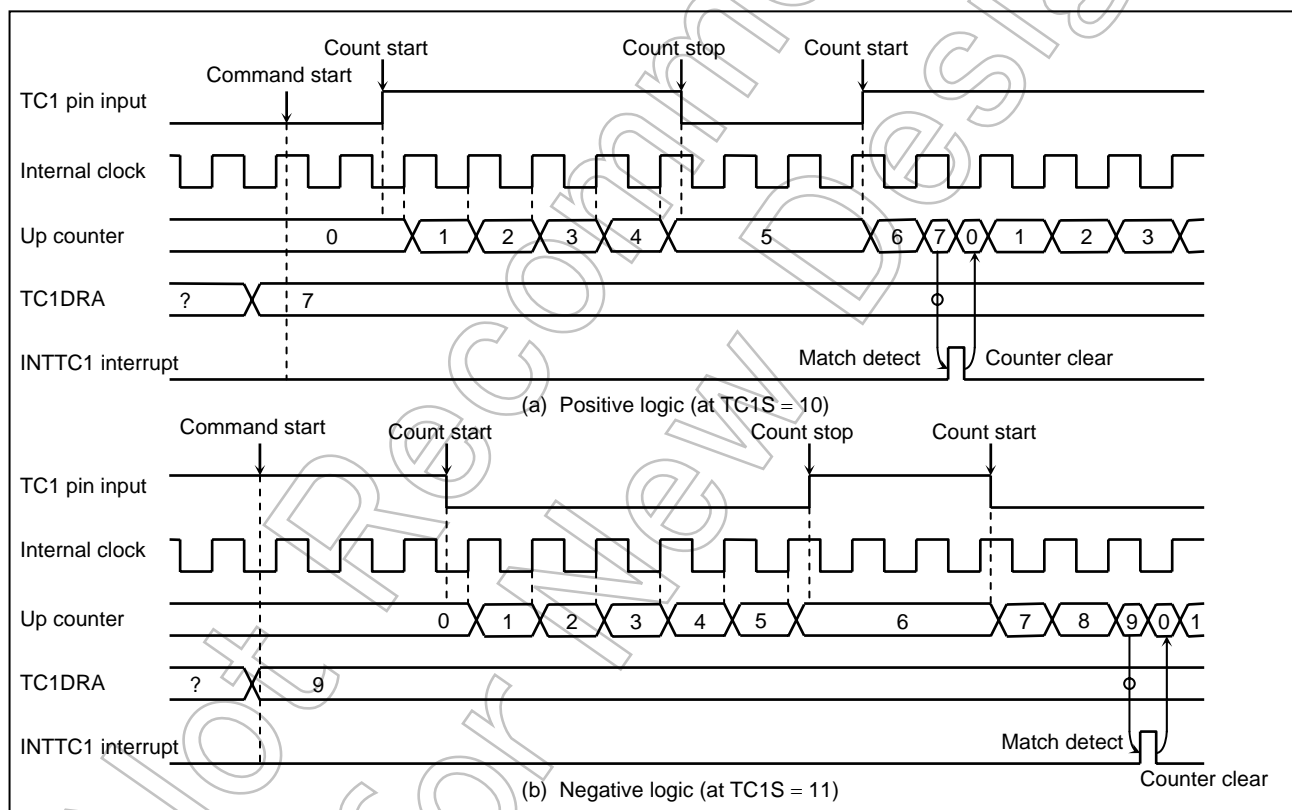
Figure 2.6.6 Window Mode Timing Chart

(5) Pulse width measurement mode

In this mode, counting is started by the external trigger (Set to external trigger start by TC1CR<TC1S>). The trigger can be selected either the rising or falling edge of the TC1 pin input. The source clock is used an internal clock. On the next falling (rising) edge, the counter contents are transferred to TC1DRB and an INTTC1 interrupt is generated. The counter is cleared when the single edge capture mode (TC1CR<MCAP> = "1") is set. When double edge capture (TC1CR<MCAP> = "0") is set, the counter continues and, at the next rising (falling) edge, the counter contents are again transferred to TC1DRB. If a falling (rising) edge capture value is required, it is necessary to read out TC1DRB contents until a rising (falling) edge is detected. Falling or rising edge is selected with the external trigger TC1CR<TC1S>, and single edge or double edge is selected with TC1CR<MCAP1>.

Note 1: Be sure to read the captured value from TC1DRB before the next trigger edge is detected. If fail to read it, it becomes undefined. It is recommended that a 16-bit access instruction be used to read from TC1DRB.

Note 2: If either the falling or rising edge is used in capturing values, the counter stops at "1" after a value has been captured until the next edge is detected. So, the value captured next will become "1" larger than the value captured right after capturing starts.

Note 3: The first captured value after the timer starts may be read incorrectively, therefore, ignore the first captured value.

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

```
            CLR     (INTTC1SW). 0          ; INTTC1 service switch initial setting.
            LD      (TC1CR), 00000110B     ; Sets the TC1 mode and source clock.
            DI                             ; IMF = "0"
            SET     (EIRL). 5              ; Enables INTTC1.
            EI                             ; IMF = "1"
            LD      (TC1CR), 00100110B     ; Starts TC1 with an external trigger at
                                             MCAP1 = 0.
PINTTC1:    CPL     (INTTC1SW). 0          ; Inverts INTTC1 service switch.
            JRS     F, SINTTC1
            LD      A, (TC1DRBL)           ; Reads TC1DRB ("H" level pulse width).
            LD      W, (TC1DRBH)
            RETI
SINTTC1:    LD      L, (TC1DRBL)           ; Reads TC1DRB (Period).
            LD      H, (TC1DRBH)
                                           ; Duty calculation.
            RETI
VINTTC1:    DW      PINTTC1
```
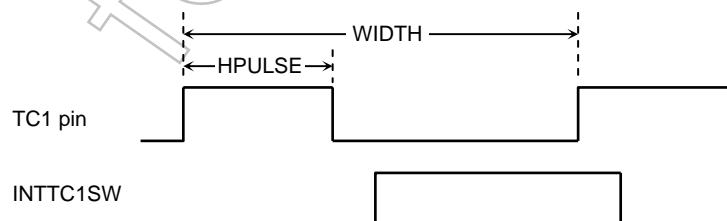
Figure 2.6.7 Pulse Measurement Mode Timing Chart

(6) Programmable pulse generate (PPG) output mode

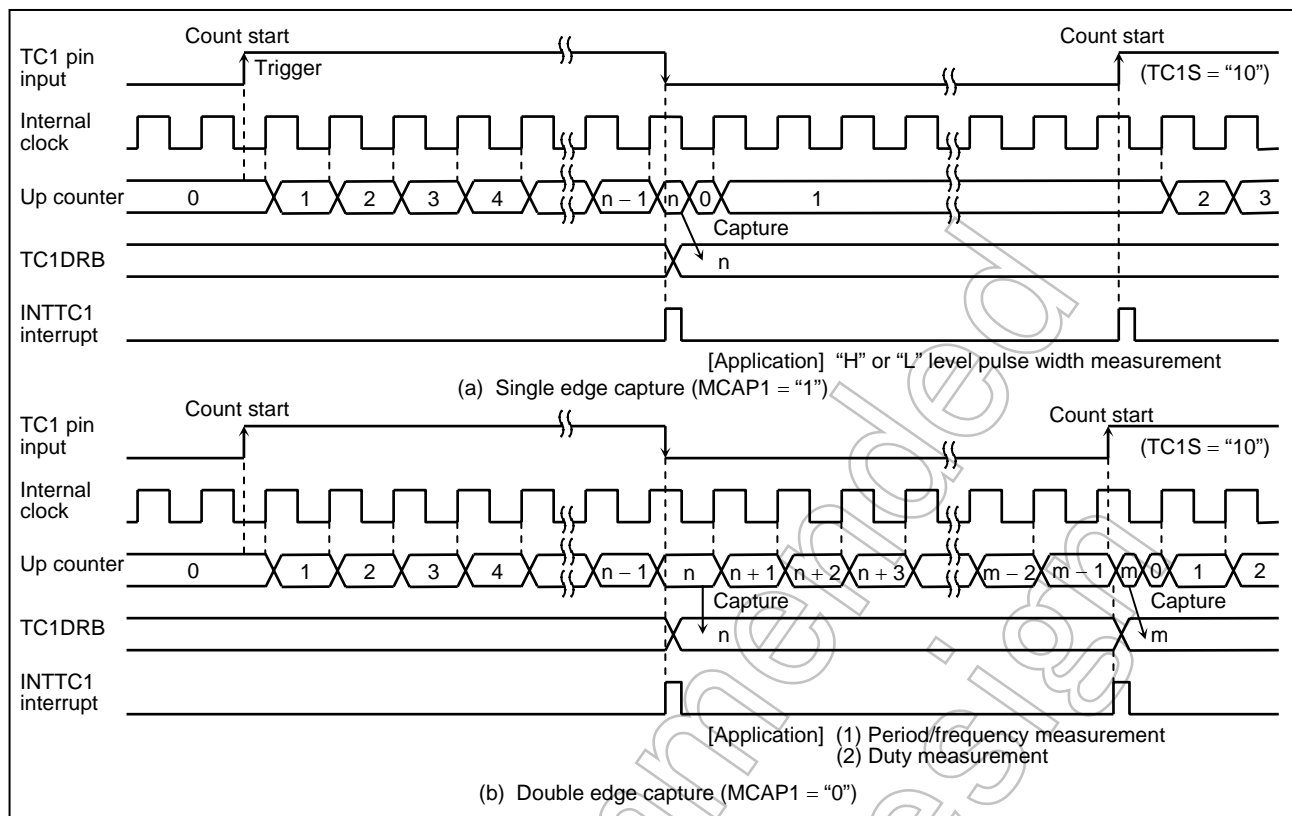The PPG output mode is intended to output pulses having an arbitrary duty cycle selected using two timer registers.

The timer starts at an edge (Rising or falling edge), that is, the same edge type as selected with the external trigger edge select bits (TC1CR<TC1S>) or on a command. Its source clock is an internal clock. Once the timer starts running, the timer F/F1 is inverted when the counter matches TC1DRB, generating the INTTC1 interrupt. The counter keeps up counting, and when counter matches TC1DRA, the timer F/F1 is inverted, generating an INTTC1 interrupt. If TC1CR<MPPG1> was previously set to "1" (one shot), TC1CR<TC1S> is cleared to "00" automatically, causing the timer to stop. If TC1CR<MPPG1> was previously cleared to "0" (Continuous pulse generation), the counter is cleared, resulting in the counter keeping to run and the PPG output being continued. If TC1CR<TC1S> is reset to "00" (One-shot-based automatic stop is included) during PPG output, the P50 ($\overline{PPG}$) pin holds the same level that it does just before the counter stops. In PPG output mode, set the output latch of port P50 to "1". The timer F/F1 is cleared to "0" at a reset. In addition, a positive or negative pulse can be output because the output level can be set up at a start, using TC1CR<TFF1>. The P50 ($\overline{PPG}$) pin outputs an inversion of the timer F/F1 output level. It is impossible to write to TC1DRB unless the PPG output mode is set.

Note 1: To change the content of the timer register when the timer is running, change it to a sufficiently large value, compared with the current count. If the timer register content is changed to a value smaller than the current count when the timer is running, it is likely that unintended pulses may be output.

Note 2: Do not change TC1CR<TFF1> when the timer is running.
TC1CR<TFF1> can be set correctly only at initialization (after a reset). When the timer is stopped during PPG output, if the PPG output is at a logic state opposite to the PPG that when the timer starts, it will become impossible to set TC1CR<TFF1> correctly (an attempt to program TC1CR<TFF1> will cause a state opposite to the programmed one to be set in the bit). Once the timer has stopped, putting the PPG output securely on an arbitrary level requires initializing the timer F/F1. To initialize it, put TC1CR<TC1M> in the timer mode again (It is unnecessary to start the timer mode), and then put it in the PPG output mode again. At the same time, set TC1CR<TFF1>.

Note 3: In the PPG output mode, a value set in the timer register must satisfy: TC1DRA > TC1DRB

Example: Pulse output "H" level 800 μs, "L" level 200 μs (at fc = 16 MHz, DV7CK = 0).

SET	(P5DR). 0	;	P50 output latch ← 1
LD	(TC1CR), 10001011B	;	Sets the PPG output mode.
LDW	(TC1DRA), 07D0H	;	Sets the period (1 ms ÷ 2^3/fc = 07D0H).
LDW	(TC1DRB), 0190H	;	Sets "L" level pulse width
			(200 μs ÷ 2^3/fc = 0190H).
LD	(TC1CR), 10011011B	;	Starts.

Figure 2.6.8 $\overline{PPG}$ Output

(a) Continuous pulse generation (with TC1S = 01)

(b) One shot (with TC1S = 10)
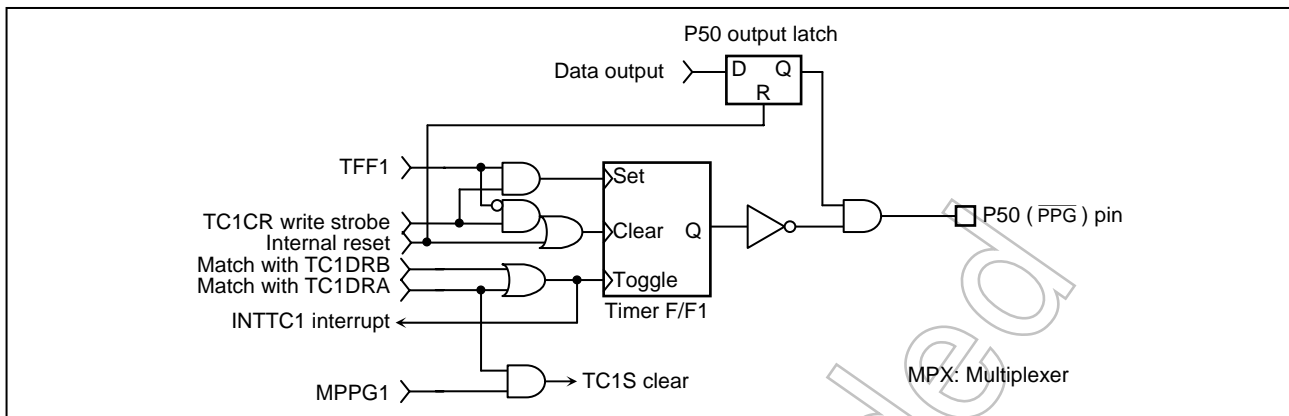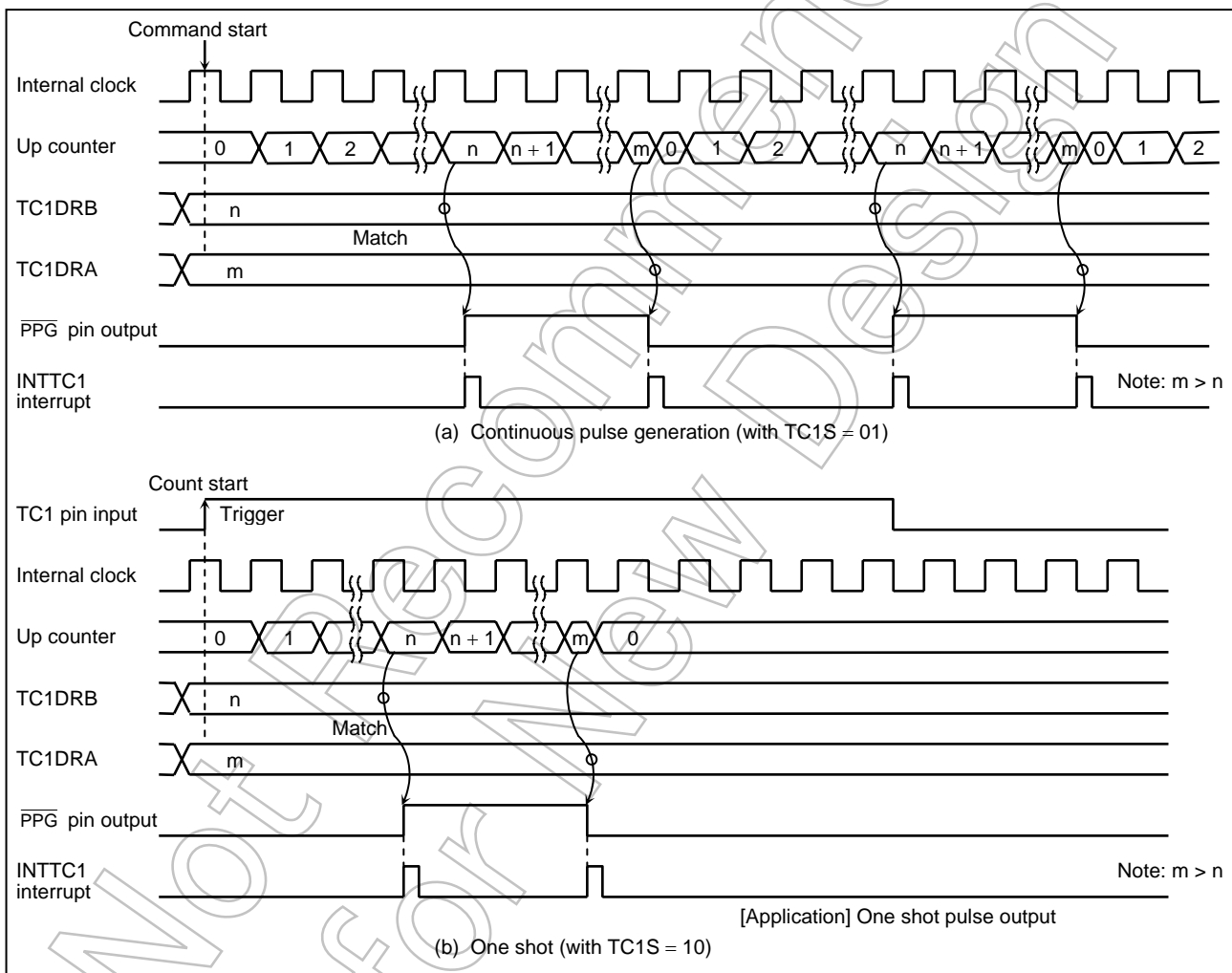
Figure 2.6.9 PPG Output Mode Timing Chart

2.7   16-Bit Timer/Counter 2
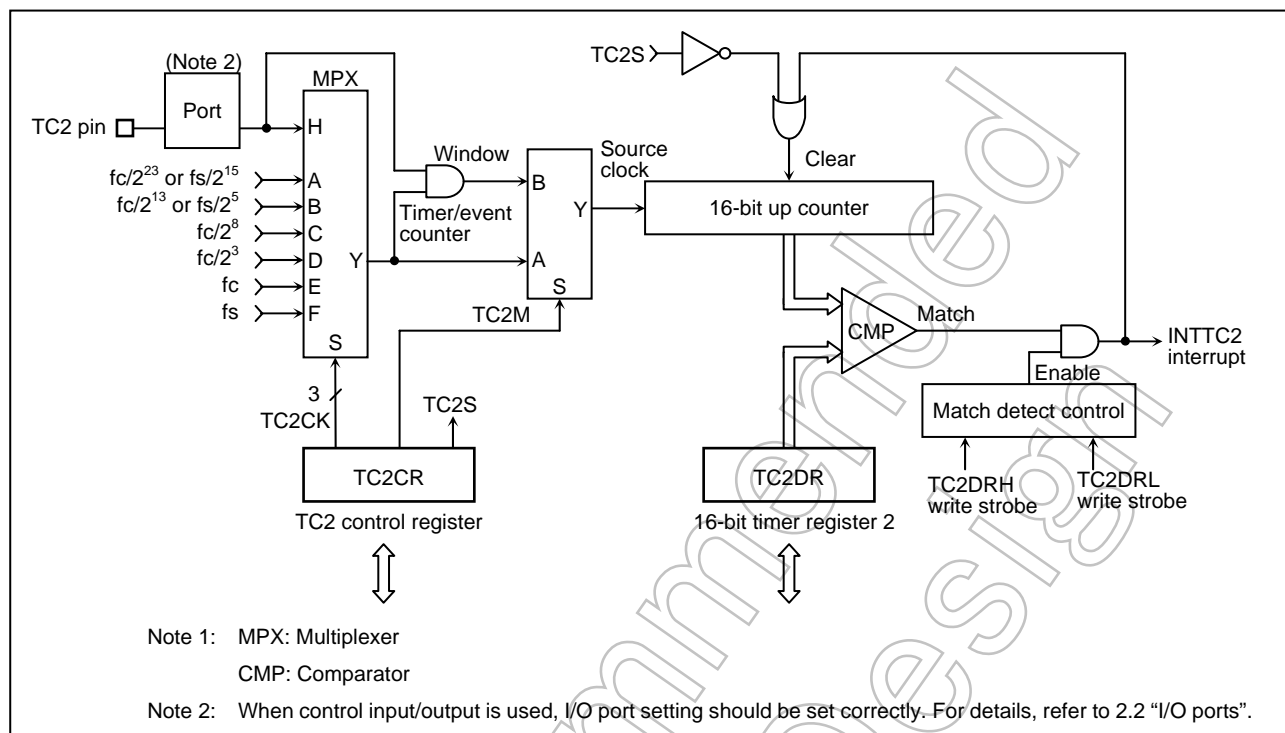
2.7.1   Configuration

Figure 2.7.1 Timer/Counter 2 (TC2A)

2.7.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR). Reset does not affect TC2DR.
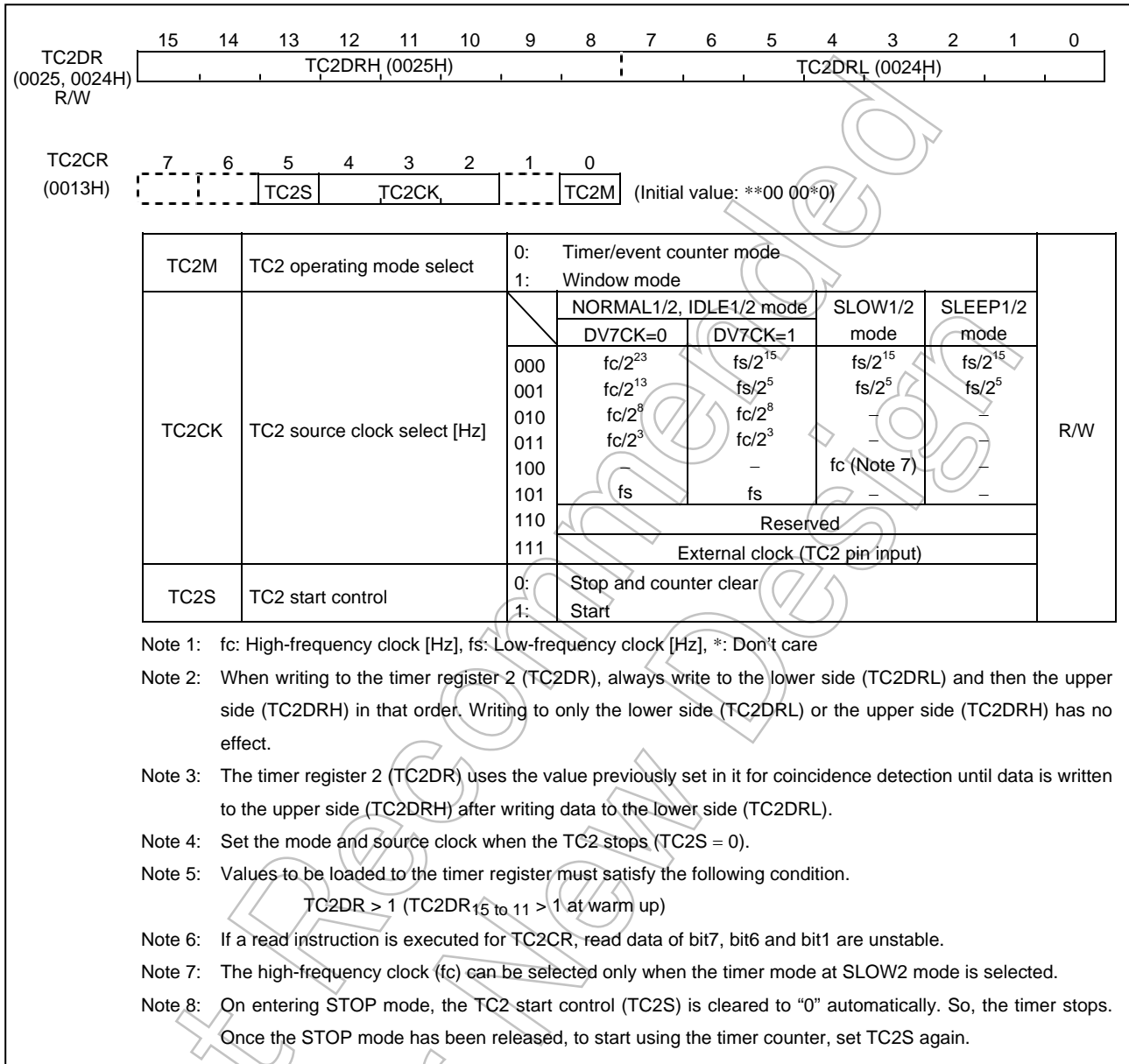
TC2DR
(0025, 0024H)
R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
			TC2DRH (0025H)								TC2DRL (0024H)				

TC2CR
(0013H)

7	6	5	4	3	2	1	0
		TC2S		TC2CK			TC2M

(Initial value: **00 00*0)

TC2M	TC2 operating mode select	0: Timer/event counter mode 1: Window mode					R/W
TC2CK	TC2 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW1/2 mode	SLEEP1/2 mode	
			DV7CK=0	DV7CK=1			
		000	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$	$fs/2^{15}$	
		001	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$	
		010	$fc/2^{8}$	$fc/2^{8}$	—	—	
		011	$fc/2^{3}$	$fc/2^{3}$	—	—	
		100	—	—	fc (Note 7)	—	
		101	fs	fs	—	—	
		110	Reserved				
		111	External clock (TC2 pin input)				
TC2S	TC2 start control	0: Stop and counter clear 1: Start					

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

Note 2: When writing to the timer register 2 (TC2DR), always write to the lower side (TC2DRL) and then the upper side (TC2DRH) in that order. Writing to only the lower side (TC2DRL) or the upper side (TC2DRH) has no effect.

Note 3: The timer register 2 (TC2DR) uses the value previously set in it for coincidence detection until data is written to the upper side (TC2DRH) after writing data to the lower side (TC2DRL).

Note 4: Set the mode and source clock when the TC2 stops (TC2S = 0).

Note 5: Values to be loaded to the timer register must satisfy the following condition.

$$TC2DR > 1 \ (TC2DR_{15 \ to \ 11} > 1 \ at \ warm \ up)$$

Note 6: If a read instruction is executed for TC2CR, read data of bit7, bit6 and bit1 are unstable.

Note 7: The high-frequency clock (fc) can be selected only when the timer mode at SLOW2 mode is selected.

Note 8: On entering STOP mode, the TC2 start control (TC2S) is cleared to "0" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC2S again.

Figure 2.7.2 Timer Register 2 and TC2 Control Register

2.7.3 Function

The timer/counter 2 has three operating modes: timer, event counter and window modes.

（1） Timer mode

In this mode, the internal clock is used for counting up. The contents of TC2DR are compared with the contents of up counter. If a match is found, a timer/counter 2 interrupt (INTTC2) is generated, and the counter is cleared. Counting up is resumed after the counter is cleared.

When fc is selected for source clock at SLOW2 mode, lower 11 bits of TC2DR are ignored and generated a interrupt by matching upper 5 bits. Though, in this situation, it is necessary to set TC2DRH only.

Table 2.7.1 Source Clock (Internal clock) for Timer/Counter 2 (at fc = 16 MHz)

TC2CK	NORMAL1/2, IDLE1/2 Modes				SLOW1/2 Mode		SLEEP1/2 Mode	
	DV7CK = 0		DV7CK = 1		Resolution	Maximum Time Setting	Resolution	Maximum Time Setting
	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting				
000	524.29 ms	9.54 h	1.00 s	18.20 h	1.00 s	18.20 h	1.00 s	18.20 h
001	512.00 μs	33.55 s	0.98 ms	1.07 min	0.98 ms	1.07 min	0.98 ms	1.07 min
010	16.00 μs	1.05 s	16.00 μs	1.05 s	–	–	–	–
011	0.50 μs	32.77 ms	0.50 μs	32.77 ms	–	–	–	–
100	–	–	–	–	62.5 ns (Note)	–	–	–
101	30.52 μs	2.00 s	30.52 μs	2.00 s	–	–	–	–

Note: When fc is selected as the source clock in timer mode, it is used at warm up for switching from SLOW2 mode to NORMAL2 mode.

Example: Sets the timer mode with source clock $fc/2^3$ [Hz] and generates an interrupt every 25 ms (at fc = 16 MHz).

```
LDW    (TC2DR), 0C350H      ;  Sets TC2DR (25 ms ÷ 2³/fc = C350H).
DI                          ;  IMF = "0"
SET    (EIRE). 4            ;  Enables INTTC2 interrupt.
EI                          ;  IMF = "1"
LD     (TC2CR), 00001100B   ;  TC2CK ← "011", TC2M ← "0"
LD     (TC2CR), 00101100B   ;  Starts TC2.
```

(2) Event counter mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TC2DR are compared with the contents of the up counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The minimum input pulse width of the TC2 pin is shown in Table 2.7.2. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width. Match detect is executed on the falling edge of the TC2 pin. A match can not be detected and INTTC2 is not generated when the pulse is still in a falling state.

Example: Sets the event counter mode and generates an INTTC2 interrupt 640 counts later.

LDW	(TC2DR), 640	;	Sets TC2DR.
DI		;	IMF = "0"
SET	(EIRE). 4	;	Enables INTTC2 interrupt.
EI		;	IMF = "1"
LD	(TC2CR), 00011100B	;	TC2CK ← "111", TC2M ← "0"
LD	(TC2CR), 00111100B	;	Starts TC2.

Table 2.7.2 Timer/Counter 2 External Clock Source

	Minimum Input Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Modes	SLOW1/2, SLEEP1/2 Modes
"H" Width	2^3/fc	2^3/fs
"L" Width	2^3/fc	2^3/fs

(3) Window mode

In this mode, counting up performed on the rising edge of an internal clock during TC2 external pin input (window pulse) is "H" level. The contents of TC2DR are compared with the contents of up counter. If a match found, an INTTC2 interrupt is generated, and the up counter is cleared.

The maximum applied frequency (TC2 input) must be considerably slower than the selected internal clock.

Note: In the window mode, before the SLOW/SLEEP mode is entered, the timer should be halted by setting TC2CR<TC2S> to "0".

Example: Generates an interrupt, inputting "H" level pulse width of 120 ms or more
           (at fc = 16 MHz, DV7CK = 0).
           LDW   (TC2DR), 00EAH   ;   Sets TC2DR (120 ms ÷ 2^{13}/fc = 00EAH).
           DI   ;   IMF = "0"
           SET   (EIRE). 4   ;   Enables INTTC2 interrupt.
           EI   ;   IMF = "1"
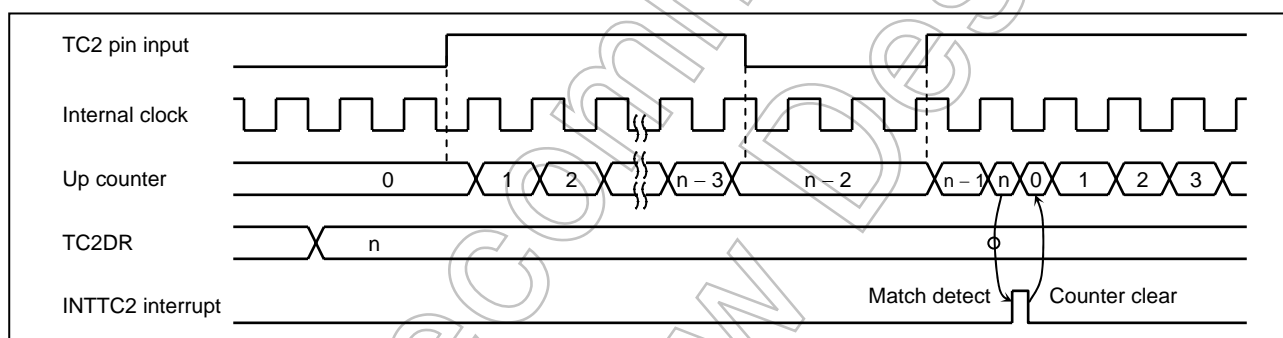   LD   (TC2CR), 00000101B   ;   TC2CK ← "001", TC1M ← "1"
           LD   (TC2CR), 00100101B   ;   Starts TC2.

Figure 2.7.3 Window Mode Timing Chart

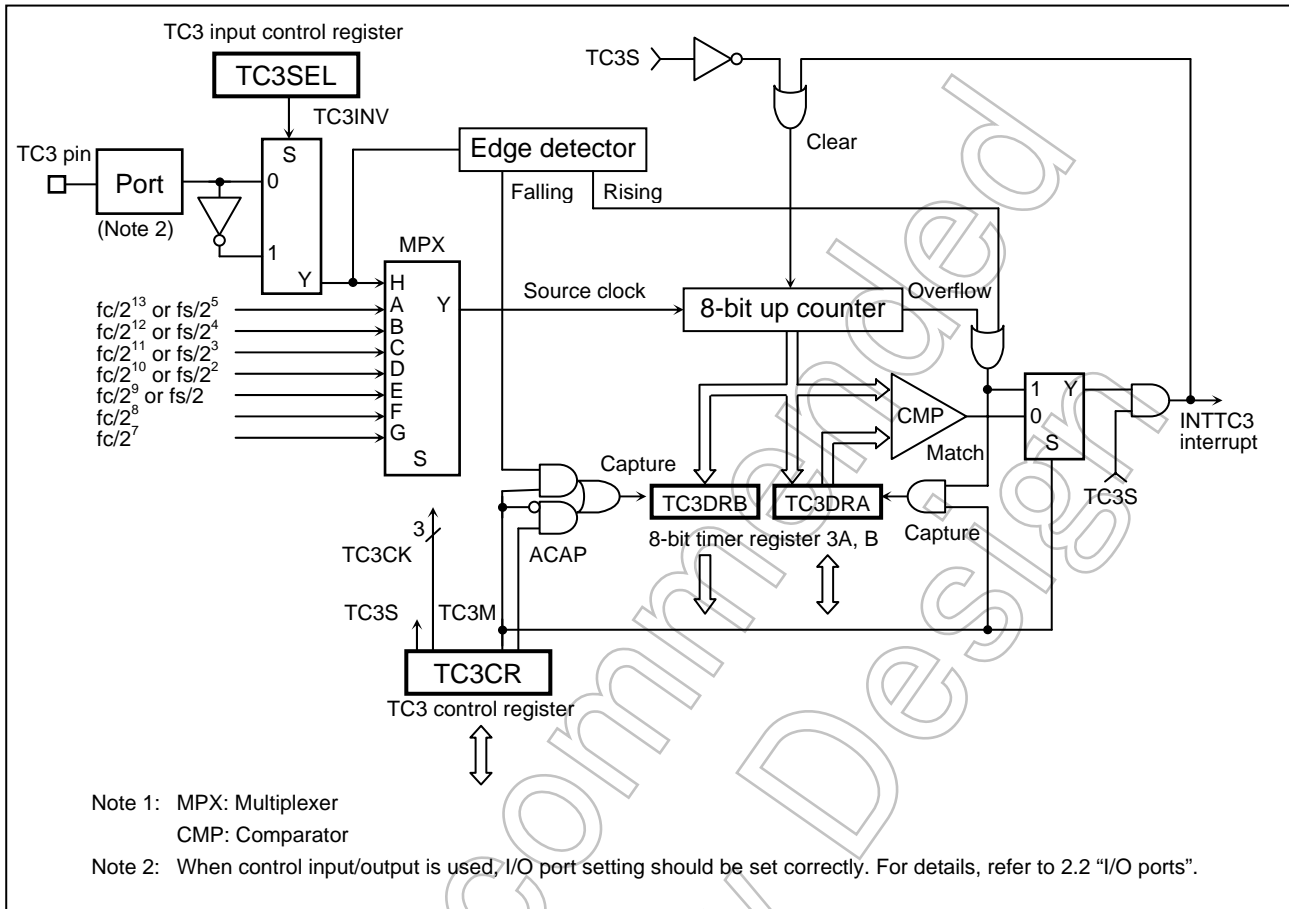## 2.8   8-Bit Timer/Counter 3
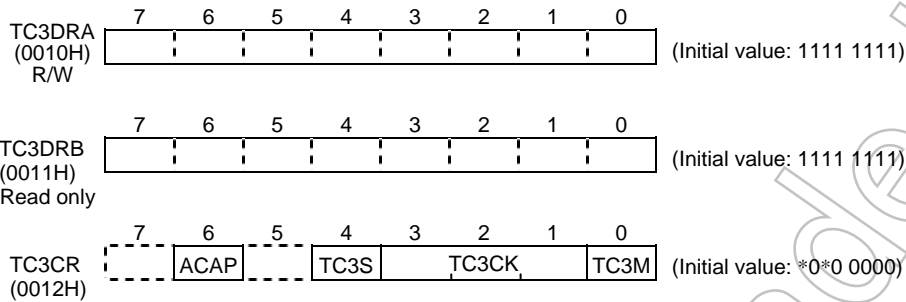
2.8.1   Configuration

Figure 2.8.1 Timer/Counter 3 (TC3)

2.8.2 Control

The timer/counter 3 is controlled by a timer/counter 3 control register (TC3CR) and two 8-bit timer registers (TC3DRA and TC3DRB).

TC3DRA (0010H) R/W — bits 7 6 5 4 3 2 1 0 (Initial value: 1111 1111)

TC3DRB (0011H) Read only — bits 7 6 5 4 3 2 1 0 (Initial value: 1111 1111)

TC3CR (0012H) — bit 6 ACAP, bit 4 TC3S, bits 3-1 TC3CK, bit 0 TC3M (Initial value: ∗0∗0 0000)

			NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode	
TC3M	TC3 operation mode set	0: Timer/event counter 1: Capture				
			DV7CK = 0	DV7CK = 1		
TC3CK	TC3 source clock select [Hz]	000	$fc/2^{13}$	$fs/2^5$	$fs/2^5$	R/W
		001	$fc/2^{12}$	$fs/2^4$	$fs/2^4$	
		010	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		011	$fc/2^{10}$	$fs/2^2$	$fs/2^2$	
		100	$fc/2^9$	$fs/2$	$fs/2$	
		101	$fc/2^8$	$fc/2^8$	−	
		110	$fc/2^7$	$fc/2^7$	−	
		111	External clock (TC3 pin input)			
TC3S	TC3 start select	0: Stop and clear 1: Start				
ACAP	Auto-capture control	0: − 1: Auto-capture enable				

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

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

Note 3: Values to be loaded into timer register 3A must satisfy the following condition.

TC3DRA > 1 (in the timer and event counter mode)

Note 4: Auto capture can be used only in the timer and event counter mode.
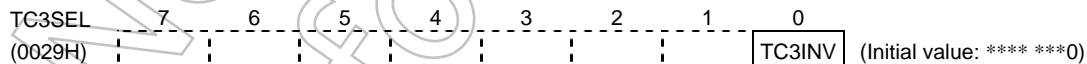
Note 5: If a read instruction is executed for TC3CR, read data for bits 7 and 5 are unstable.

Note 6: During TC3 operation, do not change TC3DRA.

Note 7: On entering STOP mode, TC3 start control (TC3S) is cleared to "0" automatically, so the timer stops.

Once the STOP mode has been released, to start using the timer counter, set TC3S again.

TC3 input control (normal/invert)

TC3SEL (0029H) — bit 0 TC3INV (Initial value: ∗∗∗∗ ∗∗∗0)

TC3INV	TC3 input control	0: Normal 1: Invert	R/W

Note: If a read instruction is executed for TC3SEL, read data for bits 7 to 1 are unstable.

Figure 2.8.2 Timer Register 3 and TC3 Control Register

2.8.3 TC3 Input Control Register

This microcomputer has the function to invert or not to invert the waveform entered from the TC3 pin. This selection is made by using TC3SEL<TC3INV>.

2.8.4 Function

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

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of TC3DRA 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.

The current contents of up counter are loaded into TC3DRB by setting TC3CR<ACAP> to "1" (Auto-capture function). The contents of up counter can be easily confirmed by executing the read instruction (RD instruction) of TC3DRB. Loading the contents of up counter is not synchronized with counting up. The contents of over flow (FFH) and 00H can not be loaded correctly. It is necessary to consider the count cycle.
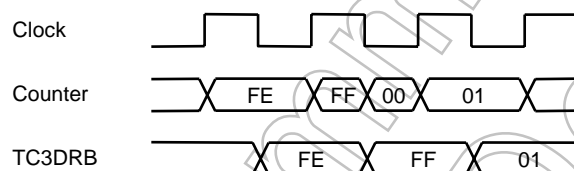

Table 2.8.1 Source Clock (Internal clock) for Timer/counter 3 (Example: at fc = 16 MHz)

| TC3CK | NORMAL1/2, IDLE1/2 Modes | | | | SLOW1/2 Modes | |
| | DV7CK = 0 | | DV7CK = 1 | | | |
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]
000	512.0	130.6	976.6	249.0	976.6	249.0
001	256.0	65.3	488.3	124.5	488.3	124.5
010	128.0	32.6	244.1	62.3	244.1	62.3
011	64.0	16.3	122.0	31.1	122.0	31.1
100	32.0	8.2	61.0	15.6	61.0	15.6
101	16.0	4.1	16.0	4.1	–	–
110	8.0	2.0	8.0	2.0	–	–

(2) Event counter mode

In this mode, events are counted on the edge of the TC3 pin input. The input pulse at the TC3 pin can have its polarity inverted using the TC3SEL register TC3INV bit. When TC3SEL<TC3INV> = "0", the counter counts up on the rising edge of the TC3 pin input and when its value matches the TC3DRA set value, it is cleared while at the same time generating an INTTC3 interrupt. When TC3SEL<TC3INV> = "1", the counter counts up on the falling edge of the TC3 pin input and when its value matches the TC3DRA set value, it is cleared while at the same time generating an INTTC3 interrupt.

When TC3SEL<TC3INV> = "0", the detection of match is executed at the falling edge of the TC3 pin. Therefore, if the TC3 pin keeps high level after the rising, the detection of match is not executed and INTTC3 is not generated until the level of TC3 pin becomes low. When TC3SEL<TC3INV> = "1", the detection of match is executed at the rising edge of the TC3 pin. Therefore, if the TC3 pin keeps low level after the falling, the detection of match is not executed and INTTC3 is not generated until the level of TC3 pin becomes high.

The minimum input pulse width of the TC3 pin is shown in Table 2.8.2. One 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 TC3DRB by setting TC3CR<ACAP> to "1" (Auto-capture function).

The contents of up counter can be easily confirmed by executing the read instruction (RD instruction) of TC3DRB. Loading the contents of up counter is not synchronized with counting up. The contents of over flow (FFH) and 00H can not be loaded correctly. It is necessary to consider the count cycle.

Table 2.8.2 Source Clock (External clock) for Timer/Counter

	Minimum Input Pulse Width [s]	
	NORMAL1/2, IDLE1/2 Modes	SLOW1/2, SLEEP1/2 Modes
"H" Width	$2^2/fc$	$2^2/fs$
"L" Width	$2^2/fc$	$2^2/fs$

(3) Capture mode

In this 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 or distinguishing AC 50/60 Hz, etc. The TC3 pin input can have its polarity changed between normal and inverse by using the TC3SEL Register.

a. If TC3SEL<TC3INV> = "0" (Non-inverting input)

Once command operation has started, the counter free runs on an internal source clock.

When the falling edge of the TC3 pin input is detected, the counter value is loaded into TC3DRB. When the rising edge is detected, the counter value is loaded into TC3DRA, and the counter is cleared, generating an INTTC3 interrupt.

If the rising edge is detected right after command operation has started, no capture to TC3DRB and an INTTC3 interrupt occurs only on capture to TC3DRA. If a read instruction is executed for TC3DRB, the value that exists at the end of the previous capture (immediately after a reset, "FF") is read.

b. If TC3SEL<TC3INV> = "1" (inverse input)

Once command operation has started, the counter free-runs on an internal clock.

When the rising edge of the TC3 pin input is detected, the counter value is loaded into TC3DRB. When the falling edge is detected, the counter value is loaded into TC3DRA, and the counter is cleared, generating an INTTC3 interrupt.

If the falling edge is detected right after command operation has started, the counter value is not captured into TC3DRB and an INTTC3 interrupt occurs only on capture to TC3DRA. If a read instruction is executed for TC3DRB, the value that exists at end of the previous capture (Immediately after a reset, "FF") is read.

The minimum acceptable input pulse width is equal to the length of one source clock period selected by TC3CR<TC3CK>.

