TOSHIBA

TOSHIBA Original CMOS 8-Bit Microcontroller

TLCS-870/C Series

TMP86CM25AFG

TOSHIBA CORPORATION

Semiconductor Company

Revision History

Date	Revision	
2008/3/6	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

Document Change Notification

The purpose of this notification is to inform customers about the launch of the Pb-free version of the device. The introduction of a Pb-free replacement affects the datasheet. Please understand that this notification is intended as a temporary substitute for a revision of the datasheet.

Changes to the datasheet may include the following, though not all of them may apply to this particular device.

1. Part number

   Example: TMPxxxxxxF → TMPxxxxxxFG

   All references to the previous part number were left unchanged in body text. The new part number is indicated on the prelims pages (cover page and this notification).

2. Package code and package dimensions

   Example: LQFP100-P-1414-0.50C → LQFP100-P-1414-0.50F

   All references to the previous package code and package dimensions were left unchanged in body text. The new ones are indicated on the prelims pages.

3. Addition of notes on lead solderability

   Now that the device is Pb-free, notes on lead solderability have been added.

4. RESTRICTIONS ON PRODUCT USE

   The previous (obsolete) provision might be left unchanged on page 1 of body text. A new replacement is included on the next page.

5. Publication date of the datasheet

   The publication date at the lower right corner of the prelims pages applies to the new device.

1. Part number

2. Package code and dimensions

Previous Part Number (in Body Text)	Previous Package Code (in Body Text)	New Part Number	New Package Code	OTP
TMP86CM25AF	P-QFP100-1420-0.65A	TMP86CM25AFG	QFP100-P-1420-0.65A	–

*: For the dimensions of the new package, see the attached Package Dimensions diagram.

3. Addition of notes on lead solderability

The following solderability test is conducted on the new device.

Lead solderability of Pb-free devices (with the G suffix)

Test	Test Conditions	Remark
Solderability	(1) Use of Lead (Pb) ·solder bath temperature = 230°C ·dipping time = 5 seconds ·the number of times = once ·use of R-type flux (2) Use of Lead (Pb)-Free ·solder bath temperature = 245°C ·dipping time = 5 seconds ·the number of times = once ·use of R-type flux	Leads with over 95% solder coverage till lead forming are acceptable.

4. RESTRICTIONS ON PRODUCT USE

The following replaces the “RESTRICTIONS ON PRODUCT USE” on page 1 of body text.

RESTRICTIONS ON PRODUCT USE 20070701-EN

- The information contained herein is subject to change without notice.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the “Handling Guide for Semiconductor Devices,” or “TOSHIBA Semiconductor Reliability Handbook” etc.
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury (“Unintended Usage”). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in his document shall be made at the customer’s own risk.
- The products described in this document shall not be used or embedded to any downstream products of which manufacture, use and/or sale are prohibited under any applicable laws and regulations.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patents or other rights of TOSHIBA or the third parties.
- Please contact your sales representative for product-by-product details in this document regarding RoHS compatibility. Please use these products in this document in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances. Toshiba assumes no liability for damage or losses occurring as a result of noncompliance with applicable laws and regulations.
- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance/Handling Precautions.

5. Publication date of the datasheet

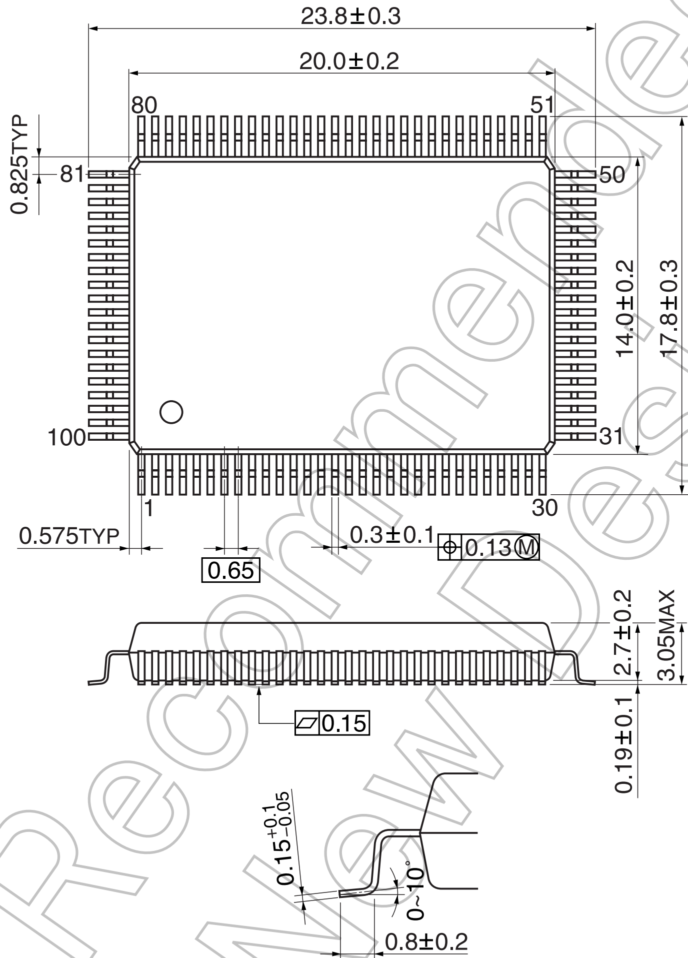
The publication date of this datasheet is printed at the lower right corner of this notification.