Table 2.8.3 TC3INV-based Capture Input Edges

TC3SEL <TC3INV>	Capture into TC3DRB	Capture into TC3DRA	INTTC3 Interrupt
"0" (Non-inverting input)	Falling edge	Rising edge	
"1" (Inverting input)	Rising edge	Falling edge	

When the overflow occurs before detecting the edge, the INTTC3 interrupt is generated, setting "FFH" to TC3DRA and clearing the counter. It is possible to confirm whether the overflow has occurred or not by reading TC3DRA in interrupt routine. After generating of interrupt, the capture function and overflow detection stop until the TC3DRA is read, but the counting is continued. Because the capture function and overflow detection are restarted by reading TC3DRA, read the TC3DRB before the reading TC3DRA.
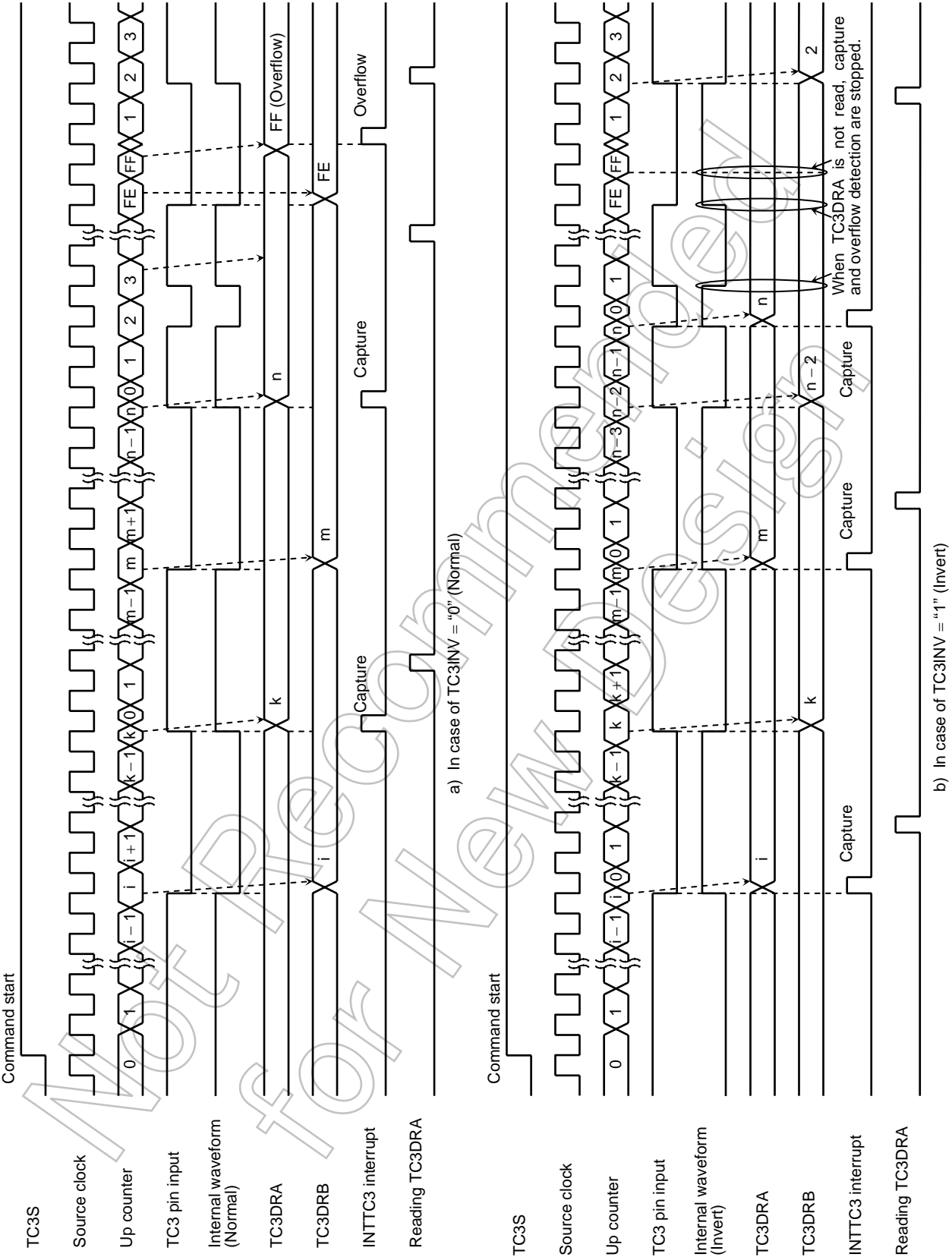
Figure 2.8.3 Capture Mode Timing Chart

2.9   8-Bit Timer/Counter 5
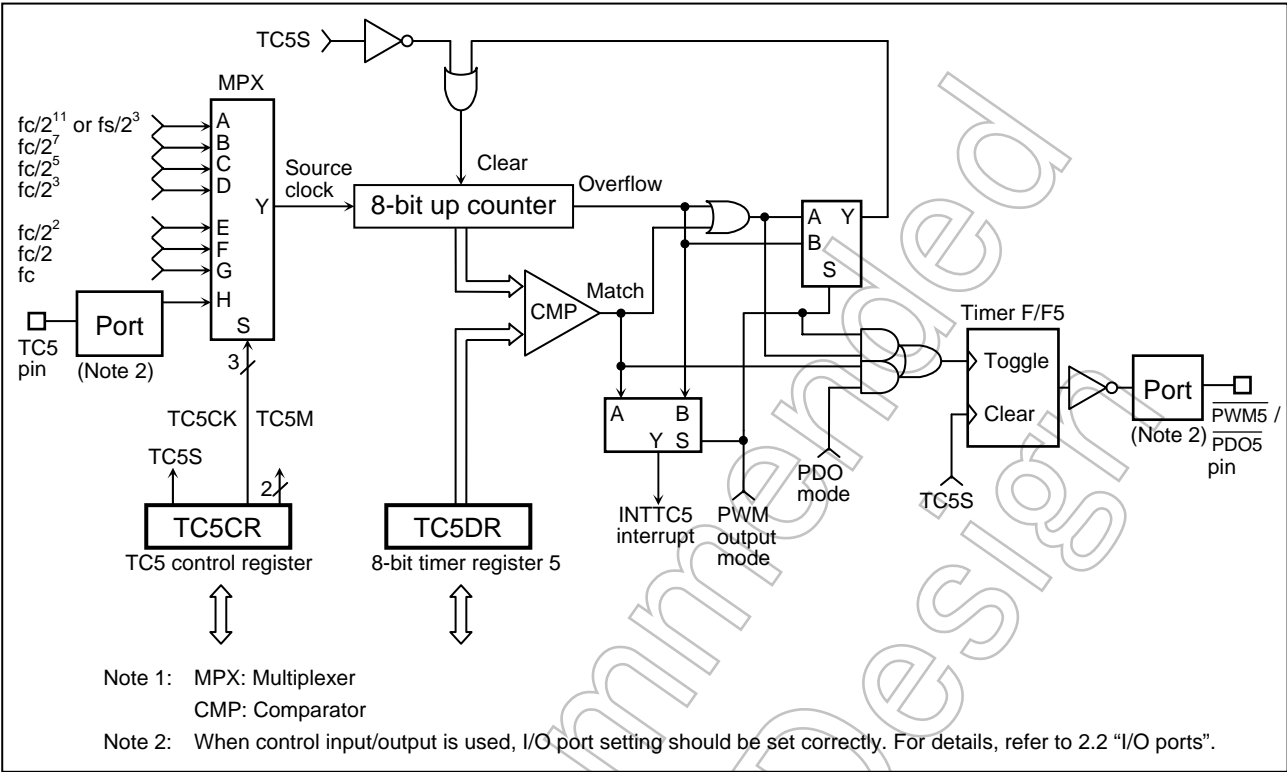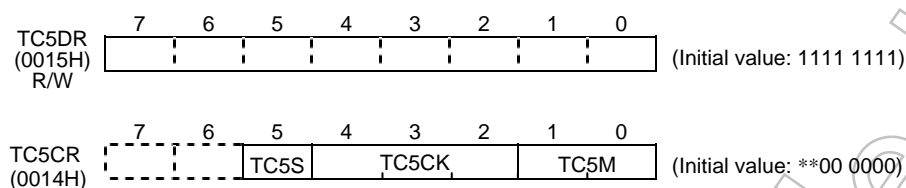
2.9.1   Configuration

Figure 2.9.1 Timer/Counter 5 (TC5)

2.9.2   Control

The timer/counter 5 is controlled by a timer/counter 5 control register (TC5CR) and an 8-bit timer register 5 (TC5DR). Reset does not affect TC5DR.

TC5DR
(0015H)
R/W

7	6	5	4	3	2	1	0

(Initial value: 1111 1111)

TC5CR
(0014H)

7	6	5	4	3	2	1	0
		TC5S		TC5CK		TC5M	

(Initial value: **00 0000)

TC5S	TC5 start control	0: Stop and counter clear 1: Start			
TC5CK	TC5 source clock select [Hz]		NORMAL1/2, IDLE1/2 Modes		SLOW1/2, SLEEP1/2 Modes
			DV7CK = 0	DV7CK = 1	
		000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$
		001	$fc/2^7$	$fc/2^7$	–
		010	$fc/2^5$	$fc/2^5$	–
		011	$fc/2^3$	$fc/2^3$	–
		100	$fc/2^2$	$fc/2^2$	–
		101	$fc/2$	$fc/2$	–
		110	fc	fc	–
		111	External clock (TC5 pin input)		
TC5M	TC5 operating mode select	00: Timer/event counter mode 01: Reserved 10: Programmable divider output (PDO) mode 11: Pulse width modulation (PWM) output mode			

R/W

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

Note 2:   Values to be loaded to the timer register must satisfy the following condition.

$$1 \leq TC5DR \leq 255$$

Note 3:   When TC5 operation is started (TC5S = "0" → "1") or TC5 operation is stopped (TC5S = "1" → "0"), do not change TC5CR <TC5M, TC5CK>. Also, during TC5 operation (TC5S = "1" → "1"), do not change TC5CR <TC5M, TC5CK>.

Note 4:   Available source clocks for each operation mode is referred to the following table.

TC5CK		Timer Mode	Event Counter Mode	PDO Mode	PWM Mode
	000	○	×	○	×
	001	○	×	○	×
	010	○	×	○	×
	011	○	×	×	○
	100	×	×	×	○
	101	×	×	×	○
	110	×	×	×	○
	111	×	○	×	×

Note 5:   The TC5S is automatically cleared to "0" after starting STOP mode.

Note 6:   If a read instruction is executed for TC5CR, read data of bits 7 and 6 are unstable.

Note 7:   During TC5 operation except PWM mode, do not change TC5DR.

Figure 2.9.2 Timer Register 5 and TC5 Control Register

2.9.3   Function

The timer/counter 5 has four operating modes: Timer, event counter, programmable divider output, and PWM output mode.

(1) Timer mode

In this mode, the internal clock is used for counting up. The contents of TC5DR is compared with the contents of up counter. If a match is found, an INTTC5 interrupt is generated and the up counter is cleared to "0". Counting up resumes after the up counter is cleared.

Table 2.9.1 Source Clock (internal clock) for Timer/Counter 5 (Example: at fc = 16 MHz)

| TC5CK | NORMAL1/2, IDLE1/2 Modes | | | | SLOW1/2 Modes | |
| | DV7CK = 0 | | DV7CK = 1 | | Resolution [μs] | Maximum Time Setting [ms] |
	Resolution [μs]	Maximum Time Setting [ms]	Resolution [μs]	Maximum Time Setting [ms]		
000	128.0	32.6	244.14	62.3	244.14	62.3
001	8.0	2.0	8.0	2.0	–	–
010	2.0	0.510	2.0	0.510	–	–
011	0.5	0.128	0.5	0.128	–	–

(2) Event counter mode

In this mode, events are counted on the rising edge of the TC5 pin input (External clock).

The contents of the TC5DR is compared with the contents of the up counter. If a match is found, an INTTC5 interrupt is generated and the counter is cleared. Counting up resumes after the up counter is cleared. The minimum input pulse width of the TC5 pin is shown in Table 2.9.2. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

Match detect is executed on the falling edge of the TC5 pin. A match can not be detected and INTTC5 interrupt is not generated when the pulse is still in a falling state.

Note: In SLOW1/2 and SLEEP1/2 modes, because external clock is not received, can not use the event counter mode.

Table 2.9.2 Timer/Counter 5 External Clock Source

| | Minimum Input Pulse Width [s] |
	NORMAL1/2, IDLE1/2 Modes
"H" Width	2^3/fc
"L" Width	2^3/fc

(3) Programmable divider output (PDO) mode

The programmable divider output (PDO) mode is intended to output a pulse having a duty cycle of about 50%. The counter counts up on an internal source clock. If the timer value matches TC5DR, the timer F/F5 is inverted, and the counter is cleared, generating an INTTC5 interrupt. The counter keeps counting up, and the timer F/F5 is inverted each time the timer value matches TC5DR. The P13 ($\overline{PDO5}$) pin outputs an inversion of the timer F/F5 output level.

At a reset or when the timer stops, the timer F/F5 is cleared to "0". So, stopping the timer when the PDO output is low may cause the duty cycle to become smaller than the set value.

To use the programmable divider output mode, set the output latch of the P13 port to "1".

```
Example:  Output a 1024 Hz pulse (at fc = 16 MHz).
          LD      (TC5CR), 00000110B    ;   Sets PDO mode.
                                            (TC5M = 10, TC5CK = 001)
          SET     (P1DR). 3             ;   P13 output latch ← 1
          LD      (TC5DR), 3DH          ;   1/1024 ÷ 2⁷/fc ÷ 2 = 3DH
          LD      (TC5CR), 00100110B    ;   Starts TC5.
```
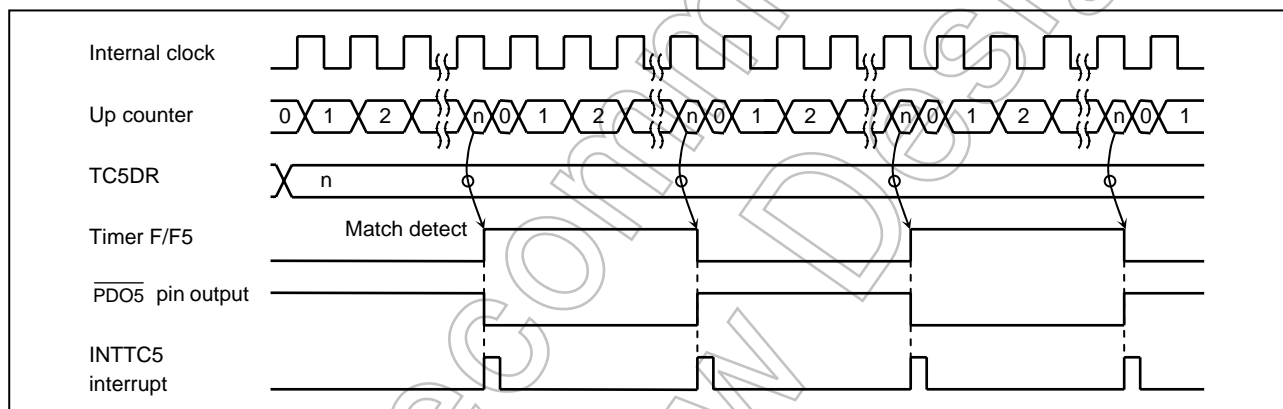

Figure 2.9.3 PDO Mode Timing Chart

(4) Pulse width modulation (PWM) output mode

The pulse width modulation (PWM) output mode is intended to output pulses at constant intervals with a resolution of 8 bits. The counter counts up on the internal source clock. If the timer value matches TC5DR, the timer F/F5 is inverted, and the counter keeps up counting. If an overflow is detected, the timer F/F5 is inverted again, generating an INTTC5 interrupt. The P13 ($\overline{\text{PWM5}}$) pin outputs an inversion of the timer F/F5 output level.

At a reset or when the timer stops, the timer F/F5 is cleared to "0". So, stopping the timer when the PWM output is low may cause one cycle to become smaller than the set value.

To use the pulse width modulation (PWM) output mode, set the output latch of the P13 port to "1".

TC5DR is configured a 2-stage shift register and, during pulse width, will not switch until one output cycle is completed even if TC5DR is overwritten; therefore, pulse width can be altered continuously. Also, the first time, TC5DR is shifted by setting TC5CR<TC5S> to "1" after data are loaded to TC5DR.

Note: In PWM mode, writing to the timer register TC5DR should be performed only right after an INTTC5 interrupt occurs (Usually, within the INTTC5 interrupt service routine). If writing to the timer register TC5DR occurs at the same timing as the INTTC5 interrupt, pulses having a value other than the set value may be output before another INTTC5 interrupt occurs, because an unstable value that is being written is shifted.
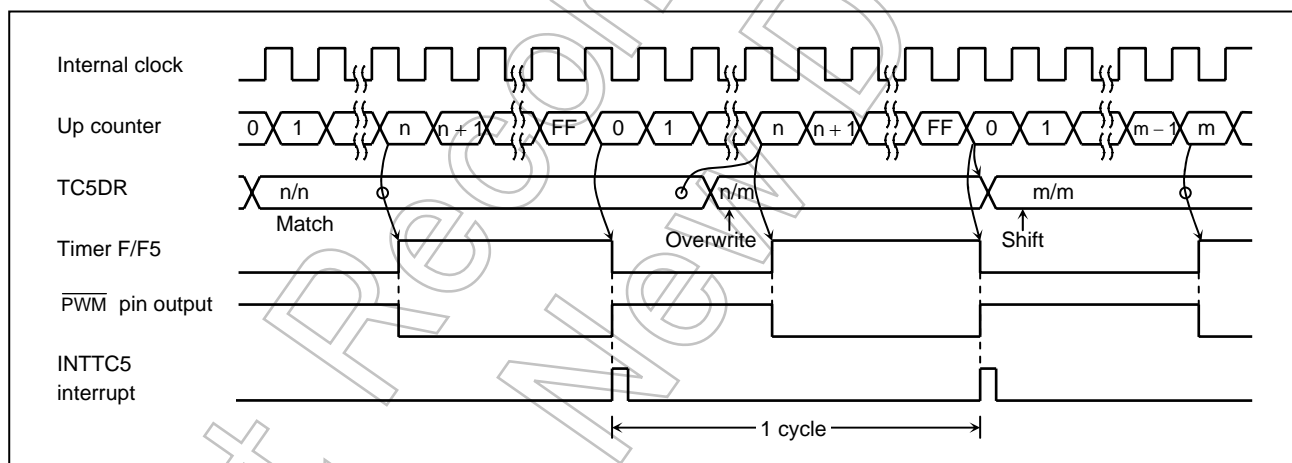

Figure 2.9.4 PWM Output Mode Timing Chart

Table 2.9.3 PWM Output Mode (Example: at fc = 16 MHz)

TC5CK	NORMAL1/2, IDLE1/2 Modes	
	Resolution [ns]	Repeat Cycle [μs]
000	−	−
001	−	−
010	−	−
011	500	128
100	250	64
101	125	32
110	62.5	16

2.10 UART (Asynchronous serial interface)

The TMP86FP24 has 1 channel of UART (Asynchronous serial interface).

The UART is connected to external devices via RXD and TXD. RXD is also used as P05; TXD, as P06. To use P05 or P06 as the RXD or TXD pin, set P0 port output latches to "1".
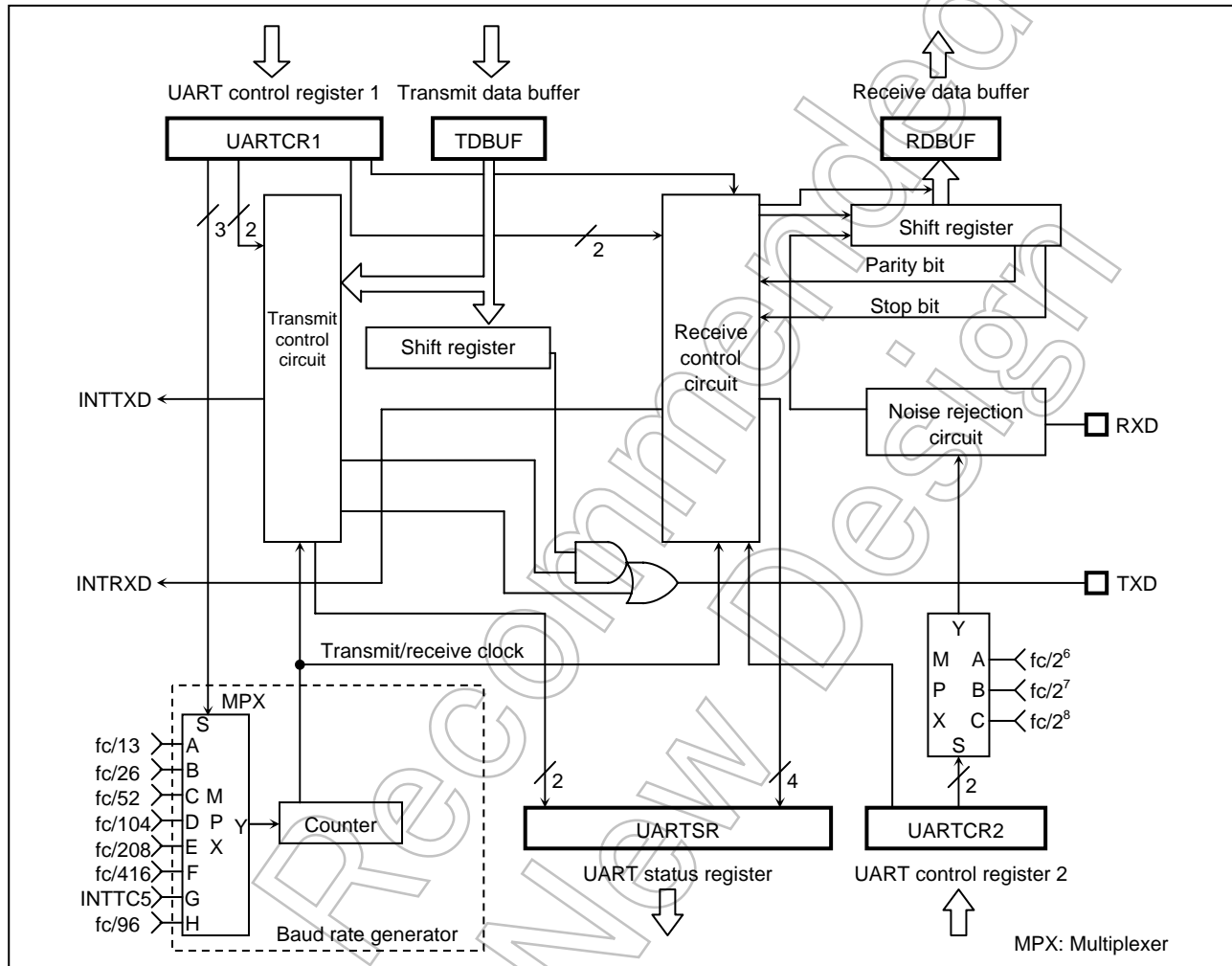
2.10.1 Configuration

Figure 2.10.1 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
(1FDDH)

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: TC5 (INTTC5) 111: fc/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.

Note 3: UARTCR1<RXE> and UARTCR1<TXE> should be set to "0" before UARTCR1<BRG> is changed

UARTCR2
(1FDEH)

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: When UARTCR2<RXDNC> = "01", pulses longer than 96/fc [s] are always regarded as signals; when UARTCR2<RXDNC> = "10", longer than 192/fc [s]; and when UARTCR2<RXDNC> = "11", longer than 384/fc [s]
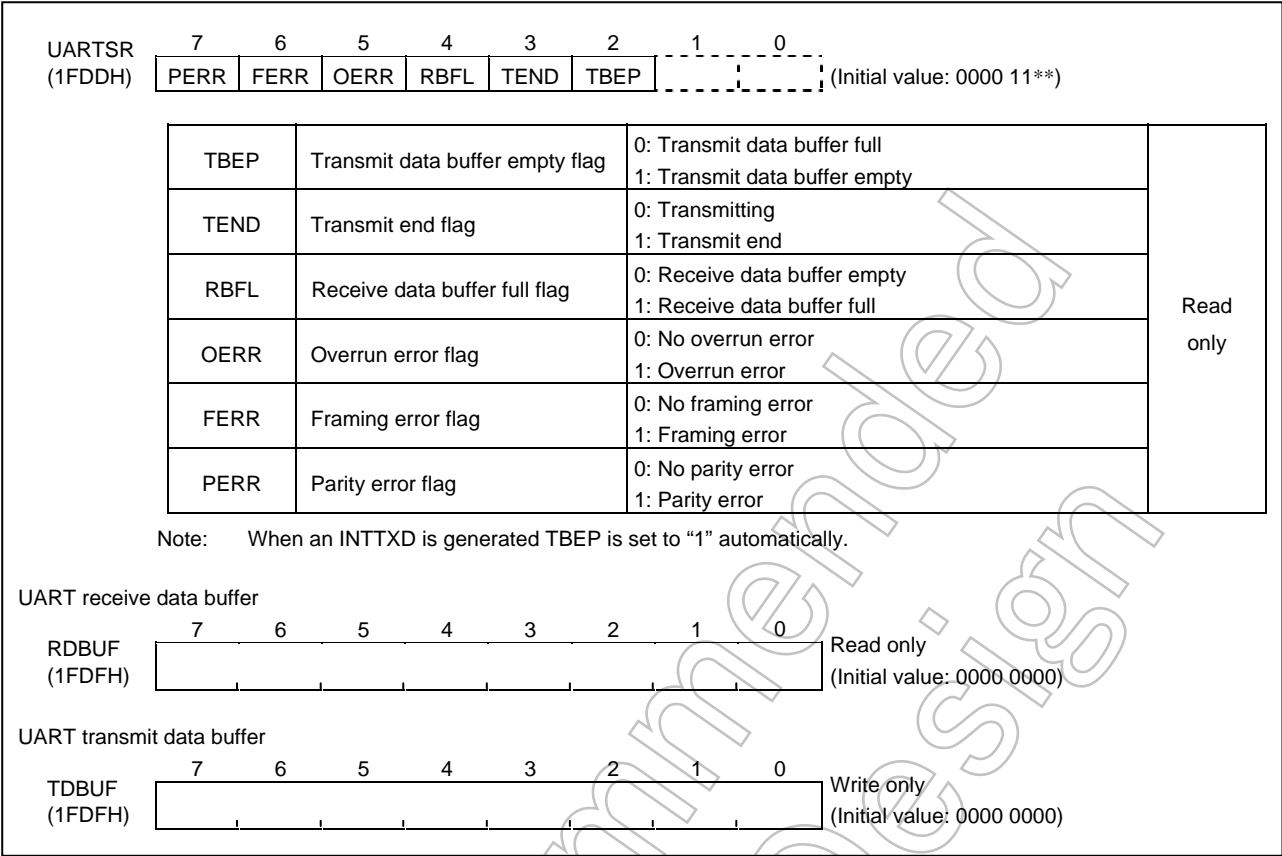
Figure 2.10.2 UART Control Register

UARTSR (1FDDH)	7 PERR	6 FERR	5 OERR	4 RBFL	3 TEND	2 TBEP	1	0	(Initial value: 0000 11∗∗)

TBEP	Transmit data buffer empty flag	0: Transmit data buffer full	
		1: Transmit data buffer empty	
TEND	Transmit end flag	0: Transmitting	
		1: Transmit end	
RBFL	Receive data buffer full flag	0: Receive data buffer empty	Read only
		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	

Note: When an INTTXD is generated TBEP is set to "1" automatically.

UART receive data buffer

RDBUF (1FDFH)	7	6	5	4	3	2	1	0	Read only (Initial value: 0000 0000)

UART transmit data buffer

TDBUF (1FDFH)	7	6	5	4	3	2	1	0	Write only (Initial value: 0000 0000)

Figure 2.10.3 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 UARTCR1<STBT>), and parity (Select parity in UARTCR1<PE>; even- or odd-numbered parity by UARTCR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follow.

Table 2.10.1 Transfer Data Format

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

Note: In order to switch the transmit data format, perform transmit operations in the following sequence except for the initial setting.
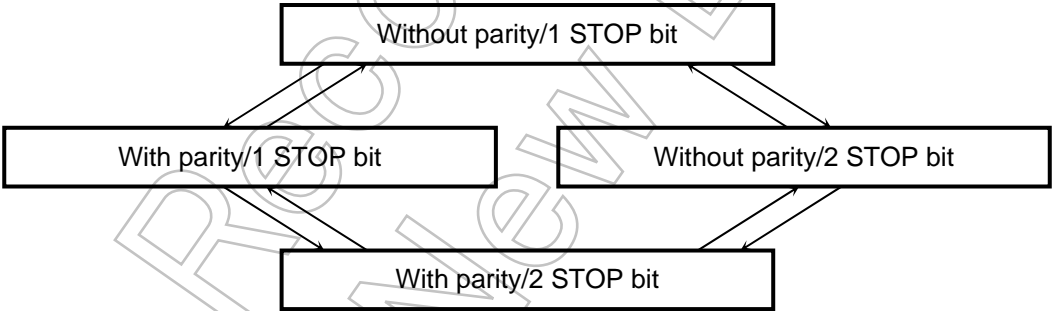
2.10.4 Transfer Rate

The baud rate of UART is set of UARTCR1<BRG>. The example of the baud rate shown as follows.

Table 2.10.2 Transfer Rate

BRG	Source Clock		
	16 MHz	8 MHz	4 MHz
000	76800 [baud]	38400 [baud]	19200 [baud]
001	38400	19200	9600
010	19200	9600	4800
011	9600	4800	2400
100	4800	2400	1200
101	2400	1200	600

When TC5 is used as the UART transfer rate (when UARTCR1<BRG> = "110"), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock} = \frac{\text{TC5 source clock}}{\text{TTREG5 set value}}$$

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

2.10.5 Data Sampling

The UART receiver keeps sampling input using the clock selected by UARTCR1<BRG> until a start bit is detected in RXD pin input. RT clock starts detecting "L" level 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).
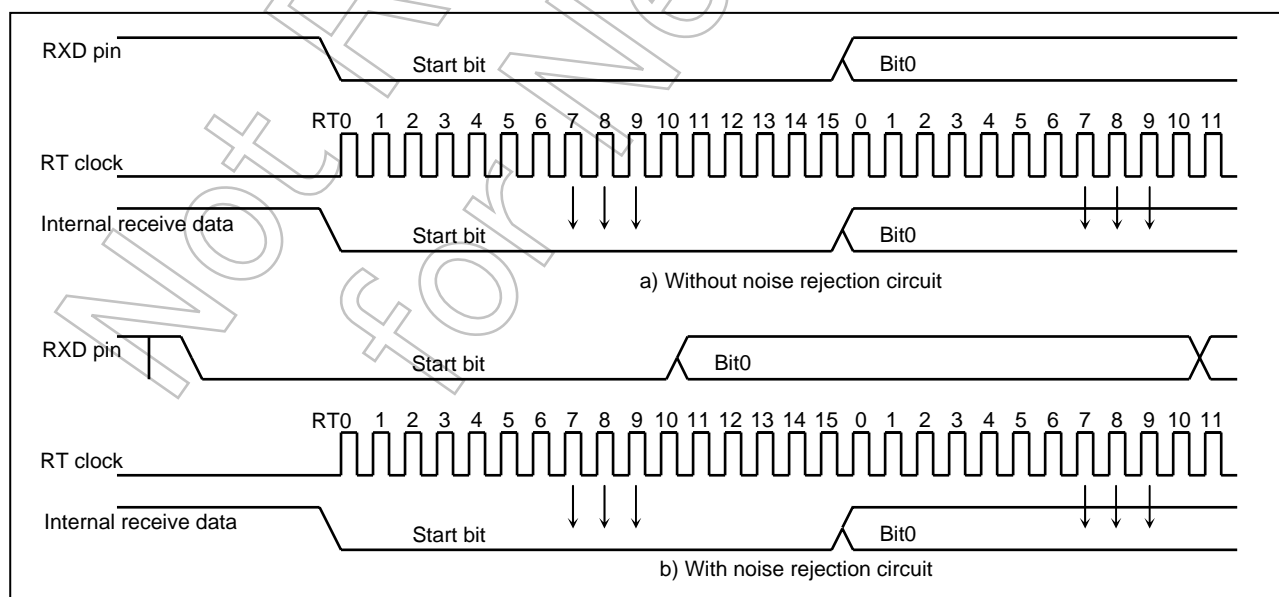

Figure 2.10.4 Data Sampling

2.10.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UARTCR1<STBT>.

2.10.7 Parity

Set parity/no parity by UARTCR1<PE>; set parity type (odd- or even-numbered) by UARTCR1<EVEN>.

2.10.8 Transmit/Receive

(1) Data transmit

Set UARTCR1<TXE> to "1". Read UARTSR to check UARTSR<TBEP> = "1", then write data in TDBUF (Transmit data buffer). Writing data in TDBUF zero-clears UARTSR<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 UARTCR1<STBT> 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 UARTSR<TBEP> is set to "1" and an INTTXD interrupt is generated.

While UARTCR1<TXE> = "0" and from when "1" is written to UARTCR1<TXE> to when send data are written to TDBUF, the TXD pin is fixed at high level. When transmitting data, first read UARTSR, then write data in TDBUF. Otherwise, UARTSR<TBEP> is not zero cleared and transmit does not start.

(2) Data receive

Set UARTCR1<RXE> 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 UARTSR<RBFL> is set and an INTRXD 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.

Note: When a receive operation is disabled by setting UARTCR1<RXE> bit to "0", the setting becomes valid when data receive is completed. However, if a framing error occurs in data receive, the receive-disabling setting may not become valid. if a framing error occurs, be sure to perform a re-receive operation.

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 UARTSR<PERR> is set to "1". The UARTSR<PERR> is cleared to "0" when the RDBUF is read after reading the UARTSR.
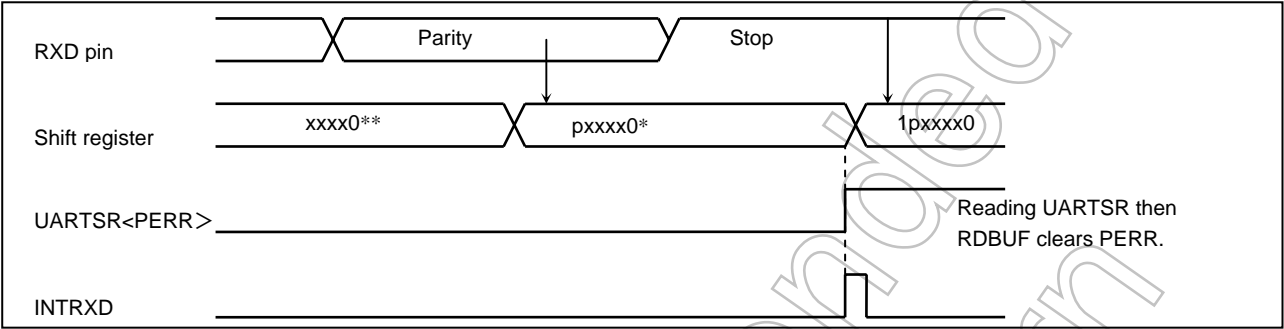

Figure 2.10.5 Generation of Parity Error

（2） Framing error

When "0" is sampled as the stop bit in the receive data, framing error flag UARTSR<FERR> is set to "1". The UARTSR<FERR> is cleared to "0" when the RDBUF is read after reading the UARTSR.
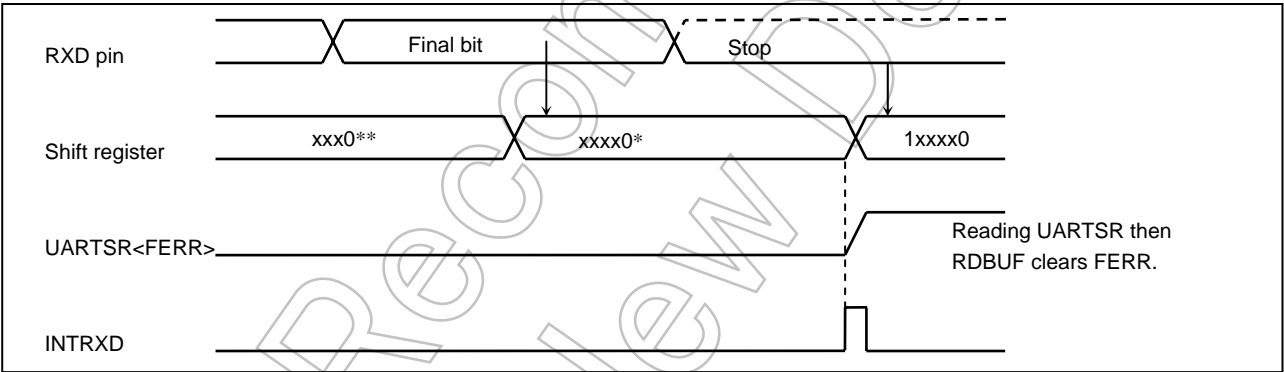

Figure 2.10.6 Generation of Framing Error

(3) Overrun error

When all bits in the next data are received while unread data are still in RDBUF, overrun error flag UARTSR<OERR> is set to "1". In this case, the receive data is discarded; data in RDBUF are not affected. The UARTSR<OERR> is cleared to "0" when the RDBUF is read after reading the UARTSR.
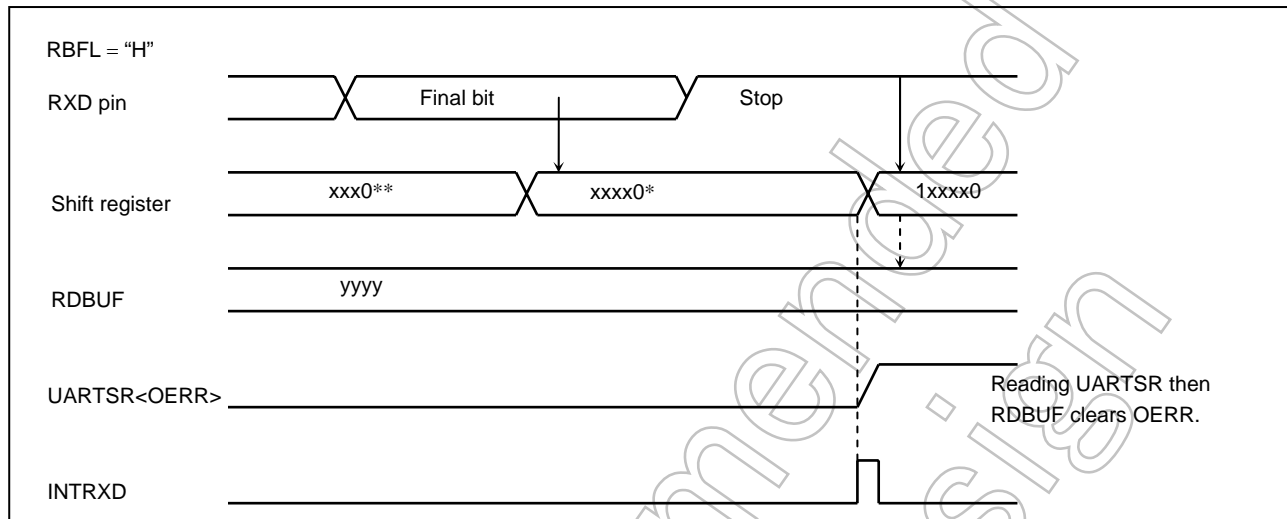

Figure 2.10.7 Generation of Overrun Error

(4) Receive data buffer full

Loading the received data in RDBUF sets receive data buffer full flag UARTSR<RBFL>. The UARTSR<RBFL> is cleared to "0" when the RDBUF is read after reading the UARTSR.
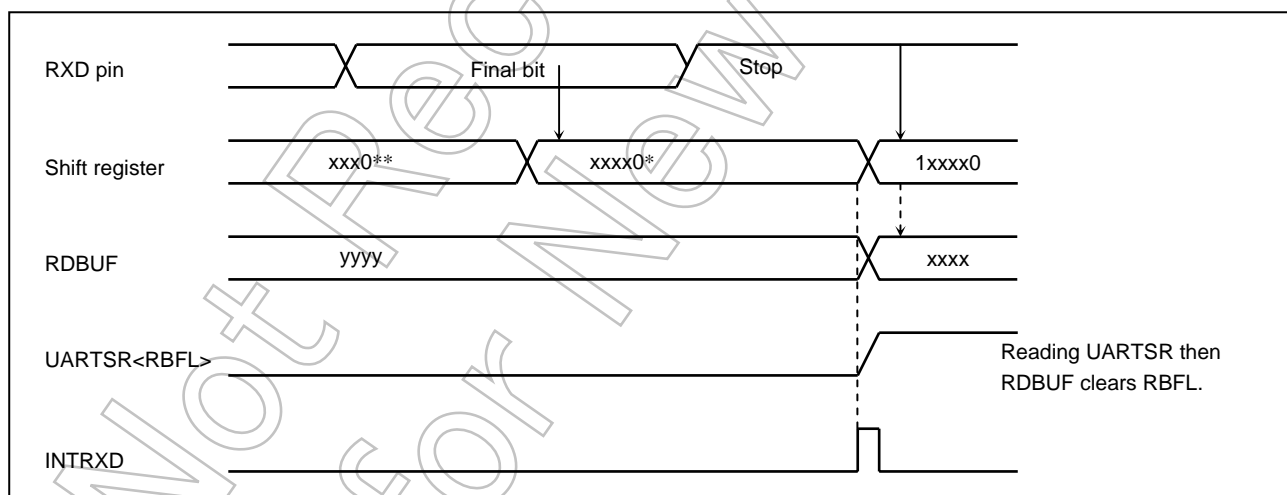

Figure 2.10.8 Generation of Receive Buffer Full

(5) Transmit data buffer empty

When no data is in the transmit buffer TDBUF, UARTSR<TBEP> is set to "1", that is, when data in TDBUF are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UARTSR<TBEP> is set to "1". The UARTSR<TBEP> is cleared to "0" when the TDBUF is written after reading the UARTSR.
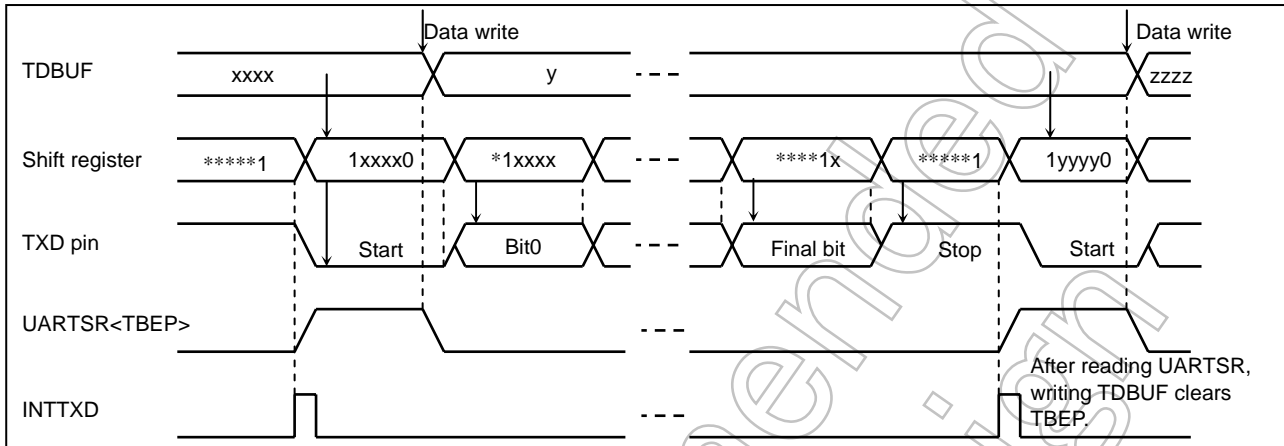

Figure 2.10.9 Generation of Transmit Buffer Empty

(6) Transmit end flag

When data are transmitted and no data is in TDBUF (UARTSR<TBEP> = "1"), transmit end flag UARTSR<TEND> is set to "1". The UARTSR<TEND> is cleared to "0" the data transmit is stated after writing the TDBUF.
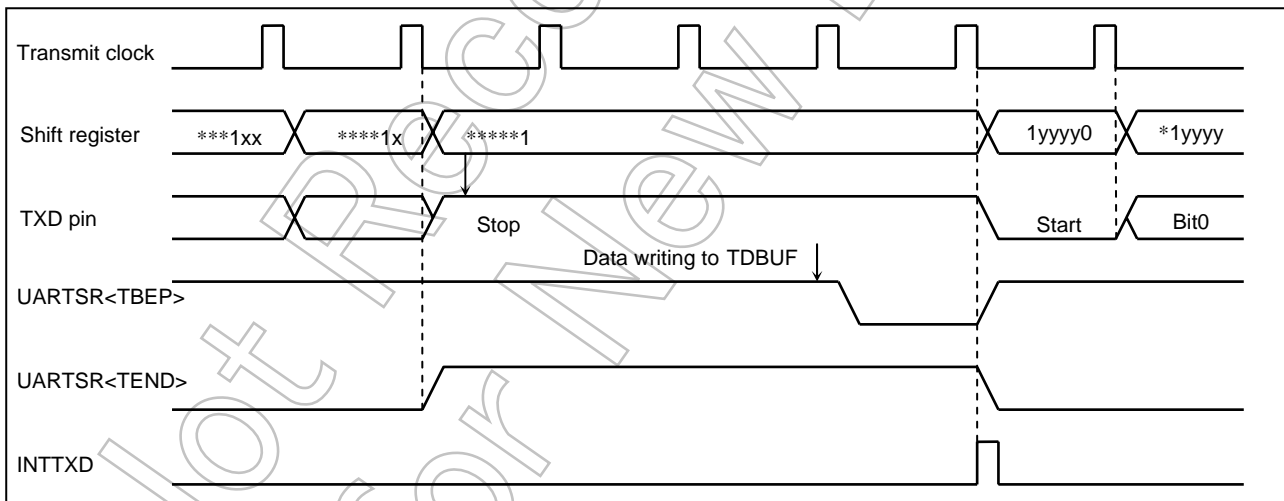

Figure 2.10.10 Generation of Transmit Buffer Empty

2.11 Key-on wakeup (KWU)

In the TMP86FP24, the STOP mode must be released by not only P20 ($\overline{INT5}$ / $\overline{STOP}$) pin but also P64 to P67 and P40 pins.

When the STOP mode is released by P40, P64 to P67 pins, the P20 ($\overline{INT5}$ / $\overline{STOP}$) pin needs to be used.