(Annex)

Package Dimensions

QFP100-P-1420-0.65A

Unit: mm



Comparison table of TMP86CM25F/CS25F/PS25F/C925XB and TMP86CM25AF/FM25F (shaded cells: Difference)

	TMP86CM25F/ TMP86CS25F	TMP86PS25F	TMP86C925XB (Emulation chip)	TMP86FM25F	TMP86CM25AF
ROM	32 K (Mask ROM) 60 K (Mask ROM)	60 K (OTP)	–	32 K (Flash)	32 K (Mask ROM)
RAM	2 K		–	2 K	
I/O	42 pin		42 pin (MCU part)	42 pin	
External Interrupt	5 pin			5 pin	
AD Converter	8-bit AD converter × 8 ch			8-bit AD converter × 8 ch (Note 3)	
Timer Counter	18-bit timer × 1 ch 8-bit timer × 4 ch			18-bit timer × 1 ch 8-bit timer × 4 ch	
Serial Interface	8-bit SIO × 2 ch UART × 1 ch			8-bit SIO × 2 ch UART × 1 ch	
LCD	60 seg × 16 com			60 seg × 16 com (Note 4)	
Key-on Wakeup	4 ch			4 ch	
Operating Voltage in MCU Mode	1.8 to 5.5 V at 4.2 MHz 2.7 to 5.5 V at 8 MHz 4.5 to 5.5 V at 16 MHz		1.8 to 5.25 V at 4.2 MHz 2.7 to 5.25 V at 8 MHz 4.5 to 5.25 V at 16 MHz	1.8 to 3.6 V at 4.2 MHz (External clock) 1.8 to 3.6 V at 8 MHz (Resonator) 2.7 to 3.6 V at 16 MHz	
Operating Temperature in MCU Mode	−40 to 85°C		0 to 60°C	−40 to 85°C	
Writing to Flash Memory	–			2.7 to 3.6V at 16 MHz 25°C ± 5°C	–
Package	P-QFP100-1420-0.65A		FBGA272	P-QFP100-1420-0.65A	
CPU Wait (Note 1)	N/A			Available (Note 2)	

Note 1: The CPU wait is a CPU halt function for stabilizing of power supply of Flash memory. The CPU wait period is as follows. In the CPU wait period except RESET, CPU is halted but peripheral functions are not halted. Therefore, if the interrupt occurs during the CPU wait period, the interrupt latch is set. In this case, if the IMF has been set to "1", the interrupt service routine is executed after CPU wait period. For details refer to 1.1 "Flash Memory" in TMP86FM25F data sheet.

Condition	Wait Time	Halt/Operate	
		CPU	Peripherals
After reset release	2^{10}/fc[s]	Halt	Halt
Changing from STOP mode to NORMAL mode (at EEPCR<MNPWDW> = "1")	2^{10}/fc[s]	Halt	Operate
Changing from STOP mode to SLOW mode (at EEPCR<MNPWDW> = "1")	2^{3}/fs[s]	Halt	Operate
Changing from IDLE0/1/2 mode to NORMAL mode (at EEPCR<ATPWDW> = "0")	2^{10}/fc[s]	Halt	Operate
Changing from SLEEP0/1/2 mode to SLOW mode (at EEPCR<ATPWDW> = "0")	2^{3}/fs[s]	Halt	Operate

Note 2: Though the TMP86CM25AF does not have a Flash memory, the CPU wait function is inserted in TMP86CM25A to keep the compatibility with Flash product (TMP86FM25F).

Note 3: AD conversion time of TMP86CM25A/FM25 is different from that of TMP86CM25/CS25/PS25/C925. For details, refer to 2.12 "8-Bit AD Converter (ADC)".

Note 4: The reference voltage of TMP86CM25A/FM25 is different from that of TMP86CM25/CS25/PS25/C925. For details, refer to "Electrical Characteristics".

CMOS 8-Bit Microcontroller

TMP86CM25AF

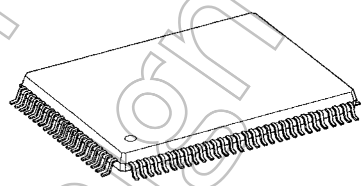
The TMP86CM25A is the high-speed, high-performance and low power consumption 8-bit microcomputer, including ROM, RAM, dot matrix LCD driver, multi-function timer/counter, serial interface (UART/SIO), a 8-bit AD converter and two clock generators on chip.

Product No.	ROM	RAM	Package	Flash MCU
TMP86CM25AF	32 K × 8 bits	2 K × 8 bits	P-QFP100-1420-0.65A	*TMP86FM25F

*: Under development

Feautures

P-QFP100-1420-0.65A



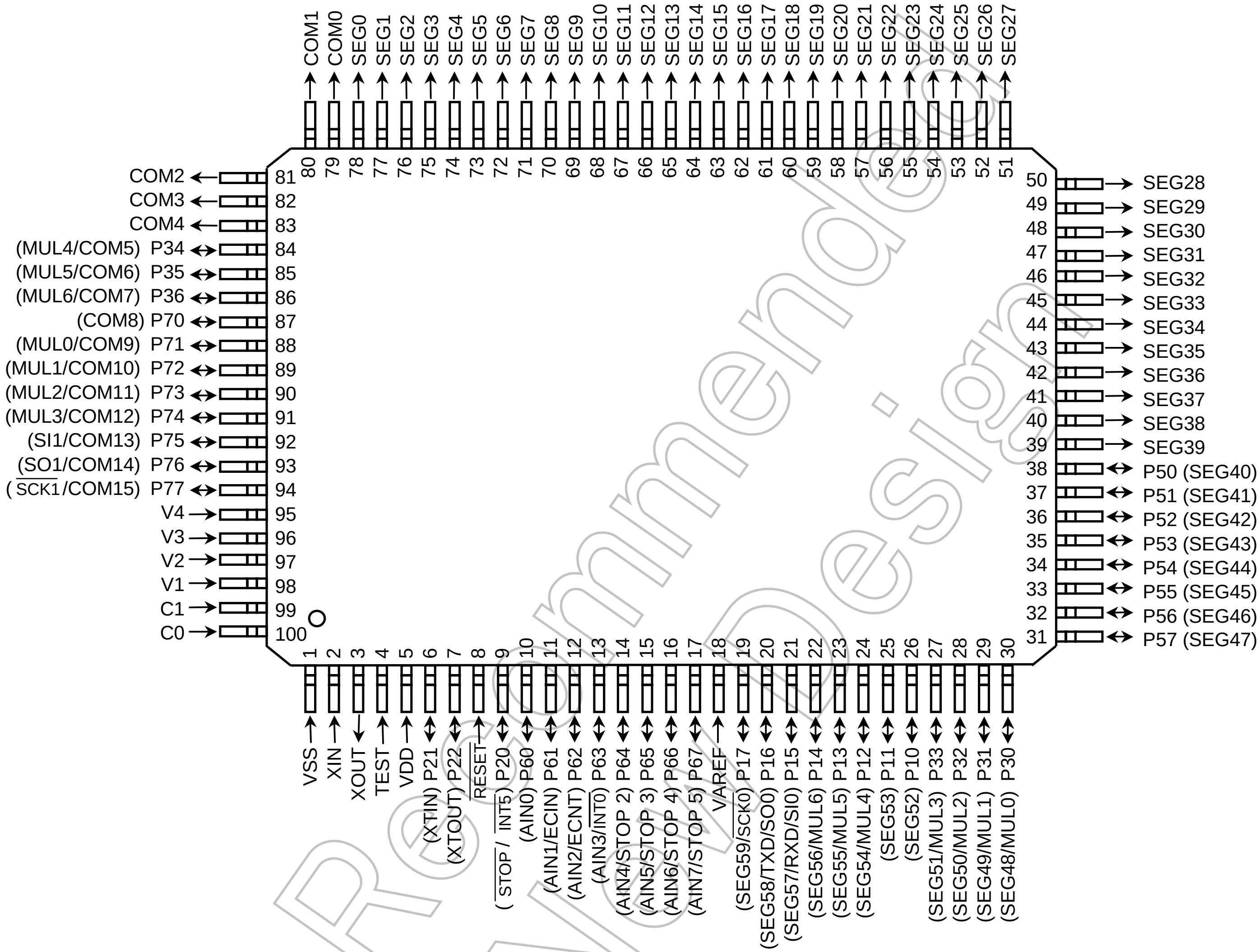
TMP86CM25AF

- 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
- 20 interrupt sources (External: 5, Internal: 15)
- Input/output ports (42 pins)
  (Out of which 20 pins are also used as SEG pins)
  (Out of which 11 pins are also used as COM pins)
- 18-bit timer counter: 1 ch
  - Timer, Event counter, Pulse width measurement, Frequency measurement modes
- 8-bit timer counter: 4 ch
  - Timer, Event counter, PWM output, Programmable divider output, PPG modes
- Time Base Timer
- Divider output function
- Watchdog Timer
  - Interrupt source/internal reset generate (Programmable)