2.11.1 Configuration
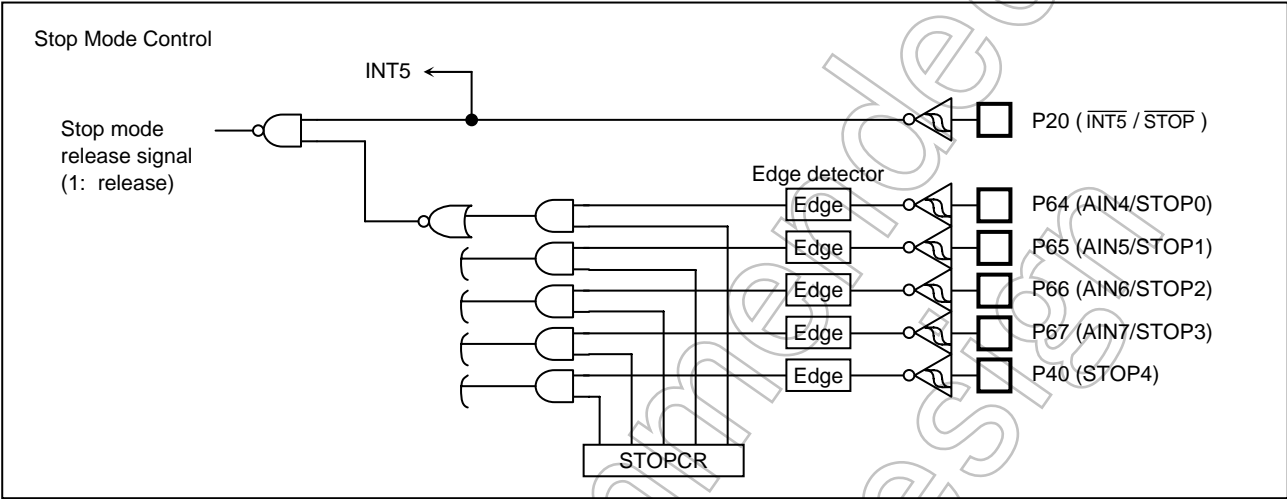
Figure 2.11.1 Key-on wakeup Circuit

Table 2.11.1 Input Edge (Level) of STOP Mode Release

Terminal Name	As Both Terminal	SYSCR1<RELM> = "1"	SYSCR1<RELM> = "0"
		Release Edge (Level)	
STOP	P20/$\overline{INT5}$	"H" level	Rising edge
STOP0	P64/AIN4	Rising/falling edge	Do not use key-on wakeup function (Note)
STOP1	P65/AIN5		
STOP2	P66/AIN6		
STOP3	P67/AIN7		
STOP4	P40		

Note: When the SYSCR1<RELM> is "0", all bit of STOPCR should be cleared to "0".

2.11.2 Control

P64 to P67 (STOP0 to STOP3) and P40(STOP4) pins can controlled by key-on wakeup control register (STOPCR). It can be configured as enable/disable in one-bit unit.

The STOP mode is started by SYSCR1<STOP> = "1". In this case, the release method for STOP mode should be set to level mode (SYSCR1<RELM> = "1").

STOP mode is released by the "H" level of STOP pin, the release edge of STOP0 to STOP4 pins which enabled setting by STOPCR.

(1) When only a key-on wakeup pin is used.

When using only the key-on wakeup pin for releasing STOP mode, P20 should be fixed to low externally or protected against high-level input.

(2) When both P20 pin and key-on wakeup pins are used.

When stop mode is entered while the P20 pin input is high, the microcomputer immediately starts warm-up without entering stop mode. Therefore, it is recommended that before entering stop mode, the P20 pin be read in software to see that its input is at the low level.

Note 1: The P20 pin is shared with external interrupt 5 (INT5). Even when using P20 as the STOP pin, inputs to the interrupt controller are effective, so that the INT5 interrupt latch may possibly be set entering or exiting stop mode.

Note 2: When stop mode is entered, the P20 output goes to a high-impedance state regardless of how SYSCR1<OUTEN> is set. Therefore, the P20 pin cannot be fixed low with its own output latch. When fixing the P20 pin low during stop mode, make sure it is fixed low external to the chip.

Note 3: When using a key-on wakeup function, SYSCR1<RELM> should be set to "1" (Level mode). The edge mode should not be used.

Key-on Wakeup Control Register

STOPCR (1FFEH)	7 STOP0	6 STOP1	5 STOP2	4 STOP3	3 STOP4	2 −	1 −	0 −	(Initial value: 0000 0***)

STOP0	Stop mode released by P64 port	0: Disable / 1: Enable	
STOP1	Stop mode released by P65 port	0: Disable / 1: Enable	
STOP2	Stop mode released by P66 port	0: Disable / 1: Enable	R/W
STOP3	Stop mode released by P67 port	0: Disable / 1: Enable	
STOP4	Stop mode released by P40 port	0: Disable / 1: Enable	

Figure 2.11.2 Key-on Wakeup Control Register

2.12 10-Bit AD Converter (ADC)

The TMP86FP24 has a 10-bit successive approximation type AD converter.

2.12.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 2.12.1.

It consists of control registers ADCCR1 and ADCCR2, conversion result registers ADCDR1 and ADCDR2, a DA converter, a sample-and-hold circuit, a comparator, and a successive comparison circuit.
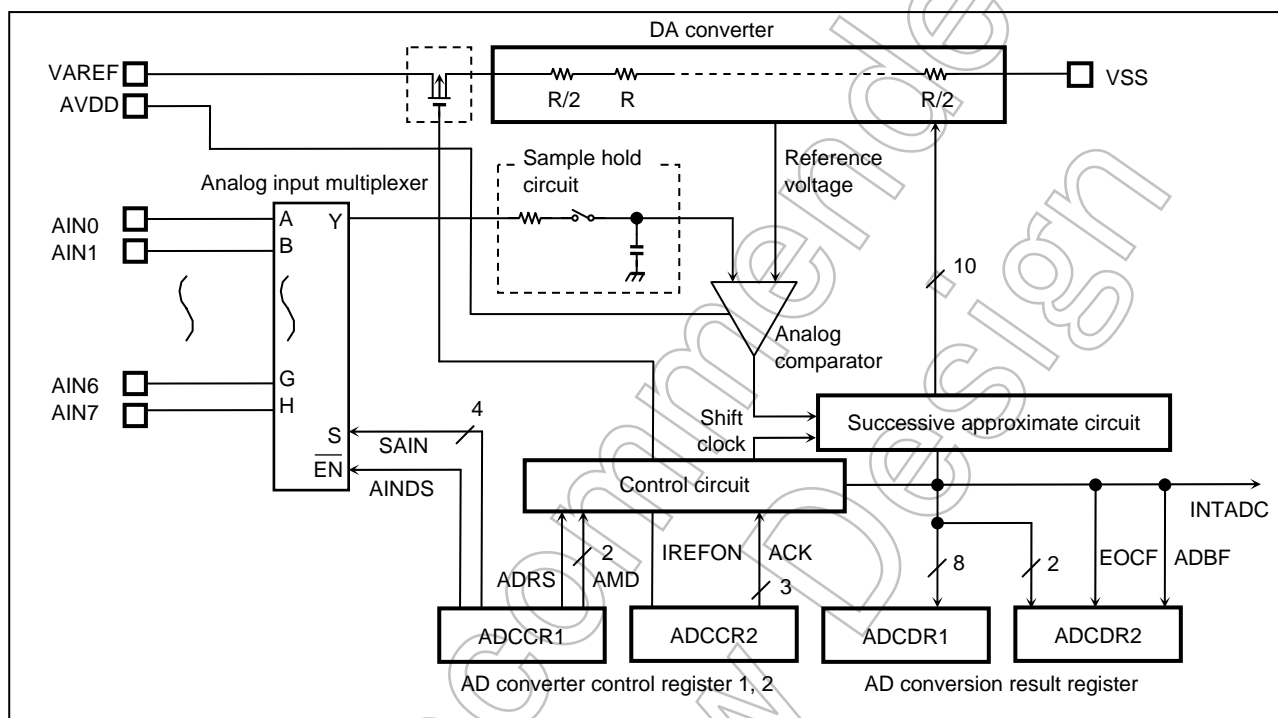

Figure 2.12.1 AD Converter (ADC)

2.12.2 Register Configuration

The AD converter consists of the following four registers:

- AD converter control register 1 (ADCCR1)
- AD converter control register 2 (ADCCR2)
- AD conversion result register 1/2 (ADCDR1/ADCDR2)

(1) AD converter control register 1 (ADCCR1)

This register selects the analog channels and operation mode (Software start or repeat) in which to perform AD conversion and controls the AD converter as it starts operating.

(2) AD converter control register 2 (ADCCR2)

This register selects the AD conversion time and controls the connection of the DA converter (Ladder resistor network).

(3) AD conversion result register (ADCDR1)

This register is used to store the digital value (Bit9 to bit2) after being converted by the AD converter.

(4) AD conversion result register (ADCDR2)

This register is used to store the digital value (Bit1 and bit0) after being converted by the AD converter, and then this register is also used to monitor the operating status of the AD converter.

The AD converter control register configurations are shown in Figure 2.12.2 and Figure 2.12.3.

AD Converter Control Register 1

ADCCR1	7	6	5	4	3	2	1	0	
(000EH)	ADRS	AMD		AINDS		SAIN			(Initial value: 0001 0000)

				R/W
ADRS	AD conversion start	0: − 1: Start		
AMD	AD Operating mode	00: AD operation disable 01: Software start mode 10: Reserved 11: Repeat mode		
AINDS	Analog input control	0: Analog input enable 1: Analog input disable		
SAIN	Analog input channel select	0000: Selects AIN0 0001: Selects AIN1 0010: Selects AIN2 0011: Selects AIN3 0100: Selects AIN4 0101: Selects AIN5 0110: Selects AIN6 0111: Selects AIN7 1***: Reserved		

Note 1: Select analog input when AD converter stops (ADCDR2<ADBF> = "0").
Note 2: When the analog input is all use disabling, the AINDS should be set to "1".
Note 3: During conversion, do not perform output instruction to maintain a precision for all of the pins. And port near to analog input, do not input intense signaling of change.
Note 4: The ADRS is automatically cleared to "0" after starting conversion.
Note 5: Do not set ADRS (ADCCR1 bit7) newly again during AD conversion. Before setting ADRS newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).
Note 6: After STOP or SLOW mode are started, AD converter control register 1 (ADCCR1) is all initialized. Therefore, set the ADCCR1 newly again after exiting these modes.

AD Converter Control Register 2

ADCCR2	7	6	5	4	3	2	1	0	
(000FH)			IREFON	"1"		ACK		"0"	(Initial value: **00 0000)

								R/W
IREFON	DA converter (Ladder resistor) connection control	Inputting current to the ladder resistor 0: Connected only during AD conversion 1: Always connected						
ACK	AD conversion time select	ACK	Conversion time	fc = 16 MHz	fc = 8 MHz	fc = 4 MHz	fc = 1 MHz	
		000	39/fc	−	−	−	39.0 μs	
		001	Reserved					
		010	78/fc	−	−	−	78.0 μs	
		011	156/fc	−	−	39.0 μs	156.0 μs	
		100	312/fc	−	39.0 μs	78.0 μs	−	
		101	624/fc	39.0 μs	78.0 μs	156.0 μs	−	
		110	1248/fc	78.0 μs	156.0 μs	−	−	
		111	Reserved					

Note 1: Settings for "−" in the above table are inhibited.
Note 2: Set conversion time by analog reference voltage (V_{AREF}) as follows.

V_{AREF} = 2.7 to 3.6 V (31.2 μs or more)

V_{AREF} = 1.8 to 3.6 V (124.8 μs or more)

Note 3: Always set bit0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".
Note 4: When a read instruction for ADCCR2, bit6 to bit7 in ADCCR2 read in as undefined data.
Note 5: fc; High-frequency clock [Hz]
Note 6: After STOP or SLOW mode are started, AD converter control register 2 (ADCCR2) is all initialized. Therefore, set the ADCCR2 newly again after exiting these modes.

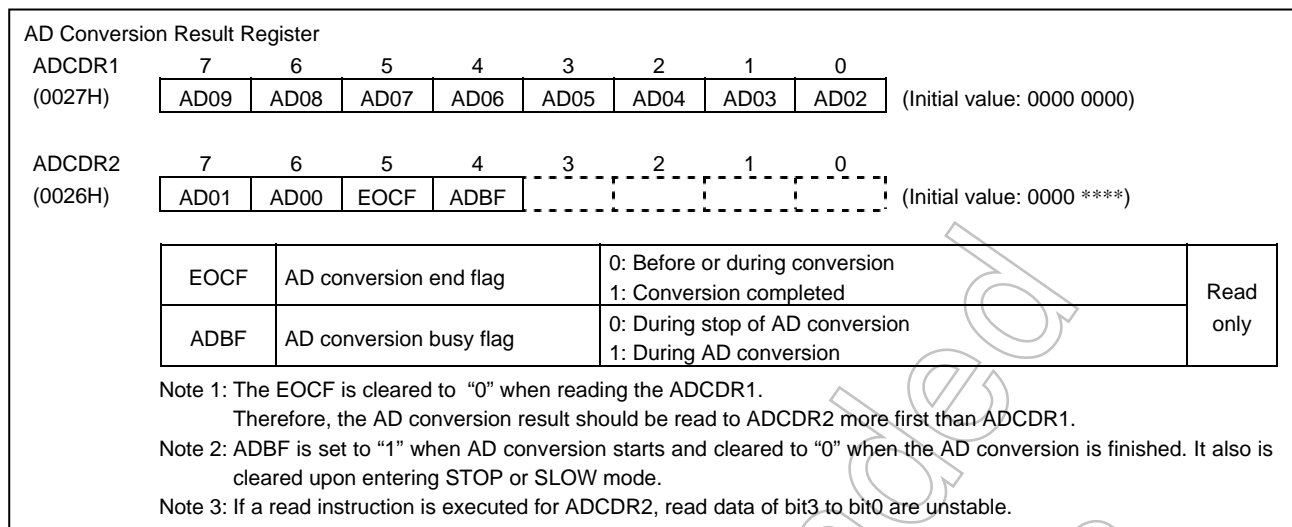Figure 2.12.2 AD Converter Control Register

AD Conversion Result Register

ADCDR1	7	6	5	4	3	2	1	0	
(0027H)	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	(Initial value: 0000 0000)

ADCDR2	7	6	5	4	3	2	1	0	
(0026H)	AD01	AD00	EOCF	ADBF					(Initial value: 0000 ****)

EOCF	AD conversion end flag	0: Before or during conversion 1: Conversion completed	Read only
ADBF	AD conversion busy flag	0: During stop of AD conversion 1: During AD conversion	

Note 1: The EOCF is cleared to "0" when reading the ADCDR1.
Therefore, the AD conversion result should be read to ADCDR2 more first than ADCDR1.

Note 2: ADBF is set to "1" when AD conversion starts and cleared to "0" when the AD conversion is finished. It also is cleared upon entering STOP or SLOW mode.

Note 3: If a read instruction is executed for ADCDR2, read data of bit3 to bit0 are unstable.

Figure 2.12.3 AD Converter Result Register

2.12.3 AD Converter Operation

(1) Set up the AD converter control register 1 (ADCCR1) as follows:

- Choose the channel to AD convert using AD input channel select (SAIN).

- Specify analog input enable for analog input control (AINDS).

- Specify AMD for the AD converter control operation mode (Software or repeat mode).

(2) Setup the AD converter control register 2 (ADCCR2) as follows:

- Set the AD conversion time using AD conversion time (ACK). For details on how to set the conversion time, refer to Note 2 for AD converter control register 2.

- Choose IREFON for DA converter control.

(3) After setting up (1) and (2) above, set AD conversion start (ADRS) of AD converter control register 1 (ADCCR1) to "1".

(4) After an elapse of the specified AD conversion time, the AD converted value is stored in AD conversion result register 1 (ADCDR1), AD conversion result register 2 (ADCDR2) and then the AD conversion end flag (EOCF) of AD conversion result register 2 (ADCDR2) is set to "1", upon which time AD conversion interrupt INTADC is generated.

(5) EOCF is cleared to "0" by a read of the conversion result. However, if reconverted before a register read, although EOCF is cleared the previous conversion result is retained until the next conversion is completed.

2.12.4 AD Converter Operation Modes

There are following two AD converter operation modes:

- Software start: AD conversion is performed once by setting AMD to "01B" and ADRS to "1".

- Repeat mode: AD conversion is performed repeatedly by setting AMD to "11B" and ADRS to "1".

(1) Software start mode

After setting ADCCR1<AMD> to "01B" (Software start mode), set ADCCR1<ADRS> to "1". AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is thereby started.

After completion of the AD conversion, the conversion result is stored in AD conversion result registers (ADCDR1, 2) and at the same time ADCDR2<EOCF> is set to "1", the AD conversion finished interrupt (INTADC) is generated.

ADCCR1<ADRS> is automatically cleared to "0" after AD conversion has started. Do not set ADCCR1<ADRS> newly again (restart) during AD conversion. Before setting ADCCR1<ADRS> newly again, check ADCDR2<EOCF> to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).
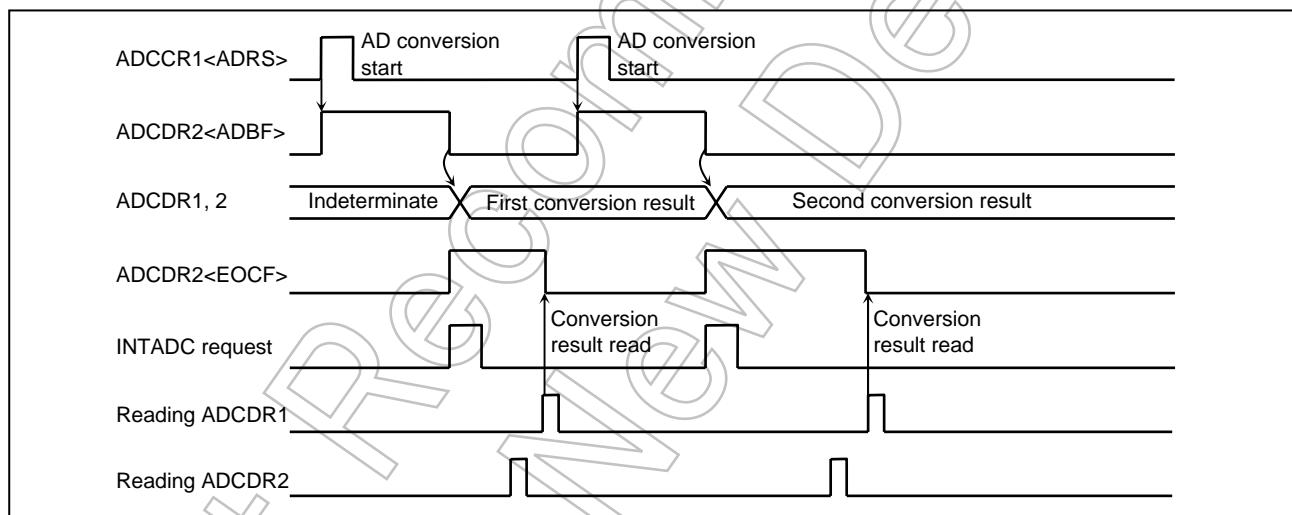

Figure 2.12.4 Operation in Software Start Mode

Example: After selecting the conversion time of 39.0 μs at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 009EH and store the upper 8 bits in address 009FH on RAM. The operation mode is software start mode.

```
            ; AIN SELECT
            LD      (P6CR1), 00000000B      ;   P6CR1 bit3 = 0
            LD      (P6CR2), 00000000B      ;   P6CR2 bit3 = 0
            LD      (ADCCR1), 00100011B     ;   Select AIN3.
            LD      (ADCCR2), 11011010B     ;   Select conversion time (624/fc) and
                                            ;   operation mode.
            ; AD CONVERT START
            SET     (ADCCR1). 7             ;   ADRS = 1
SLOOP:      TEST    (ADCDR2). 5             ;   EOCF = 1 ?
            JRS     T, SLOOP
            ; RESULT DATA READ
            LD      A, (ADCDR2)
            LD      (9EH), A
            LD      A, (ADCDR1)
            LD      (9FH), A
```

(2) Repeat mode

AD conversion of the voltage at the analog input pin specified by ADCCR1<SAIN> is performed repeatedly. In this mode, AD conversion is started by setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "11B".

After completion of the AD conversion, the conversion result is stored in AD conversion result registers (ADCDR1, 2) and at the same time ADCDR2<EOCF> is set to "1", the AD conversion finished interrupt (INTADC) is generated.

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set ADCCR1<AMD> to "00B" (Disable mode). The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD conversion result register.
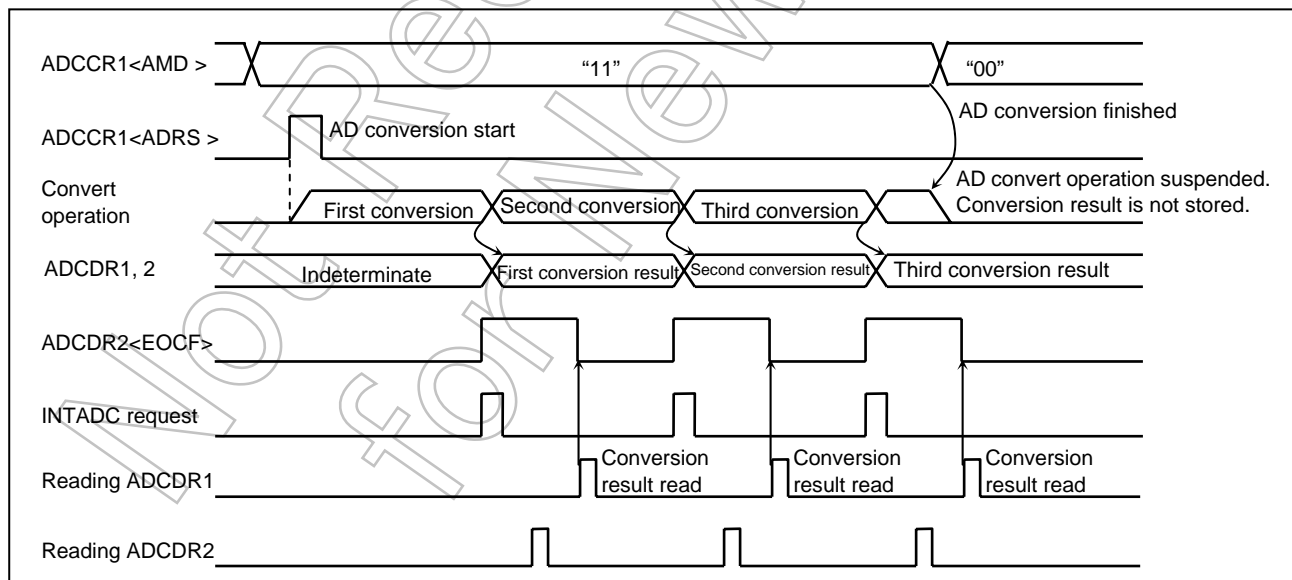

Figure 2.12.5 Operation in Repeat Mode

2.12.5 STOP and SLOW Modes during AD Conversion

When the STOP or SLOW mode is entered forcibly during AD conversion, the AD convert operation is suspended and the AD converter is initialized (ADCCR1 and ADCCR2 are initialized to initial value). Also, the conversion result is indeterminate. (Conversion results up to the previous operation are cleared, so be sure to read the conversion results before entering STOP or SLOW mode.) When released from STOP or SLOW mode, AD conversion is not automatically restarted. Therefore, when the AD converter is used again, it is necessary to restart AD conversion (Set ADCCR1<ADRS> to "1"). Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

2.12.6 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 2.12.6.
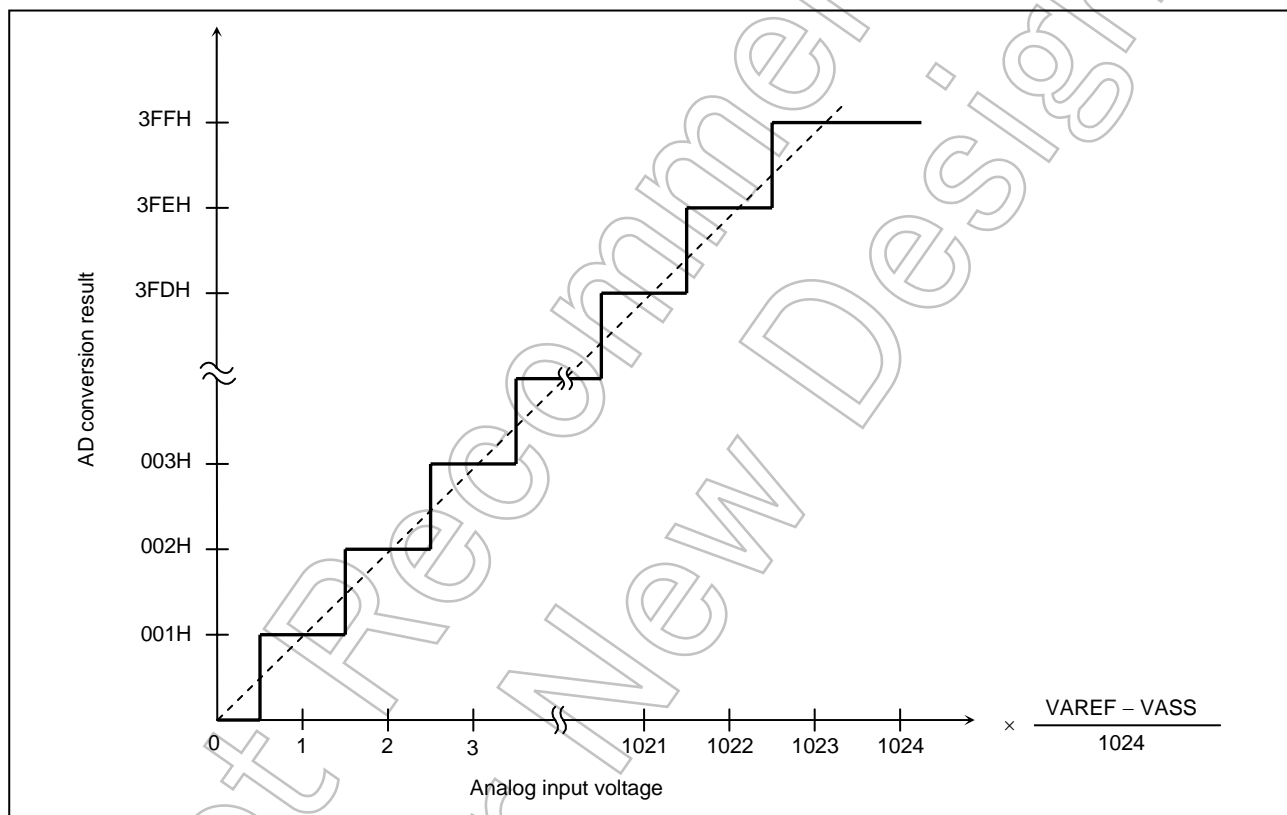

Figure 2.12.6 Analog Input Voltage and AD Conversion Result (typ.)

2.12.7 Precautions about AD Converter

（1） Analog input pin voltage range

Make sure the analog input pins (AIN0 to AIN7) are used at voltages within VSS below VAREF. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain. The other analog input pins also are affected by that.

（2） Analog input shared pins

The analog input pins (AIN0 to AIN7) are shared with input/output ports. When using any of the analog inputs to execute AD conversion, do not execute input/output instructions for all other ports. This is necessary to prevent the accuracy of AD conversion from degrading. Not only these analog input shared pins, some other pins may also be affected by noise arising from input/output to and from adjacent pins.

（3） Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 2.12.7. The higher the output impedance of the analog input source, more easily they are susceptible to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. Toshiba also recommends attaching a capacitor external to the chip.
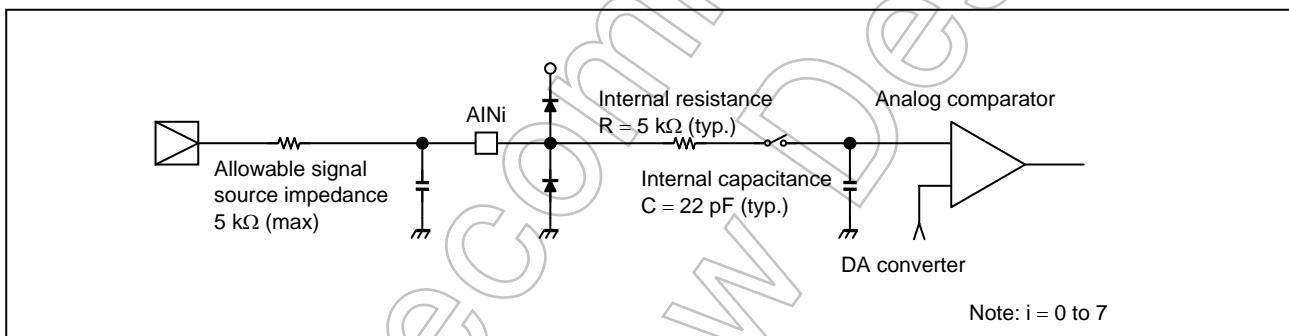

Figure 2.12.7 Analog Input Equivalent Circuit and Example of Input Pin Processing

2.13 LCD Driver

The TMP86FP24 has a driver and control circuit to directly drive the liquid crystal device (LCD). The pins to be connected to LCD are as follows:

a.	Segment output port	8 pins (SEG7 to SEG0)
b.	Segment output or P4, P9 input/output port	16 pins (SEG23 to SEG8)
c.	Common output port	4 pins (COM3 to COM0)

In addition, C0, C1, V1, V2, V3 pin are provided for the LCD driver's booster circuit.

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

a.	1/4 Duty (1/3 Bias) LCD	Max 96 Segments (8 segments × 12 digits)
b.	1/3 Duty (1/3 Bias) LCD	Max 72 Segments (8 segments × 9 digits)
c.	1/2 Duty (1/2 Bias) LCD	Max 48 Segments (8 segments × 6 digits)
d.	Static LCD	Max 24 Segments (8 segments × 3 digits)

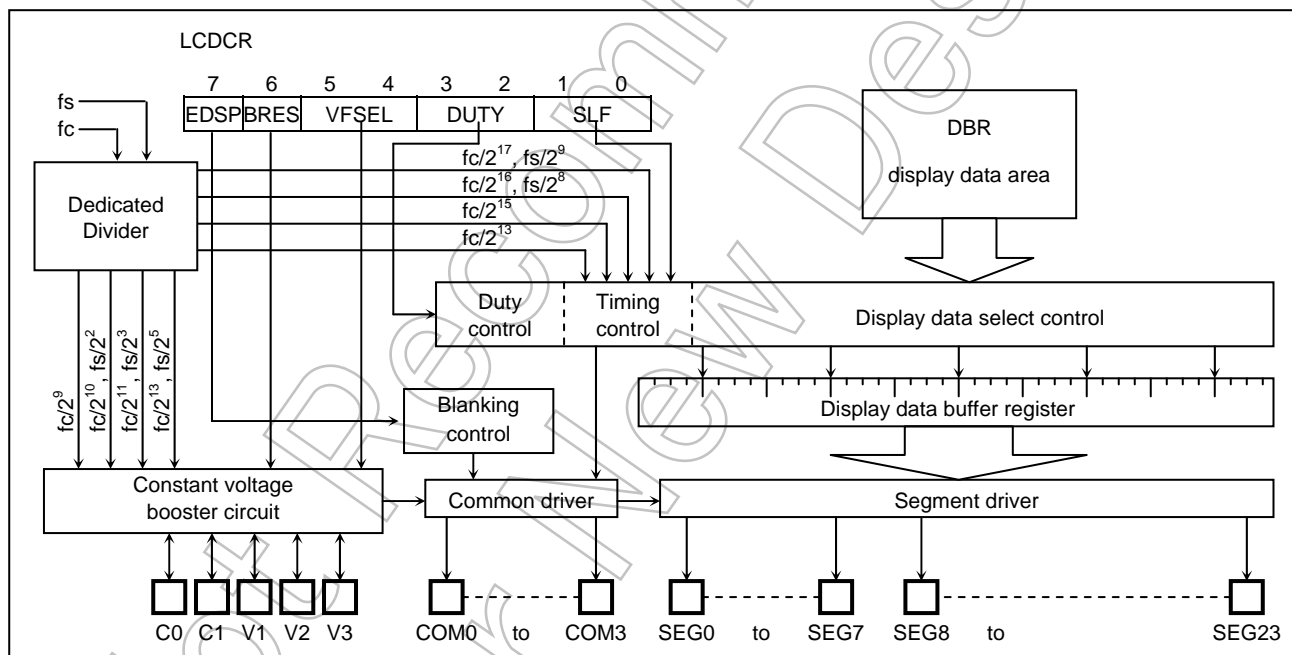### 2.13.1 Configuration

Figure 2.13.1 LDC Driver

Note: The LCD driver circuit has a built-in dedicated divider circuit. Thus, during use of the tool, LCD outputting is not stopped by debugger break processing.

2.13.2 Control

The LCD driver is controlled using the LCD control register (LCDCR). The LCD driver's display is enabled using the EDSP.

LCDCR	7	6	5	4	3	2	1	0	
(1FE3H)	EDSP	BRES	VFSEL		DUTY		SLF		(Initial value: 0000 0000)

			NORMAL1/2,IDLE0/1/2 Modes		SLOW1/2, SLEEP01/2 Modes	
EDSP	LCD display control	0: Blanking 1: Enables LCD display (Blanking is released)				R/W
BRES	Booster circuit control	0: Disable (Use divider resistance) 1: Enable				
VFSEL	Selection of boost frequency		DV7CK = 0	DV7CK = 1		
		00:	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	
		01:	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	
		10:	$fc/2^{10}$	$fs/2^{2}$	$fs/2^{2}$	
		11:	$fc/2^{9}$	$fc/2^{9}$	Reserved	
DUTY	Selection of driving methods	00: 1/4 Duty (1/3 Bias) 01: 1/3 Duty (1/3 Bias) 10: 1/2 Duty (1/2 Bias) 11: Static				
SLF	Selection of LCD frame frequency		DV7CK = 0	DV7CK = 1		
		00:	$fc/2^{17}$	$fs/2^{9}$	$fs/2^{9}$	
		01:	$fc/2^{16}$	$fs/2^{8}$	$fs/2^{8}$	
		10:	$fc/2^{15}$	$fc/2^{15}$	Reserved	
		11:	$fc/2^{13}$	$fc/2^{13}$	Reserved	

Note 1: When <BRES>(Booster circuit control) is set to "0", $V_{DD} \geq V_3 \geq V_2 \geq V_1 \geq V_{SS}$ should be satisfied.
When <BRES> is set to "1", 3.6 [V] $\geq V_3 \geq V_{DD}$ should be satisfied.
If these conditions are not satisfied, it not only affects the quality of LCD display but also may damage the device due to over voltage of the port.

Note 2: When used as the booster circuit, the VLCD setting should be composed to 1/3 bias. Therefore, do not set LCDCR<DUTY> to "10" or "11" when the booster circuit is enable.

Figure 2.13.2 LCD Driver Control Register

(1) LCD driving methods

As for LCD driving method, 4 types can be selected by DUTY (Bit3 to bit2 of LCDCR). The driving method is initialized in the initial program according to the LCD used.
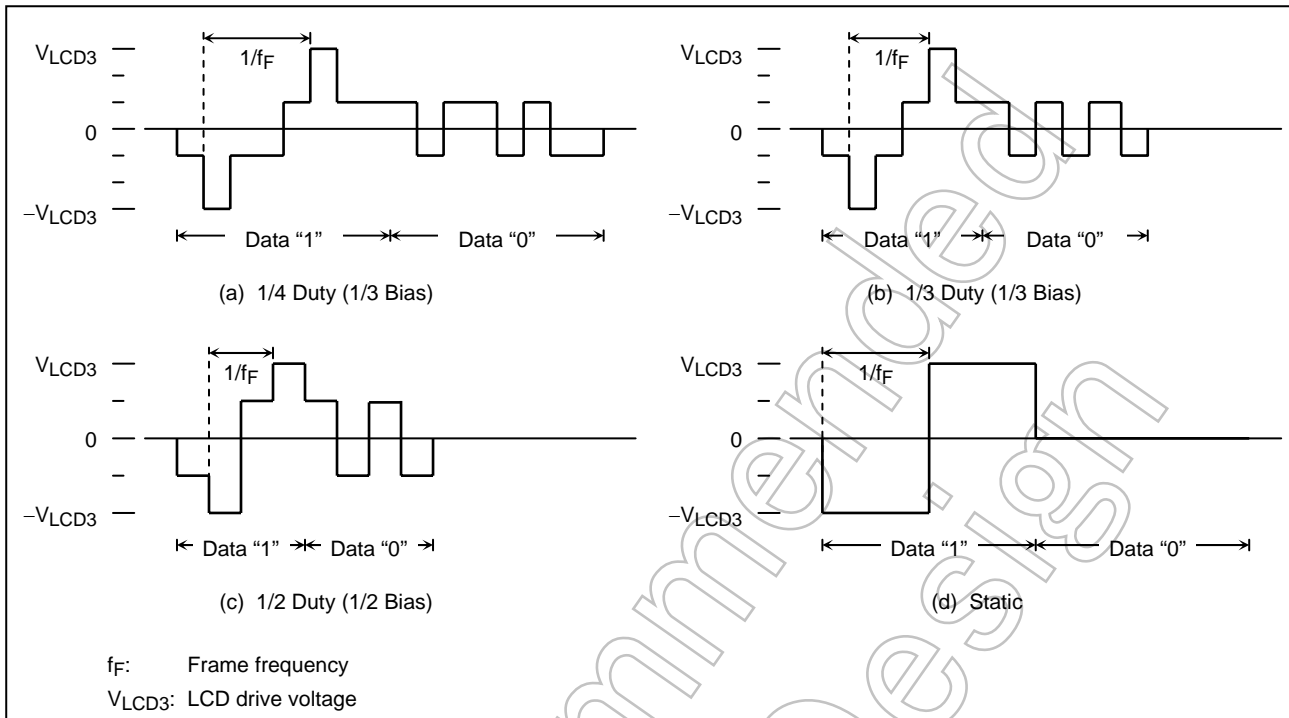

Figure 2.13.3 LCD Drive Waveform (COM-SEG pins)

(2) Frame frequency

Frame frequency (f_F) is set according to driving method and base frequency as shown in the following Table 2.13.1. The base frequency is selected by SLF (Bit1 and bit0 of LCDCR) according to the frequency fc and fs of the basic clock to be used.

Table 2.13.1 Setting of LCD Frame Frequency

a. At the single clock mode. At the dual clock mode (DV7CK = 0).

SLF	Base Frequency [Hz]	Frame Frequency [Hz]			
		1/4 duty	1/3 duty	1/2 duty	Static
00	$\dfrac{fc}{2^{17}}$	$\dfrac{fc}{2^{17}}$	$\dfrac{4}{3} \cdot \dfrac{fc}{2^{17}}$	$\dfrac{4}{2} \cdot \dfrac{fc}{2^{17}}$	$\dfrac{fc}{2^{17}}$
	(fc = 16 MHz)	122	163	244	122
	(fc = 8 MHz)	61	81	122	61
01	$\dfrac{fc}{2^{16}}$	$\dfrac{fc}{2^{16}}$	$\dfrac{4}{3} \cdot \dfrac{fc}{2^{16}}$	$\dfrac{4}{2} \cdot \dfrac{fc}{2^{16}}$	$\dfrac{fc}{2^{16}}$
	(fc = 8 MHz)	122	163	244	122
	(fc = 4 MHz)	61	81	122	61
10	$\dfrac{fc}{2^{15}}$	$\dfrac{fc}{2^{15}}$	$\dfrac{4}{3} \cdot \dfrac{fc}{2^{15}}$	$\dfrac{4}{2} \cdot \dfrac{fc}{2^{15}}$	$\dfrac{fc}{2^{15}}$
	(fc = 4 MHz)	122	163	244	122
	(fc = 2 MHz)	61	81	122	61
11	$\dfrac{fc}{2^{13}}$	$\dfrac{fc}{2^{13}}$	$\dfrac{4}{3} \cdot \dfrac{fc}{2^{13}}$	$\dfrac{4}{2} \cdot \dfrac{fc}{2^{13}}$	$\dfrac{fc}{2^{13}}$
	(fc = 1 MHz)	122	163	244	122

Note: fc: High-frequency clock [Hz]

b. At the dual clock mode (DV7CK = 1 or SYSCK = 1)

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

Note: fs: Low-frequency clock [Hz]

(3) Booster circuit for LCD driver

The LCD voltage booster pin can select the booster circuit or the divider resistance.

The booster circuit control is selected by LCDCR<BRES>.

The booster circuit boosts the output voltage for the segment/common signal by double (V2) and triple (V3) in relation to the on-chip output voltage (1 V typ.).

If the reference supply voltage is at the V2 pin, the reference voltage is reduced to 1/2(V1) or boosted to 3/2(V3).

When used as the divider resistance, the V1, V2 and V3 pins are input by divider voltage of external supply.

The selection of boost frequency is selected by LCDCR<VFSEL>.

Selecting the fast frequency using the VFSEL in the command register (LCDCR) raises the drive capability of segment/common.

Note: When used as the booster circuit, the VLCD setting should be composed to 1/3 bias. Therefore, do not set LCDCR<DUTY> to "10" or "11" when the booster circuit is enable.
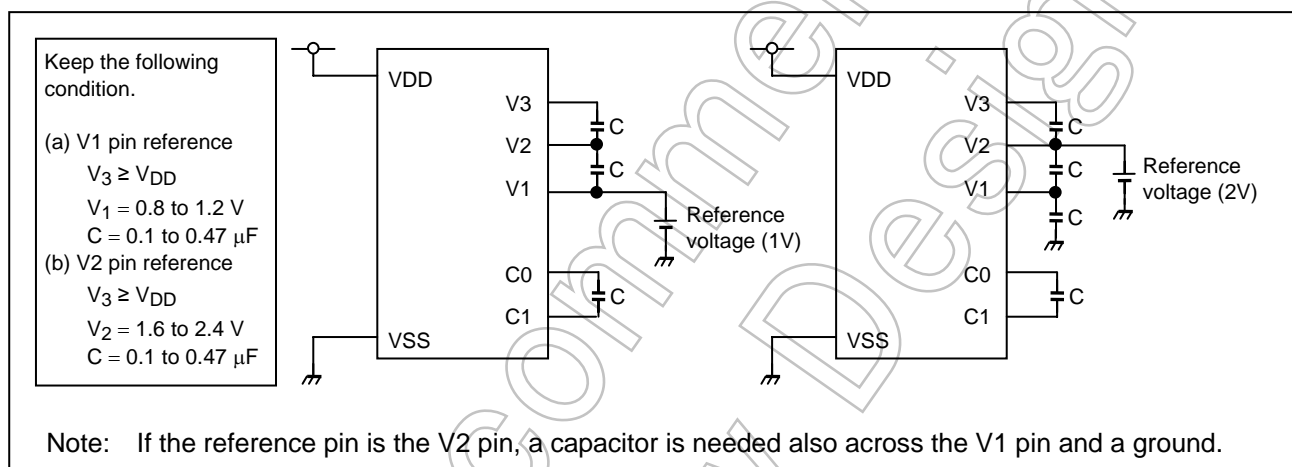

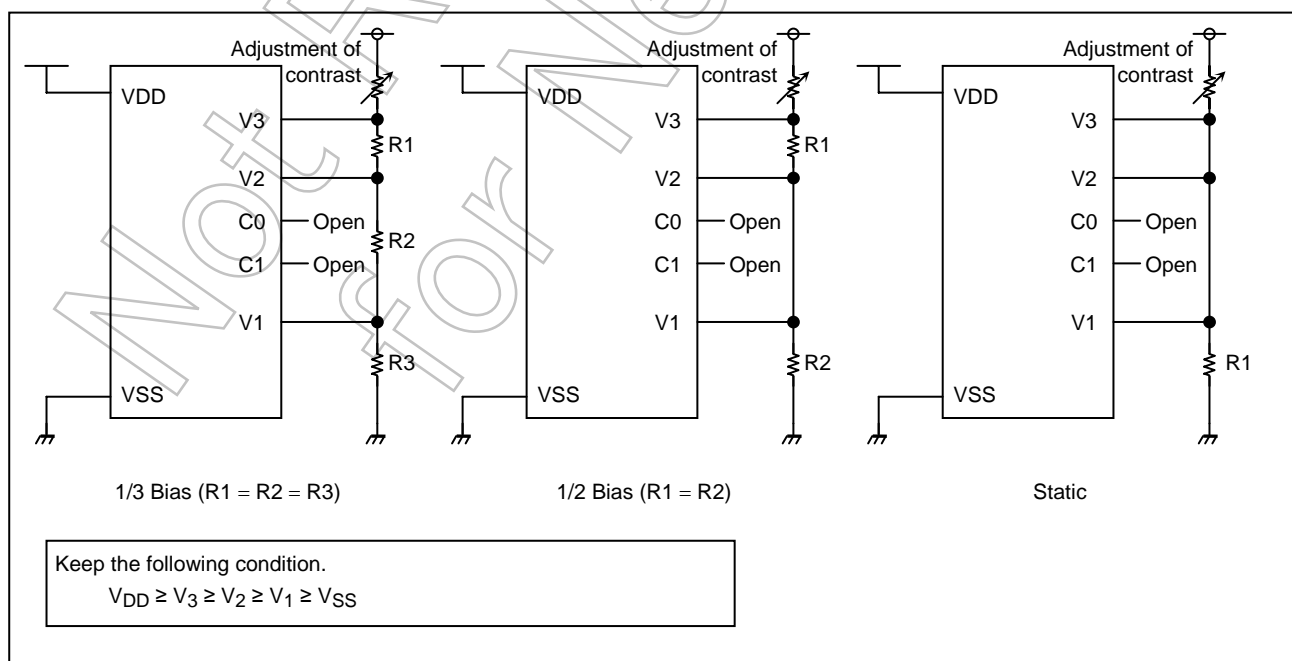Figure 2.13.4 Example of a Booster Circuit (LCDCR<BRES> = "1")

Figure 2.13.5 Example of Divider Resistance (LCDCR<BRES> = "0")

2.13.3 LCD Display Operation

(1) Display data setting

Display data is stored to the display data area (Assigned to address 1F80H to 1F8BH) 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 2.13.6 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 2.13.2).