- Serial interface
  - UART: 1 ch
  - SIO: 2 ch
- 8-bit successive approximation type AD converter
  - Analog input: 8 ch
- Four key-on wakeup pins
- LCD driver/controller
  - Built-in voltage booster for LCD driver
  - With display memory
  - LCD direct drive capability (60 seg × 16 com, 60 seg × 8 com, 60 seg × 4 com)
  - 1/16, 1/8, 1/4 duties drive are programmably selectable
- Dual clock operation
  - Single/Dual-clock mode
- Nine power saving operating modes
  - STOP mode: Oscillation stops. Battery/capacitor back-up. 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 3.6 V at 8 MHz/32.768 kHz
  2.7 to 3.6 V at 16 MHz/32.768 kHz

Pin Assignments (Top view)

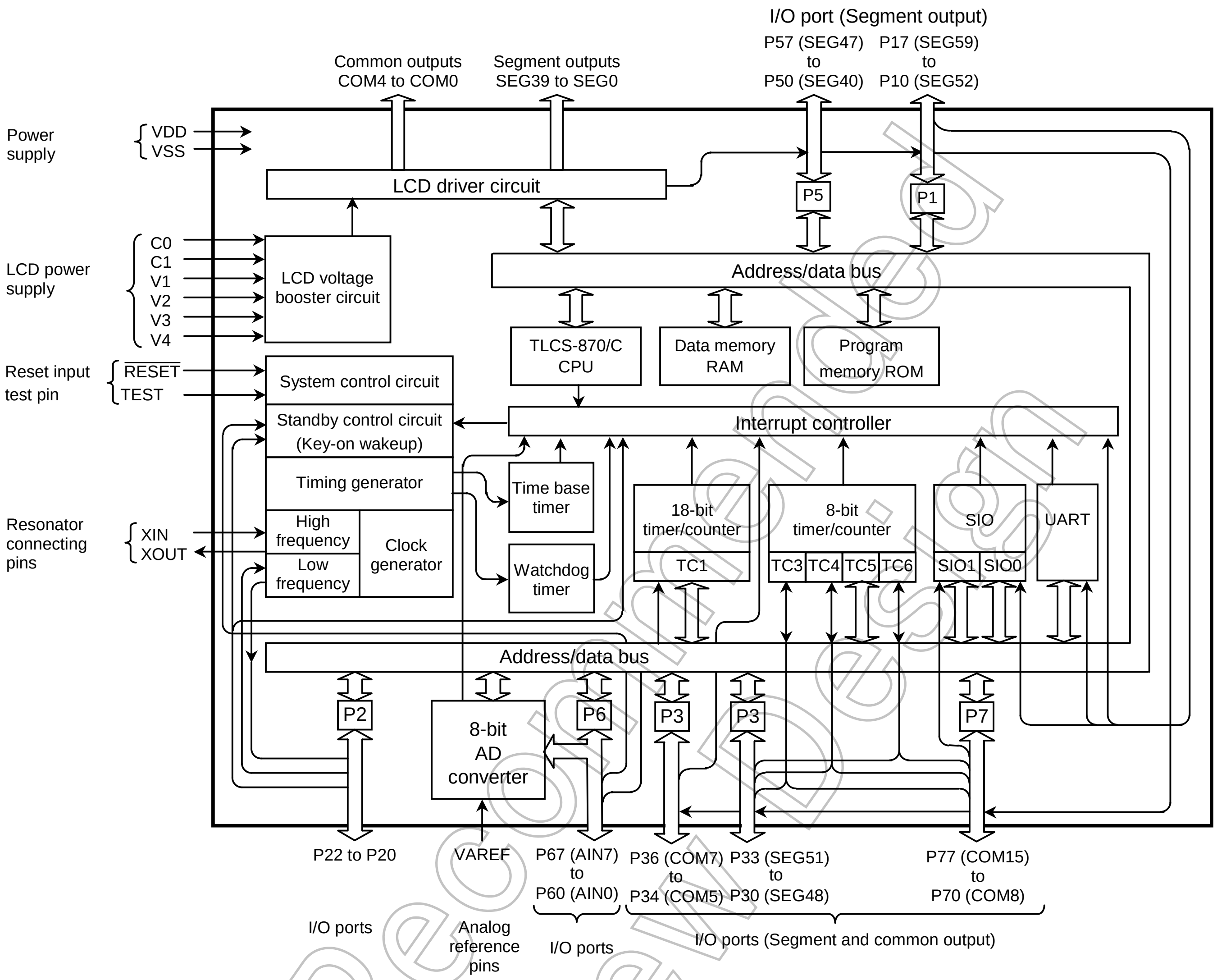
P-QFP100-1420-0.65A



Note: Ports assigned as MUL6 to MUL0 can switch pin assignment by the multifunction register (MULSEL). For functions assigned to each pin, see the table below.

Pin Name	Function	Pin Assignment
MUL0	DVO	P30 or P71
MUL1	PWM3 , PDO3 ,TC3	P31 or P72
MUL2	PPG4 , PWM4 , PDO4 , TC4	P32 or P73
MUL3	PPG6 , PWM6 , PDO6 , TC6	P33 or P74
MUL4	INT1	P12 or P34
MUL5	INT2	P13 or P35
MUL6	INT3	P14 or P36

Block Diagram



Pin Functions

Pin Name	Input/Output	Functions		
P17 (SEG59, $\overline{SCK0}$)	I/O (I/O)	8-bit input/output port with latch. When used as input port, serial clock input/output, serial data input/output or UART data input/output, the output latch must be set to "1". When an external interrupt input, the corresponding bit of MULSEL should be cleared to "0" and the output latch must be set to "1". When used as a LCD segment output, the P1LCR must be set to "1".	Serial clock input/output	LCD segment outputs
P16 (SEG58, TXD, SO0)	I/O (Output)		UART data output Serial data output	
P15 (SEG57, RXD, SI0)	I/O (I/O)		UART data input Serial data input	
P14 (SEG56, MUL6)	I/O(I/O)		External interrupt 3 input	
P13 (SEG55, MUL5)	I/O (I/O)		External interrupt 2 input	
P12 (SEG54, MUL4)	I/O (I/O)		External interrupt 1 input	
P11 (SEG53)	I/O (Output)			
P10 (SEG52)	I/O (Output)			
P22 (XTOUT)	I/O (Output)	3-bit input/output port with latch. When used as an input port, the output latch must be set to "1".	Resonator connecting pins (32.768 kHz) For inputting external clock, XTIN is used and XTOUT is opened.	
P21 (XTIN)	I/O (Input)			
P20 ($\overline{INT5}$, $\overline{STOP}$)	I/O (Input)		External interrupt input 5 or STOP mode release signal input.	
P36 (COM7, MUL6)	I/O (I/O)	7-bit I/O port with latch. When used as input port, the output latch must be set to "1". When an external interrupt input, the corresponding bit of MULSEL should be set to "1" and the output latch must be set to "1". When a timer/counter input/output or divider output, the corresponding bit of MULSEL should be cleared to "0" and the output latch must be set to "1". When used as a LCD output, the P3LCR must be set to "1".	External interrupt 3 input	LCD common outputs
P35 (COM6, MUL5)	I/O (I/O)		External interrupt 2 input	
P34 (COM5, MUL4)	I/O (I/O)		External interrupt 1 input	
P33 (SEG51, MUL3)	I/O (I/O)		Timer/counter 6 input/output	LCD segment outputs
P32 (SEG50, MUL2)	I/O(I/O)		Timer/counter 4 input/output	
P31 (SEG49, MUL1)	I/O(I/O)		Timer/counter 3 input/output	
P30 (SEG48, MUL0)	I/O(Output)		Divider output	
P57 (SEG47) to P50 (SEG40)	I/O (Output)	8-bit input/output port with latch. When used as a LCD segment output, the P5LCR must be set to "1".	LCD segment outputs	
P67 (AIN7, STOP5)	I/O (Input)	8-bit programmable input/output ports (Tri-state). Each bit of this port can be individually configured as an input or an output under software control. When used as a key-on wakeup input, an external interrupt input and timer/counter input, the P6CR must be cleared to "0". When used as an analog input, the P6DR and P6CR should be cleared to "0".	STOP5 input	AD converter analog inputs
P66 (AIN6, STOP4)	I/O (Input)		STOP4 input	
P65 (AIN5, STOP3)	I/O (Input)		STOP3 input	
P64 (AIN4, STOP2)	I/O (Input)		STOP2 input	
P63 (AIN3, $\overline{INT0}$)	I/O (Input)		External interrupt 0 input	
P62 (AIN2, ECNT)	I/O (Input)		Timer/counter 1 input	
P61 (AIN1, ECIN)	I/O (Input)			
P60 (AIN0)	I/O (Input)			
P70 (COM8)	I/O(Output)	8-bit input/output port with latch. When used as input port, serial clock input/output or serial data input/output, the output latch must be set to "1". When a timer/counter input/output or divider output, the corresponding bit of MULSEL should be set to "1" and the output latch must be set to "1". When used common output, P7 port control register (P7LCR) should be set to "1".		LCD common outputs
P71 (COM9, MUL0)	I/O(Output)		Divider Output	
P72 (COM10, MUL1)	I/O(I/O)		Timer/counter 3 input/output	
P73 (COM11, MUL2)	I/O(I/O)		Timer/counter 4 input/output	
P74 (COM12, MUL3)	I/O(I/O)		Timer/counter 6 input/output	
P75 (COM13, SI1)	I/O(I/O)		Serial data input	
P76 (COM14, SO1)	I/O(Output)		Serial data output	
P77 (COM15, $\overline{SCK1}$)	I/O(I/O)		Serial clock input/output	
SEG39 to SEG0	Output	LCD segment outputs		
COM4 to COM0		LCD common outputs		
V4 to V1	LCD voltage booster pin	LCD voltage booster pin. Capacitors are required between C0 and C1 pin and V1/V2/V3/ V4 pin and GND.		
C1 to C0				
XIN, XOUT	Input Output	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.		
$\overline{RESET}$	Input	Reset signal input		
TEST	Input	Test pin for out-going test. Be fixed to low.		
VDD, VSS	Power Supply	+1.8 to +3.6 V, 0 (GND)		
VAREF		Analog reference voltage input.		