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

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
1F80H		SEG1				SEG0		
81		SEG3				SEG2		
82		SEG5				SEG4		
83		SEG7				SEG6		
84		SEG9				SEG8		
85		SEG11				SEG10		
86		SEG13				SEG12		
87		SEG15				SEG14		
88		SEG17				SEG16		
89		SEG19				SEG18		
8A		SEG21				SEG20		
8B		SEG23				SEG22		
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0

Figure 2.13.6 LCD Display Data Area (DBR)

Table 2.13.2 Driving Method and Bit for Display Data

Driving Methods	Bit7/3	Bit6/2	Bit5/1	Bit4/0
1/4 Duty	COM3	COM2	COM1	COM0
1/3 Duty	–	COM2	COM1	COM0
1/2 Duty	–	–	COM1	COM0
Static	–	–	–	COM0

–: 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 GND level to SEG/COM pin.

When in STOP mode, EDSP is cleared to "0" and automatically blanked. To redisplay LCD after exiting STOP mode, it is necessary to set EDSP back to "1".

Note: During reset, the LCD segment outputs (SEG0 to SEG7) and LCD common outputs are fixed "0" level. But the multiplex terminal (P4 and P9 ports) of input/output port and LCD segment output becomes high impedance. Therefore, when the reset input is long remarkably, ghost problem may appear in LCD display.

2.13.4 Control Method of LCD Driver

（1） Initial setting

shows the flowchart of initialization.

Example: To operate a 1/4 duty LCD of 24 segments $\times$ 4 commons at frame fc/2^{16} [Hz].

```
LD        (LCDCR), 01000001B    ; Sets LCD driving method, frame
                                    frequency and Boost frequency.
LD        (P4LCR), 0FFH         ; Sets P4 port as segment output .
LD        (P9LCR), 0FFH         ; Sets P9 port as segment output .
                                ; Sets the initial value of display data.
LD        (LCDCR), 11000001B    ; Display enable
```
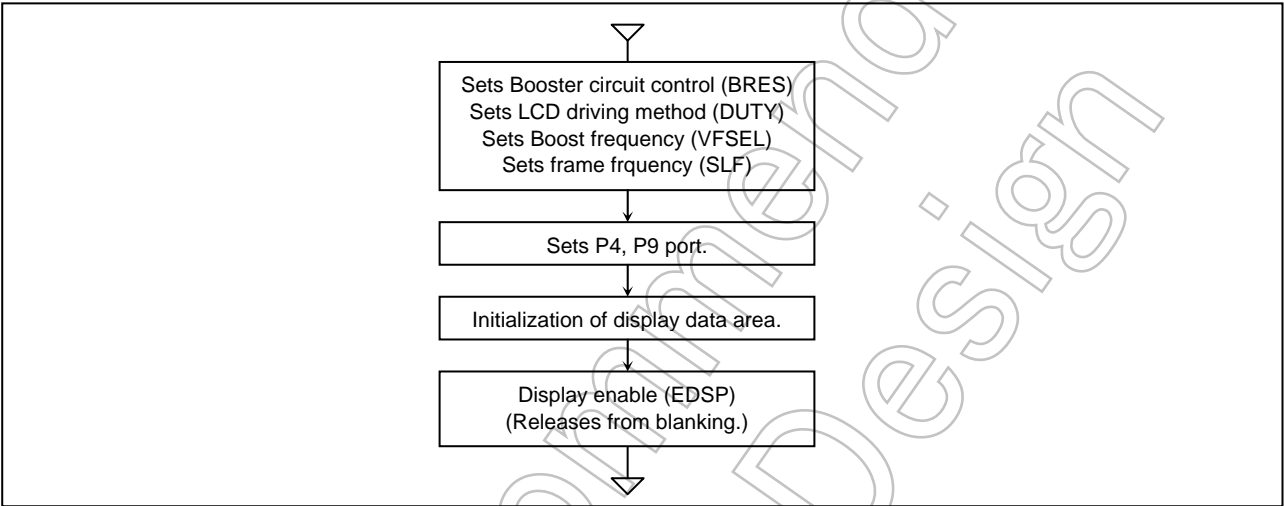

Figure 2.13.7 Initial Setting of LCD Driver

（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 80H (when pins COM and SEG are connected to LCD as in Figure 2.13.8), display data become as shown in Table 2.12.2.

```
          LD     A, (80H)
          ADD    A, TABLE − $ − 7
          LD     HL, 1F80H
          LD     W, (PC + A)
          LD     (HL), W
          RET
TABLE:    DB     11011111B, 00000110B,
                 11100011B, 10100111B,
                 00110110B, 10110101B,
                 11110101B, 00010111B,
                 11110111B, 10110111B
SNEXT:
```

Note: DB is a byte data difinition instruction.
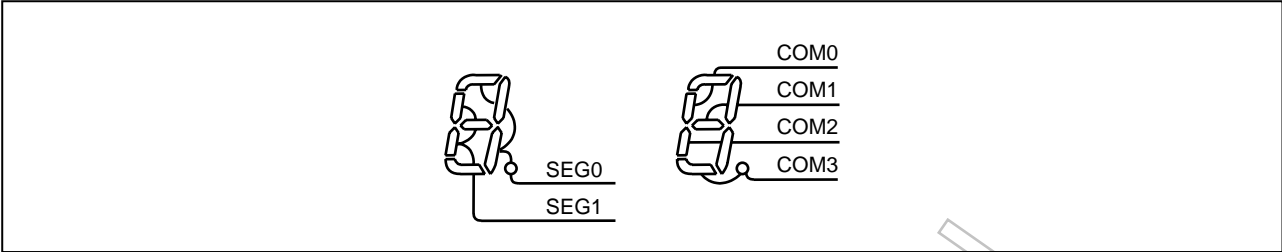
Figure 2.13.8 Example of COM, SEG Pin Connection (1/4 duty)

Table 2.13.4 Example of Display Data (1/4 duty)

Number	Display	Display Data	Number	Display	Display Data
0.		11011111	5		10110101
1		00000110	6		11110101
2		11100011	7		00000111
3		10100111	8		11110111
4		00110110	9		10110111

Example 2: Table 2.13.4 shows an example of display data which are displayed using 1/2 duty LCD in the same way as Table 2.13.5. The connection between pins COM and SEG are the same as shown in Figure 2.13.9.
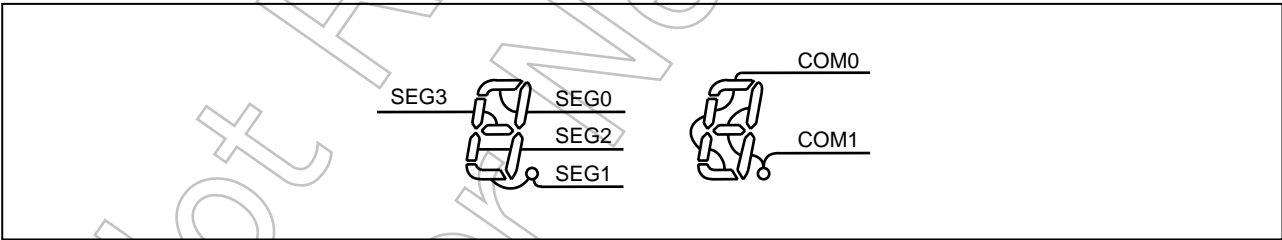
Figure 2.13.9 Example of COM, SEG Pin Connection

Table 2.13.5 Example of Display Data (1/2 duty)

Number	Display Data		Number	Display Data	
	High Order Address (1F81H)	Low Order Address (1F80H)		High Order Address (1F81H)	Low Order Address (1F80H)
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

*: Don't care

(3) Example of LCD drive output
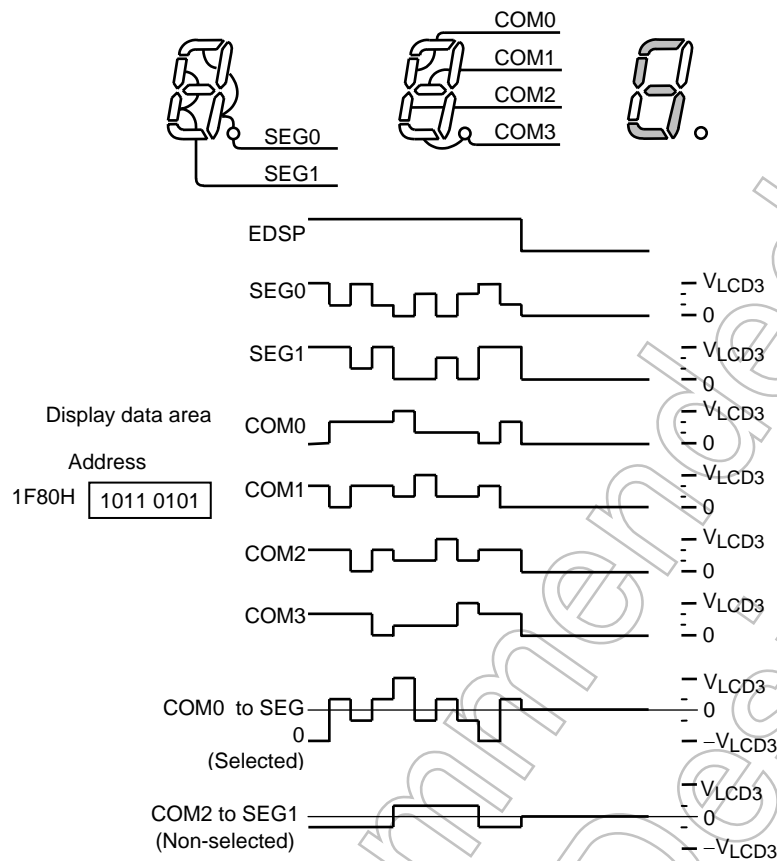
Figure 2.13.10 1/4 Duty (1/3 bias) Drive
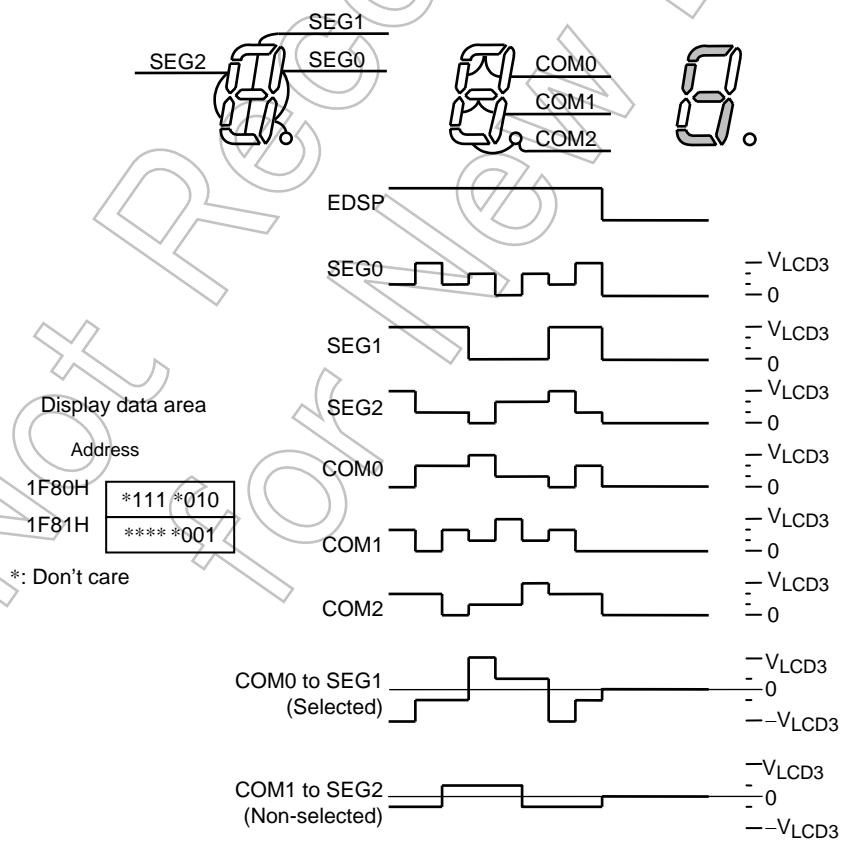
Figure 2.13.11 1/3 Duty (1/3 bias) Drive
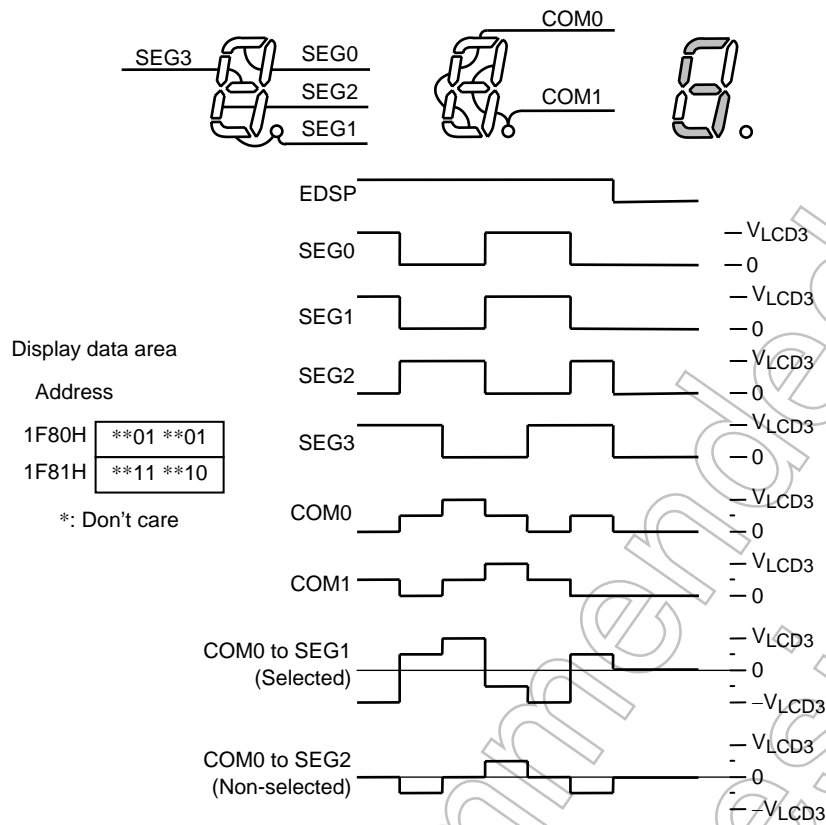
Figure 2.13.12 1/2 Duty (1/3 bias) Drive
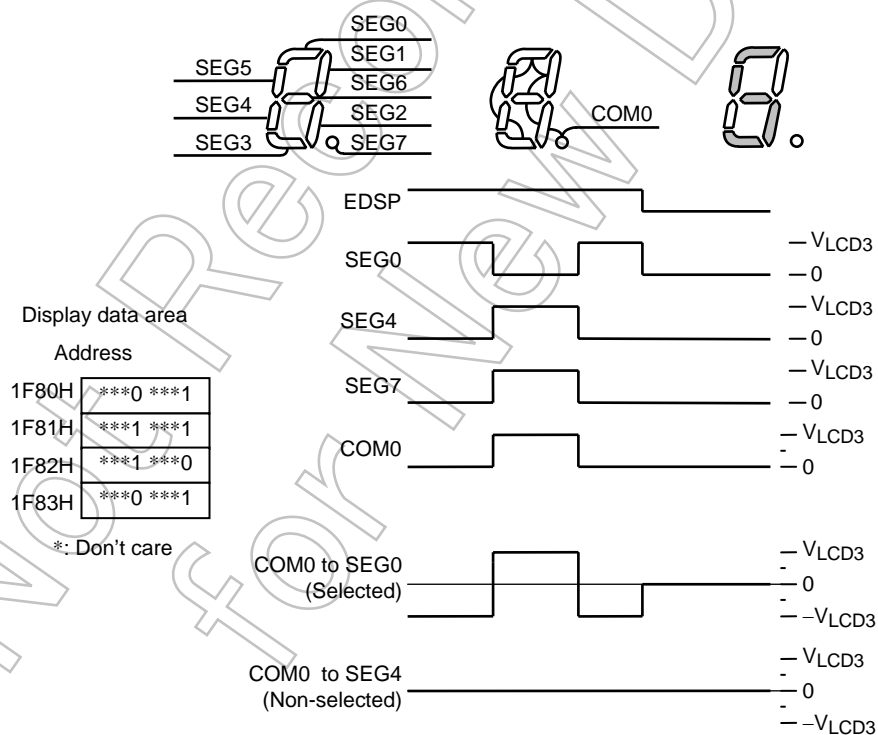
Figure 2.13.13 Static Drive

2.14 SIO (Synchronous serial interface)

The TMP86FP24 contains two SIO (Synchronous serial interface) channel. They are connected to external devices via the SI1, SI2, SO1, SI2, $\overline{SCK2}$ and $\overline{SCK1}$ pins. The SI1 (SI2) pin is used also as the P05 (P11) pin, the SO1 (SO2) pin is used also as the P06(P10) pin, and the $\overline{SCK1}$ ($\overline{SCK2}$) pin is used also as the P07 (P12) pin. Using these pins for serial interfacing requires setting the output latches of the port P0 and P1 to "1".

Because SIO1 and SIO2 are the same except that the register for each SIO are located at different addresses, explanation here is made of only SIO1. The registers for SIO1 and SIO2 are listed in Table 2.14.1 below.

Table 2.14.1 Control Registers

	SIO1		SIO2	
	Register Name	Address	Register Name	Address
SIO control register 1	SIO1CR1	0016H	SIO2CR1	001AH
SIO control register 2	SIO1CR2	0017H	SIO2CR2	001BH
SIO status register	SIO1SR	0018H	SIO2SR	001CH
SIO data buffer	SIO1BUF	0019H	SIO2BUF	001DH

2.14.1   Configuration
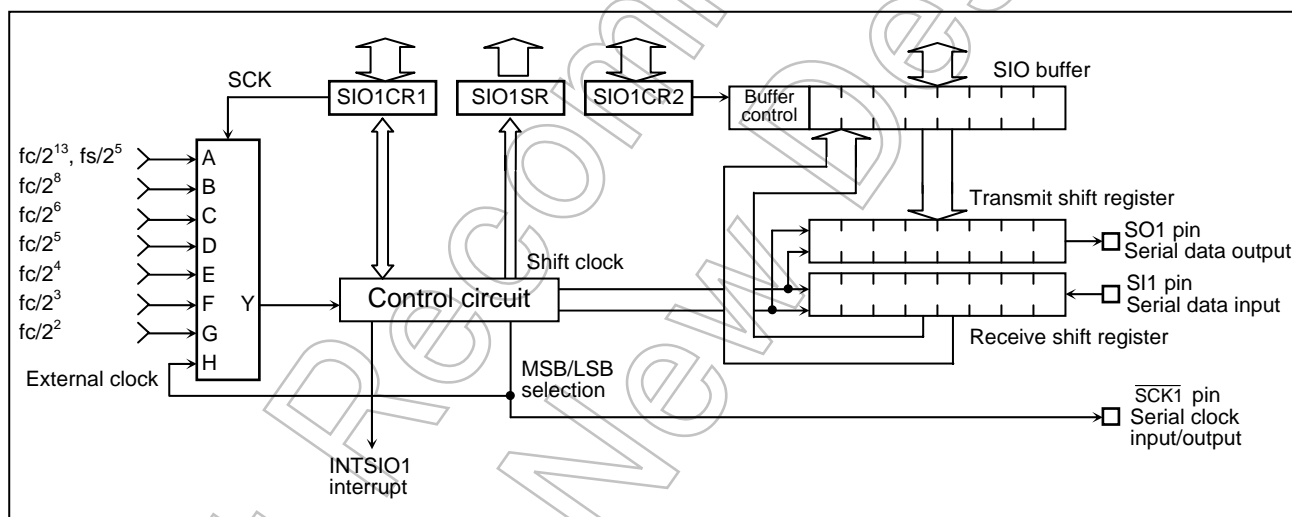
Figure 2.14.1 Configuration of the Serial Interface

2.14.2 Control

SIO is controlled using serial interface control register 1 (SIO1CR1) and serial interface control register 2 (SIO1CR2). The operating status of the serial interface can be determined by reading the serial interface status register (SIO1SR).

Serial Interface Control Register 1

	7	6	5	4	3	2	1	0	
SIO1CR1	SIOS	SIOINH	SIOM		SIODIR	SCK			(Initial value: 0000 0000)

SIOS	Start/stop a transfer.	0: Stop 1: Start			
SIOINH	Continue/abort a transfer (Note 1).	0: Continue transfer. 1: Abort transfer (automatically cleared to "0" after abort).			
SIOM	Select transfer mode.	00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved			
SIODIR	Select direction of transfer.	0: MSB (Transfer beginning with bit7) 1: LSB (Transfer beginning with bit0)			R/W
SCK	Select a serial clock. (Note 2)		NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode
			DV7CK = 0	DV7CK = 1	
		000	$fc/2^{13}$	$fs/2^5$	$fs/2^5$
		001	$fc/2^8$	$fc/2^8$	–
		010	$fc/2^6$	$fc/2^6$	–
		011	$fc/2^5$	$fc/2^5$	–
		100	$fc/2^4$	$fc/2^4$	–
		101	$fc/2^3$	$fc/2^3$	–
		110	$fc/2^2$	$fc/2^2$	–
		111	External clock (Supplied via the $\overline{SCK1}$ pin)		

Note 1: If SIO1CR1<SIOINH> is set to "1", SIO1CR1<SIOS>, SIO1SR<SIOF>, SIO1SR<SEF>, SIO1SR<TXF>, SIO1SR<RXF>, SIO1SR<TXERR>, and SIO1SR<RXERR> are initialized.

Note 2: When selecting a serial clock, do not make such a setting that the serial clock rate will exceed 1 Mbps.

Note 3: Before setting SIO1CR1<SIOS> to "1" or setting SIO1CR1<SIOM>, SIO1CR1<SIODIR> or SIO1CR1<SCK> to any value, make sure the SIO is idle (SIO1SR<SIOF> = "0").

Note 4: Reserved: Setting prohibited

Figure 2.14.2 Serial Interface Control Register 1

Serial Interface Control Register 2

	7	6	5	4	3	2	1	0	
SIO1CR2							SIORXD		(Initial value: **** *000)

	SIORXD	Set the number of data bytes to transmit/receive.	000: 1-byte transfer 001: 2-byte transfer 010: 3-byte transfer 011: 4-byte transfer 100: 5-byte transfer 101: 6-byte transfer 110: 7-byte transfer 111: 8-byte transfer	R/W

Note 1: Before setting the number of data bytes to transfer, make sure the SIO is idle (SIO1SR<SIOF> = "0").

Note 2: The number of data bytes to transfer is used for transmit and receive operations in common.

Note 3: Always write "0" to bits 7 to 3.

Serial Interface Status Register

	7	6	5	4	3	2	1	0	
SIO1SR	SIOF	SEF	TXF	RXF	TXERR	RXERR			(Initial value: 0010 00**)

	SIOF	Monitor the operating status of serial transfer.	0: Transfer ended (Note 1) 1: Transfer in process	
	SEF	Number of clocks monitor.	0: 8 clocks 1: 1 to 7 clocks	
	TXF	Transmit buffer flag.	0: The transmit buffer contains data. 1: The transmit buffer contains no data.	
	RXF	Receive buffer flag.	0: The receive buffer contains no data. 1: As many data bytes specified in SIO1CR2<SIORXD> have been received. (The flag is reset to "0" when as many data bytes as specified in SIO1CR2<SIORXD> have been read.)	Read only
	TXERR	Transmit error flag (Note 2)	0: Transmit operation was normal. 1: Error occurred during transmission.	
	RXERR	Receive error flag (Note 2)	0: Receive operation was normal. 1: Error occurred during reception.	

Note 1: The SIO1SR<SIOF> bit is cleared to "0" by clearing SIO1CR1<SIOS> to stop transferring or by setting SIO1CR1<SIOINH> to "1" to abort transfer.

Note 2: Neither the SIO1SR<TXERR> nor SIO1SR<RXERR> bit can be cleared when transfer ends on SIO1CR1<SIOS> = "0". To clear them, set SIO1CR1<SIOINH> to "1".

Note 3: Do not write to the SIO1SR register.

Serial Interface Data Buffer

	7	6	5	4	3	2	1	0	
SIO1BUF									(Initial value: **** ****)

	SIOBUF	Transmit/receive data buffer	Transmit data are set, or received data are stored.	R/W

Note 1: Setting SIO1CR1<SIOINH> causes the contents of SIOBUF to be lost.

Note 2: When setting transmit data or storing received data, be sure to handle as many bytes as specified in SIO1CR2<SIORXD> at a time.

Figure 2.14.3 Serial Interface Control Register 2, Status Register, and Data Buffer Register

2.14.3 Configuration

(1) Serial clock

a. Clock source

One of the following clocks can be selected using SIO1CR1<SCK>.

1. Internal clock

A clock having the frequency selected with SIO1CR1<SCK> (except for "111") is used as the serial clock. The $\overline{\text{SCK1}}$ pin output goes high when transfer starts or ends.

Table 2.14.2 Serial Clock Rate (fc = 4 MHz)

SIO1CR1<SCK>	Clock	Baud Rate
000	$fc/2^{13}$	0.47 Kbps
001	$fc/2^{8}$	15.25 Kbps
010	$fc/2^{6}$	61.04 Kbps
011	$fc/2^{5}$	122.07 Kbps
100	$fc/2^{4}$	244.14 Kbps
101	$fc/2^{3}$	488.28 Kbps
110	$fc/2^{2}$	976.56 Kbps
111	External	External

(1 Kbit = 1,024 bits)

Note: Do not make such a setting that the serial clock rate will exceed 1 Mbps.

(2) External clock

Setting SIO1CR1<SCK> to "111" causes an external clock to be selected. A clock supplied to the $\overline{\text{SCK1}}$ pin is used as the serial clock.

For a shift operation to be performed securely, both the high and low levels of the serial clock pulse must be at least 4/fc. If fc = 4 MHz, therefore, the maximum available transfer rate is 488.28 Kbits/s.
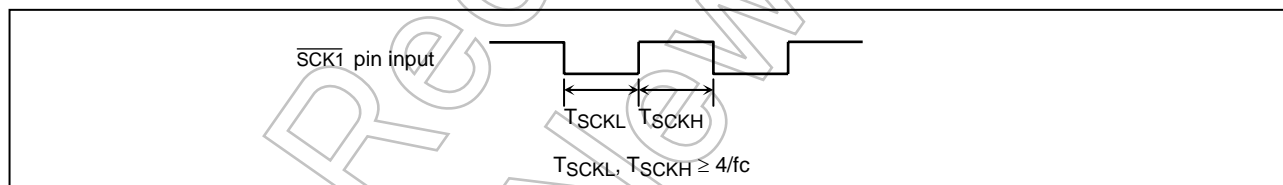
Figure 2.14.4 External Clock

b. Shift edges

The SIO uses leading-edge shift for transmission and trailing-edge shift for reception.

1. Leading-edge shift

   Data are shifted on each leading edge of the serial clock pulse (Falling edge of the $\overline{SCK1}$ pin input/output).

2. Trailing-edge shift

   Data are shifted on each trailing edge of the serial clock pulse (Rising edge of the $\overline{SCK1}$ pin input/output).
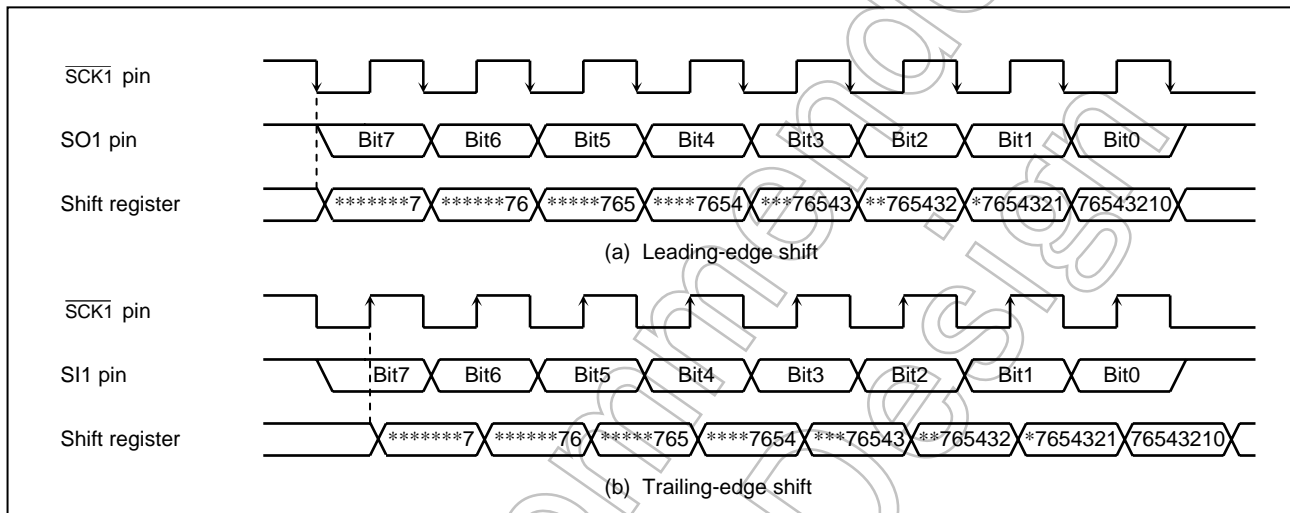
Figure 2.14.5 Shift Edges

(2) Transfer bit direction

The direction in which 8-bit serial data are transferred can be selected using SIO1CR1<SIODIR>. The direction of data transfer applies in common to both transmission and reception, and cannot be set individually.

1. MSB transfer

MSB transfer is assumed by clearing SIO1CR1<SIODIR> to "0". In MSB transfer, data are transferred sequentially beginning with the most significant bit (MSB). As for received data, the first data bit to receive is stored as the MSB.

2. LSB transfer

LSB transfer is assumed by setting SIO1CR1<SIODIR> to "1". In LSB transfer, data are transferred sequentially beginning with the least significant bit (LSB). As for received data, the first data bit to receive is stored as the LSB.
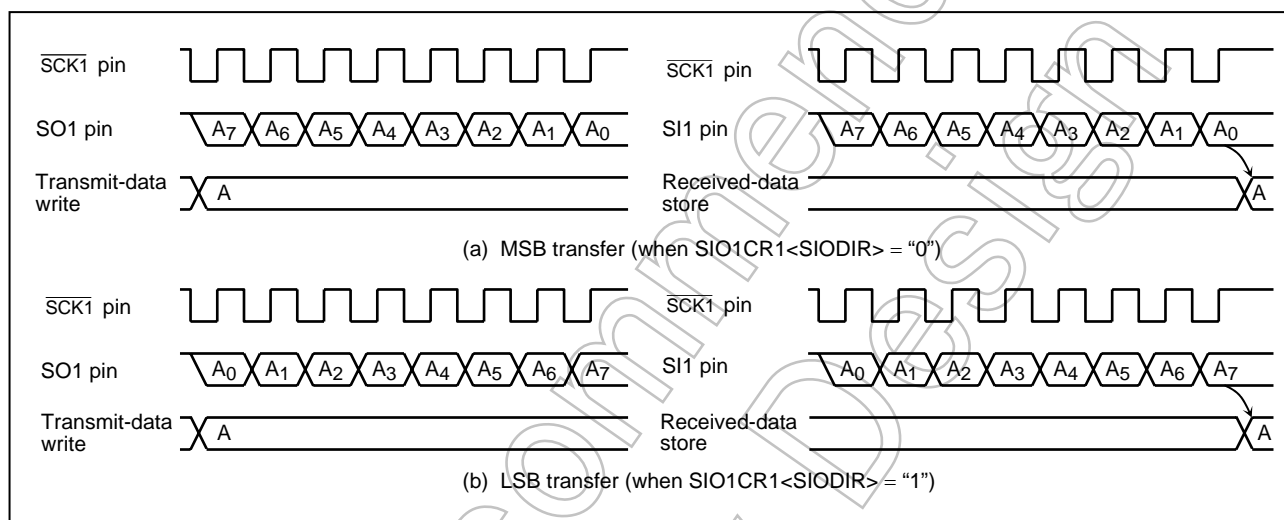
Figure 2.14.6 Transfer Bit Direction

(3) Transfer modes

SIO1CR1<SIOM> is used to select a transfer mode (Transmit, receive, or transmit/receive mode).

<u>a. Transmit mode</u>

Transmit mode is assumed by setting SIO1CR1<SIOM> to "00".

Causing the SIO to start transmitting

1. Set the transmit mode, serial clock rate, and transfer direction, respectively, in SIO1CR1<SIOM>, SIO1CR1<SCK>, and SIO1CR1<SIODIR>.

2. Set the number of data bytes to transfer in SIO1CR2<SIORXD>.

3. Set, in SIOBUF, as many transmit data bytes as specified in SIO1CR2<SIORXD>.

4. Set SIO1CR1<SIOS> to "1".

   • If the selected serial clock is an internal clock, the SIO immediately starts transmitting data sequentially in the direction selected using SIO1CR1<SIODIR>.

   • If the selected serial clock is an external clock, the SIO immediately starts transmitting data, upon external clock input, sequentially in the direction selected using SIO1CR1<SIODIR>.

Causing the SIO to stop transferring

5. When as many data bytes as specified in SIO1CR2<SIORXD> have been transmitted, be sure to clear SIO1CR1<SIOS> to "0" to halt the SIO. Clearing of SIO1CR1<SIOS> should be executed within the INTSIO1 service routine or should be executed after confirmation of SIO1SR<TXF> = "1". Before starting to transfer the next data after clearing SIO1CR1<SIOS> to "0", make sure SIO1SR<SIOF> = "0" and SIO1SR<TXERR> = "0" in the external clock mode (No transfer error), write the data to be transferred, and then set SIO1CR1<SIOS> = "1".
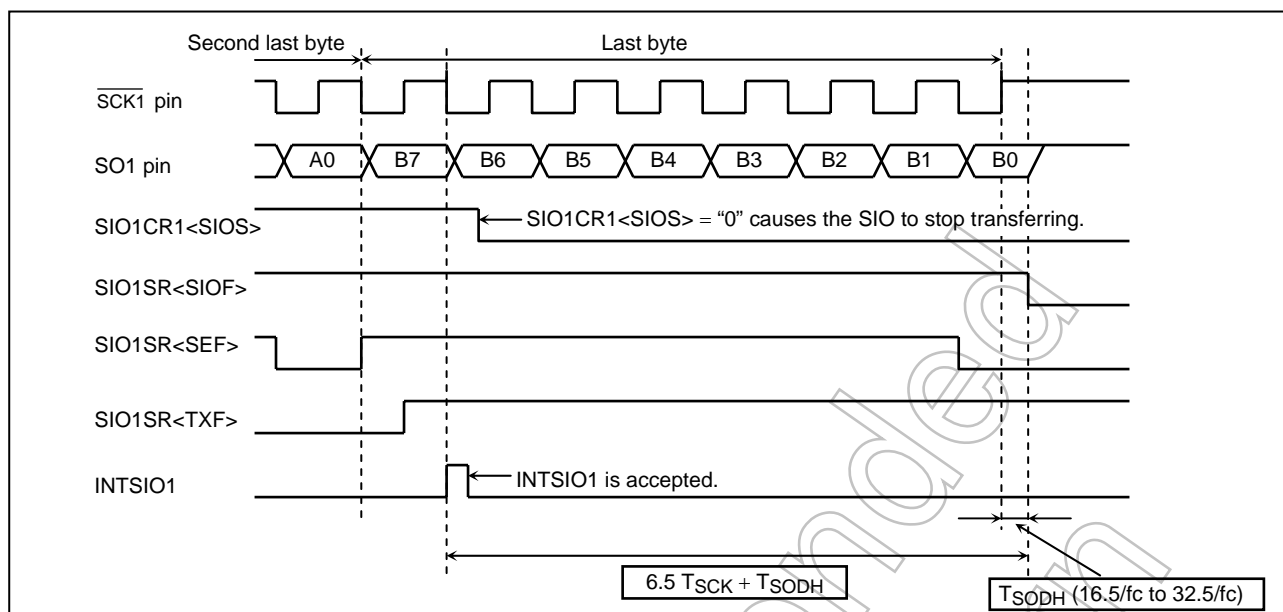
Figure 2.14.7 Time from INTSIO1 Occurrence to Transfer End (SIO1SR<SIOF> = "0")
when the SIO is Directed to Stop Transferring (SIO1CR1<SIOS> = "0")
upon the Occurrence of a Transmit Interrupt

Note 1: Be sure to write as many bytes as specified in SIO1CR2<SIORXD> to SIO1BUF. If the number of data bytes to be written to SIO1BUF is not equal to the value specified in SIO1CR2<SIORXD>, the SIO fails to work normally.

Note 2: Before starting the SIO, be sure to write as many data bytes as specified in SIO1CR2<SIORXD> to SIO1BUF.

Note 3: In the transmit mode, an INTSIO1 interrupt occurs when the transmission of the second bit of the last byte begins.

Note 4: If an attempt is made to write SIO1CR1<SIOS> = "0" within the INTSIO1 interrupt service routine, the SIO stops transferring (SIO1SR<SIOF> = "0") after the last data byte is transmitted (the signal at the $\overline{SCK1}$ pin rises).
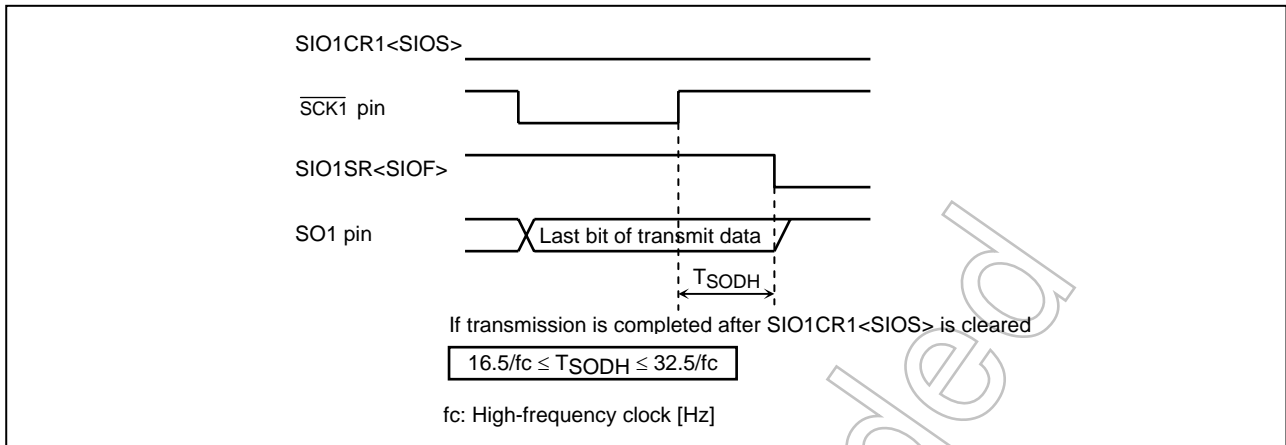
Figure 2.14.8 Last-bit Hold Time

- Setting SIO1CR1<SIOINH> to "1" causes the SIO to immediately stop a transmission sequence even if any byte is being transmitted.
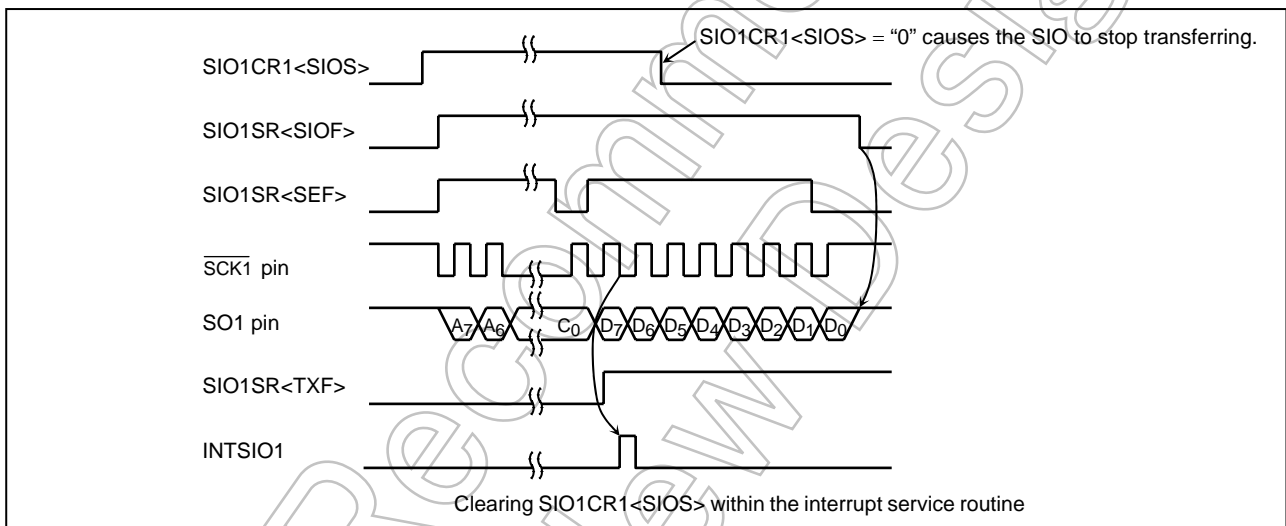
Figure 2.14.9 SIO1CR1<SIOS> Clear Timing

<u>Transmit error</u>

During operation on an external clock, the following case may be detected as a transmit error, causing the transmit error flag (SIO1SR<TXERR>) to be set to "1". If a transmit error occurs, the SO1 pin goes high.

- If the $\overline{SCK1}$ pin goes low when the SIO is running (SIO1SR<SIOF> = "1") but there is no transmit data in SIO1BUF (SIO1SR<TXF> = "1").

If a transmit error is detected, be sure to set SIO1CR1<SIOINH> to "1" to force the SIO to halt. Setting SIO1CR1<SIOINH> to "1" initializes the SIO1CR1<SIOS> and SIO1SR registers; no other registers or bits are initialized.

Example of setting the transmit mode (Transmit mode, external clock, and 8-byte transfer).

```
                LD      (P0OUTCR), 01******B    ;   PORT setting.
                                                    P07 (SCK1) input and P06 (SO1) output.
                LD      (P0DR), 11******B       ;   Sets SCK1 and SO1.
                DI                              ;   IMF ← 0
                LDW     (EIRL), *****1*********0B ;  Enables INTSIO1 (EF10).
                EI                              ;   Enables interrupts.
                LD      (SIO1CR1), 01******B    ;   Initializes the SIO (forces the SIO halt).
WAIT:           TEST    (SIO1SR). 7             ;   Checks to see if the SIO has halted
                                                    (SIO1SR<SIOF> = 0).
                JRS     F, WAIT                 ;   Jumps to START if the SIO is already at a
                                                    halt.
START:          LD      (SIO1CR1), 00000111B    ;   Sets the transmit mode, selects the
                                                    direction of transfer, and sets a serial
                                                    clock.
                LD      (SIO1CR2), 00000111B    ;   Sets the number of bytes (8 bytes) to
                                                    transfer.

                Transmit data setting

                LD      (SIO1CR1), 10000111B    ;   Directs the SIO to start transferring.
INTSIO1 (INTSIO1 service routine):
                LD      (SIO1CR1), 00000111B    ;   Directs the SIO to stop transferring.
                TEST    (SIO1SR). 3             ;   Checks SIO1SR<TXERR>.
                JRS     T, NOERR
                LD      (SIO1CR1), 01000111B    ;   Forces the SIO to halt (clears
                                                    SIO1SR<TXERR>).

                Error handling

NOERR:
END:                                            ;   End of transfer.
```
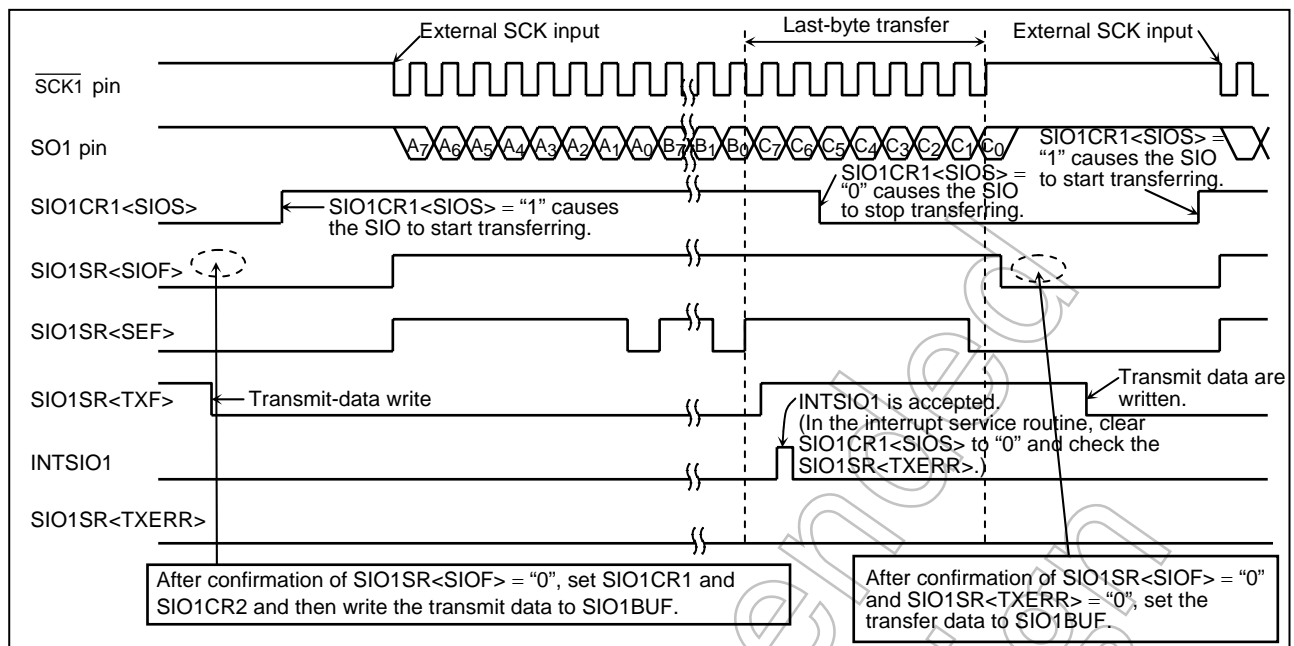
Figure 2.14.10 Transmit Mode Operation (Where 3 bytes are transferred on an external source clock)
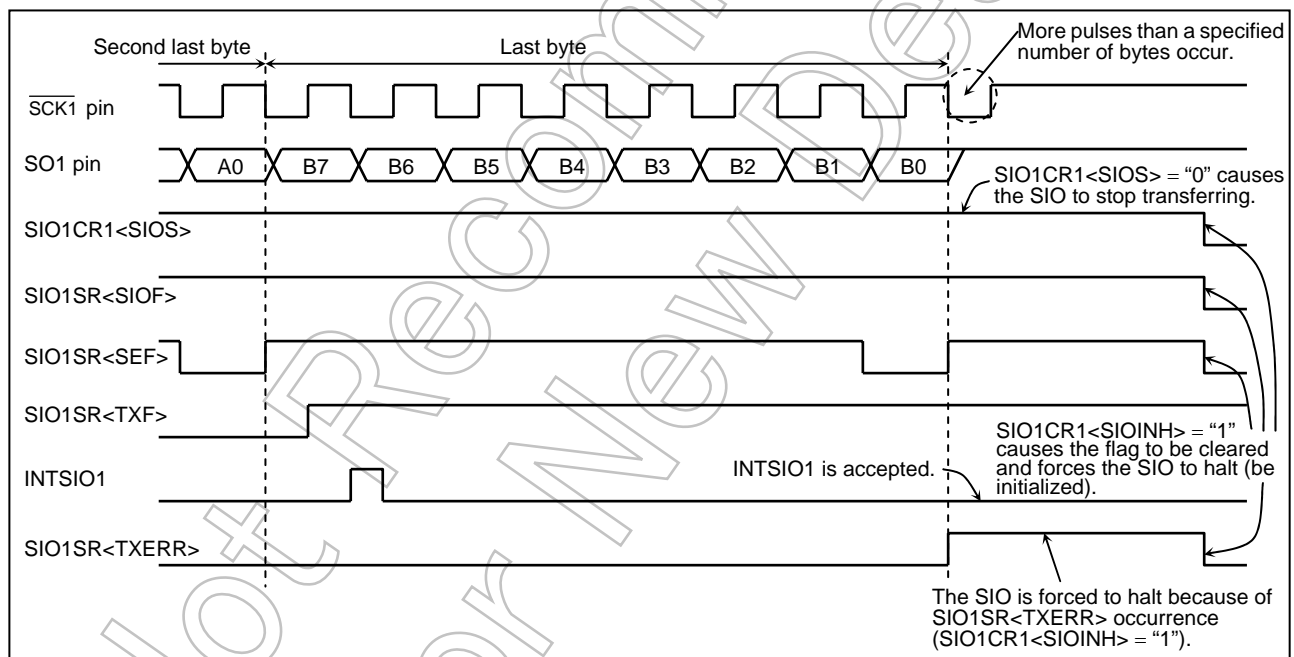
Figure 2.14.11 Occurrence of Transmit Error (Where, before the SIO is directed to stop transferring
(SIO1CR1<SIOS> = "0" is written), the transfer of the last byte is completed and more
pulses than a specified number of bytes occur.)