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, and the reset circuit.

1.1 Memory Address Map

The TMP86CM25A memory consists of 4 blocks: 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 TMP86CM25A memory address map. The general-purpose registers are not assigned to the RAM address space.

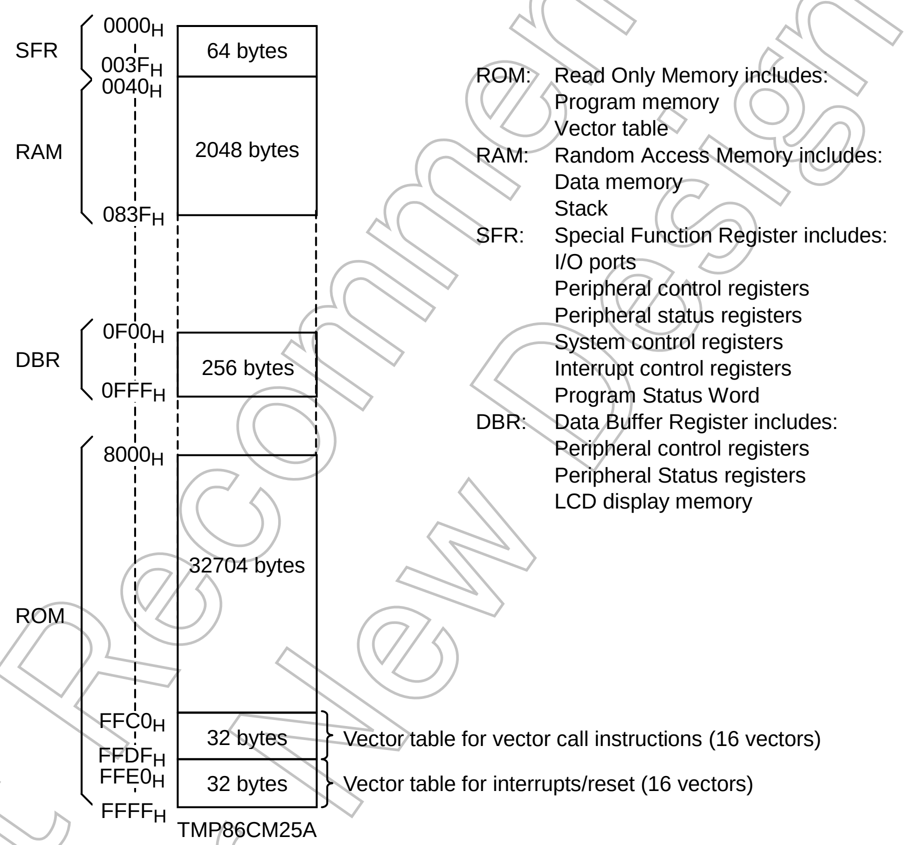


Figure 1.1.1 Memory Address Maps

1.2 Program Memory (ROM)

The TMP86CM25A has a 32 K × 8 bits (Address 8000_H to $FFFF_H$) of program memory (Mask programmed ROM). However, placing program memory on the internal RAM is deregulated if a certain procedure is executed (See 2.5.5 Address Trap).

1.3 Data Memory (RAM)

The TMP86CM25A has 2048 bytes of internal RAM (Address 0040_H to $083F_H$). The first 192 bytes (0040_H to $00FF_H$) 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.

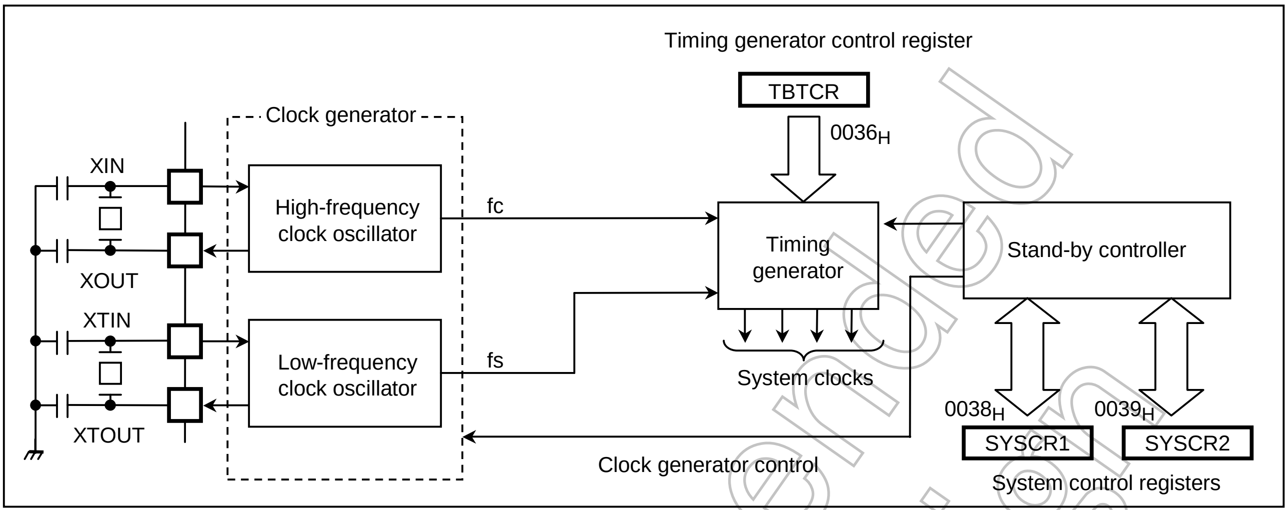


Figure 1.4.1 System Clock Control

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.

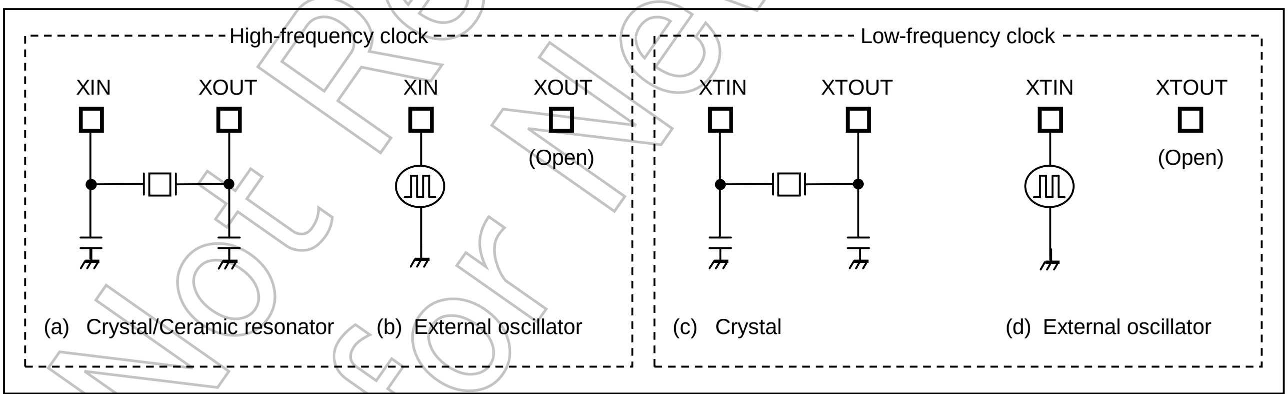


Figure 1.4.2 Examples of Resonator Connection

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”.