Note: When the SIO is running (SIO1SR<SIOF> = "1"), do not supply more transfer clock pulses than the
number of bytes specified in SIO1CR2<SIORXD> to the $\overline{\text{SCK1}}$ pin.

<u>b. Receive mode</u>

Receive mode is assumed by setting SIO1CR1<SIOM> to "01".

Causing the SIO to start receiving

1. Set the receive mode, serial clock rate, and transfer direction, respectively, in SIO1CR1<SIOM>, SIO1CR1<SCK>, and SIO1CR1<SIODIR>.

2. Set the number of data bytes to transfer in SIO1CR2<SIORXD>.

3. Set SIO1CR1<SIOS> to "1".

- If the selected serial clock is an internal clock, the SIO immediately starts receiving data sequentially in the direction selected using SIO1CR1<SIODIR>.

- If the selected serial clock is an external clock, the SIO immediately starts receiving data, upon external clock input, sequentially in the direction selected using SIO1CR1<SIODIR>.

Causing the SIO to stop receiving

4. When as many data bytes as specified in SIO1CR2<SIORXD> have been received, be sure to clear SIO1CR1<SIOS> to "0" to halt the SIO. Clearing of SIO1CR1<SIOS> should be executed within the INTSIO1 service routine or should be executed after confirmation of SIO1SR<RXF> = "1".

- Setting SIO1CR1<SIOINH> to "1" causes the SIO to immediately stop a reception sequence even if any byte is being received.

Received-data read timing

Before reading received data, make sure SIO1BUF is full (SIO1SR<RXF> = "1") or clear SIO1CR1<SIOS> to "0" to halt the SIO in the INTSIO1 interrupt service routine. Before reading the receive data after clearing SIO1CR1<SIOS> to "0", make sure SIO1SR<SIOF> = "0" and SIO1SR<RXERR> = "0" in the external clock mode (No receive error), and then read the receive data. SIO1SR<RXF> is cleared to "0" when as many received data bytes as specified in SIO1CR2<SIORXD> are read. To restart to receive the next data after clearing SIO1CR1<SIOS> to "0", first read the received data and set SIO1CR1<SIOS> = "1" to start receiving data.

Note 1: Be sure to read, from SIO1BUF, as many received data bytes as specified in SIO1CR2<SIORXD>. If the number of data bytes to be read from SIO1BUF is not equal to the value specified in SIO1CR2<SIORXD>, the SIO fails to work normally.

Note 2: If an attempt is made to read data before the end of reception (SIO1SR<RXF> = "0"), the SIO fails to work normally.

Note 3: In the receive mode, an INTSIO1 interrupt occurs when the reception of the last bit of the last data byte is completed.

Note 4: If an attempt is made to start transferring after a receive error has been detected, the SIO fails to work normally. Before starting transferring, set SIO1CR1<SIOINH> = "1" to force the SIO to halt.
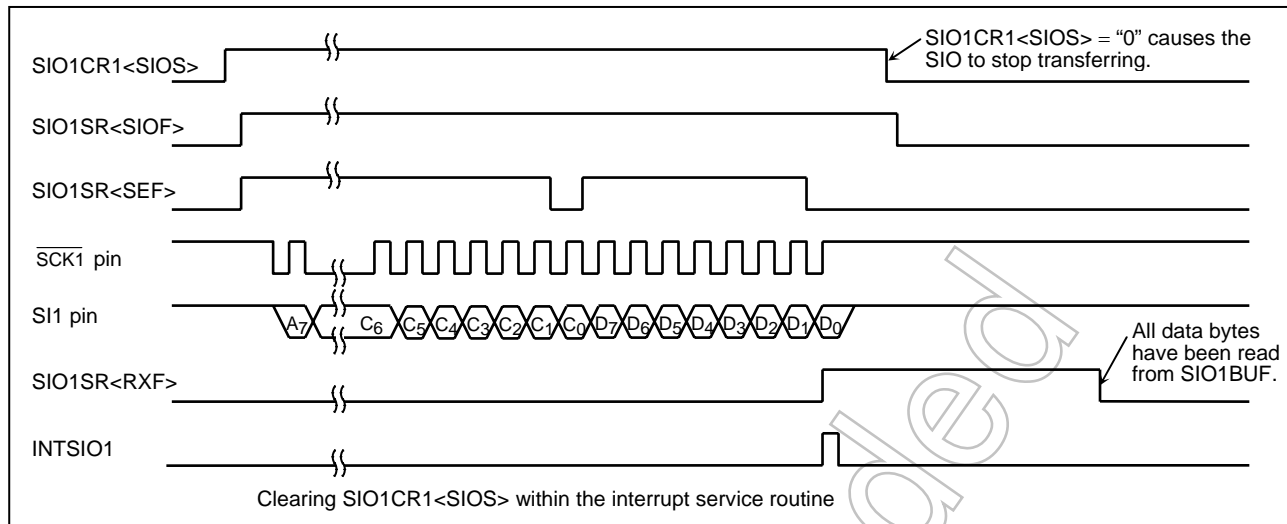
Figure 2.14.12 SIO1CR1<SIOS> Clear Timing

Receive error

During operation on an external clock, the following case is detected as a receive error, causing the receive error flag (SIO1SR<RXERR>) to be set to "1". If a receive error occurs, discard all data from the receive buffer.

- If the reception of the next data byte ends with SIO1BUF full (SIO1SR<RXF> = "1") (if eight clock pulses are supplied to the $\overline{\text{SCK1}}$ pin.)

If a receive error is detected, be sure to set SIO1CR1<SIOINH> to "1" to force the SIO to halt. Setting SIO1CR1<SIOINH> to "1" initializes the SIO1CR1<SIOS> and SIO1SR registers; no other registers or bits are initialized.

Note: If as many clocks of data bytes as specified in SIO1CR2<SIORXD> have been received after the occurrence of receive error, the INTSIO1 is generated. And then, if as many clocks of data bytes as specified in SIO1CR2<SIORXD> have been received continuously again without clearing the SIO1SR<RXERR> flag, the INTSIO1 isn't generated.

Example of setting the receive mode (receive mode, external clock, and 8-byte transfer).

```
                LD      (P0OUTCR), 0*0*****B     ;   PORT setting
                                                     P07 ( SCK1 ) input and P05 (SI1) input.
                LD      (P0DR), 1*1*****B        ;   Sets SCK1 and SI1.
                DI                               ;   IMF ← 0
                LDW     (EIRL), *****1*********0B ;   Enables INTSIO1 (EF10)
                EI                               ;   Enables interrupts.
                LD      (SIO1CR1), 01******B     ;   Initializes the SIO (Forces the SIO halt).
WAIT:           TEST    (SIO1SR). 7              ;   Checks to see if the SIO has halted
                                                     (SIO1SR<SIOF> = 0).
                JRS     F, WAIT                  ;   Jumps to START if the SIO is already at a
                                                     halt.
START:
                LD      (SIO1CR1), 00010111B     ;   Sets the receive mode, selects the
                                                     direction of transfer, and sets a serial
                                                     clock.
                LD      (SIO1CR2), 00000111B     ;   Sets the number of bytes to transfer.
                LD      (SIO1CR1), 10010111B     ;   Directs the SIO to start transferring.
INTSIO1 (INTSIO1 service routine):
                LD      (SIO1CR1), 00010111B     ;   Directs SIO to stop transferring.
                LD      (SIO1CR1), 01010111B     ;   Forces the SIO to halt.

                Receive data reading
                Checks a checksum or the like to see if the received data are normal.

END:                                                 End of transfer.
```
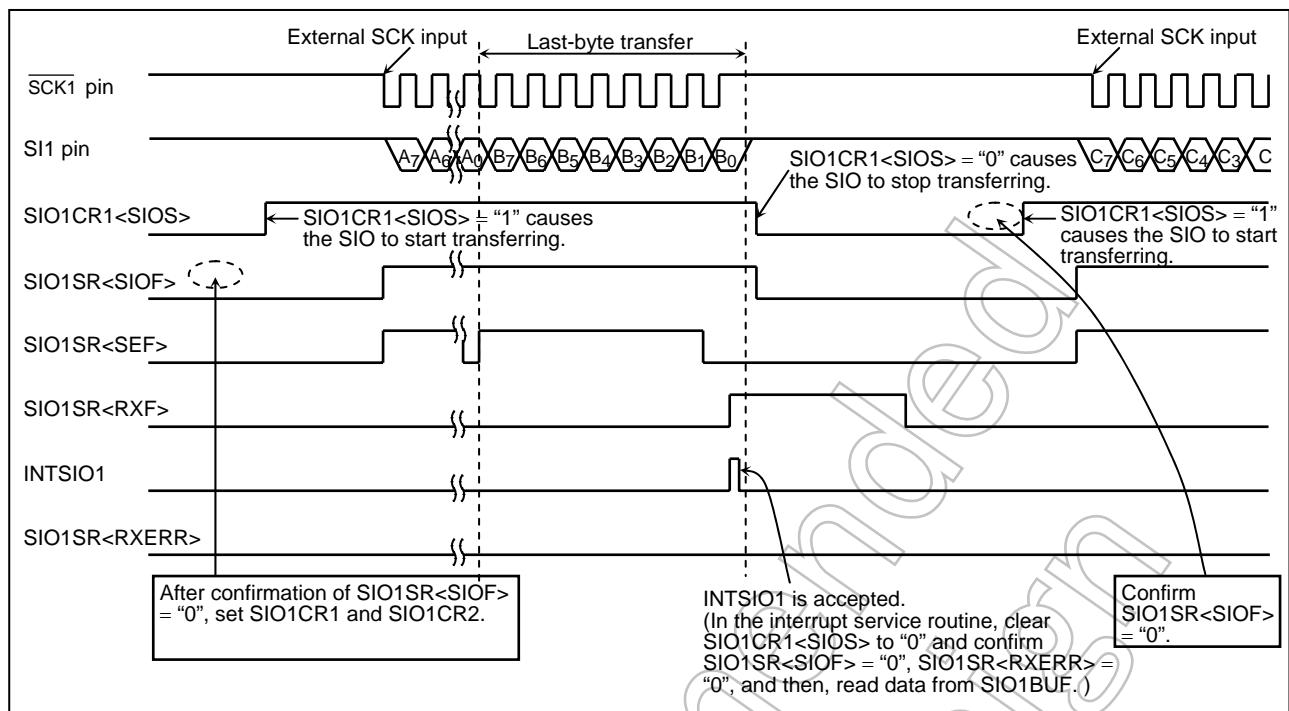
Figure 2.14.13 Receive Mode Operation (Where 2 bytes are transferred on an external source clock)
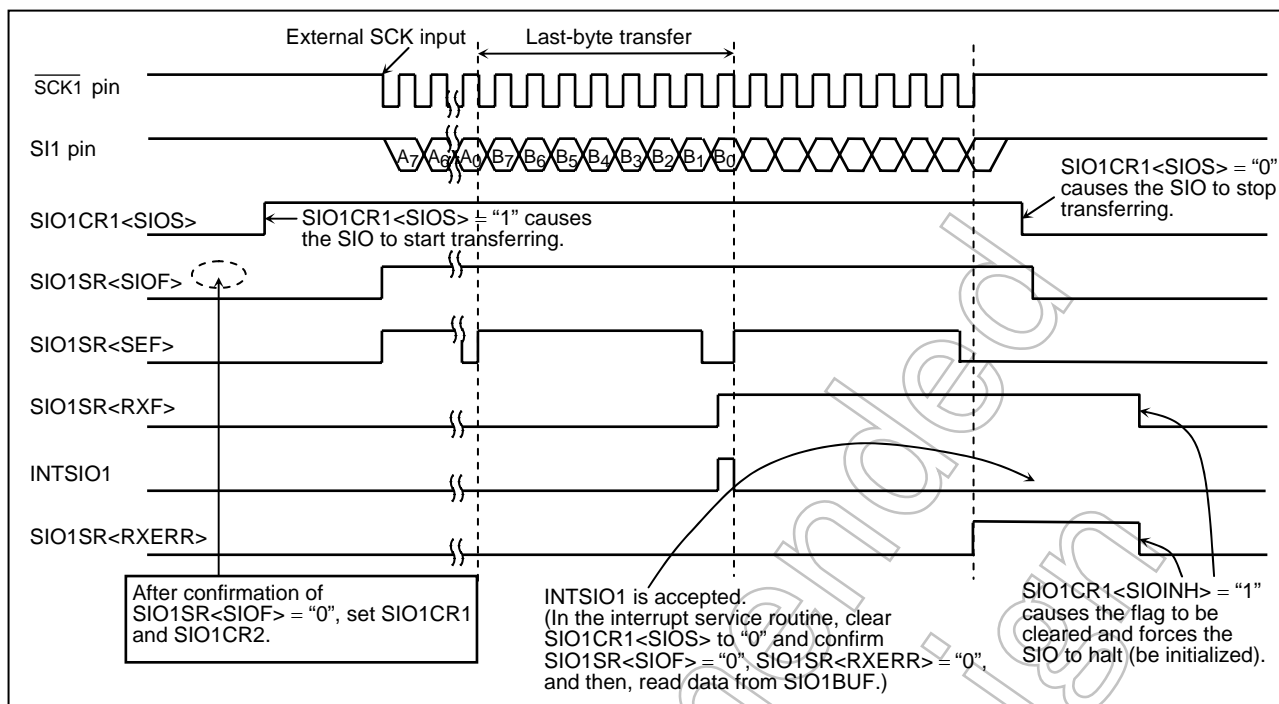
Figure 2.14.14 Occurrence of Receive Error (2 bytes are transferred on an external source clock)

Note 1: When the SIO is running (SIO1SR<SIOF> = "1"), do not supply more transfer clock pulses than the number of bytes specified in SIO1CR2<SIORXD> at $\overline{SCK1}$ pin.

Note 2: After data reception is completed, a receive error occurs if eight clock pulses are supplied to the $\overline{SCK1}$ pin before a direction to stop the SIO becomes valid (SIO1CR1<SIOS> = "0"). Figure 2.14.14 shows a case in which a receive error occurs when eight clock pulses are supplied to the $\overline{SCK1}$ pin before the INTSIO1 interrupt service routine writes SIO1CR1<SIOS> = "0".

<u>c. Transmit/receive mode</u>

Transmit/receive mode is assumed by setting SIO1CR1<SIOM> to "10".

Causing the SIO to start transmitting/receiving

1. Set the transmit/receive mode, serial clock rate, and transfer direction, respectively, in SIO1CR1<SIOM>, SIO1CR1<SCK>, and SIO1CR1<SIODIR>.

2. Set the number of data bytes to transfer in SIO1CR2<SIORXD>.

3. Set, in SIO1BUF, as many transmit data bytes as specified in SIO1CR2<SIORXD>.

4. Set SIO1CR1<SIOS> to "1".

   • If the selected serial clock is an internal clock, the SIO immediately starts transmitting/receiving data sequentially in the direction selected using SIO1CR1<SIODIR>.

   • If the selected serial clock is an external clock, the SIO starts transmitting/receiving data, in synchronization with a clock input to the $\overline{SCK1}$ pin sequentially in the direction selected using SIO1CR1<SIODIR>.

Note 1: SIO1CR2<SIORXD>, SIO1CR1<SIODIR>, and SIO1CR1<SCK> are used in common to both transmission and reception. They cannot be set individually.

Note 2: Transmit data are output in synchronization with the falling edge of a signal at the $\overline{SCK1}$ pin. The data are received in synchronization with the rising edge of a signal at the $\overline{SCK1}$ pin.

Causing the SIO to stop transmitting/receiving

5. When as many data bytes as specified in SIO1CR2<SIORXD> have been transmitted and received, be sure to clear SIO1CR1<SIOS> to "0" to halt the SIO. Clearing of SIO1CR1<SIOS> should be executed within the INTSIO1 service routine or should be executed after confirmation of SIO1SR<RXF> = "1".

   • Setting SIO1CR1<SIOINH> to "1" causes the SIO to immediately stop the transmission/reception sequence even if any byte is being transmitted or received.

Received-data read and transmit-data set timing

After as many bytes as specified in SIO1CR2<SIORXD> have been transmitted and received, reading the received data and writing the next transmit data should be executed after confirmation of SIO1SR<RXF> = "1" or should be executed after SIO1CR1<SIOS> is cleared to "0" in the INTSIO1 interrupt service routine. Before transferring the next data after clearing SIO1CR1<SIOS> to "0", make sure SIO1SR<SIOF> = "0", SIO1SR<TXERR> = "0" and SIO1SR<RXERR> = "0" in the external clock mode (No transfer error and no receive error), read the received data, and then write the transmit data and set SIO1CR1<SIOS> = "1" to start transferring.

Note 1: An INTSIO1 interrupt occurs when the last bit of the last data byte is received.

Note 2: When writing to and reading from SIO1BUF, make sure that the number of data bytes to transfer is as specified in SIO1CR2<SIORXD>. If the number is not equal to the value specified in SIO1CR2<SIORXD>, the SIO does not run normally.

Note 3: When as many data bytes as specified in SIO1CR2<SIORXD> are read, SIO1SR<RXF> is cleared to "0".

Note 4: In the transmit/receive mode, setting SIO1CR1<SIOINH> to "1" to force the SIO to halt will cause received data to be discarded.

Note 5: If a transfer sequence is started after a transmit or receive error has been detected, the SIO does not run normally. Before starting transferring, set SIO1CR1<SIOINH> = "1" to force the SIO to halt.
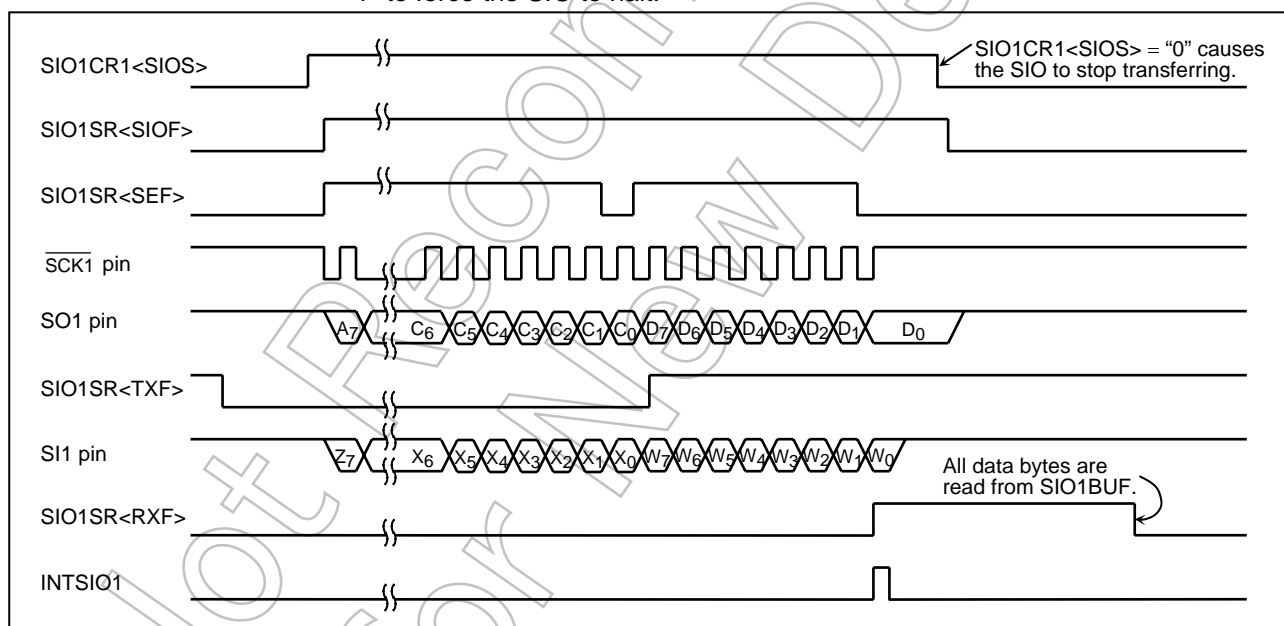
Figure 2.14.15 SIO1CR1<SIOS> Clear Timing (Transmit/receive mode)

Transmit/receive error

During operation on an external clock, the following cases may be detected as a transmit or receive error, causing an error flag (SIO1SR<TXERR> or SIO1SR<RXERR>) to be set. If an error occurs, the SO1 pin goes high.

- If the $\overline{\text{SCK1}}$ pin goes low when the SIO is running (SIO1SR<SIOF> = "1") but there is no transmit data in SIO1BUF (SIO1SR<TXF> = "1").

- If the reception of the next data byte is completed when the SIO is running (SIO1SR<SIOF> = "1") and SIO1BUF is full (SIO1SR<RXF> = "1") (if eight clock pulses are supplied to the $\overline{\text{SCK1}}$ pin) (SIO1SR<RXERR>).

If a transmit or receive error is detected, be sure to set SIO1CR1<SIOINH> to "1" to force the SIO to halt.

Note: If as many clocks of data bytes as specified in SIO1CR2<SIORXD> have been received after the occurrence of receive error, the INTSIO1 is generated. And then, if as many clocks of data bytes as specified in SIO1CR2<SIORXD> have been received continuously again without clearing the SIO1SR<RXERR> flag, the INTSIO1 isn't generated.

Example of setting the transmit/receive mode (Transmit/receive mode, external clock, and 8-byte transfer).

```
                LD       (P0OUTCR), 010*****B      ;  PORT setting.
                                                      P07 (SCK1) input, P06 (SO1) output, and
                                                      P05 (SI1) input.
                LD       (P0DR), 111*****B         ;  Sets SCK1, SO1, and SI1.
                DI                                 ;  IMF ← 0
                LDW      (EIRL), *****1*********0B  ;  Enables INTSIO (EF10)
                EI                                 ;  Enables interrupts.
                LD       (SIO1CR1), 01******B      ;  Initializes the SIO (forces the SIO halt).
WAIT:           TEST     (SIO1SR). 7               ;  Checks to see if the SIO has halted
                                                      (SIO1SR<SIOF> = 0).
                JRS      F, WAIT                   ;  Jumps to START if the SIO is already at a
                                                      halt.
START:          LD       (SIO1CR1), 00100111B      ;  Sets the transmit/receive mode, selects
                                                      the direction of transfer, and sets a serial
                                                      clock.
                LD       (SIO1CR2), 00000111B      ;  Sets the number of bytes (8 bytes) to
                                                      transfer.

                         Transmit data setting

                LD       (SIO1CR1), 10100111B      ;  Starts transferring.
INTSIO1 (INTSIO1 service routine):
                LD       (SIO1CR1), 00100111B      ;  Directs the SIO to stop transferring.
                TEST     (SIOSR). 3                ;  Checks SIO1SR<TXERR>.
                JRS      T, TXNOER
                LD       (SIO1CR1), 01100111B      ;  Forces the SIO to halt (clears
                                                      SIO1SR<TXERR>).

                         Error handling

                JP       END
TXNOER:

                         Receive-data reading
                         Checks a checksum or the like to see if the received data are correct.

END:                                               ;  End of transfer.
```
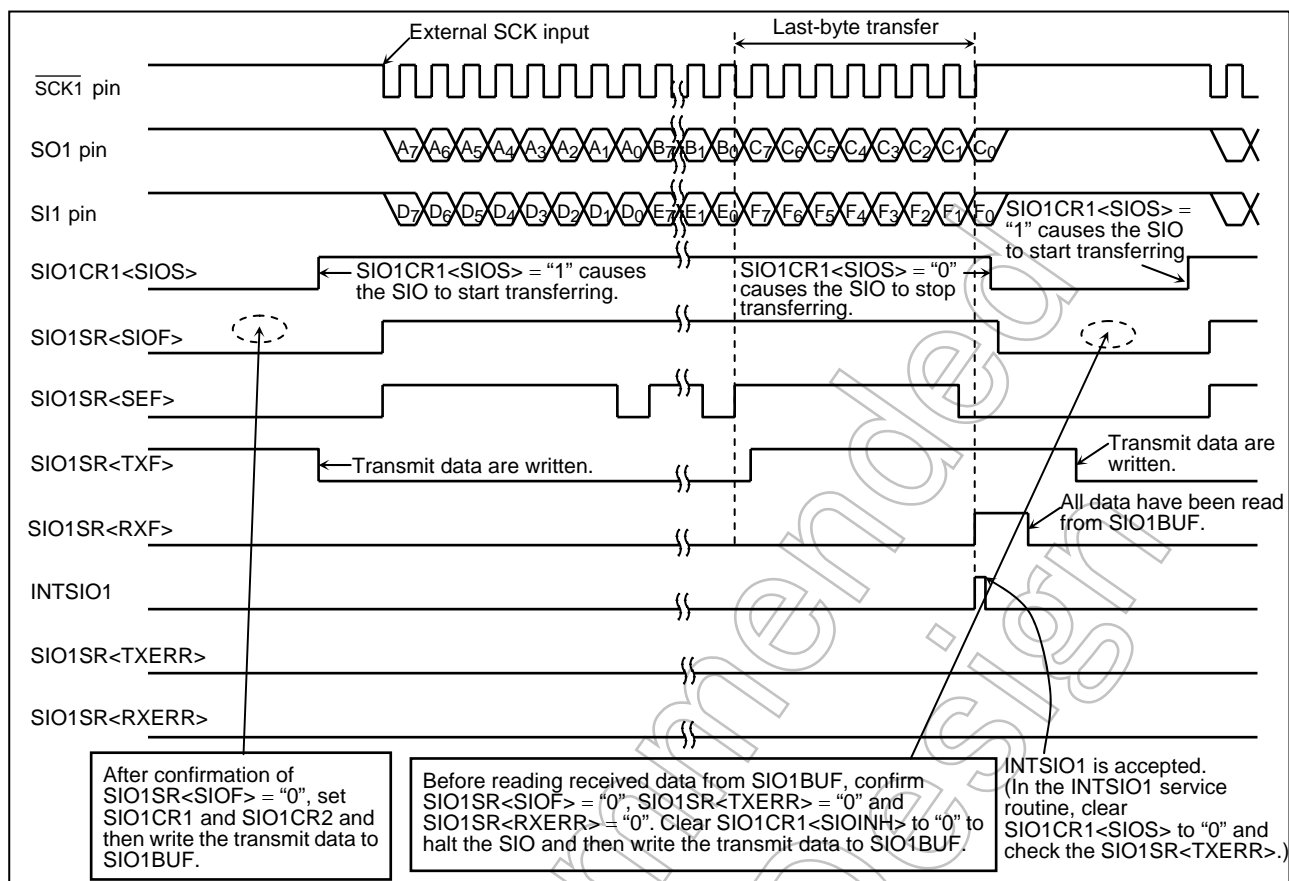
Figure 2.14.16 Transmit/Receive Mode Operation (Where 3 bytes are transferred on an external source clock)
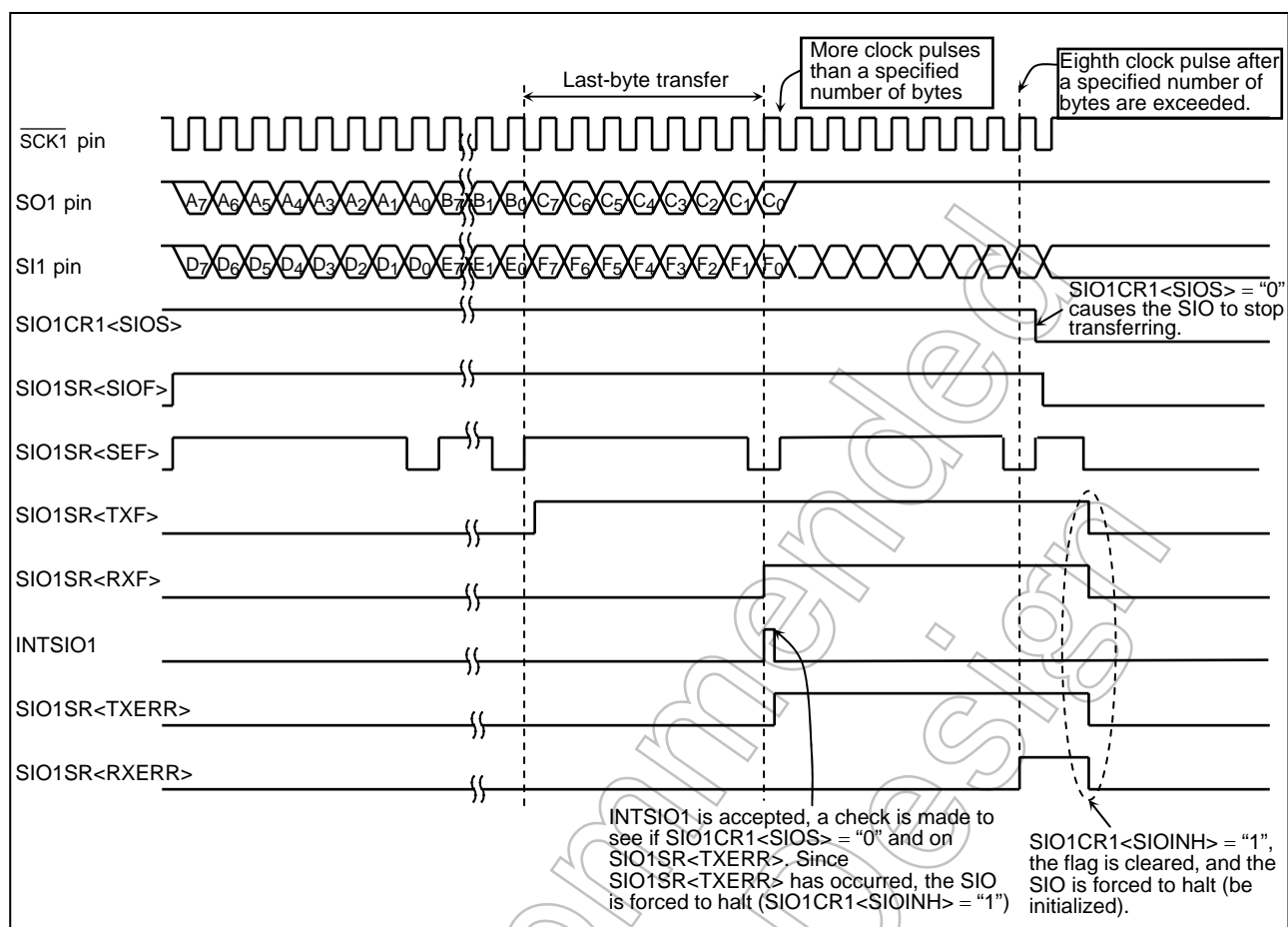
Figure 2.14.17 Occurrence of Transmit/Receive Error (3 bytes are transferred on an external source clock)

Note: When the SIO is running (SIO1SR<SIOF> = "1"), do not supply more transfer clock pulses than the number of bytes specified in SIO1CR2<SIORXD> to the $\overline{SCK1}$ pin.

2.15 Program Patch Logic

The program patch logic provides a function to fix the program code in the on-chip ROM with a bug.

This logic has two modes of operation: Address jump and data replacement. In the address jump mode, the three consecutive bytes starting at the address specified in the program correction address registers can be replaced with an absolute jump instruction. In the data replacement mode, the data at the address specified in the program correction address registers can be replaced with the specified one or two data.

The two modes of operation allow up to four memory locations to be patched.

Note 1: Before using the program patch logic, a routine for this logic must be embedded in the initial routine.

Note 2: To set the jump target in the RAM for the address jump mode, an address trap of RAM must be disabled through the WDTCR1 and WDTCR2 before setting the program correction control register (ROMCCR).

2.15.1 Configuration
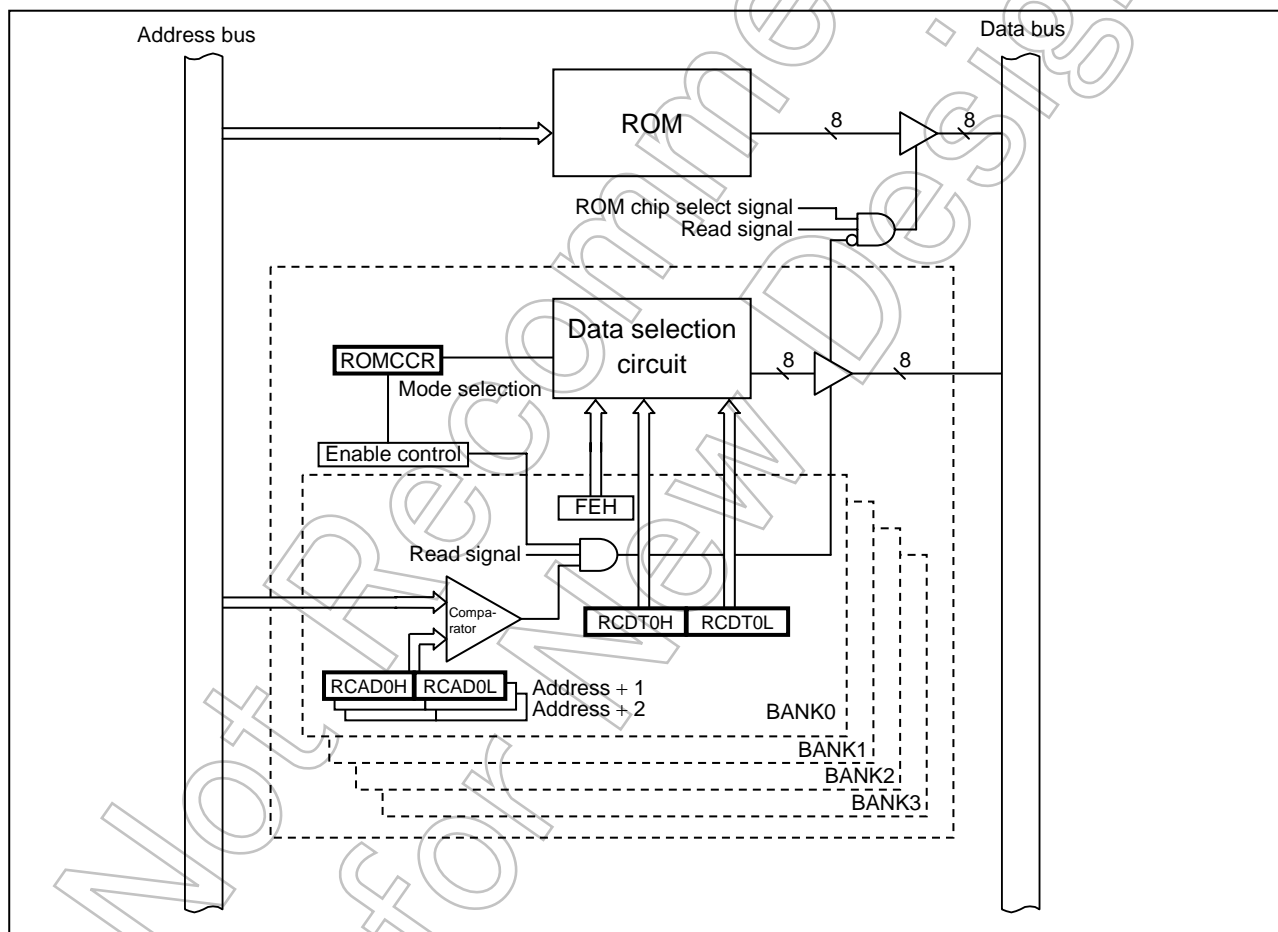
Figure 2.15.1 Program Patch Logic

2.15.2 Control

The program patch logic is controlled by the program correction control register (ROMCCR), program correction address registers (RCADxL, RCADxH), and program correction data registers (RCDTxL, RCDTxH). (x: 0 to 3)

Program Correction Control Register

ROMCCR	7	6	5	4	3	2	1	0	
(1FC0H)	BANK3CNT		BANK2CNT		BANK1CNT		BANK0CNT		(Initial value: 0000 0000)

BANK3CNT	BANK3 control	00: Disable 01: Address jump mode 10: 1-byte data replacement mode 11: 2-byte data replacement mode	R/W
BANK2CNT	BANK2 control	00: Disable 01: Address jump mode 10: 1-byte data replacement mode 11: 2-byte data replacement mode	
BANK1CNT	BANK1 control	00: Disable 01: Address jump mode 10: 1-byte data replacement mode 11: 2-byte data replacement mode	
BANK0CNT	BANK0 control	00: Disable 01: Address jump mode 10: 1-byte data replacement mode 11: 2-byte data replacement mode	

Note 1: If the BANKxCNT is set to a value other than "00B" for the bank, it is impossible to write to the program correction address registers (RCADxL, RCADxH) and program correction data registers (RCDTxL, RCDTxH) (Although the write instruction is executed). Thus, a mode must be selected through the ROMCCR after all these registers are set.

Note 2: The reset vectors (FFFEH, FFFFH) immediately after reset can not be replaced with a patch program because the ROMCCR is initialized by reset.

Figure 2.15.2 Program Correction Control Register

BANK0 Program Correction Address Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCAD0				RCAD0H (1FC2H)								RCAD0L (1FC1H)				

BANK0 Program Correction Data Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCDT0				RCDT0H (1FC4H)								RCDT0L (1FC3H)				

BANK1 Program Correction Address Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCAD1				RCAD1H (1FC6H)								RCAD1L (1FC5H)				

BANK1 Program Correction Data Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCDT1				RCDT1H (1FC8H)								RCDT1L (1FC7H)				

BANK2 Program Correction Address Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCAD2				RCAD2H (1FCAH)								RCAD2L (1FC9H)				

BANK2 Program Correction Data Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCDT2				RCDT2H (1FCCH)								RCDT2L (1FCBH)				

BANK3 Program Correction Address Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCAD3				RCAD3H (1FCEH)								RCAD3L (1FCDH)				

BANK3 Program Correction Data Registers   (Initial value: 0000 0000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCDT3				RCDT3H (1FD0H)								RCDT3L (1FCFH)				

Note 1:   The program correction address registers (RCADx) can only be used for specifying addresses of the user ROM area. If they are used for specifying addresses of other areas (Such as SFR, RAM, and BOOT ROM), the address jump or data replacement mode is not enabled even though it is selected through the ROMCCR. (x: 0 to 3)

Note 2:   Set the RCADx so the addresses of the locations that must be patched will not overlap each other in the banks. If they overlap, the program operation is unpredictable. (x: 0 to 3)

Note 3:   Do not set the RCADx to the areas which are specified in the RCADx, RCDTx, and ROMCCR, or those of the 2-byte areas following the instruction that sets the ROMCCR. If it is set to any one of these areas, the program might not be patched properly.

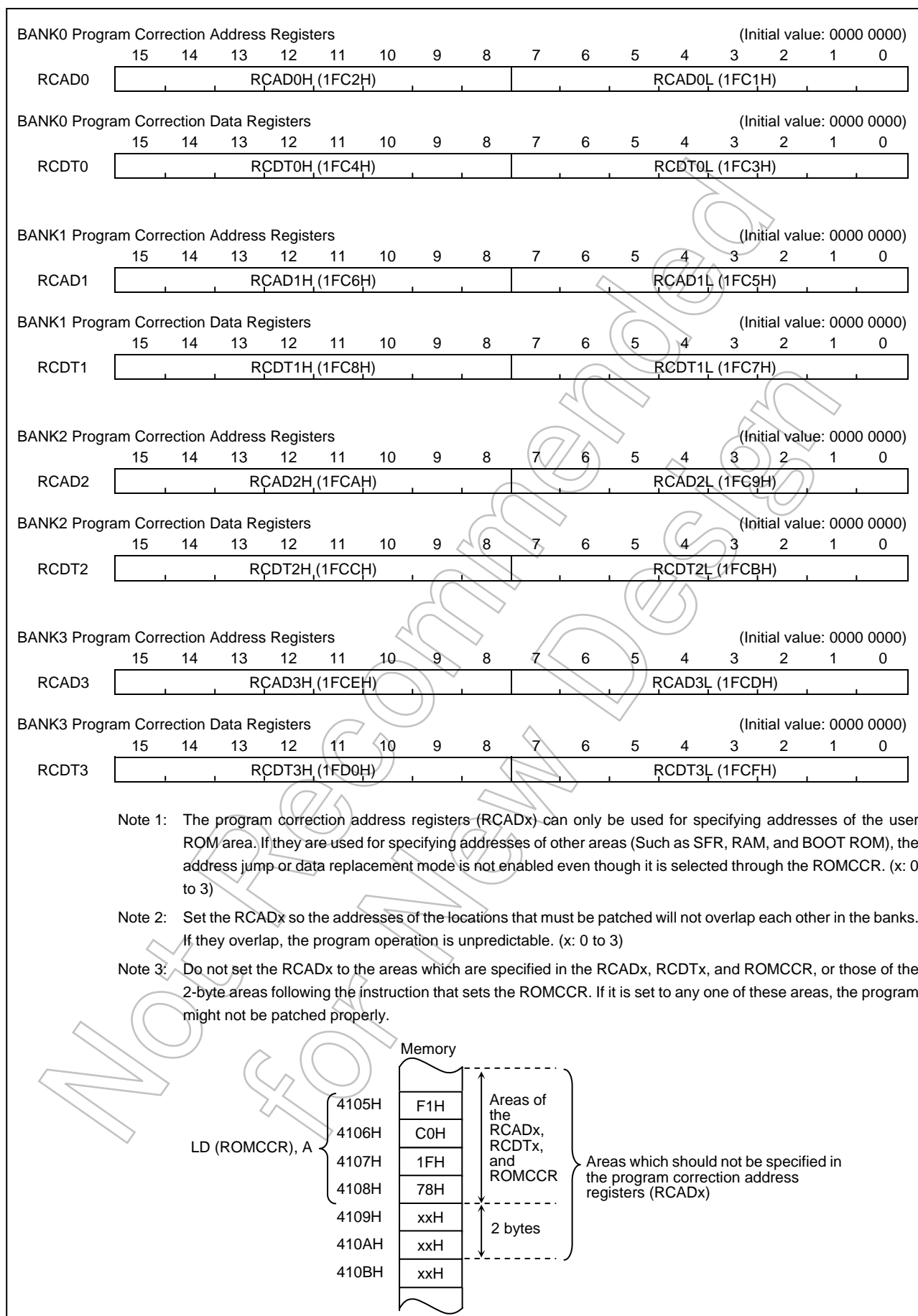|	Memory	
4105H	F1H	Areas of the RCADx, RCDTx, and ROMCCR
LD (ROMCCR), A 4106H	C0H	
4107H	1FH	Areas which should not be specified in the program correction address registers (RCADx)
4108H	78H	
4109H	xxH	2 bytes
410AH	xxH	
410BH	xxH	

Figure 2.15.3 Program Correction Address Register and Data Register

2.15.3 Function

The program patch logic includes a total of four banks of the program correction address registers and program correction data registers. This logic can patch one memory location through a bank. Three correction modes are available for this logic: Address jump, 1-byte data replacement, and 2-byte data replacement. One correction mode should be set for each bank by using the program correction control register (ROMCCR).

(1) Address jump mode

In the address jump mode, any three consecutive bytes of data in ROM can be replaced with an absolute JUMP instruction. When this mode is enabled, the three consecutive bytes starting at the address specified in the program correction address registers are replaced with an absolute JUMP instruction code (FEH, (RCDTxL), (RCDTxH)). (x is defined as 0 to 3 hereinafter.)

- Setting the registers

Set the first ROM address of the location that must be patched in the program correction address registers (RCADxL, RCADxH). (These two registers are called RCADx hereinafter.) Set the jump target address in the program correction data registers (RCDTxL, RCDTxH). The program patch logic enables the address jump mode when the BANKxCNT in the ROMCCR is set to "01B" after all these registers are set.

Note 1: Be sure to specify the address of the first operation address (Start address of the instruction) in the RCADx. If it is specified incorrectly, the result is unpredictable.

Note 2: The three bytes starting at the address specified in the RCADx are replaced with an absolute jump instruction code (FEH (operation code), xxH (operand), xxH (operand)). Therefore, if a JUMP or CALL instruction is executed for the addresses corresponding to these replaced operands (Second and third bytes), the result is unpredictable.

Note 3: To set the jump target in the RAM for the address jump mode, an address trap of the RAM address must be disabled through the WDTCR1 and WDTCR2 before setting the ROMCCR.

Note 4: In the address jump mode, when a read instruction is executed for the address specified in the RCADx, FEH is read from the address.

Example 1: Replacing the three bytes starting at D254H with a JUMP instruction (JP 0300H) through the BANK0.

LD	(WDTCR1), 09H	;	Disable the address trap of RAM.
LD	(WDTCR2), 0D2H	;	Sets the ATRAP control code to WDTCR2.
LD	HL,RCAD0L		
LDW	(HL), 0D254H	;	Sets the program correction address registers.
LD	HL,RCDT0L		
LDW	(HL), 0300H	;	Sets the program correction data registers.
LD	(ROMCCR), 00000001B	;	Sets the program correction control register.
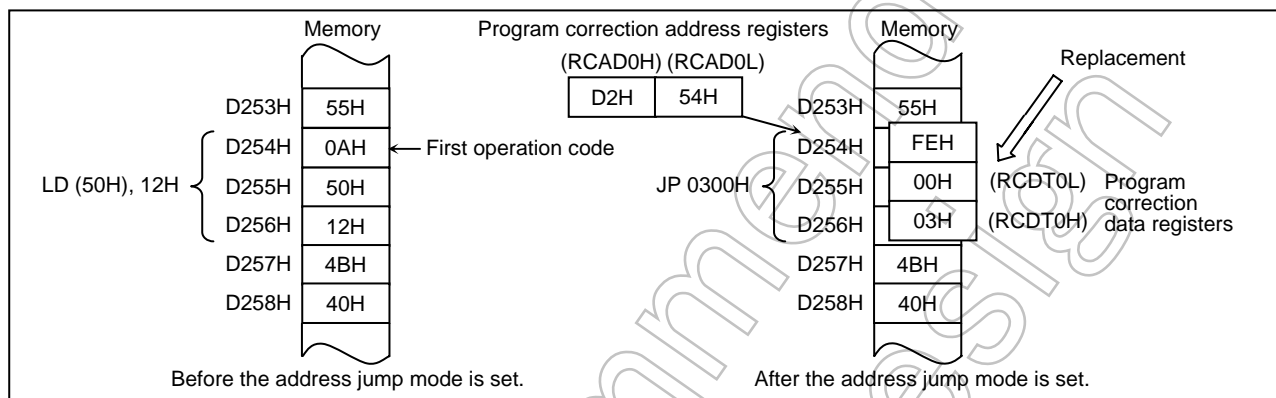
Figure 2.15.4 Example of Replacing the Three Byte (Address jump mode)

Example 2: Setting a sequence for the program patch logic in the initial routine (in the address jump mode).

In the initial routine (normally after reset), set the program patch logic's registers and store a patch program into the on-chip RAM as follows.

(1) Read the flag, which indicates whether to use the program patch logic, from the external memory (Instead of the flag, pin information can be used to determine whether to use the program patch logic).

(2) If the logic is not used, perform normal initial processing.

(3) If it is used, read the ROM address that must be patched and jump target address from the external memory.

(4) Set the ROM address that must be patched in the RCADx, and the jump target address in the RCDTxL and RCDTxH.

(5) Read the patch program from the external memory, and store it into the on-chip RAM. (This step is not required if the jump target is within the ROM.)

(6) Repeat steps (3) through (5) as many times as required banks.

(7) Set the ROMCCR to address jump mode.

Example 3: Executing the patch program in the RAM areas 0200H through 022FH when there is a bug in the ROM areas C020H through C085H.

During the initial routine immediately after reset, read the ROM address that must be patched (C020H) and jump target address (0200H) from the external memory, and set the data in the registers. Store the patch program into the on-chip RAM, and then set the ROMCCR to address jump mode.

When the program execution reaches the instruction at address C020H, the absolute jump instruction is fetched and executed instead of the instruction at address C020H. So, program execution is transferred to the patch program. It is recommended to set the jump instruction to return to the main routine in the end of the patch program.

```
LD      (WDTCR1), 09H          ;  Disable the address trap of RAM.
LD      (WDTCR2), 0D2H         ;  Sets the ATRAP control code to
                                  WDTCR2.
LD      HL, RCAD0L
LDW     (HL), 0C020H           ;  Sets the program correction address
                                  registers.
LD      HL, RCDT0L
LDW     (HL), 0200H            ;  Sets the program correction data
                                  registers.
LD      (ROMCCR), 00000001B    ;  Sets the program correction control
                                  register.
```
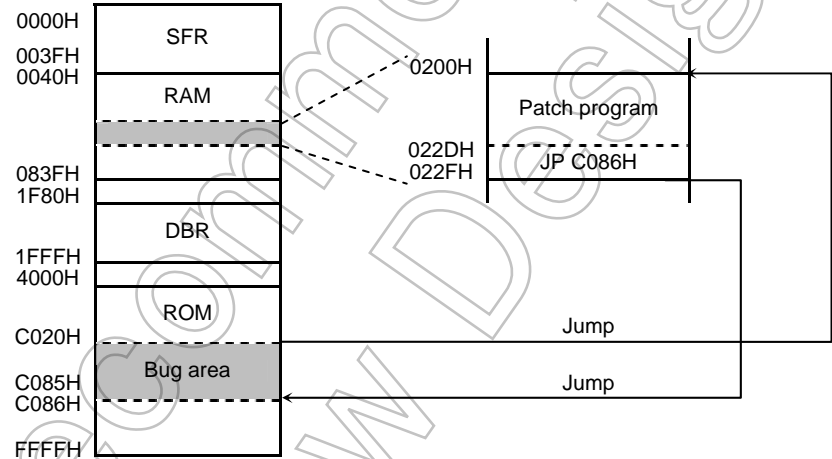

Figure 2.15.5 Example of Executing the Patch Program in the RAM Areas (Address jump mode)

(2) 1-byte data replacement mode

In the 1-byte data replacement mode, any one byte of data in ROM can be replaced with specified data. When this mode is enabled, the data at the address specified in the program correction address registers is replaced with the specified data (RCDTxL). The data of RCDTxH has no effect to ROM in the 1-byte data replacement mode.

Either an operation code or operand can be specified in the program correction address registers.

- Setting the registers

Set the ROM address that must be patched in the program correction address registers (RCADx). Set the corrective data in the program correction data registers (RCDTxL). The program patch logic enables the 1-byte data replacement mode when the BANKxCNT in the ROMCCR is set to "10B" after all these registers are set.

Example 1: Replacing the data at D256H with 34H through the BANK0

LD	HL, RCAD0L	;	Sets the program correction address registers.
LDW	(HL), 0D256H	;	Sets the program correction data registers.
LD	(RCDT0L), 34H		
LD	(ROMCCR), 00000010B	;	Sets the program correction control register.
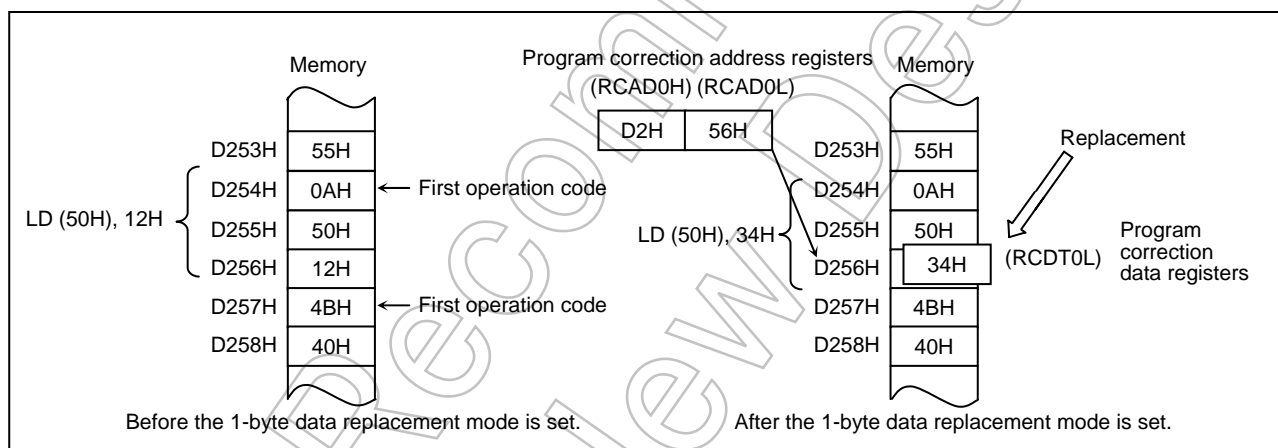
Figure 2.15.6 Example of Replacing the Data (1-byte data replacement mode)

Example 2: Setting a sequence for the program patch logic in the initial routine

In the initial routine (Normally after reset), set the program patch logic's registers as follows.

(1) Read the flag, which indicates whether to use the program patch logic, from the external memory. (Instead of the flag, pin information can be used to determine whether to use the program patch logic.)
(2) If the logic is not used, perform normal initial processing.
(3) If it is used, read the ROM address that must be patched and corrective data from the external memory.
(4) Set the address in the RCADx, and the corrective data in the RCDTxL.
(5) Repeat steps (3) and (4) as many times as required banks.
(6) Set the ROMCCR to 1-byte data replacement mode.

(3) 2-byte data replacement mode

In the 2-byte data replacement mode, any two consecutive bytes of data in the ROM can be replaced with specified data. When this mode is enabled, the 2-byte data starting at the address specified in the program correction address registers is replaced with the specified data (RCDTxL, RCDTxH).

Either an operation code or operand can be specified in the program correction address registers.

- Setting the registers

Set the first ROM address that must be patched in the program correction address registers (RCADx). Set the corrective data in the program correction data registers (RCDTxL, RCDTxH). The program patch logic enables the 2-byte data replacement mode when the BANKxCNT in the ROMCCR is set to "11B" after all these registers are set.

Example 1: Skipping over the LD instruction (LD (50H), 12H) at addresses D254H, D255H, and D256H
with a JR instruction through the BANK0

LD	HL, RCAD0L	;	Sets the program correction address registers.
LDW	(HL), 0D254H		
LD	HL, RCDT0L	;	Sets the program correction data registers.
LDW	(HL), 01FCH		
LD	(ROMCCR), 00000011B	;	Sets the program correction control register.

Figure 2.15.7 Example of Replacing the Data (2-byte data replacement mode)

Note: To change to a relative JUMP instruction in data replacement mode as shown above, set the jump target in the first operation code.

Example 2: Setting a sequence for the program patch logic in the initial routine

In the initial routine (Normally after reset), set the program patch logic's registers as follows.

(1) Read the flag, which indicates whether to use the program patch logic, from the external memory.
(Instead of the flag, pin information can be used to determine whether to use the program patch logic.)

(2) If the logic is not used, perform normal initial processing.

(3) If it is used, read the ROM address that must be patched and corrective data from the external memory.

(4) Set the address in the RCADx, and the corrective data in the RCDTxL and RCDTxH.

(5) Repeat steps (3) and (4) as many times as required banks.

(6) Set the ROMCCR to the 2-byte data replacement mode.

Example 3: Replacing data 55H at address C020H with 33H, and replacing data AAH at address C021H with CCH.

During the initial routine immediately after reset, read the first ROM address that must be patched (C020H) and the corrective data (33H and CCH) from the external memory, and set the data in the registers. Then, set the ROMCCR to the 2-byte replacement mode.

When the data is fetched or read at address C020H, 33H instead of 55H is brought into the CPU. When the data is fetched or read at address C021H, CCH instead of AAH is brought into the CPU.

```
LD      HL, RCAD0L          ;   Sets the program correction address
                                registers.
LDW     (HL), 0C020H
LD      HL, RCDT0L
LDW     (HL), 0CC33H        ;   Sets the program correction data
                                registers.
LD      (ROMCCR), 00000011B ;   Sets the program correction control
                                register.
```
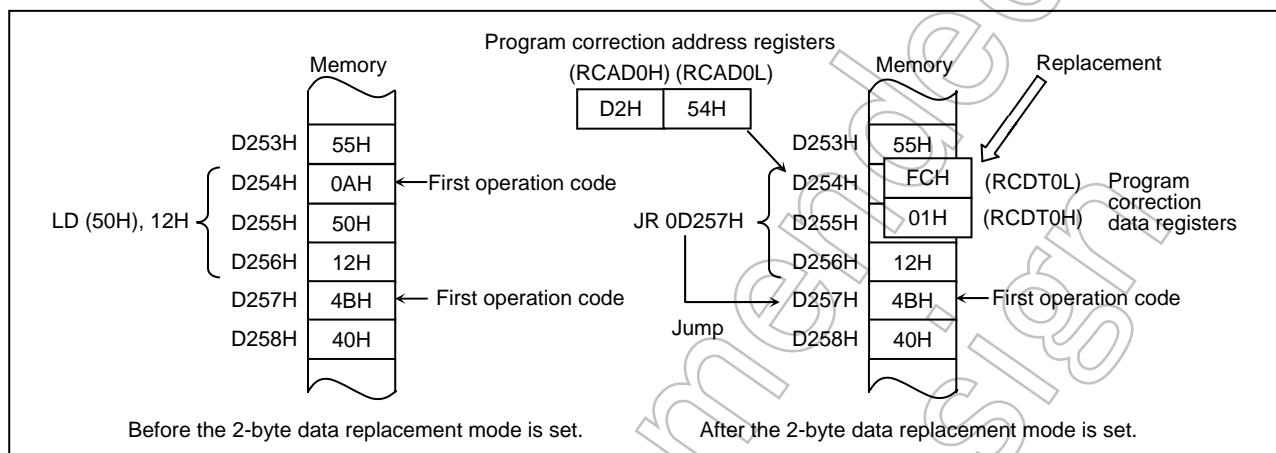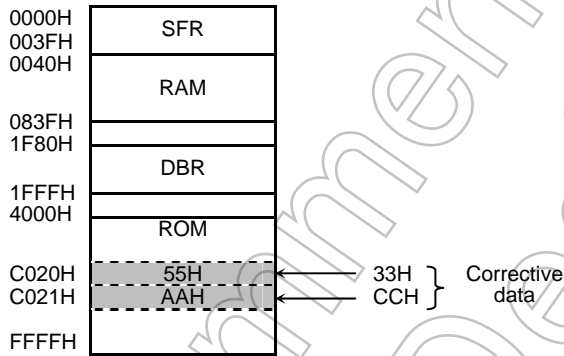

Figure 2.15.8 Example of Replacing the Data (2-byte data replacement mode)

2.16 FLASH Memory

2.16.1 Outline

The TMP86FP24 incorporates 49152 bytes of FLASH memory (Address 4000H to FFFFH). The writing to FLASH is controlled by FLASH control register (EEPCR), FLASH status register (EEPSR) and FLASH write emulate time control register (EEPEVA).

To write data to the FLASH, execute the serial PROM mode. For details about the serial PROM mode, refer to 2.1 "Serial PROM mode".

The FLASH memory of the TMP86FP24 features:

- The FLASH memory is constructed of 384 pages FLASH and one page size is 128 bytes (384 pages × 128 bytes = 49512 bytes).

- The TMP86FP24 incorporates a 128-byte temporary data buffer. The data written to FLASH is temporarily stored in this data buffer. After 128 bytes data have been written to the temporary data buffer, the writing to FLASH automatically starts by page writing (the 128 bytes data are written to specified page of FLASH simultaneously). At the same time, page-by-page erasing occurs automatically. So, it is unnecessary to erase individual pages in advance.

- The FLASH control circuit incorporates an oscillator dedicated to the FLASH. So FLASH writing time is independent of the system clock frequency (fc). In addition, because an FLASH control circuit controls writing time for each FLASH cell, the writing time varies in each page (Typically 4 ms per page).

- Controlling the power for the FLASH control circuit (Regulator and voltage step-up circuit) achieves low power consumption if the FLASH is not in use (e.g., when the program is executed in RAM area).

2.16.2 Conditions for Accessing the FLASH Areas

The conditions for accessing the FLASH areas vary depending on each operation mode. The following tables shows FLASH are access conditions.

Table 2.16.1 FLASH Area Access Conditions

	Area	Operation Mode	
		MCU Mode (Note 1)	Serial PROM Mode (Note 2)
FLASH Memory	4000H to FFFFH	Read/fetch only	Write/read/fetch supported

Note 1: "MCU mode" shows NORMAL1/2 and SLOW1/2 modes.

Note 2: "Serial PROM mode" shows the FLASH controlling mode. For details, refer to 2.1 "Serial PROM mode".

Note 3: "Fetch" means reading operation of FLASH data as an instruction by CPU.

2.16.3 Differences among Product Series

The specifications of the FLASH product (TMP86FP24) are different from those of the emulation chip (TMP86C948) and masked ROM product (TMP86CP24) as listed below. See 2.17.2 "Control" for explanations about the control registers.

	FLASH Product (TMP86FP24F)	Emulation Chip (TMP86C948XB)	Masked ROM Product (TMP86CP24F)
Rewriting the EEPCR register <EEPMD, EEPRS, MNPWDW>	It is possible to rewrite the EEPCR register only when the program execution area in use is RAM/BOOT ROM	In the debugger memory window, it is impossible to rewrite the EEPCR register.	Neither the EEPMD nor EEPRS itself does not function.
Accessing the EEPEVA register	It is possible only to write- and read-access the EEPEVA register. The writing to this register does not affect the function.	The time required to emulate FLASH writing is put under control.	It is possible only to write- and read-access the EEPEVA register. The writing to this register does not affect the function.
FLASH write time (The emulation chip is written to emulation memory instead of FLASH)	Typically 4 ms (Independent of the system clock)	The FLASH write time is setup using the EEPEVA register (Dependent on the system clock).	- (Writing to an area that corresponds to the FLASH area causes nothing.)
Executing a read instruction/fetch to the 4000H to FFFFH area when EEPSR<BFBUSY> = "1".	FFH is always read regardless of the ROM data. Fetching FFH results in a software interrupt occurring.	The debugger memory window always displays ROM data.	Always masked ROM data is read.
Executing a write instruction to the 4000H to FFFFH area when EEPCR<EEPMD> = "0011" EEPSR<EWUPEN> = "1" and EEPSR<BFBUSY> = "0" — MCU mode	The EEPSR<BFBUSY> stays at "0" (Write disabled).	In the debugger memory window, it is possible to rewrite the 4000H to FFFFH area (the EEPSR<BFBUSY> remains unchanged).	
Executing a write instruction to the 4000H to FFFFH area when EEPCR<EEPMD> = "0011" EEPSR<EWUPEN> = "1" and EEPSR<BFBUSY> = "0" — Serial PROM mode	The EEPSR<BFBUSY> is set to "1" (Write enabled).		
BOOT ROM	2 Kbytes are included in the 3800H to 3FFFH area.		No BOOT ROM is included. Executing a read/fetch to the 3800H to 3FFFH area causes "FFH" to be read. Fetching "FFH" results in a software interrupt occurring.
Operating voltage	VDD = 1.8 to 3.6 V	VDD = 1.8 to 3.3 V	VDD = 1.8 to 3.6 V

2.16.4   FLASH Memory Configuration

128 consecutive bytes in the FLASH area are treated as one group, which is defined as a page. The TMP86FP24 incorporates a one-page temporary data buffer. Writing data to FLASH is temporarily stored in this 128-byte data buffer. After 128 bytes data have been written to the temporary data buffer, these data are written to specified page of FLASH at a time. However, data can be read from any address byte by byte.

2.16.4.1 Page Configuration

The FLASH area has a page configuration of 128 bytes/page as shown below. The total number of bytes in it is 384 pages × 128 bytes (= 49152 bytes). The writeable area is 4000H to FFFFH in serial PROM mode.

Note: The FLASH area (4000H to FFFFH) can be written only in the serial PROM mode. For details of the serial PROM mode, refer to 2.1 "Serial PROM Mode".
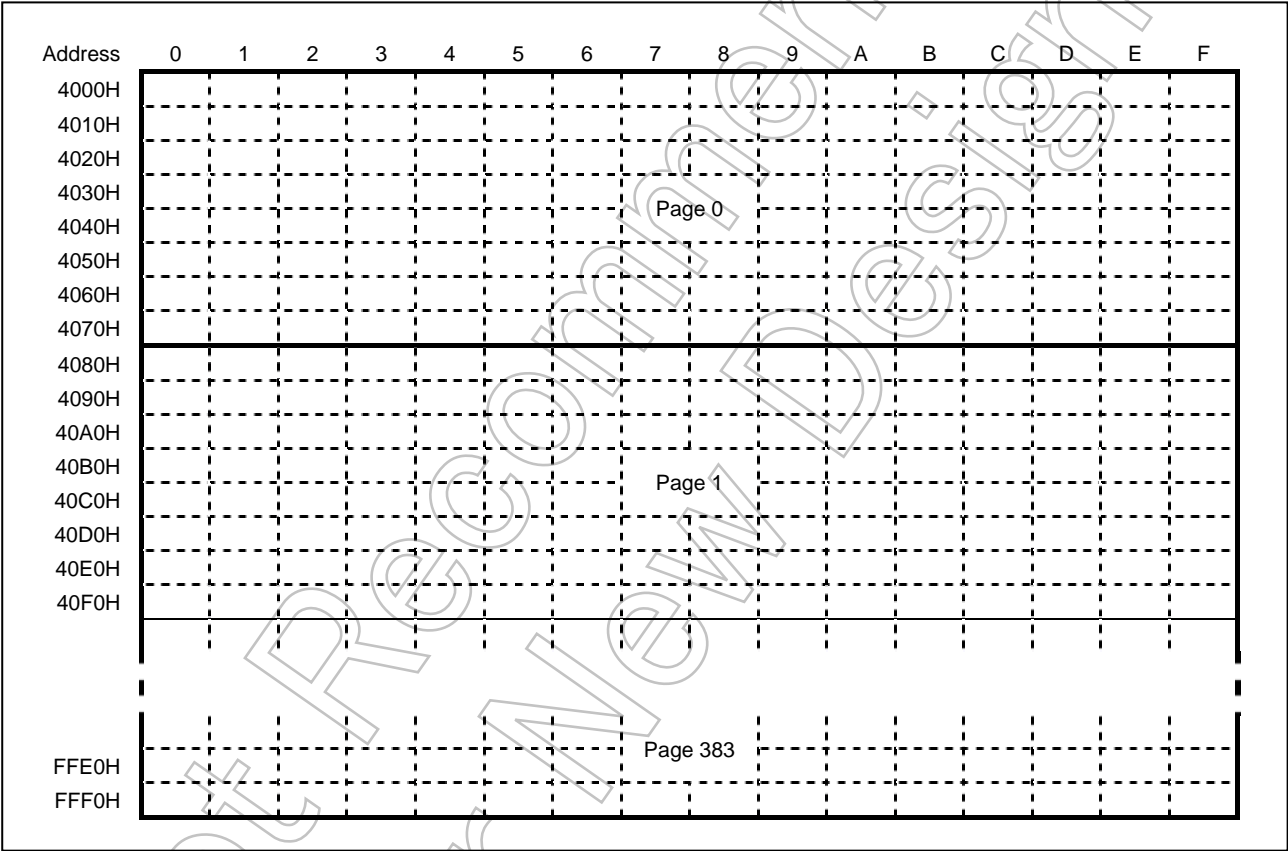

Figure 2.16.1 Page Configuration

2.17 FLASH Memory Control Circuit

2.17.1 Configuration
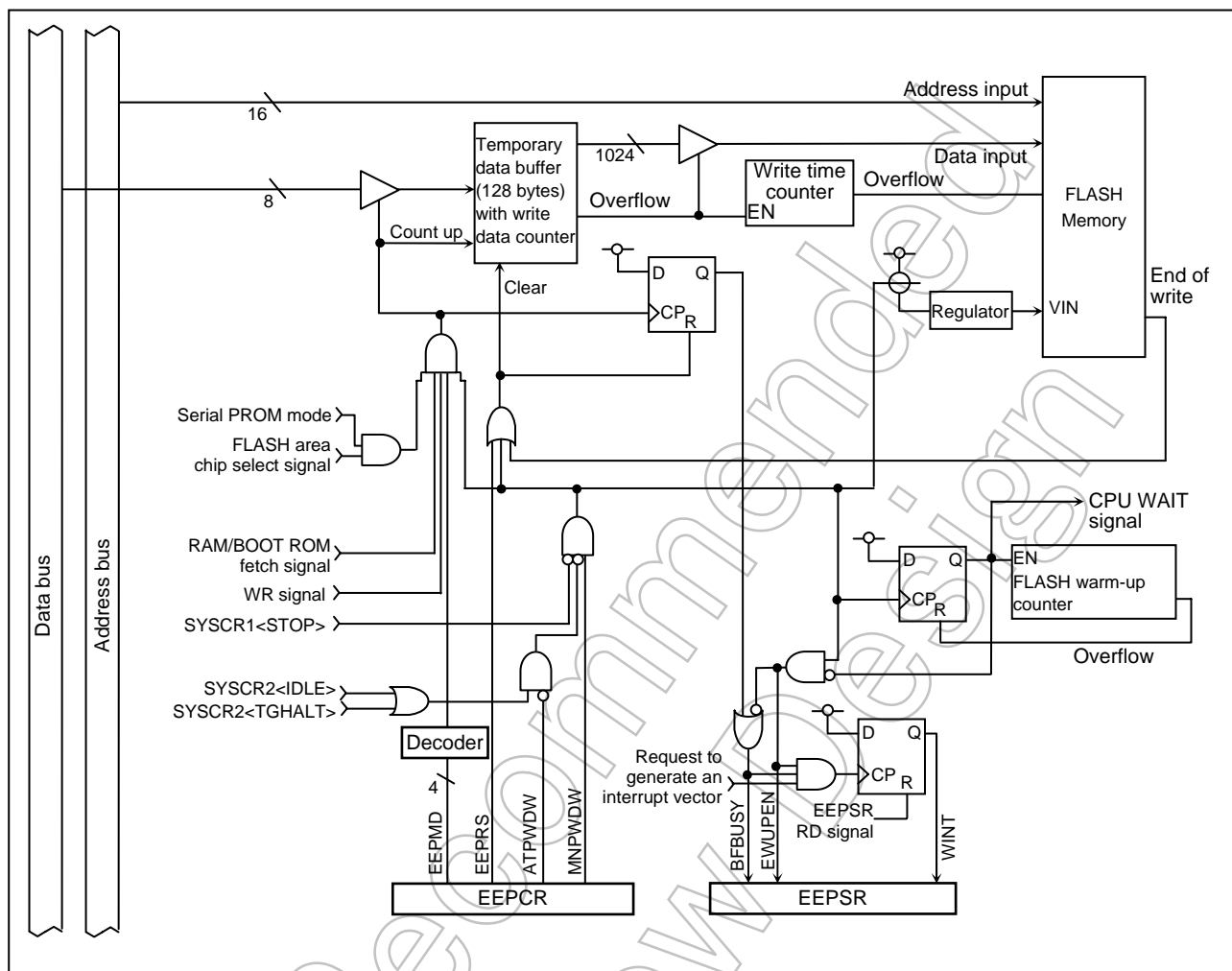
Figure 2.17.1 FLASH Memory Control

2.17.2 Control

The FLASH is controlled by FLASH control register (EEPCR), FLASH status register (EEPSR) and FLASH write emulate time control register (EEPEVA).

FLASH control register

	7	6	5	4	3	2	1	0	
EEPCR (1FE0H)			EEPMD			EEPRS	ATPWDW	MNPWDW	(Initial value: 1100 *011)

			Program Execution Area	
			RAM/ BOOT	FLASH
EEPMD	FLASH write enable control (Write protect)	1100: FLASH write disable 0011: FLASH write enable Other values: Reserved		Read only
EEPRS	FLASH write forcible stop	0: − 1: FLASH writing is forced to stop. (The write data counter is initialized.) * After writing "1" to EEPRS, it is automatically cleared to "0".	R/W	Read only
ATPWDW	Automatic power control for the FLASH control circuit in the IDLE0/1/2, SLEEP0/1/2 modes. (This bit is available only when MNPWDW is set to "1".)	0: Automatic power shut down is executed in IDLE0/1/2 and SLEEP0/1/2 modes. 1: Automatic power shut down is not executed in IDLE0/1/2 and SLEEP0/1/2 modes. (The power is always supplied in these modes.)		R/W
MNPWDW	Software-based power control for the FLASH control circuit	0: The power for the FLASH control circuit is turned off. 1: The power for the FLASH control circuit is turned on		Read only

Note 1: The EEPMD, EEPRS, and MNPWDW can be rewritten only when a program fetch is taking place in the RAM or BOOT ROM area. If an attempt is made to rewrite the EEPCR register when a program is being executed in the FLASH area, the EEPMD, EEPRS, and MNPWDW keep holding the previous data; they are not rewritten.

Note 2: To write to the FLASH, set the EEPMD with "0011B" in advance when a program fetch is taking place in the RAM area.

Note 3: To forcibly stop writing of FLASH, set the EEPRS to "1" when a program fetch is taking place in the RAM area.

Note 4: The ATPWDW functions only if the MNPWDW is "1". If the MNPWDW is "0", the power for the FLASH control circuit is kept turned off regardless of the setting of the ATPWDW.

Note 5: When a STOP mode is executed, the power for the FLASH control circuit is turned off regardless of the setting of the ATPWDW. If the MNPWDW is "0", entering/exiting the STOP mode allows the power for the FLASH control circuit to be kept turned off.

Note 6: Executing a read instruction to the EEPCR register results in bit3 being read as undefined. Bit2 is always read as "0".

Note 7: The following attention is necessary when the MNPWDW is set or cleared.

When the MNPWDW is changed from "1" to "0"	Clear the interrupt master enable flag (IMF) to "0" in advance to disable an interrupt. After that, do not set IMF to "1" during EEPSR<EWUPEN> = "0". If a watchdog timer is used as interrupt request, clear the binary counter for the watchdog timer just before MNPWDW is changed from "1" to "0".
When the MNPWDW is changed from "0" to "1"	When write to or read from the flash memory, make sure that the EEPSR<EWUPEN> is "1" by software. Once the MNPWDW is rewritten from "0" to "1" by software, keep performing software based polling until the EEPSR<EWUPEN> becomes "1".

Figure 2.17.2 FLASH Control Register

FLASH status register

	7	6	5	4	3	2	1	0	
EEPSR (1FE1H)						WINT	EWUPEN	BFBUSY	(Initial value: **** *010)

			Operating (Power on)		Halt (Power off) or warm-up	
WINT	Interrupt detection during a write to the FLASH		0: Not detected 1: Detected (Interrupt occurred) * WINT is automatically cleared to "0" when read instruction is executed to EEPSR.			Read only
EWUPEN	FLASH control circuit status monitor	Control circuit status	Operating (Power on)		Halt (Power off) or warm-up	
		FLASH status	Temporary data buffer empty	Writing	Disable	
			1	1	0	
BFBUSY	FLASH write busy flag		0	1	1	

Note 1: If a non-maskable interrupt occurs during a write to the FLASH, the WINT is set to "1" and the writing is discontinued, and then warm-up (CPU wait) for the control circuit of flash memory is executed. (The write data counter is initialized.) If WINT = "1" is detected in the non-maskable interrupt service routine, a write is not completed successfully. So, it is necessary to try a write again. The content of the page to which a write is taking place may be changed to an unexpected value depending on the timing when the WINT becomes "1".

Note 2: Even if a non-maskable interrupt occurs during an FLASH warm-up, the CPU stays at a halt until the warm-up is finished.

Note 3: The WINT is automatically cleared to "0" when a read instruction is executed to the EEPSR register.

Note 4: When MNPWDW is changed from "0" to "1", EWUPEN becomes "1" after taking $2^{10}/fc$ [s] (if SYSCK = "0") or $2^{3}/fs$ [s] (if SYSCK = "1"). Before accessing the FLASH, make sure that the EWUPEN is "1" in the RAM area.

Note 5: If the BFBUSY is "1", executing a read instruction or fetch to the FLASH area causes FFH to be read. Fetching FFH results in a software interrupt occurring.

Note 6: In masked ROM product, if the EWUPEN is "1", writing to the masked ROM area that corresponds to the FLASH area does not set the BFBUSY of the masked ROM product to "1".

Figure 2.17.3 FLASH Status Register

FLASH Write Emulation Time Control Register (Setting of this register functions only in emulation chip (TMP86C948).)

	7	6	5	4	3	2	1	0	
EEPEVA (1FE2H)							EEPSUCR		(Initial value: **** *000)
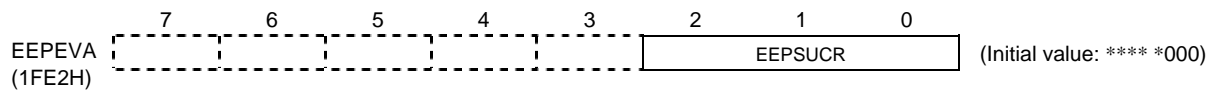
			Emulation Chip (TMP86C948)		FLASH Product (TMP86FP24)	
			NORMAL1/2 IDLE1/2 Modes	SLOW1/2 SLEEP1/2 Modes	All operation Modes	
EEPSUCR	Controlling the FLASH write emulation time [s] for emulation chip	000 001 010 011 100 101 110 111	2^{16}/fc 2^{15}/fc 2^{14}/fc 2^{13}/fc 2^{12}/fc Reserved Reserved Reserved	2^{7}/fs 2^{6}/fs 2^{5}/fs 2^{4}/fs 2^{3}/fs Reserved Reserved Reserved	Typ. 4 ms (Regardless of register settings and the system clock.)	R/W

Note 1:   Only in the emulation chip, the EEPSUCR functions. In the FLASH and masked ROM products, it is possible only to write- and read-access the register. It does not actually function. Because the FLASH product incorporates a dedicated oscillator, its write time is independent of the system clock.

Note 2:   Executing a read instruction to the EEPEVA register results in bits 7 to 3 being read as undefined.

Note 3:   The following table lists the write emulation time specified by the setting of the EEPSUCR. Select an appropriate value according to the operating frequency used. The shading indicates recommended settings.

	EEPSUCR Setting	NORMAL1/2 Mode					SLOW1/2 Mode
		fc = 16 MHz	fc = 8 MHz	fc = 4 MHz	fc = 2 MHz	fc = 1 MHz	fs = 32.768 kHz
Write Time [ms]	000	4.10	8.19	16.38	32.77	65.54	3.91
	001	2.05	4.10	8.19	16.38	32.77	1.95
	010	1.02	2.05	4.10	8.19	16.38	0.98
	011	0.51	1.02	2.05	4.10	8.19	0.49
	100	0.26	0.51	1.02	2.05	4.10	0.24

Figure 2.17.4 FLASH Control Register and FLASH Status Register

2.17.3 FLASH Write Enable Control (EEPCR<EEPMD>)

In the FLASH product, the control register can be used to disable a write to the FLASH (Write protect) in order to prevent a write to the FLASH from occurring by mistake because of a program error or microcontroller malfunction. To enable a write to the FLASH, set the EEPCR<EEPMD> with 0011B. To disable a write to the FLASH, set the EEPCR<EEPMD> with 1100B. A reset initializes the EEPCR<EEPMD> to 1100B to disable a write to the FLASH. Usually, set the EEPCR<EEPMD> with 1100B, except when it is necessary to write to the FLASH.

Note 1: The FLASH memory (4000H to FFFFH) can be written only in the serial PROM mode.

Note 2: The EEPCR<EEPMD> can be rewritten only when a program is being executed in the RAM area. Executing a write instruction to the EEPCR<EEPMD> in the FLASH area does not change its setting.

Note 3: In the masked ROM product, executing a write instruction to the EEPCR<EEPMD> changes its setting; however, the new setting does not take effect.

2.17.4   FLASH Write Forcible Stop (EEPCR<EEPRS>)

To forcibly stop a write to the FLASH, set the EEPCR<EEPRS> to "1". Setting the EEPCR<EEPRS> to "1" initializes the write data counter of data buffer and forcibly stops a write, and then a warm-up (CPU wait) for the control circuit of flash memory is executed. After warm-up period, the EEPSR<BFBUSY> is cleared to "0". The warm-up period is $2^{10}/fc$ (SYSCK = "0") or $2^3/fs$ (SYSCK = "1"). After this, if writing to FLASH starts again, data is stored as the 1st byte of the temporary data buffer and sets the EEPSR<BFBUSY> to "1". Therefore, it is necessary to write 128 bytes data to the temporary data buffer.

After 1 to 127 bytes are saved to the temporary data buffer, if the EEPCR<EEPRS> is set to "1" the specified page of FLASH is not written. (It keeps previous data.)

Note 1: After 128 bytes are written to the temporary data buffer, the setting the EEPCR<EEPRS> to "1" may cause the writing the page of FLASH to an unexpected value.

Note 2: The EEPCR<EEPRS> can be rewritten only when a program is being executed in the RAM area. In the FLASH area, executing a write instruction to the EEPCR<EEPRS> does not affect its setting.

Note 3: During the warm-up period for flash memory (CPU wait), the peripheral circuits continue operating, but the CPU stays at a halt until the warm-up is finished. Even if an interrupt latch is set to "1" by generating of interrupt request, an interrupt se'quence doesn't start till the end of warm-up. If interrupts occur during a warm-up period with IMF = "1", the interrupt sequence which depends on interrupt priority will start after warm-up period.

Note 4: When the EEPCR<EEPRS> is set to "1" with EEPSR<BFBUSY> = "0", a warm-up is not executed.

Note 5: If executed a write or read instruction to the flash area immediately after setting EEPCR<EEPRS>, insert one or more machine cycle instructions after setting EEPCR<EEPRS>.

```
Example: Reads the flash memory data immediately after setting EEPCR<EEPRS> to "1"
        LD        HL, 8000H
        LD        (EEPCR), 3FH              ; Set EEPCR<EEPRS> to "1".
        NOP                                 ; NOP
                                             (Do not execute write or read instruction immediately
                                              after setting EEPCR<EEPRS>.)
        LD        A,(HL)                    ; Reads the data of address 8000H
                                             (Write or read instruction to the flash memory).
```
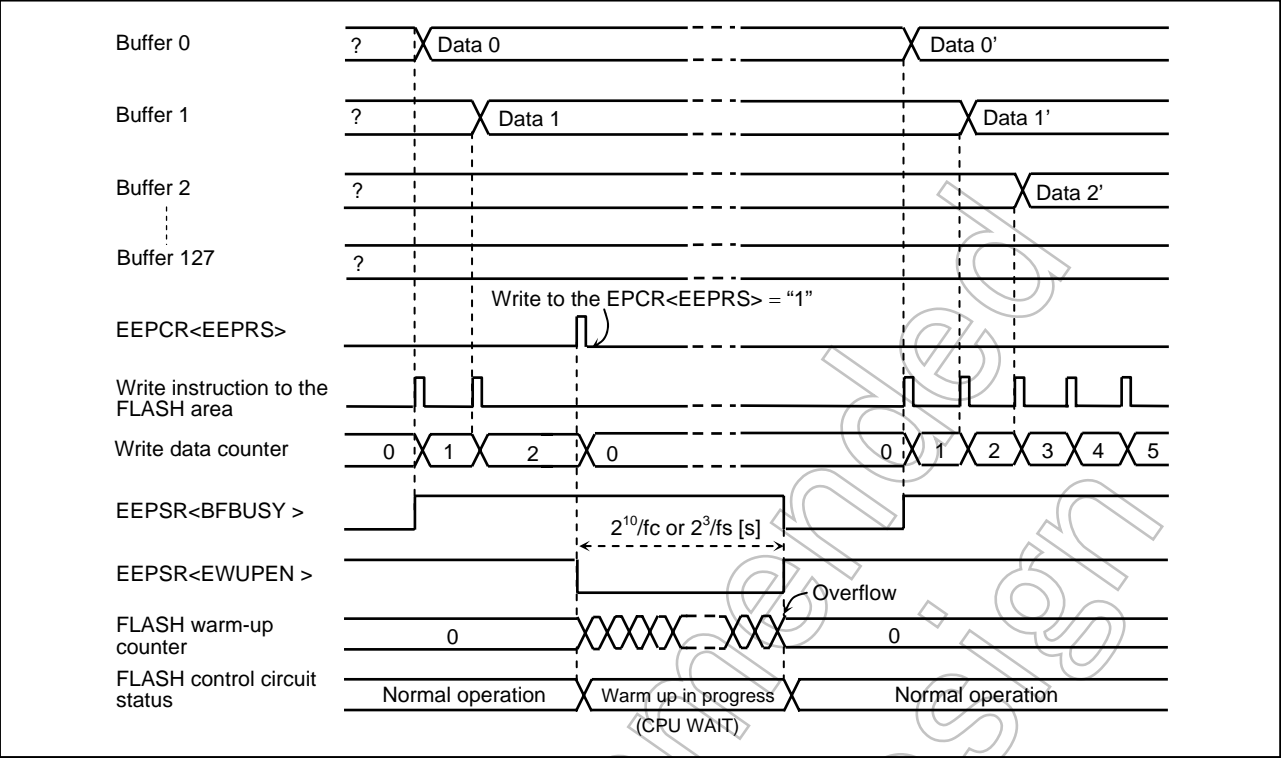
Figure 2.17.5 Write Data Counter Initialization and Write Forcible Stop

2.17.5 Power Control for the FLASH Control Circuit

For the FLASH product, it is possible to turn off the power for FLASH control circuit (Such as a regulator) to suppress power consumption if the FLASH area is not accessed. For the emulation chip (TMP86C948) and the masked ROM product (TMP86CP24), the register setting and the CPU wait functions behave in the same manner as for the FLASH product to maintain compatibility; however, power consumption is not suppressed.

The EEPCR<MNPWDW> and EEPCR<ATPWDW> are used to control the power for the FLASH control circuit. If the power for the FLASH control circuit is turned off according to the setting of these registers, starting to use the circuits again needs to allow warm-up time for the power supply.

Table 2.17.1 Power Supply Warm-up Time (CPU wait) for the FLASH Control Circuit

NORMAL1/2 IDLE0/1/2 Mode	SLOW1/2 SLEEP0/1/2 Mode	STOP Mode (when EEPCR<MNPWDW> = "1")	
		To Return to a NORMAL Mode	To Return to a SLOW Mode
2^{10}/fc [s] (64 μs at 16 MHz)	2^3/fs [s] (244 μs at 32.768 kHz)	STOP warm-up time + 2^{10}/fc [s]	STOP warm-up time + 2^3/fs [s]

2.17.5.1 Software-based Power Control for the FLASH Control Circuit (EEPCR<MNPWDW>)

The EEPCR<MNPWDW> is a software-based power control bit for the FLASH control circuit. When a program is being executed in the RAM area, setting this bit enables software-based power control. Clearing the EEPCR<MNPWDW> to "0" immediately turns off the power for the FLASH control circuit. Once the EEPCR<MNPWDW> is switched from "0" to "1", before attempting a read or fetch from the FLASH area, it is necessary to insert a warm-up period by software until the power supply is stabilized. In this case, because the CPU wait is not executed, any other instructions except accessing to flash (Write or read) are available. When MNPWDW is changed from "0" to "1", EWUPEN becomes "1" after taking 2^{10}/fc [s] (SYSCK = "0") or 2^3/fs [s] (SYSCK = "1"). Usually software-based polling should be performed until the EEPSR<EWUPEN> becomes "1". An example of setting is given below.

(1) Example of controlling the EEPCR<MNPWDW>

1. Transfer a program for controlling the EEPCR<MNPWDW> to the RAM area.

2. Release an address trap in the RAM area (setup the WDTCR1 and WDTCR2 registers).

3. Jump to the control program transferred to the RAM area.

4. Clear the interrupt master enable flag (IMF ← "0").

5. Clear the binary counter if the watchdog timer is in use.

6. To turn off the power for the FLASH control circuit, clear the EEPCR<MNPWDW> to "0".

7. Perform CPU processing as required.

8. To access the FLASH area again, set the EEPCR<MNPWDW> to "1".

9. Keep program polling until the EEPSR<EWUPEN> becomes "1".
   (Upon completion of an FLASH warm up, the EEPSR<EWUPEN> is set to "1". It takes 2^{10}/fc (SYSCK = "0") or 2^3/fs (SYSCK = "1") until EWUPEN becomes "1".)

This procedure enables the FLASH area to be accessed.

If the EEPCR<MNPWDW> is "1", entering a STOP mode forcibly turns off the power for the FLASH control circuit. When the STOP mode is released, a STOP mode oscillation warm-up is carried out, and then the CPU wait period (Warm up for stabilizing of FLASH power supply circuit) is automatically performed. If the EEPCR<MNPWDW> is "0", entering/exiting the STOP mode keeps the power for the FLASH control circuit turned off.

Note 1: If the EEPSR<EWUPEN> is "0", do not access (Fetch, read, or write) the FLASH area. Executing a read instruction or fetch to the FLASH area causes FFH to be read. Fetching FFH results in a software interrupt occurring. For the masked ROM product, however, masked ROM data is always read regardless of the state of the EEPSR<EWUPEN>.

Note 2: To clear the EEPCR<MNPWDW> to "0", clear the interrupt master enable flag (IMF) to "0" in advance to disable an interrupt. After that, do not set IMF to "1" during EEPSR<EWUPEN> = "0".

Note 3: If the EEPCR<MNPWDW> is "0", generating a non-maskable interrupt automatically rewrites the MNPWDW to "1" to warm up the FLASH control circuit (CPU wait). That time, the peripheral circuits continue operating, but the CPU stays at a halt until the warm-up is finished.

Note 4: The EEPCR<MNPWDW> can be rewritten only when a program is being executed in the RAM area. In the FLASH area, executing a write instruction to the EEPCR<MNPWDW> does not affect its setting.

Note 5: If a watchdog timer is used as interrupt request, clear the binary counter for the watchdog timer just before MNPWDW is changed from "1" to "0".

Note 6: During the warm-up period with a software polling of EEPSR<EWUPEN>, if a non-maskable interrupt occurs, the CPU stays at a halt until the warm-up is finished.
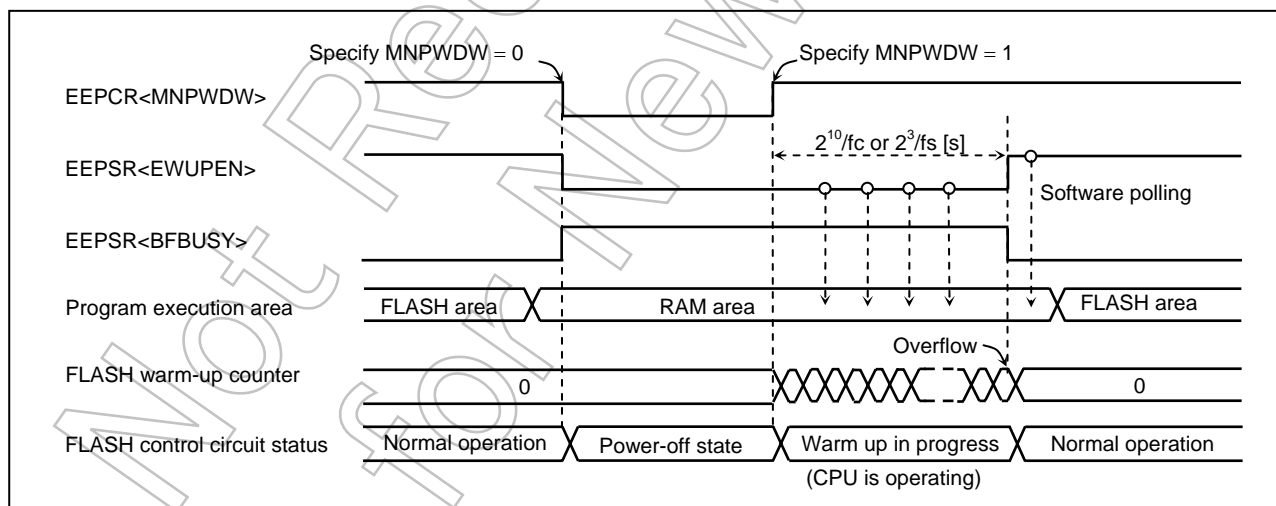
Figure 2.17.6 Software-based Power Control for the FLASH Control Circuit (EEPCR<MNPWDW>)

Example: Performing software-based power control for the FLASH control circuit.

sRAMAREA:

	DI		;	Disable an interrupt (IMF ← "0").
	LD	(WDTCR2), 4Eh	;	Clear the binary counter if the watchdog timer is in use.
	CLR	(EEPCR). 0	;	Clear the EEPCR<MNPWDW> to "0".
	SET	(EEPCR). 0	;	Set the EEPCR<MNPWDW> to "1".
sLOOP1:	TEST	(EEPSR). 1	;	Monitor the EEPSR<EWUPEN> register.
	JRS	T, sLOOP1	;	Jump to sLOOP1 if EEPSR<EWUPEN> = "0".
	JP	MAIN	;	Jump to the FLASH area.

2.17.5.2 Automatic Power Control for the FLASH Control Circuit (EEPCR<ATPWDW>)

The EEPCR<ATPWDW> is an automatic power control bit for the FLASH control circuit. It is possible to suppress power consumption by automatically shutting down the power for the FLASH control circuit when an operation mode is changed to IDLE0/1/2 and SLEEP0/1/2 modes. This bit can be specified regardless of the area in which a program is being executed.

After the EEPCR<ATPWDW> is cleared to "0", entering an operation mode (IDLE0/1/2 or SLEEP0/1/2) where the CPU is at a halt automatically turns off the power for the FLASH control circuit. Once the operation mode is released, the warm-up time (CPU wait) is automatically counted to resume normal processing. The CPU wait period is either $2^{10}/fc$ (SYSCK = "0") or $2^{3}/fs$ (SYSCK = "1"). If the EEPCR<ATPWDW> is "1", releasing the operation mode does not cause the CPU wait.

If EEPCR<MNPWDW> = "1", executing a STOP mode forcibly turns off the power for the FLASH control circuit regardless of the setting of the EEPCR<ATPWDW>. When the STOP mode is released, a STOP mode oscillation warm-up is carried out, and then an FLASH control circuit warm-up (CPU wait) is automatically performed. If the EEPCR<MNPWDW> is "0", entering/exiting a STOP mode allows the power for the FLASH control circuit to be kept turned off.

Note 1: The EEPCR<ATPWDW> functions only if the EEPCR<MNPWDW> is "1". If the EEPCR<MNPWDW> is "0", the power for the FLASH control circuit is kept turned off when an operation mode is executed or released.

Note 2: During an FLASH warm-up (CPU wait), the peripheral circuits continue operating, but the CPU stays at a halt. Even if an interrupt latch is set under this condition, no interrupt process occurs until the CPU wait is completed. If the IMF is "1" when the interrupt latch is set, interrupt process takes place according to the interrupt priority after the CPU has started operating.
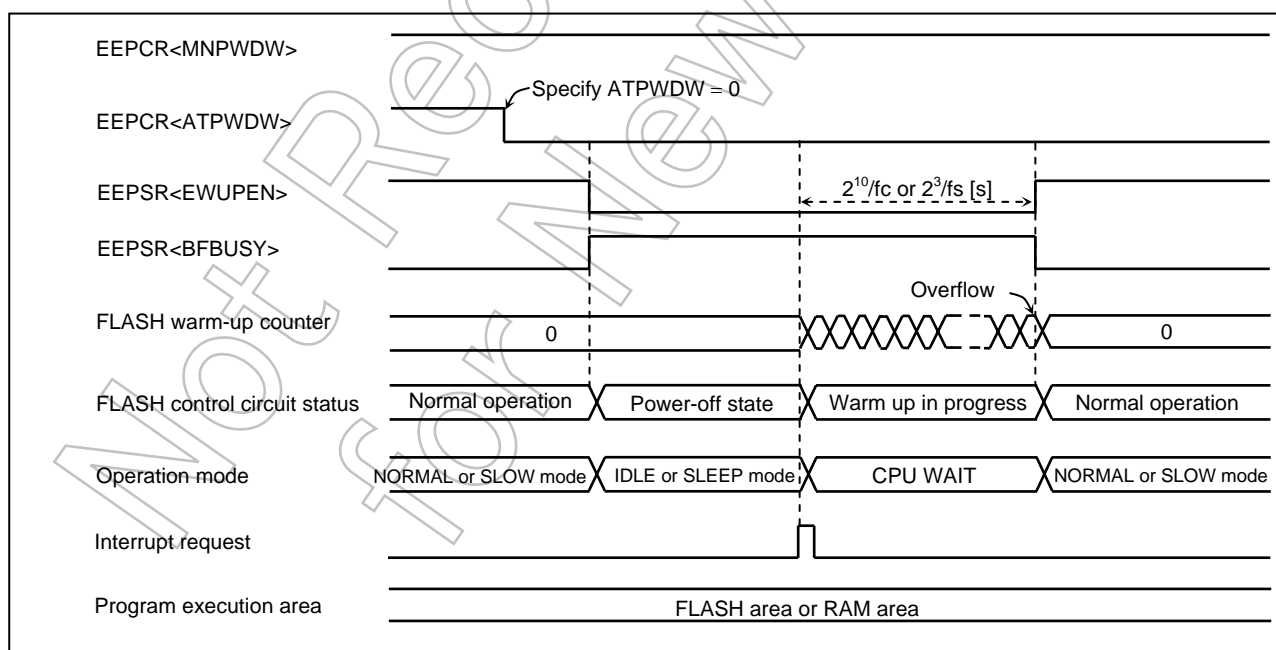

Figure 2.17.7 Automatic Power Control for the FLASH Control Circuit (EEPCR<ATPWDW>)

2.17.6 Accessing to the FLASH Memory

During the writing to the FLASH area, neither a read nor fetch can be performed for the 4000H to FFFFH area. Therefore, to write the FLASH area, the program should be executed in the BOOT ROM or RAM area. Basically, to write the FLASH area, the program can be executed in BOOT ROM area by using the FLASH writing mode of the serial PROM mode, but it can be also executed any user program in RAM area by using the RAM loader mode of the serial PROM mode.

Explanation here is made of only the method of FLASH programing in RAM area. For detail about each operation mode of the serial PROM mode, refer to 2.1 "Serial PROM Mode".

Although the writing to FLASH is executed on page-by-page, the reading from FLASH is executed on byte-by-byte.

If a non-maskable interrupt occurs during a write to the FLASH (EEPSR<BFBUSY> = "1"), the WINT is set to "1" and the writing is discontinued, and then the warm-up (CPU wait) for control circuit of flash memory is executed (the write data counter is also initialized). If WINT = "1" is detected in the non-maskable interrupt service routine, a write is not completed successfully. So, it is necessary to try a write again. The warm-up period is $2^{10}/fc$ (SYSCK = "0") or $2^3/fs$ (SYSCK = "1"). After 1 to 127 bytes are saved to the temporary data buffer, if an interrupt generates, the specified page of FLASH is not written. (It keeps previous data.)

Note 1: Writing to the FLASH area is enabled only in serial PROM mode. For details of serial PROM mode, refer to 2.1 "Serial PROM mode".

Note 2: After 128 bytes are written to the temporary data buffer, the generating of an interrupt may cause the writing the page of FLASH to an unexpected value.

Note 3: During the warm-up period for flash memory (CPU wait), the peripheral circuits continue operating, but the CPU stays at a halt until the warm-up is finished. Even if an interrupt latch is set to "1" by generating of interrupt request, an interrupt sequence doesn't start till the end of warm-up. If interrupts occur during a warm-up period with IMF = "1", the interrupt sequence which depends on interrupt priority will start after warm-up period.

Note 4: When write the data to Flash memory from RAM area, disable all the non-maskable interrupt by clearing interrupt master enable flag (IMF) to "0" beforehand.

2.17.6.1 FLASH Writing Program in the RAM Area

To develop the program in RAM, the write control program should be loaded from external device by using RAM loader mode in serial PROM mode. Given below is an example of writing the control program in the RAM area.

(1) Example of writing program in the RAM area

1. For the emulation chip, set the EEPEVA register with an optimum time value according to the operating frequency.

2. Monitor the EEPSR<EWUPEN>. If it is "0", set the EEPCR<MNPWDW> to "1", and then start and keep polling until the EEPSR<EWUPEN> becomes "1".

3. Clear the interrupt master enable flag (IMF ← "0").

4. Set the EEPCR with "3BH" (to enable a write to the FLASH).

5. Execute a write instruction for 128 bytes to the FLASH area.

6. Start and keep polling by software until the EEPSR<BFBUSY> becomes "0". (Upon completion of an erase and write to the FLASH cells, the EEPSR<BFBUSY> is set to "1". For the FLASH product, the required write time is typically 4 ms. For the emulation chip, it is the value specified in the EEPEVA register.)

7. Set the EEPCR with "CBH" (to disable a write to the FLASH).

Note: See (2), "Method of specifying an address for a write to the FLASH", for a description about the FLASH address to be specified at step 5 above.

(2) Method of specifying an address for a write to the FLASH

The FLASH page to be written is specified by the 10 high-order bits of the address of the 1st byte data. The 1st byte data is stored at the first address of the temporary data buffer. If the data to be written is, for example, 4080H, page 1 is selected, and the data is stored at the first address of the temporary data buffer. Even if the 6 low-order bits of the specified address is not 000000B, the 1st byte data is always stored at the first address of the data buffer.

Any address can be specified as the second and subsequent address within FLASH area (4000H to FFFFH). The write data bytes are stored in the temporary data buffer in the sequence they are written, regardless of what address is specified. Usually, the address that is the same as the 1st byte is specified for the second and subsequent address. A 16-bit transfer instruction (LDW) can also be used for writing to the temporary data buffer.

```
Example:  Data bytes 00 to 7FH are written to page 1.
          (Figure 2.17.9 shows the example of data buffer and pages.)
              DI                              ;  Disable an interrupt (IMF ← "0")
              LD      C, 00H
              LD      HL, EEPCR               ;  Specify the EEPCR register address.
              LD      IX, 4080H               ;  Specify a write address.
              LD      (HL), 3BH               ;  Specify the EEPCR.
sLOOP1:
              LD      (IX), C                 ;  Store data to the temporary data buffer.
                                                 (A write page is selected when the 1st
                                                 byte is written.)
              INC     C                       ;  C = C + 1
              CMP     C, 80H                  ;  Jump to sLOOP1 if C is not 80h.
              JR      NZ, sLOOP1
sLOOP2:
              TEST    (EEPSR). 0
              JRS     F, sLOOP2               ;  Jump to sLOOP2 if EEPSR<BFBUSY> =
                                                 "1".
              LD      (HL), 0CBH              ;  Specify the EEPCR.
```

Note: If the BFBUSY is "1", executing a read instruction or fetch to the FLASH area causes "FFH" to be read. Fetching "FFH" results in a software interrupt occurring.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	00H	01H	02H	03H	04H	05H	06H	07H	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
	10H	11H	12H	13H	14H	15H	16H	17H	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH
	20H	21H	22H	23H	24H	25H	26H	27H	28H	29H	2AH	2BH	2CH	2DH	2EH	2FH
	30H	31H	32H	33H	34H	35H	36H	Temporary data buffer		39H	3AH	3BH	3CH	3DH	3EH	3FH
	40H	41H	42H	43H	44H	45H	46H			49H	4AH	4BH	4CH	4DH	4EF	4FH
	50H	51H	52H	53H	54H	55H	56H	57H	58H	59H	5AH	5BH	5CH	5DH	5EH	5FH
	60H	61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH
	70H	71H	72H	73H	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH

Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4070H																
4080H	00H	01H	02H	03H	04H	05H	06H	07H	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
4090H	10H	11H	12H	13H	14H	15H	16H	17H	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH
40A0H	20H	21H	22H	23H	24H	25H	26H	27H	28H	29H	2AH	2BH	2CH	2DH	2EH	2FH
40B0H	30H	31H	32H	33H	34H	35H	36H	Page 1		39H	3AH	3BH	3CH	3DH	3EH	3FH
40C0H	40H	41H	42H	43H	44H	45H	46H			49H	4AH	4BH	4CH	4DH	4EF	4FH
40D0H	50H	51H	52H	53H	54H	55H	56H	57H	58H	59H	5AH	5BH	5CH	5DH	5EH	5FH
40E0H	60H	61H	62H	63H	64H	65H	66H	67H	68H	69H	6AH	6BH	6CH	6DH	6EH	6FH
40F0H	70H	71H	72H	73H	74H	75H	76H	77H	78H	79H	7AH	7BH	7CH	7DH	7EH	7FH

Figure 2.17.8 Data Buffer and Write Page (Example)
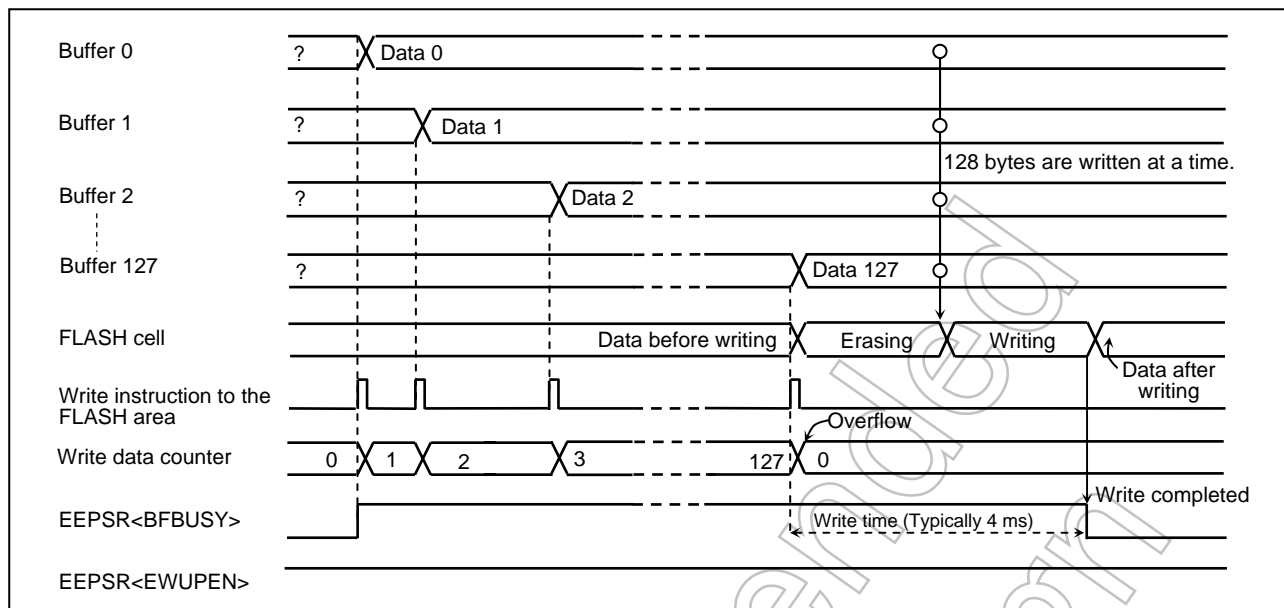
Figure 2.17.9 Write to the FLASH Area (In case of FLASH product)
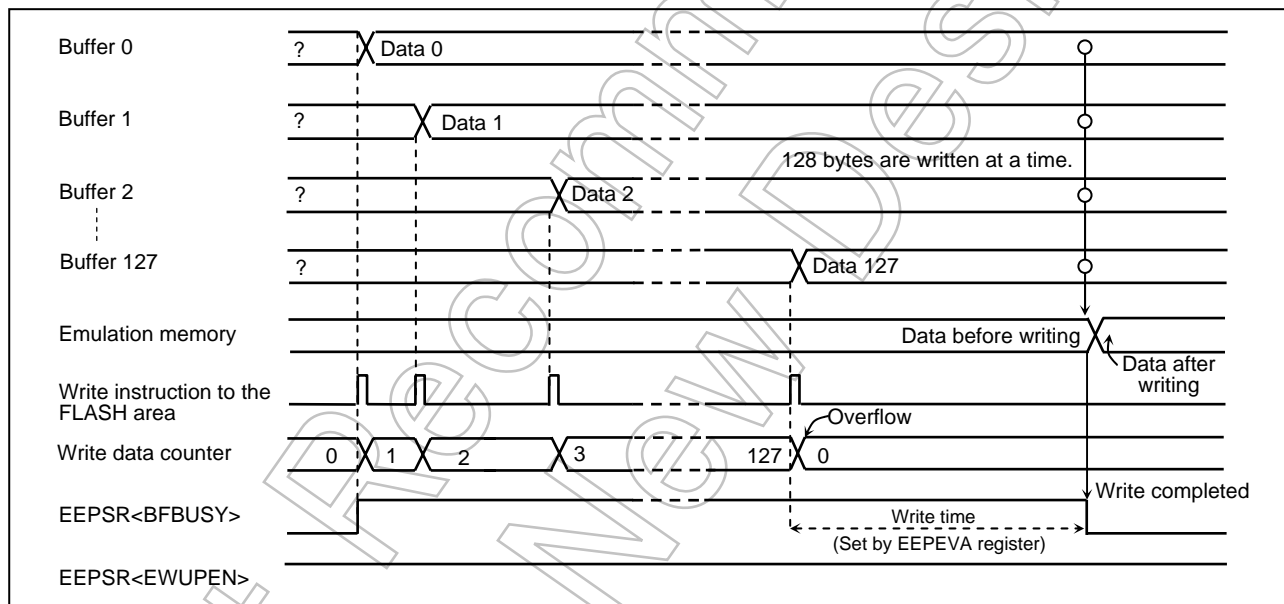
Figure 2.17.10 Write to the FLASH Area (In case of emulation chip)

Note 1: The emulation chip is written to emulation memory instead of FLASH cell.

Note 2: In case of emulation chip, the data stacked to data buffer is written to emulation memory just before EEPSR<BFBUSY> is changed from "1" to "0". Therefore, if the writing of FLASH is stopped forcibly after the write data counter becomes overflow, the memory value on the page subjected to a write may be different from FLASH product.

2.18 Serial PROM Mode

2.18.1 Outline

The TMP86FP24 has a 2 Kbytes BOOT ROM for programming to FLASH memory. This BOOT ROM is a mask ROM that contains a program to write the FLASH memory on board. The BOOT ROM is available in a serial PROM mode and it is controlled by P11 pin, BOOT pin (P23), TEST pin and $\overline{\text{RESET}}$ pin, and is communicated via TXD (P06) and RXD (P05) pins. There are four operation modes in a serial PROM mode: FLASH writing mode, RAM loader mode, FLASH memory SUM output mode and product discrimination code output mode. Operating area of serial PROM mode differs from that of MCU mode. The operating area of serial PROM mode shows in Table 2.18.1.

Table 2.18.1 Operating Area of Serial PROM Mode

Parameter	Min	Max	Unit
Operating voltage	2.7	3.6	V
High frequency [Note]	2	16	MHz
Temperature	25 ± 5		°C

Note: Even though included in above operating area, part of frequency can not be supported in serial PROM mode. For details, refer to Table 2.18.6.

2.18.2 Memory Mapping

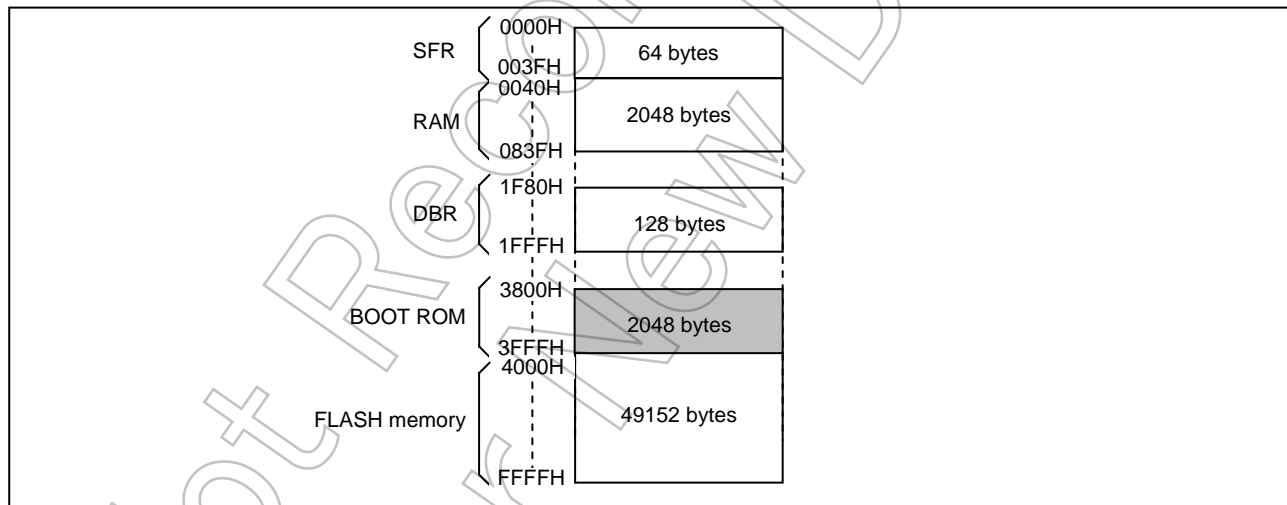The BOOT ROM is mapped in address 3800H to 3FFFH. The Figure 2.18.1 shows a memory mapping.

Figure 2.18.1 Memory Address Maps

2.18.3 Serial PROM Mode Setting

2.18.3.1 Serial PROM Mode Control Pins

To execute on-board programming, start the TMP86FP24 in serial PROM mode. Setting of a serial PROM mode is shown in Table 2.1.2.

Table 2.18.2 Serial PROM Mode Setting

Pin	Setting
BOOT pin (P23)	High
P11 pin	Low
$\overline{\text{RESET}}$, TESTpin	___/‾

2.18.3.2 Pin Function

In the serial PROM mode, TXD (P06) and RXD (P05) pins are used as a serial interface pin.

Table 2.18.3 Pin function in the Serial PROM Mode

Pin Name (Serial PROM mode)	Input/ Output	Function		Pin Name (MCU mode)
TXD	Output	Serial data output		P06
RXD	Input	Serial data input		P05
BOOT	Input	Serial PROM mode control (Fix to "H" level)	(Note 1)	P23
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
TEST	Input	Serial PROM mode control		TEST
P11	Input	Serial PROM mode control (Fix to "L" level)		P11
VDD, AVDD	Power supply	2.7 V to 3.6 V		
VSS		0 V		
VAREF		Open or equal with VDD		
P00 to P04,P07	I/O	Placed in High-Z state during serial PROM mode.		
P10, P12 to P15				
P20 to P22				
P30 to P37				
P40 to P47				
P50 to P53				
P60 to P67				
P90 to P97				
$\overline{\text{WAKE}}$	Output	Placed in High-Z state during serial PROM mode.		
SEG7 to SEG0	Output	Open		
COM3 to COM0				
C0,C1,V3 to V1	LCD voltage booster pin			
XIN	Input	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.	(Note 2)	
XOUT	Output			

Note 1: When the device is used as on-board writing and other parts are already mounted in place, be careful no to affect these communication control pins.

Note 2: Operating area of high frequency in serial PROM mode is from 2 MHz to 16 MHz.

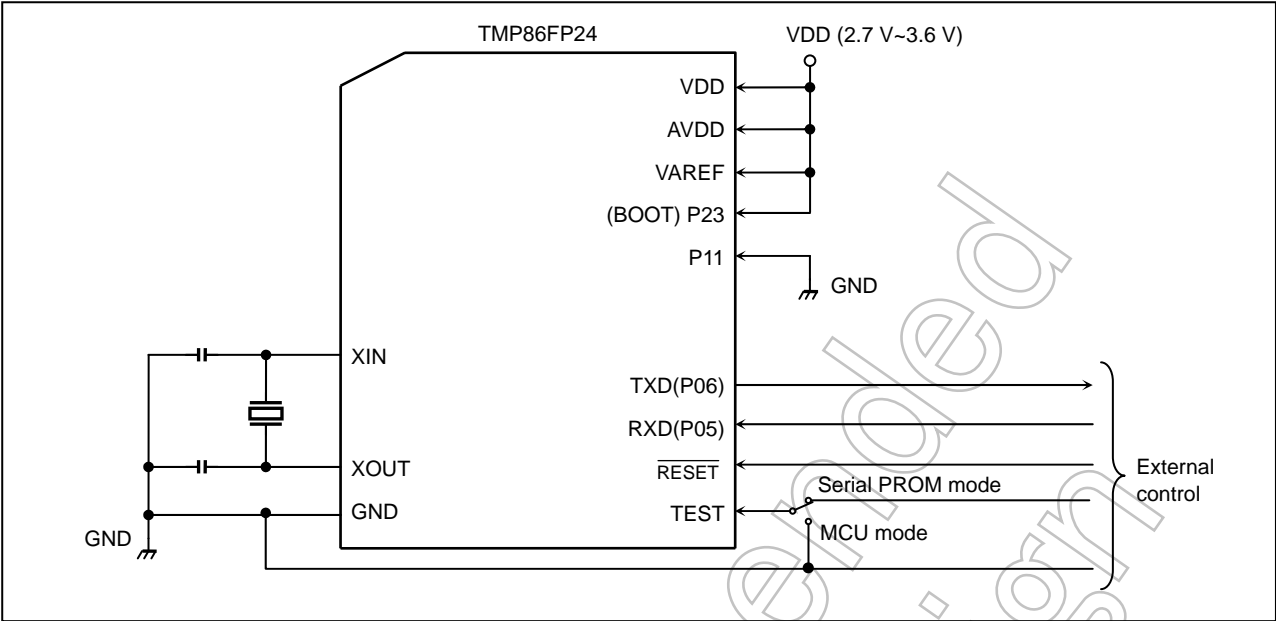To set a serial PROM mode, connect device pins as shown in Figure 2.18.2.

Figure 2.18.2 Serial PROM Mode Port Setting

2.18.3.3 Activating Serial PROM Mode

The following is a procedure of setting of serial PROM mode. Figure 2.18.3 shows a serial PROM mode timing.

(1) Turn on the power to the VDD pin.

(2) Set the P11 pin, TEST pin and $\overline{\text{RESET}}$ pin to low level.

(3) Set the BOOT pin (P23) and RXD pin (P05) to high level.

(4) Wait until the power supply and clock sufficiently stabilize.

(5) Set the TEST pin from low level to high level.

(6) Release the $\overline{\text{RESET}}$ (Set to high level).

(7) Input a matching data (5AH) to RXD pin after waiting for setup sequence.
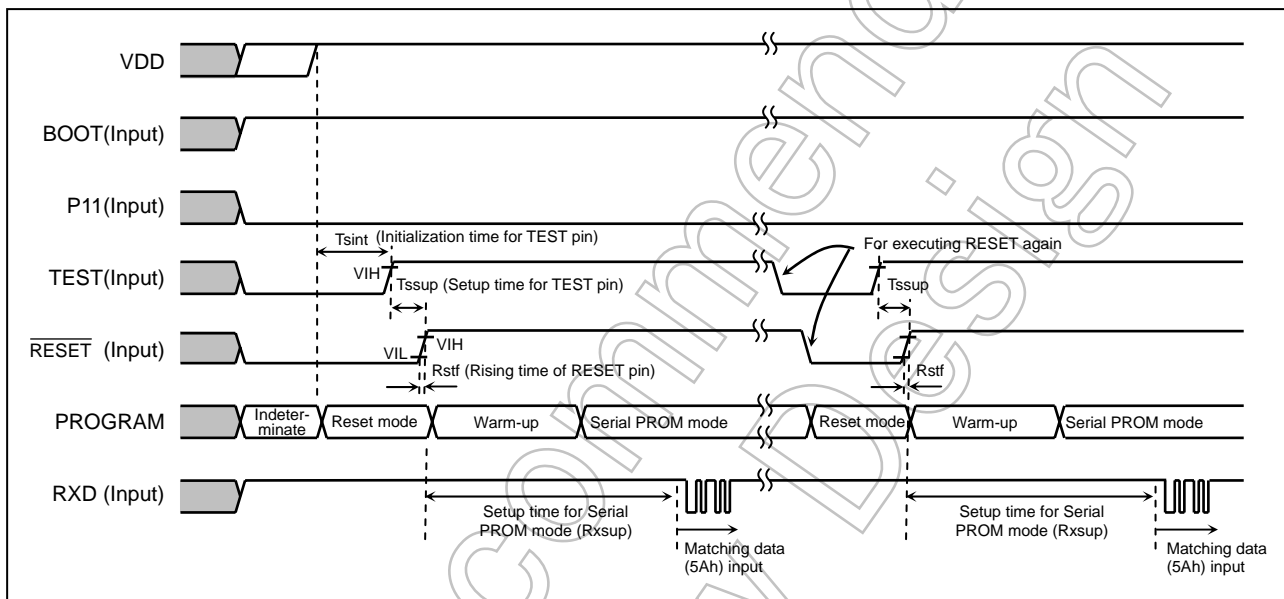
Figure 2.18.3 Serial PROM Mode Timing

Table 2.18.4 Serial PROM Mode Timing characteristics

Parameter		Symbol	The Number of Clock (fc)	Required Minimum Time	
				at fc = 2 MHz	at fc = 16 MHz
Setup time for TEST pin	Rstf > 512 / fc [s]	Tssup	-	1 ms	
	Rstf < 512 / fc [s]		-	0 [*Note1]	
Initialization time for TEST pin		Tsint	-	1ms	
Time from reset release until acceptance of start bit of RXD pin		RXsup	110000	55 ms	6.9 ms

Note 1: If Rstf is shorter than 512 / fc[s] due to using CMOS-type reset IC or Logic IC, the TEST pin can input the same pulse as the $\overline{\text{RESET}}$ pin input. (TEST pin can be directly connected to the $\overline{\text{RESET}}$ pin.) However, drive the pins carefully not to affect the pin's input level, as the TEST pin and the $\overline{\text{RESET}}$ pin have pull-down resistor and pull-up resistor built-in.

Note 2: fc; High-frequency clock

2.18.3.4   Examples of On-board writing

Figure 2.18.4 shows examples of On-board writing.
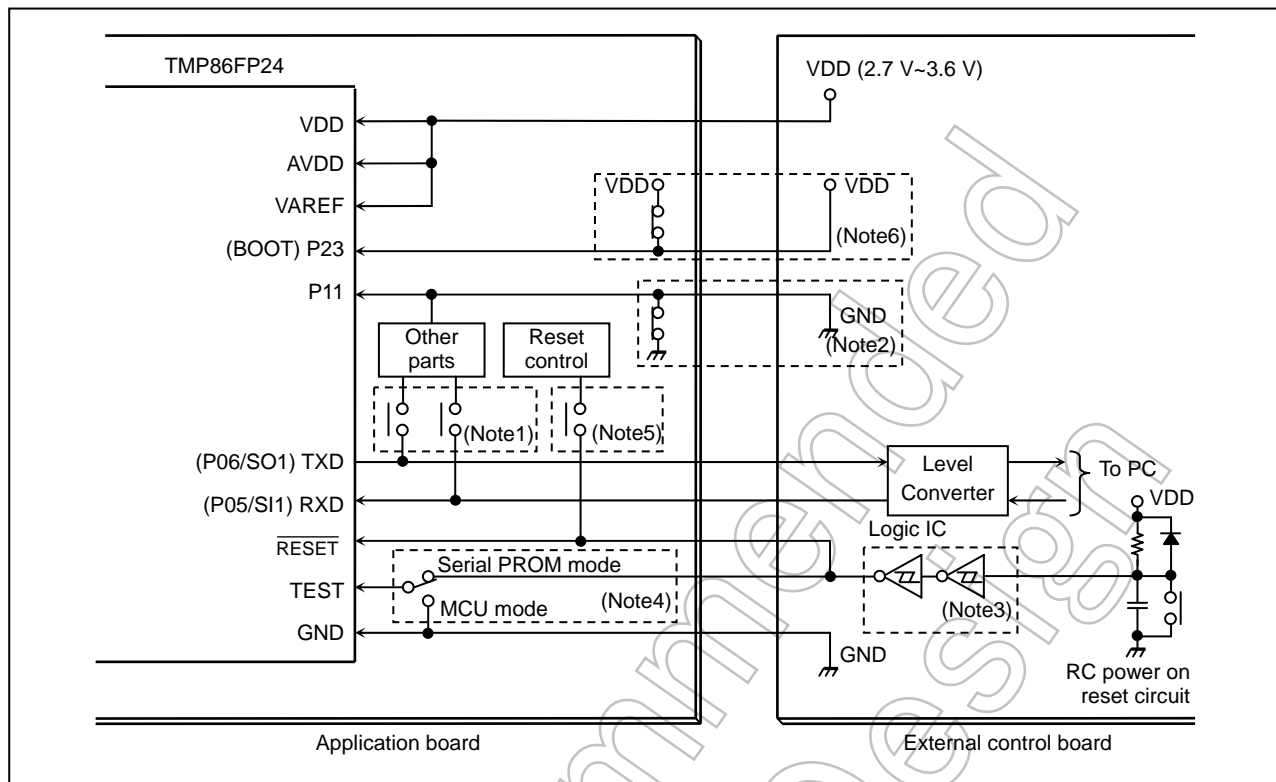
Figure 2.18.4 Examples of Onboard writing

Note 1: If capacity for LCD panel and other devices on the application board affect UART communication in Serial PROM mode, disconnect these pins by using a jumper or a switch.

Note 2: Set the P11 pin to GND. There are two ways. Set P11 pin to GND on the external board, or set it to GND by setting a jumper on the application board.

Note 3: If input signal has analog delay due to the use of such circuit as RC power on reset circuit, connect both TEST pin and $\overline{\text{RESET}}$ pin to logic ICs (Schmitt input IC such as TC74HC14). In this case, control the pin capacity to require the condition Rstf<512/fc[s].

Note4: In MCU mode, the TEST pin can be disconnected as it has a pull-down resistor built-in. However, we recommend connecting it to GND level to avoid noise influence.

Note5: If the RESET control circuit on the application board affects the Serial PROM mode to start, disconnect it by using a jumper, etc.

Note6: Set the P23 pin to VDD. There are two ways. Set P23 pin to VDD on the external board, or set it to VDD by setting a jumper on the application board.

2.18.4 Interface Specifications for UART

The following shows the UART communication format used in serial PROM mode.

Before on-board programming can be executed, the communication format on the external controller side must also be setup in the same way as for this product.

Note that although the default baud rate is 9600 bps, it can be changed to other values as shown in Table 2.18.5. The Table 2.18.6 shows an operating frequency and baud rate in serial PROM mode. Except frequency which is not described in Table 2.18.6 can not use in serial PROM mode.

Baud rate (Default): 9600 bps
Data length: 8 bits
Parity addition: None
Stop bit length: 1 bit

Table 2.18.5 Baud Rate Modification Data

Baud Rate Modification Data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

Table 2.18.6 Operating Frequency and Baud Rate in Serial PROM Mode

(Note 3)	Reference Baud Rate (bps)		76800		62500		57600		38400		31250		19200		9600	
	Baud Rate Modification Data		04H		05H		06H		07H		0AH		18H		28H	
	Refer Frequency (MHz)	Area (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
1	2	1.91~2.10	–	–	–	–	–	–	–	–	–	–	–	–	9615	+0.16
2	4	3.82~4.19	–	–	–	–	–	–	–	–	31250	0.00	19231	+0.16	9615	+0.16
	4.19	3.82~4.19	–	–	–	–	–	–	–	–	32734	+4.75	20144	+4.92	10072	+4.92
3	4.9152	4.70~5.16	–	–	–	–	–	–	38400	0.00	–	–	19200	0.00	9600	0.00
	5	4.70~5.16	–	–	–	–	–	–	39063	+1.73	–	–	19531	+1.73	9766	+1.73
4	6	5.87~6.45	–	–	–	–	–	–	–	–	–	–	–	–	9375	−2.34
	6.144	5.87~6.45	–	–	–	–	–	–	–	–	–	–	–	–	9600	0.00
5	7.3728	7.05~7.74	–	–	–	–	57600	0.00	–	–	–	–	19200	0.00	9600	0.00
6	8	7.64~8.39	–	–	62500	0.00	–	–	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16
7	9.8304	9.40~10.32	76800	0.00	–	–	–	–	38400	0.00	–	–	19200	0.00	9600	0.00
	10	9.40~10.32	78125	+1.73	–	–	–	–	39063	+1.73	–	–	19531	+1.73	9766	+1.73
8	12	11.75~12.90	–	–	–	–	57692	+0.16	–	–	31250	0.00	18750	−2.34	9375	−2.34
	12.288	11.75~12.90	–	–	–	–	59077	+2.56	–	–	32000	+2.40	19200	0.00	9600	0.00
	12.5	11.75~12.90	–	–	60096	−3.85	60096	+4.33	–	–	30048	−3.85	19531	+1.73	9766	+1.73
9	14.7456	14.10~15.48	–	–	–	–	57600	0.00	38400	0.00	–	–	19200	0.00	9600	0.00
10	16	15.27~16.77	76923	+0.16	62500	0.00	–	–	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16

Note 1: "Refer Frequency" and "Area" show the high-frequency area supported in serial PROM mode. Except the above frequency can not be supported in serial PROM mode even though the high frequency is included in area from 2 MHz to 16 MHz.

Note 2: The total error of frequency must be kept within +/−3% so that the auto detection of frequency is executed correctly.

Note 3: An external controller should transmit a matching data repeatedly till the TMP86FP24 transmit an echo back data. Above number indicates a transmission number of times of matching data till transmission of echo back data.

2.18.5 Command

There are five commands in serial PROM mode. After reset release, the TMP86FP24 waits a matching data (5AH).

Table 2.18.7 Command in Serial PROM Mode

Command Data	Operation Mode	Remarks
5AH	Setup	Matching data. Always start with this command after reset release.
30H	FLASH memory writing	Writing to area from 4000H to FFFFH is enable.
60H	RAM loader	Writing to area from 0050H to 082FH is enable.
90H	FLASH memory SUM output	The checksum of entire FLASH area (from 4000H to FFFFH) is output in order of the upper byte and the lower byte.
C0H	Product discrimination code output	Product discrimination code, that is expressed by 13 bytes data, is output.

2.18.6 Operation Mode

There are four operating modes in serial PROM mode: FLASH memory writing mode, RAM loader mode, FLASH memory SUM output mode and Product discrimination code output mode. For details about these modes, refer to (1) FLASH memory writing mode through (4) Product discrimination code output mode.

(1) FLASH memory writing mode

The data are written to the specified FLASH memory addresses. The controller should send the write data in the Intel Hex format (binary). For details of writing data format, refer to "2.18.7 FLASH Memory Writing Data Format".

If no errors are encountered till the end record, the SUM of 48 Kbytes of FLASH memory is calculated and the result is returned to the controller.

To execute the FLASH writing mode, the TMP86FP24 checks the passwords except a blank product. If the passwords did not match, the program is not executed.

(2) RAM loader mode

The RAM loader transfers the data into the internal RAM that has been sent from the controller in Intel Hex format. When the transfer has terminated normally, the RAM loader calculates the SUM and sends the result to the controller before it starts executing the user program. After sending of SUM, the program jumps to the start address of RAM in which the first transferred data has been written. This RAM loader function provides the user's own way to control on-board programming.

To execute the RAM loader mode, the TMP86FP24 checks the passwords except a blank product. If the passwords did not match, the program is not executed.

(3) FLASH memory SUM output mode

The SUM of 48 Kbytes of FLASH memory is calculated and the result is returned to the controller.

The BOOT ROM does not support the reading function of the FLASH memory. Instead, it has this SUM command to use. By reading the SUM, it is possible to manage Revisions of application programs.

(4) Product discrimination code output mode

The product discrimination code is output as a 13-byte data, that includes the start address and the end address of ROM (In case of TMP86FP24, the start address is 4000H and the end address is FFFFH). Therefore, the controller can recognize the device information by using this function.

2.18.6.1   FLASH Memory Writing Mode (Operation Command: 30H)

Table 2.18.8 shows FLASH Memory writing mode process.

Table 2.18.8 FLASH Memory Writing Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FP24	Baud Rate	Transfer Data from TMP86FP24 to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) −	9600 bps 9600 bps	− (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 2.18.5) −	9600 bps 9600 bps	− OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) −	Changed new baud rate Changed new baud rate	− OK: Echo back data (30H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	− OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	− OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	− OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	− OK: Nothing transmitted Error: Nothing transmitted
	15th byte ⋮ m'th byte	Password string (Note 5) −	Changed new baud rate Changed new baud rate	− OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte ⋮ n'th − 2 byte	Intel Hex format (Binary) (Note 2)	Changed new baud rate	−
	n'th − 1 byte	−	Changed new baud rate	OK: SUM (High) (Note 3) Error: Nothing transmitted
	n'th byte	−	Changed new baud rate	OK: SUM (Low) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation) (Command data)	Changed new baud rate	−

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to 2.18.8 "Error Code".

Note 2: Refer to 2.18.10 "Intel Hex Format (Binary)".

Note 3: Refer to 2.18.9 "Checksum (SUM)".

Note 4: Refer to 2.18.11 "Passwords".

Note 5: If all data of FFE0H to FFFFH area are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. However, it is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. If a password error occurs, the UART function of TMP86FP24 stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FP24 should be reset by $\overline{\text{RESET}}$ pin input.

Description of FLASH memory writing mode

1. The receive data in the 1st byte is the matching data. When the boot program starts in serial PROM mode, TMP86FP24 (Mentioned as "device" hereafter) waits for the matching data (5AH) to receive. Upon receiving the matching data, it automatically adjusts the UART's initial baud rate to 9.600 bps.

2. When the device has received the matching data, the device transmits the data "5AH" as an echo back to the controller. If the device can not receive the matching data, the device does not transmit the echo back data and waits for the matching data again with changing baud rate. Therefore, the controller should send the matching data continuously until the device transmits the echo back data. An external controller should transmit a matching data repeatedly till the device transmit an echo back data. The transmission number of times of matching data varies by the frequency of device. For details, refer to Table 2.18.6.

3. The receive data in the 3rd byte is the baud rate modification data. The seven kinds of baud rate modification data shown in Table 2.18.5 are available. Even if baud rate changing is no need, be sure to send the initial baud rate data (28H: 9,600 bps).

4. When the 3rd byte data is one of the baud rate modification data corresponding to the device's operating frequency, the device sends the echo back data which is the same as received baud rate modification data. Then the baud rate is changed. If the 3rd byte data does not correspond to the baud rate modification data, the device stops UART function after sending 3 bytes of baud rate modification error code: (62H). The changing of baud rate is executed after transmitting the echo back data.

5. The receive data in the 5th byte is the command data (30H) to write the FLASH memory.

6. When the 5th byte is one of the operation command data shown in Table 2.18.7, the device sends the echo back data which is the same as received operation command data (in this case, 30H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

7. The 7th byte is used as an upper bit (Bit15 to Bit8) of the password count storage address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occur, the device does not send any data and stops UART function.

8. The 9th byte is used as a lower bit (Bit7 to Bit0) of the password count storage address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occur, the device does not send any data and stops UART function.

9. The 11th byte is used as an upper bit (Bit15 to Bit8) of the password comparison start address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occur, the device does not send any data and stops UART function.

10. The 13th byte is used as a lower bit (Bit7 to Bit0) of the password comparison start address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occur, the device does not send any data and stops UART function.

11. The 15th through the m'th bytes are the password data. The number of passwords is the data (N) indicated by the password count storage address. The password data are compared for N entries beginning with the password comparison start address. The controller should send N bytes of password data to the device. If the passwords do not match, the device stops UART function without returning error code to the controller. If the data of addresses from FFE0H to FFFFH are all "FFH", the comparison of passwords is not executed because the device is considered as a blank product.

12. The receive data in the m'th + 1 through n'th – 2 byte are received as binary data in Intel Hex format. No received data are echoed back to the controller. The data which is not the start mark (3AH for ":") in Intel Hex format is ignored and does not send an error code to the controller until the device receives the start mark. After receiving the start mark, the device receives the data record, that consists of length of data, address, record type, writing data and checksum. After receiving the checksum of data record, the device waits the start mark data (3AH) again. The data of data record is temporarily stored to RAM and then, is written to specified FLASH memory by page (128 bytes) writing. For details of an organization of FLASH, refer to 2.18.7 "FLASH Memory Writing Data Format". Since after receiving an end record, the device starts to calculate the SUM, the controller should wait the SUM after sending the end record. If receive error or Intel Hex format error occurs, the device stops UART function without returning error code to the controller.

13. The n'th – 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to 2.18.9 "Checksum (SUM)". The SUM calculation is performed after detecting the end record, but the calculation is not executed when receive error or Intel Hex format error has occurred. The time required to calculate the SUM of the 48 Kbytes of FLASH memory area is approximately 150 ms at fc = 16 MHz. After the SUM calculation, the device sends the SUM data to the controller. After sending the end record, the controller can judge that the transmission has been terminated correctly by receiving the checksum.

14. After sending the SUM, the device waits for the next operation command data.

2.18.6.2 RAM Loader Mode (Operation command: 60H)

Table 2.18.9 shows RAM loader mode process.

Table 2.18.9 RAM Loader Mode Process

	Number of Bytes Transferred	Transfer Data from External CONTROLLER to TMP86FP24	Baud Rate	Transfer Data from TMP86FP24 to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9600 bps 9600 bps	– (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 2.18.5) –	9600 bps 9600 bps	– OK: Echo back data Error:A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (60H) –	Changed new baud rate Changed new baud rate	– OK: Echo back data (60H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted Error: Nothing transmitted
	15th byte ⋮ m'th byte	Password string (Note 5) –	Changed new baud rate Changed new baud rate	– OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte ⋮ n'th − 2 byte	Intel Hex format (Binary) (Note 2)	Changed new baud rate	–
	n'th − 1 byte	–	Changed new baud rate	OK: SUM (High) (Note 3) Error: Nothing transmitted
	n'th byte	–	Changed new baud rate	OK: SUM (Low) (Note 3) Error: Nothing transmitted
RAM	–	The program jumps to the start address of RAM in which the first transferred data has been written.		

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to 2.18.8 "Error Code".

Note 2: Refer to 2.18.10 "Intel Hex Format (Binary)".

Note 3: Refer to 2.18.9 "Checksum (SUM)".

Note 4: Refer to 2.18.11 "Passwords".

Note 5: If all data of addresses from FFE0H to FFFFH area are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. However, it is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. If a password error occurs, the UART function of TMP86FP24 stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FP24 should be reset by $\overline{\text{RESET}}$ pin input.

Note 6: Do not send only end record after transferring of password string. If the TMP86FP24 receives the end record only after reception of password string, it does not operate correctly.

Note 7: When the FLASH power supply is turned off in user's program by setting EEPCR<MNPWDW>, be sure to disable the watchdog timer (WDT) or to clear the binary counter of WDT immediately before.

Description of RAM loader mode

1. The process of the 1st byte through the 4th byte are the same as FLASH writing mode.

2. The receive data in the 5th byte is the RAM loader command data (60H) to write the user's program to RAM.

3. When the 5th byte is one of the operation command data shown in Table 2.18.7, the device sends the echo back data which is the same as received operation command data (in this case, 60H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The process of the 7th byte through the m'th byte are the same as FLASH writing mode.

5. The receive data in the m'th + 1 through n'th − 2 byte are received as binary data in Intel Hex format. No received data are echoed back to the controller.
   The data which is not the start mark (3AH for ":") in Intel Hex format is ignored and does not send an error code to the controller until the device receives the start mark. After receiving the start mark, the device receives the data record, that consists of length of data, address, record type, writing data and checksum. After receiving the checksum of data record, the device waits the start mark data (3AH) again. The data of data record is written to specified RAM by the receiving data. Since after receiving an end record, the device starts to calculate the SUM, the controller should wait the SUM after sending the end record. If receive error or Intel Hex format error occurs, the UART function of TMP86FP24 stops without returning error code to the controller.

6. The n'th − 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to 2.18.9 "Checksum (SUM)". The SUM calculation is performed after detecting the end record, but the calculation is not executed when receive error or Intel Hex format error has occurred.
   The SUM is calculated by the data written to RAM, but the length of data, address, record type and checksum in Intel Hex format are not included in SUM.

7. The boot program jumps to the first address that is received as data in Intel Hex format after sending the SUM to the controller.

2.18.6.3   FLASH Memory SUM Output Mode (Operation command: 90H)

Table 2.18.10 shows FLASH memory SUM output mode process.

Table 2.18.10 FLASH Memory SUM Output Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FP24	Baud Rate	Transfer Data from TMP86FP24 to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) –	9600 bps 9600 bps	– (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 2.18.5) –	9600 bps 9600 bps	– OK: Echo back data Error:A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (90H) –	Changed new baud rate Changed new baud rate	– OK: Echo back data (90H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	–	Changed new baud rate	OK: SUM (High) (Note 2) Error: Nothing transmitted
	8th byte	–	Changed new baud rate	OK: SUM (Low) (Note 2) Error: Nothing transmitted
	9th byte	(Wait for the next operation) (Command data)	Changed new baud rate	–

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to 2.18.8 "Error Code".

Note 2: Refer to 2.18.9 "Checksum (SUM)".

Description of FLASH memory SUM output mode

1. The process of the 1st byte through the 4th byte are the same as FLASH writing mode.

2. The receive data in the 5th byte is the FLASH memory SUM command data (90H) to calculate the entire FLASH memory.

3. When the 5th byte is one of the operation command data shown in Table 2.18.7, the device sends the echo back data which is the same as received operation command data (in this case, 90H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The 7th and the 8th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to 2.18.9 "Checksum (SUM)".

5. After sending the SUM, the device waits for the next operation command data.

2.18.6.4   Product Discrimination Code Output Mode (Operation command: C0H)

Table 2.18.11 shows product discrimination code output mode process.

Table 2.18.11 Product Discrimination Code Output Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FP24	Baud Rate	Transfer Data from TMP86FP24 to External Controller	
BOOT ROM	1st byte	Matching data (5AH)	9600 bps	– (Baud rate auto set)	
	2nd byte	–	9600 bps	OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte	Baud rate modification data (See Table 2.18.5)	9600 bps	–	
	4th byte	–	9600 bps	OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte	Operation command data (C0H)	Changed new baud rate	–	
	6th byte	–	Changed new baud rate	OK: Echo back data (C0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Changed new baud rate	3AH	Start mark
	8th byte		Changed new baud rate	0AH	The number of transfer data (from 9th to 18th byte)
	9th byte		Changed new baud rate	02H	Length of address (2 bytes)
	10th byte		Changed new baud rate	00H	Reserved data
	11th byte		Changed new baud rate	00H	Reserved data
	12th byte		Changed new baud rate	00H	Reserved data
	13th byte		Changed new baud rate	00H	Reserved data
	14th byte		Changed new baud rate	01H	The number of ROM block (1 block)
	15th byte		Changed new baud rate	80H	1st address of ROM
	16th byte		Changed new baud rate	00H	
	17th byte		Changed new baud rate	FFH	End address of ROM
	18th byte		Changed new baud rate	FFH	
	19th byte		Changed new baud rate	BFH	Checksum of transferred data (from 9th to 18th byte)
	20th byte	(Wait for the next operation) (command data)	Changed new baud rate	–	

Note: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to 2.18.8 "Error Code".

Description of product discrimination code output mode

1.   The process of the 1st byte through the 4th byte are the same as FLASH writing mode.

2.   The receive data in the 5th byte is the product discrimination code output command data (C0H).

3.   When the 5th byte is one of the operation command data shown in Table 2.18.7, the device sends the echo back data which is the same as received operation command data (in this case, C0H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4.   The 7th and the 19th bytes are the product discrimination code. For details, refer to 2.18.12 "Product Discrimination Code".

5.   After sending the SUM, the device waits for the next operation command data.

2.18.7 FLASH Memory Writing Data Format

FLASH area of TMP86FP24 consists of 384 pages and one page size is 128 bytes.

Writing to FLASH is executed by page writing. Therefore, it is necessary to send 128 bytes data (for one page) even though only a few bytes data are written. Figure 2.18.5 shows an organization of FLASH area. When the controller sends the writing data to the device, be sure to keep the format described below.

1. The address of data after receiving the FLASH memory writing command should be the first address of page. For example, in case of page 1, the first address should be 4080H.

2. If the last data's address of data record is not end address of page, the address of the next data record should be the address + 1. For example, if the last data's address is 406FH (page 0), the address of the next data record should be 4070H (page 0).
    Example:
      :104060006061626364656667686A6B6C6D6E6FD8   ; 4060H to 406FH data
      :10407000707172737475767778797A7B7C7D7E7FC8   ; 4070H to 407FH data

3. The last data's address of data record immediately before sending the end record should be the last address of page. For example, in case of page 0, the last data's address of data record should be 407FH.
    Example:
      :10407000707172737475767778797A7B7C7D7E7FC8   ; 4070H to 407FH data
      :00000001FF   ; End record

Note: Do not write only the addresses from FFE0H to FFFFH when all data of FLASH memory are the same data. If the these area are only written, the next operation can not be executed because of password error.
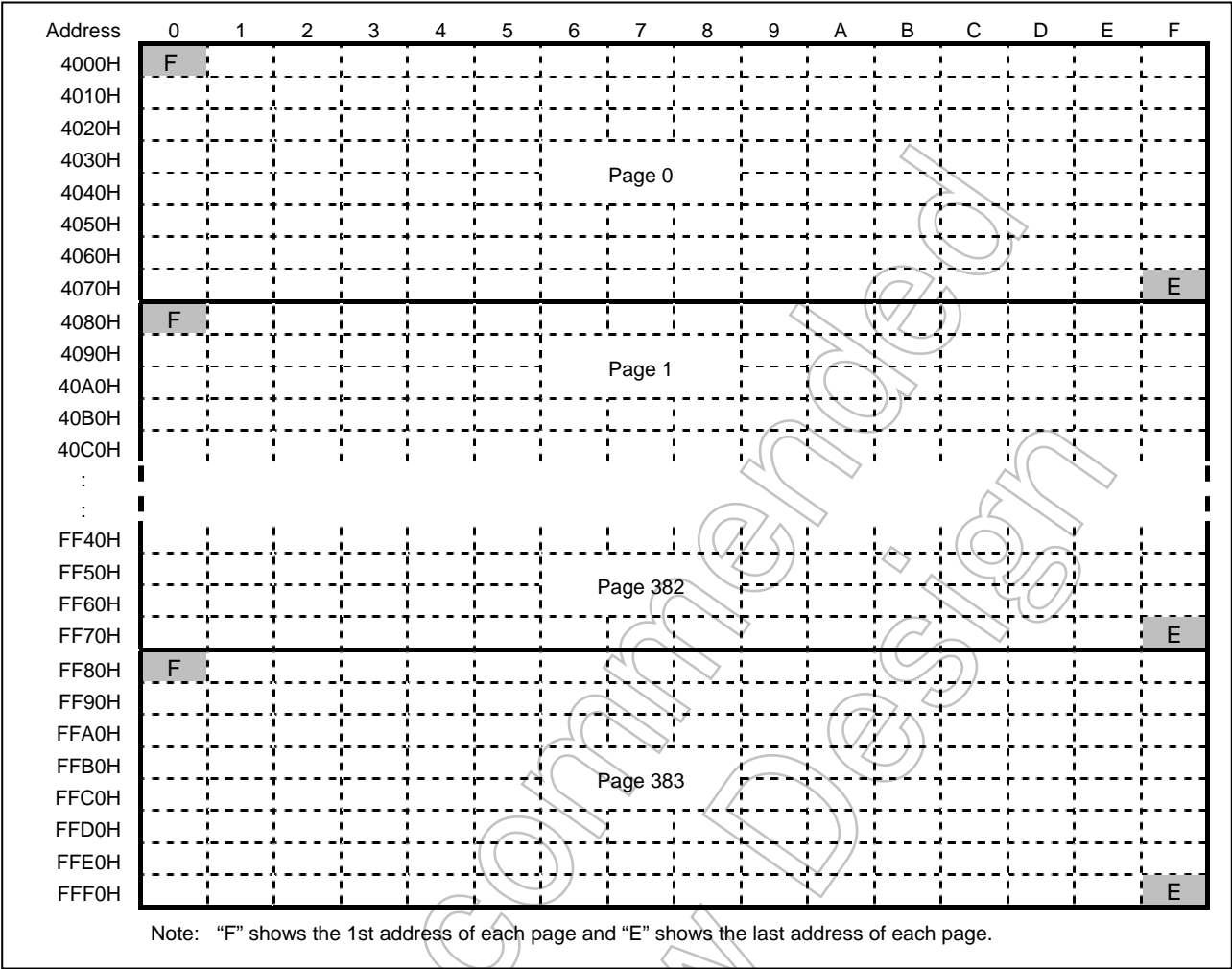
Figure 2.18.5 Organization of FLASH Area

2.18.8 Error Code

When the device detects an error, the error codes are sent to the controller.

Table 2.18.12 Error Code

Transmit Data	Meaning of Transmit Data
62H, 62H, 62H	Baud rate modification error occurred.
63H, 63H, 63H	Operating command error occurred.
A1H, A1H, A1H	Framing error in received data occurred.
A3H, A3H, A3H	Overrun error in received data occurred.

Note: If password error occurs, the TMP86FP24 doesn't send error codes.

2.18.9 Checksum (SUM)

(1) Calculation method

SUM consists of byte + byte... + byte, the checksum of which is returned in word as the result.

Namely, data is read out in byte and checksum of which is calculated, with the result returned in word.

Example:

A1H	If the data to be calculated consists of the four bytes shown to the left, SUM of the data is
B2H	A1H + B2H + C3H + D4H = 02EAH
C3H	SUM (HIGH) = 02H
	SUM (LOW) = EAH
D4H	

The SUM returned when executing the FLASH memory write command, RAM loader command, or FLASH memory SUM command is calculated in the manner shown above.

(2) Calculation data

The data from which SUM is calculated are listed in Table 2.18.13 below.

Table 2.18.13 Checksum Calculation Data

Operating Mode	Calculation Data	Remarks
FLASH memory writing mode	Data in the entire area (48 Kbytes) of FLASH memory	Even when written to part of the FLASH area, data in the entire memory area (48 Kbytes) is calculated.
FLASH checksum output mode		The length of data, address, record type and checksum in Intel Hex format are not included in SUM.
RAM loader mode	Data written to RAM	The length of data, address, record type and checksum in Intel Hex format are not included in SUM.
Product discrimination code output mode	Checksum of transferred data (from 9th to 18th byte)	For details, refer to Table 2.18.15 Product Discrimination Code Format.

2.18.10   Intel Hex Format (Binary)

1. After receiving the checksum of a record, the device waits for the start mark data (3AH for ":") of the next record. Therefore, the device ignores the data, which does not match the start mark data after receiving the checksum of a record.

2. Make sure that once the controller program has finished sending the checksum of the end record, it does not send anything and waits for two bytes of data to be received (Upper and lower bytes of checksum). This is because after receiving the checksum of the end record, the boot program calculates the checksum and returns the calculated checksum in two bytes to the controller.

3. If a receive error or Intel Hex format error occurs, the UART function of TMP86FP24 stops without returning error code to the controller. In the following cases, an Intel Hex format error occurs:

    •   When the record type is not 00H, 01H, or 02H

    •   When a SUM error occurred

    •   When the data length of an extended record (type = 02H) is not 02H

    •   When the address of an extended record (type = 02H) is larger than 1000H and after that, receives the data record

    •   When the data length of the end record (type = 01H) is not 00H

2.18.11   Passwords

The eight or more bytes consecutive data in flash memory area can be used as password. In password check, TMP86FP24 compares these data with data which are transmitted from the external controller. The area in which passwords can be specified is located at addresses from 4000H to FF9FH. The area from FFA0H to FFFFH can not be specified as passwords area. The device compares the stored passwords with the passwords, which are received from the controller. If all data of addresses from FFE0H to FFFFH are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. It is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. Table 2.1.14 shows the password setting in the blank product and non blank product.

Table 2.18.14 Password Setting in the Blank Product and Non Blank Product

Password	Blank Product (Note 1)	Non Blank Product
PNSA (Password count storage addresses)	4000H ≤ PNSA ≤ FF9FH	4000H ≤ PNSA ≤ FF9FH
PCSA (Password comparison start address)	4000H ≤ PCSA ≤ FF9FH	4000H ≤ PCSA ≤ FFA0 − N
N (Password count)	∗	8 ≤ N
Setting of password	No need	Need (Note 2)

Note 1: When all data of addresses from FFE0H to FFFFH area are "00H" or "FFH", the device is judged as blank product.

Note 2: The same three or more bytes consecutive data can not be used as password. When the password includes the same consecutive data (Three or more bytes), the password error occurs. If the password error occurred, the UART function of device stops without returning error code.

Note 3: ∗: Don't care

Note 4: When the password doesn't match the above condition, the password error occurs. If the password error occurred, the UART function of device stops without returning error code.

Note 5: In case of the blank product, the device receives Intel Hex Format immediately after receiving PCSA without receiving password strings. In this time, because the device ignores the data except the start mark data (3AH for ":") as Intel Hex Format data, even if external controller transmitted dummy password strings, process operates correctly. However, if the dummy password strings contain data "3AH", the device detects it as start mark data mistakenly, and device stops process without returning error code. Therefore, if these process becomes issue, the external controller should not transmit the dummy password strings.
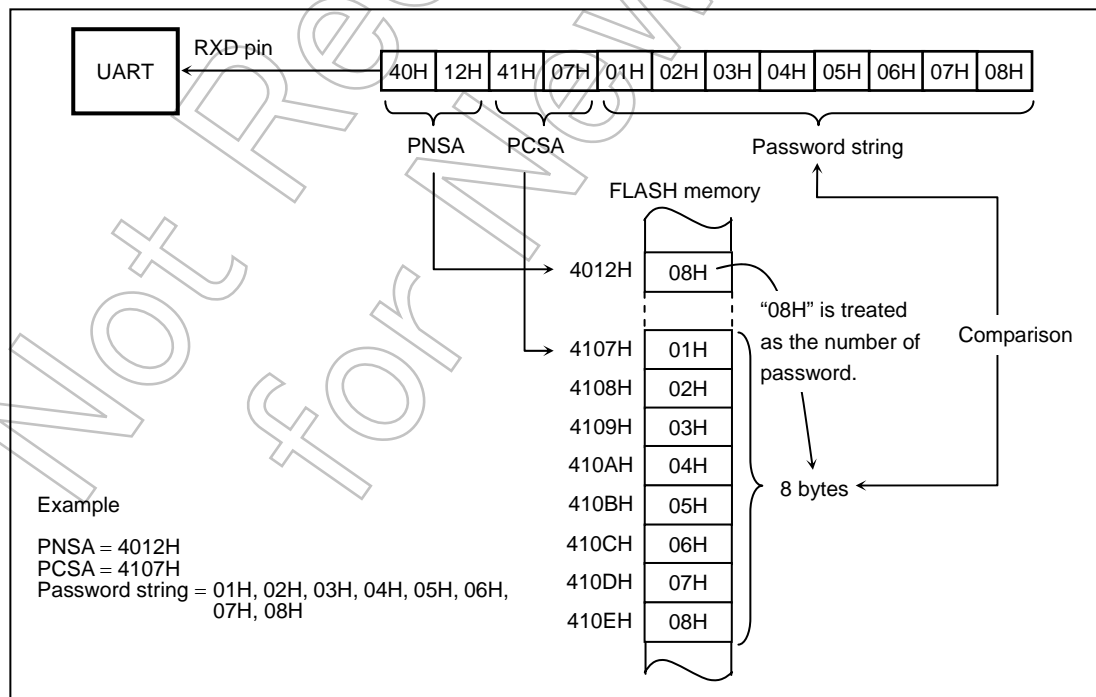

Figure 2.18.6 Example of password compare

2.18.11.1   Confirmation Method of the Blank Product and Non Blank Product

The external controller can confirm whether the device is the blank product or not, by transmission of data described below.

(1)   Executes FLASH memory writing mode or RAM loader mode.

(2)   Transmits the PNSA and PCSA.

(3)   Transmits the end record.

(4)   In case of the blank product, the device sends checksum of flash memory. In case of the non blank product, the device doesn't send checksum of flash memory but the UART function stops without sending any data.

The external controller can confirm the blank product and non blank product by receiving checksum.

Note: When the UART function stops in non blank product, the TMP86FP24 should be reset by pin reset input for restarting the serial prom mode.

2.18.11.2   Password String

A string of passwords in the received data are compared with the data in the FLASH memory. In the following cases, a password error occurs:

•   When the received data does not match the data in the FLASH memory

2.18.11.3   Handling of Password Error

If a password error occurs, the UART function of TMP86FP24 stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FP24 should be reset by $\overline{\text{RESET}}$ pin input.

2.18.12   Product Discrimination Code

The product discrimination code is a 13-byte data, that includes the start address and the end address of ROM. Table 2.18.15 shows the product discrimination code format.

Table 2.18.15 Product Discrimination Code Format

Data	The Meaning of Data	In Case of TMP86FP24
1st	Start mark (3AH)	3AH
2nd	The number of transfer data (from 3rd to 12th byte)	0AH
3rd	Length of address	02H
4th	Reserved data	00H
5th	Reserved data	00H
6th	Reserved data	00H
7th	Reserved data	00H
8th	The number of ROM block	01H
9th	The upper byte of the first address of ROM	40H
10th	The lower byte of the first address of ROM	00H
11th	The upper byte of the end address of ROM	FFH
12th	The lower byte of the end address of ROM	FFH
13th	Checksum of transferred data (from 3rd to 12th byte)	BFH

2.18.13   Flowchart

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           │
                    ┌──────┴──────┐
                    │    Setup    │
                    └──────┬──────┘
                           │ ◄───────────────────────────────────────┐
                    ┌──────┴──────────┐                               │
                    │ UART data receive│                              │
                    └──────┬──────────┘                               │
                           │                                          │
                    ╱──────┴──────╲                                   │
                   ╱   Receive     ╲        No      ┌─────────────┐   │
                   ╲ data = "5AH"   ╱──────────────►│ Change baud │   │
                    ╲──────┬──────╱                 │ rate        │───┘
                           │ Yes                    │ (Adjust to  │
                    ┌──────┴──────────┐             │ 9600 baud   │
                    │ UART data transmit│           │ source clock)│
                    │ (Transmit data =  │           └─────────────┘
                    │ "5AH")           │
                    └──────┬──────────┘
                    ┌──────┴──────────┐
                    │ UART data receive│
                    └──────┬──────────┘
                    ┌──────┴──────────┐
                    │ UART data transmit│
                    │ (Echoed back the │
                    │ baud rate        │
                    │ modification data)│
                    └──────┬──────────┘
                    ┌──────┴──────────┐
                    │ Change baud rate │
                    │ by receive data  │
                    └──────┬──────────┘
```
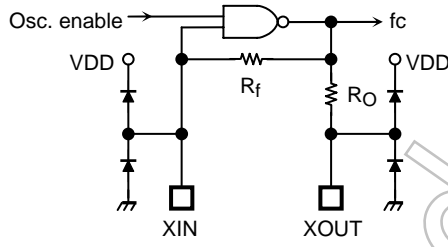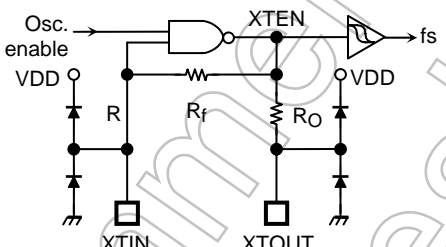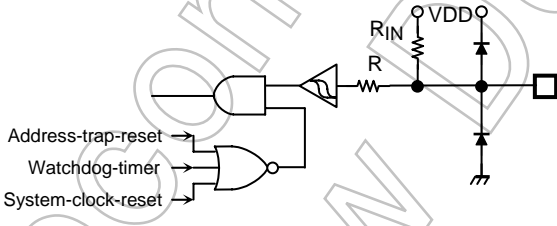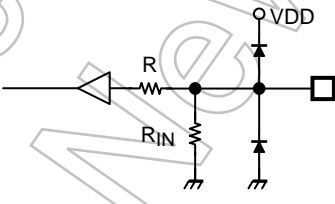
- **Receive data = 30H** (FLASH writing mode) → UART data transmit (Transmit data = 30H) → Password certification (Compare receive data and FLASH data) → UART data receive (Intel Hex format) → FLASH write process → UART data transmit (Checksum)

- **Receive data = 60H** (RAM loader mode) → UART data transmit (Transmit data = 60H) → Password certification (Compare receive data and FLASH data) → UART data receive (Intel Hex format) → RAM write process → UART data transmit (Checksum) → Jumps to start address of user program

- **Receive data = 90H** (FLASH SUM output mode) → UART data transmit (Transmit data = 90H) → UART data transmit (Checksum)

- **Receive data = C0H** (Product discrimination code output mode) → UART data transmit (Transmit data = C0H) → UART data transmit (Product discrimination code)
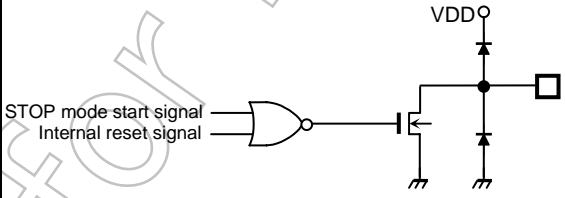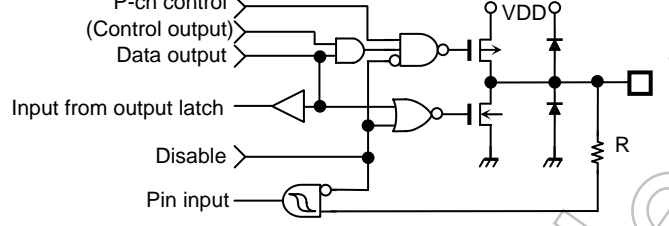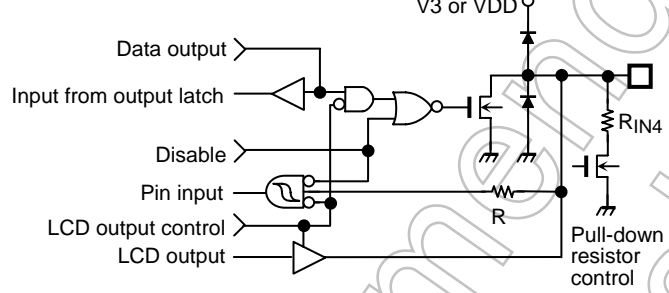
Input/Output Circuitry

(1) Control pins

The input/output circuitries of the TMP86FP24 control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output		Resonator connecting pins (High frequency) R_f = 3 MΩ (typ.) R_O = 0.5 kΩ (typ.)
XTIN XTOUT	Input Output	NORMAL1 mode / NORMAL2 mode Refer to port P2 	Resonator connecting pins (Low frequency) R_f = 20 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 100 Ω (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 100 Ω (typ.)
$\overline{\text{WAKE}}$	Output		Sink open-drain output

(2) Input/output ports

Port	I/O	Input/Output Circuitry	Remarks
P0 P1	I/O	Initial "High-Z" P-ch control (Control output) Data output Input from output latch Disable Pin input (Control input)	Sink open-drain output or CMOS output Hysteresis input R = 100 Ω (typ.)
P20 P23	I/O	Initial "High-Z" P-ch control Data output Input from output latch Disable Pin input (Control input)	Sink open-drain output or CMOS output Hysteresis input R = 100 Ω (typ.)
P21 P22	I/O	Initial "High-Z" Pull-up resistor R_{IN3} Resistor control Data output Input from output latch Disable Pin input	Sink open-drain output or CMOS output Hysteresis input Programmable pull-up resistor $R_{IN3} = 220$ kΩ (typ.)
P3	I/O	Initial "High-Z" P-ch control Data output Input from output latch Disable Pin input	Sink open-drain or CMOS output Hysteresis input High current output (N-ch) R = 100 Ω (typ.)
P6	I/O	Initial "High-Z" Data output Input from output latch I/O control Input control Pin input Disable Analog input	Tri-state I/O Hysteresis input R = 100 Ω (typ.)

Port	I/O	Input/Output Circuitry	Remarks
P5	I/O	Initial "High-Z"	Sink open-drain output or CMOS output Hysteresis input High current output (N-ch) $R = 100\,\Omega$ (typ.)
P4 P9	I/O	Initial "High-Z"	Sink open-drain output Hysteresis input Programmable pull-down resistor $R = 100\,\Omega$ (typ.)

Electrical Characteristics

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

Parameter	Symbol	Pins	Rating	Unit
Supply voltage	V_{DD}		−0.3 to 4.0	V
Input voltage	V_{IN}		−0.3 to V_{DD} + 0.3	
Output voltage	V_{OUT1}	Except V3 pin	−0.3 to V_{DD} + 0.3	
	V_{OUT2}	V3 pin	−0.3 to 4.0	
Output current (Per 1 pin)	I_{OUT1}	P0, P1, P20, P23, P3, P5, P6 ports	−2	mA
	I_{OUT2}	P0, P1, P2, P4, P6, P9, $\overline{WAKE}$ ports	2	
	I_{OUT3}	P3, P5 Ports	10	
Output current (Total)	ΣI_{OUT1}	P0, P1, P20, P23, P3, P5, P6 ports	−30	
	ΣI_{OUT2}	P0, P1, P2, P4, P6, P9, $\overline{WAKE}$ ports	80	
	ΣI_{OUT3}	P3, P5 ports	30	
Power dissipation [Topr = 85°C]	PD		350	mW
Soldering temperature (Time)	Tsld		260 (10 s)	°C
Storage temperature	Tstg		−55 to 125	
Operating temperature	Topr		−40 to 85	

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 Condition-1 (MCU mode) | (V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Max	Unit
Supply voltage	V_{DD}		fc = 16 MHz	NORMAL1, 2 mode	2.7	3.6	V
				IDLE0, 1, 2 mode			
			fc = 8 MHz (In case of connecting the resonator)	NORMAL1, 2 mode	1.8		
				IDLE0, 1, 2 mode			
			fc = 4.2 MHz (In case of external clock input)	NORMAL1, 2 mode			
				IDLE0, 1, 2 mode			
			fs = 32.768 kHz	SLOW1, 2 mode	1.8		
				SLEEP0, 1, 2 mode			
				STOP mode			
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 2.7$ V		$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 2.7$ V		$V_{DD} \times 0.80$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 2.7$ V		0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 2.7$ V			$V_{DD} \times 0.20$	
Clock frequency (In case of connecting the resonator)	fc	XIN, XOUT	V_{DD} = 1.8 V to 3.6 V		1.0	8.0	MHz
			V_{DD} = 2.7 V to 3.6 V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 1.8 V to 3.6 V		30.0	34.0	kHz
Clock frequency (In case of external clock input)	fc	XIN, XOUT	V_{DD} = 1.8 V to 3.6 V		1.0	4.2	MHz
			V_{DD} = 2.7 V to 3.6 V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 1.8 V to 3.6 V		30.0	34.0	kHz
LCD reference voltage	V1		Booster circuit is enable ($V3 \geq V_{DD}$)		0.8	1.2	V
	V2				1.6	2.4	
Capacity for LCD booster circuit	C_{LCD}		LCD booster circuit is enable ($V3 \geq V_{DD}$)		0.1	0.47	μF

Note: 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.

Recommended Operating Condition-2 (Serial PROM mode) | (V_{SS} = 0 V, Topr = 25°C ± 5°C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	VDD		2 MHz ≤ fc ≤ 16MHz	2.7	3.6	V
Clock frequency	fc	XIN, XOUT	VDD = 2.7 V to 3.6 V	2.0	16.0	MHz

Note: The operating temperature area of serial PROM mode is 25°C ± 5°C and the operating area of high frequency of serial PROM mode is different from MCU mode.

DC Characteristics (V$_{SS}$ = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Typ.	Max	Unit
Hysteresis voltage	V$_{HS}$	Hysteresis input	V$_{DD}$ = 3.3 V		–	0.4	–	V
Input current	I$_{IN1}$	TEST	V$_{DD}$ = 3.6 V, V$_{IN}$ = 0 V		–	–	−5	μA
	I$_{IN2}$	Sink open drain, Tri-state	V$_{DD}$ = 3.6 V, V$_{IN}$ = 3.6 V/0 V		–	–	±5	
	I$_{IN4}$	$\overline{RESET}$	V$_{DD}$ = 3.6 V, V$_{IN}$ = 3.6 V		–	–	+5	
Input resistor	R$_{IN1}$	TEST pull down	V$_{DD}$ = 3.6 V, V$_{IN}$ = 3.6 V		–	70	–	kΩ
	R$_{IN2}$	$\overline{RESET}$ pull up P21, P22 ports	V$_{DD}$ = 3.6 V, V$_{IN}$ = 0 V		100	220	450	
	R$_{IN3}$	Programmable pull down (P4, P9 ports)	V$_{DD}$ = 1.8V, V$_{IN}$ = 1.8 V		–	320	–	
High frequency feedback resistor	R$_{FB}$	XOUT	V$_{DD}$ = 3.6 V		–	3	–	MΩ
Low frequency feedback resistor	R$_{FBT}$	XTOUT	V$_{DD}$ = 3.6 V		–	20	–	
Output leakage current	I$_{LO}$	Sink open drain, Tri-state	V$_{DD}$ = 3.6 V, V$_{OUT}$ = 3.4V / 0.2 V		–	–	±10	μA
Output high voltage	V$_{OH}$	CMOS, Tri-state	V$_{DD}$ = 3.6 V, I$_{OH}$ = −0.6 mA		3.2	–	–	V
Output low voltage	V$_{OL}$	Except XOUT, P3 and P5 ports	V$_{DD}$ = 3.6 V, I$_{OL}$ = 0.9 mA		–	–	0.4	
Output low current	I$_{OL}$	P3, P5 ports	V$_{DD}$ = 3.6 V, V$_{OL}$ = 1.0 V		–	6	–	mA
LCD output voltage (LCD booster is enable)	V$_{2-3OUT}$	V2 pin	V3 ≥ V$_{DD}$ Reference supply pin: V1 SEG/COM pin: No load		–	V1 × 2	–	V
		V3 pin			–	V1 × 3	–	
	V$_{1-3OUT}$	V1 pin	V3 ≥ V$_{DD}$ Reference supply pin: V2 SEG/COM pin: No load		–	V2 × 1/2	–	
		V3 pin			–	V2 × 3/2	–	
LCD output current capacity (LCD booster is enable)	I$_{LCDV3}$	V3 pin	V$_{DD}$ = 1.8 V fc = 4MHz C$_{LCD}$ = 0.15 μF Reference supply pin: V1 = 1V	<VFSEL> = 00	–	72	–	mV/μA
				<VFSEL> = 01	–	20	–	
				<VFSEL> = 10	–	15	–	
				<VFSEL> = 11	–	12	–	
			V$_{DD}$ = 1.8 V fc = 4MHz C$_{LCD}$ = 0.15 μF Reference supply pin: V2 = 2 V	<VFSEL> = 00	–	28	–	
				<VFSEL> = 01	–	8	–	
				<VFSEL> = 10	–	5	–	
				<VFSEL> = 11	–	4	–	
Supply current in NORMAL1, 2 mode	I$_{DD}$	Fetch area: Flash area	V$_{DD}$ = 3.6 V V$_{IN}$ = 3.4 V/0.2 V fc = 16 MHz fs = 32.768 kHz	MNP = "1"	–	5.5	6.6	mA
		Fetch area: RAM area		MNP = "0"	–	4.3	5.6	
Supply current in IDLE0, 1, 2 mode				MNP·ATP = "1"	–	3.8	5.5	
				MNP·ATP = "0"	–	2.9	4.5	
Supply current in SLOW1 mode		Fetch area: Flash area	V$_{DD}$ = 3 V V$_{IN}$ = 2.8 V/0.2 V fs = 32.768 kHz	MNP = "1"	–	800	1400	μA
		Fetch area: RAM area		MNP = "0"	–	9	25	
Supply current in SLEEP1 mode				MNP·ATP = "1"	–	800	1400	
				MNP·ATP = "0"	–	7	25	
Supply current in SLEEP0 mode				MNP·ATP = "1"	–	800	1400	
				MNP·ATP = "0"	–	7	25	
Supply current in STOP mode			V$_{DD}$ = 3.6 V V$_{IN}$ = 3.4 V/0.2 V		–	0.5	10	

Note 1: Typical values show those at Topr = 25°C, V_{DD} = 3.3 V.

Note 2: Input current (I_{IN1}, I_{IN2}): The current through pull-up or pull-down resistor is not included.

Note 3: I_{DD} does not include I_{REF} current.

Note 4: The supply currents of SLOW 2 and SLEEP 2 modes are equivalent to IDLE0, IDLE1, IDLE2.

Note 5: Current capacity indicates the drop in pin V3 output voltage per 1 μA. Select an appropriate booster frequency setting in LCDCR<VFSEL> according to LCD panel. To maintain stable operation, the current capacity for the reference pin must be more than ten times that of the output current capacity.

Note 6: MNP (MNPWDW) shows bit0 in EEPCR register and ATP (ATPWDW) shows bit1 in EEPCR register.

Note 7: "Fetch" means reading operation of FLASH data as an instruction by CPU.

| AD Conversion Characteristics | | (V_{SS} = 0.0 V, 2.7 V ≤ V_{DD} ≤ 3.6 V, Topr = −40 to 85°C) | | | | | |

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 1.0	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.5	−	−	
Analog input voltage	V_{AIN}		V_{SS}	−	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 3.6 V V_{SS} = 0.0 V	−	0.35	0.61	mA
Non linearity error		V_{DD} = A_{VDD} 2.7 V V_{SS} = 0.0 V V_{AREF} = 2.7 V	−	−	±2	LSB
Zero point error			−	−	±2	
Full scale error			−	−	±2	
Total error			−	−	±2	

(V_{SS} = 0.0 V, 2.0 V ≤ V_{DD} < 2.7 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 0.6	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.0	−	−	
Analog input voltage	V_{AIN}		V_{SS}		V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 2.0V V_{SS} = 0.0 V	−	0.20	0.34	mA
Non linearity error		V_{DD} = A_{VDD} 2.0 V V_{SS} = 0.0 V V_{AREF} = 2.0 V	−	−	±4	LSB
Zero point error			−	−	±4	
Full scale error			−	−	±4	
Total error			−	−	±4	

(V_{SS} = 0.0 V, 1.8 V ≤ V_{DD} < 2.0 V, Topr = −10 to 85°C) (Note 5)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} − 0.1	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}			V_{DD}		
Analog reference voltage range (Note 4)	ΔV_{AREF}		1.8	−	−	
Analog input voltage	V_{AIN}		V_{SS}	−	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 1.8V V_{SS} = 0.0 V	−	0.18	0.31	mA
Non linearity error		V_{DD} = A_{VDD} 1.8 V V_{SS} = 0.0 V V_{AREF} = 1.8 V	−	−	±4	LSB
Zero point error			−	−	±4	
Full scale error			−	−	±4	
Total error			−	−	±4	

Note 1: The total error includes all errors except a quantization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is different in recommended value by power supply voltage.
About conversion time, please refer to 2.12.2 "Register Configuration".

Note 3: Please use input voltage to AIN input pin in limit of V_{AREF} − V_{SS}.
When voltage of range outside is input, conversion value becomes unsettled and gives affect to other channel conversion value.

Note 4: Analog reference voltage range: ΔV_{AREF} = V_{AREF} − V_{SS}.

Note 5: When AD is used with V_{DD} < 2.0 V, the guaranteed temperature range varies with the operating voltage.

Note 6: When AD converter is not used, fix the A_{VDD} pin on the V_{DD} level.

AC Characteristics   (V_{SS} = 0 V, V_{DD} = 2.7 to 3.6 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 mode	0.25	–	4	µs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	twcH	For external clock operation (XIN input) fc = 16 MHz	–	31.25	–	ns
Low level clock pulse width	twcL					
High level clock pulse width	twcH	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	µs
Low level clock pulse width	twcL					

(V_{SS} = 0 V, V_{DD} = 1.8 to 3.6 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 mode	0.5	–	4	µs
		IDLE1, 2 mode				
		SLOW1, 2 mode	117.6	–	133.3	
		SLEEP1, 2 mode				
High level clock pulse width	twcH	For external clock operation (XIN input) fc = 4.2 MHz	–	119.04	–	ns
Low level clock pulse width	twcL					
High level clock pulse width	twcH	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	µs
Low level clock pulse width	twcL					

Flash Characteristics   (V_{SS} = 0 V)

| Parameter	Condition	Min	Typ.	Max	Unit
Number of guaranteed writes (Page writing) to Flash memory in serial PROM mode	V_{DD} = 2.7 to 3.6 V, 2 MHz ≤ fc ≤ 16 MHz (Topr = 25°C ± 5°C)	–	–	10^5	Times

UART Timing in Serial PROM Mode

| UART Timing-1 | (VDD = 2.7 V to 3.6 V, fc = 2 MHz to 16 MHz, Ta = 25°C) |

Parameter	Symbol	The Number of Clock (fc)	Required Minimum Time	
			at fc = 2 MHz	at fc = 16 MHz
Time from the reception of a matching data until the output of an echo back	CMeb1	Approx. 600	300 μs	37.5 μs
Time from the reception of a Baud rate modification data until the output of an echo back	CMeb2	Approx. 700	350 μs	43.7 μs
Time from the reception of an operation command until the output of an echo back	CMeb3	Approx. 600	300 μs	37.5 μs
Calculation time of checksum	CKsm	Approx. 2360000	1180 ms	147.5 ms

| UART Timing-2 | (VDD = 2.7 V to 3.6 V, fc = 2 MHz to 16 MHz, Ta = 25°C) |

Parameter	Symbol	The Number of Clock (fc)	Required Minimum Time	
			at fc = 2 MHz	at fc = 16 MHz
Time from reset release until acceptance of start bit of RXD pin	RXsup	110000	55 ms	6.9 ms
Time between a matching data and the next matching data	CMtr1	28500	14.3 ms	1.8 ms
Time from the echo back of matching data until the acceptance of baud rate modification data	CMtr2	600	300 μs	37.5 μs
Time from the output of echo back of baud rate modification data until the acceptance of an operation command	CMtr3	750	375 μs	46.9 μs
Time from the output of echo back of operation command until the acceptance of password count storage addresses	CMtr4	950	475 μs	59.4 μs
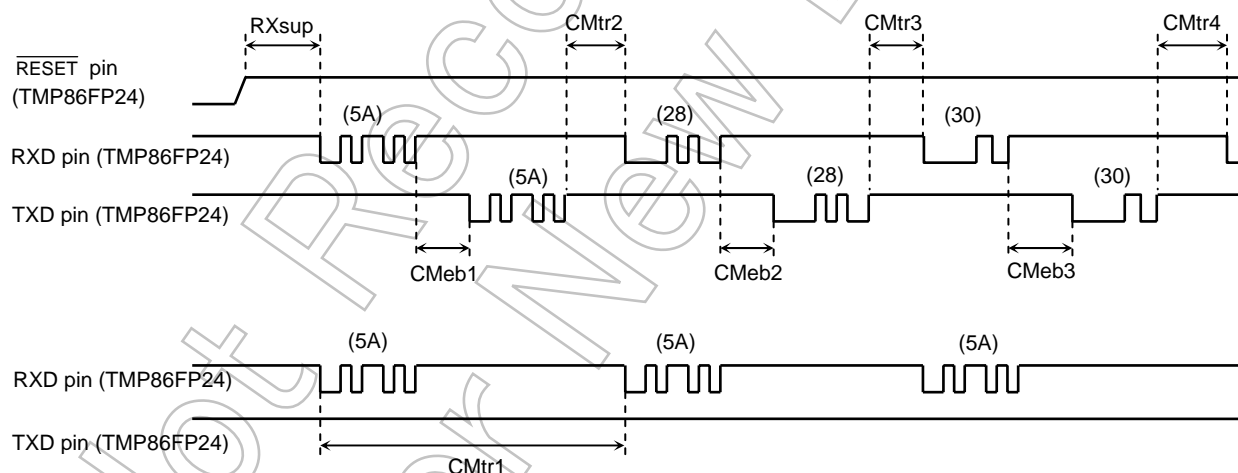
Recommended Oscillating Conditions

Note 1: An electrical shield by metal shield plate on the surface of IC package is recommended in order to protect the device from the high electric field stress applied from CRT (Cathodic ray tube) for continuous reliable operation.

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

Handling Precaution

- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.

1. When using the Sn-37Pb solder bath
   Solder bath temperature = 230°C
   Dipping time = 5 seconds
   Number of times = once
   R-type flux used

2. When using the Sn-3.0Ag-0.5Cu solder bath
   Solder bath temperature = 245°C
   Dipping time = 5 seconds
   Number of times = once
   R-type flux used

   Note : The pass criteron of the above test is as follows: Solderability rate until forming ≥ 95%

- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Package Dimensions

LQFP80-P-1212-0.50A

Unit: mm

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.
We are confident that our products can satisfy your application needs now and in the future.