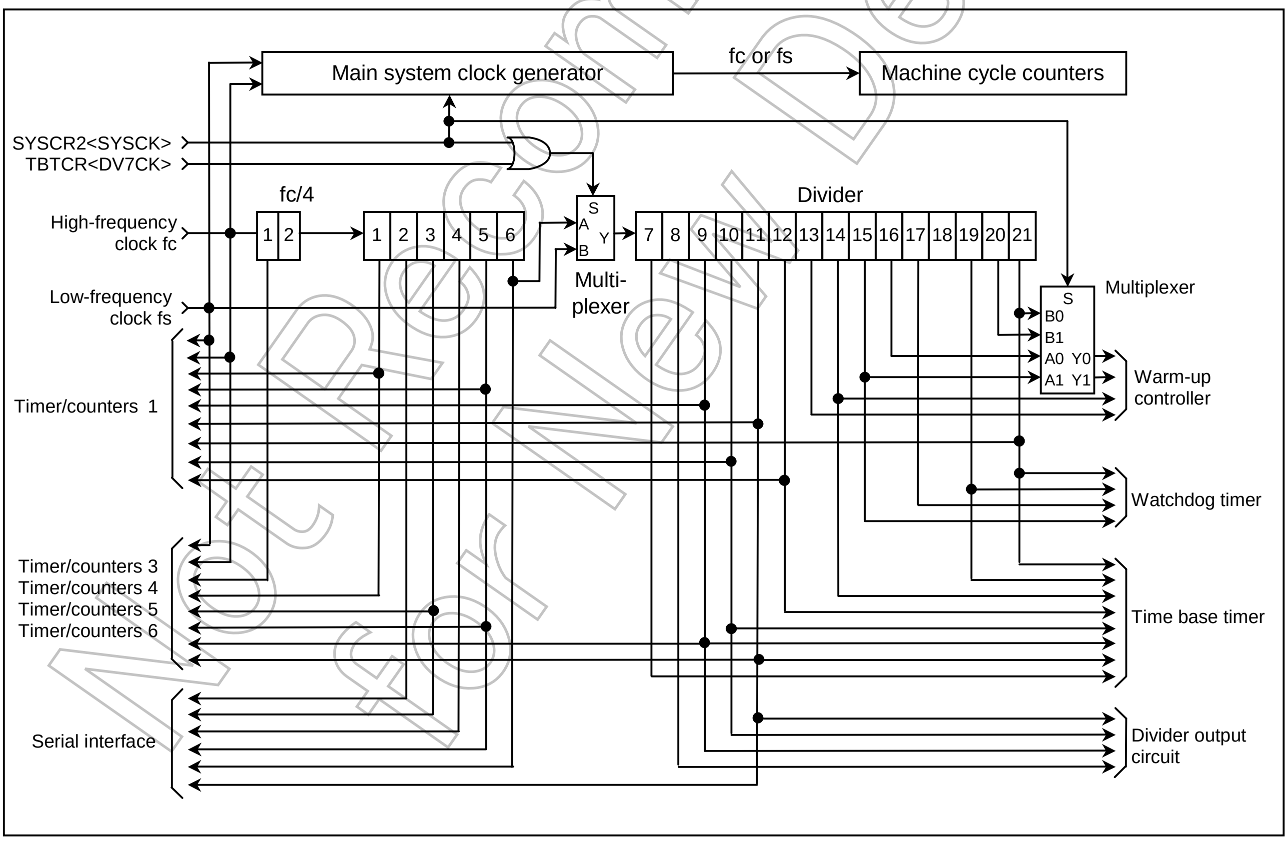


Figure 1.4.3 Configuration of Timing Generator

TBTCR (0036_H)

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

(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], fs; 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 a “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.

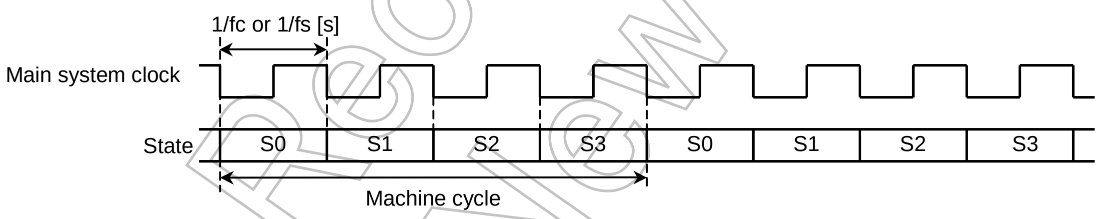


Figure 1.4.5 Machine Cycle

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 TMP86CM25A 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”, EF6 (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”, EF6 (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.

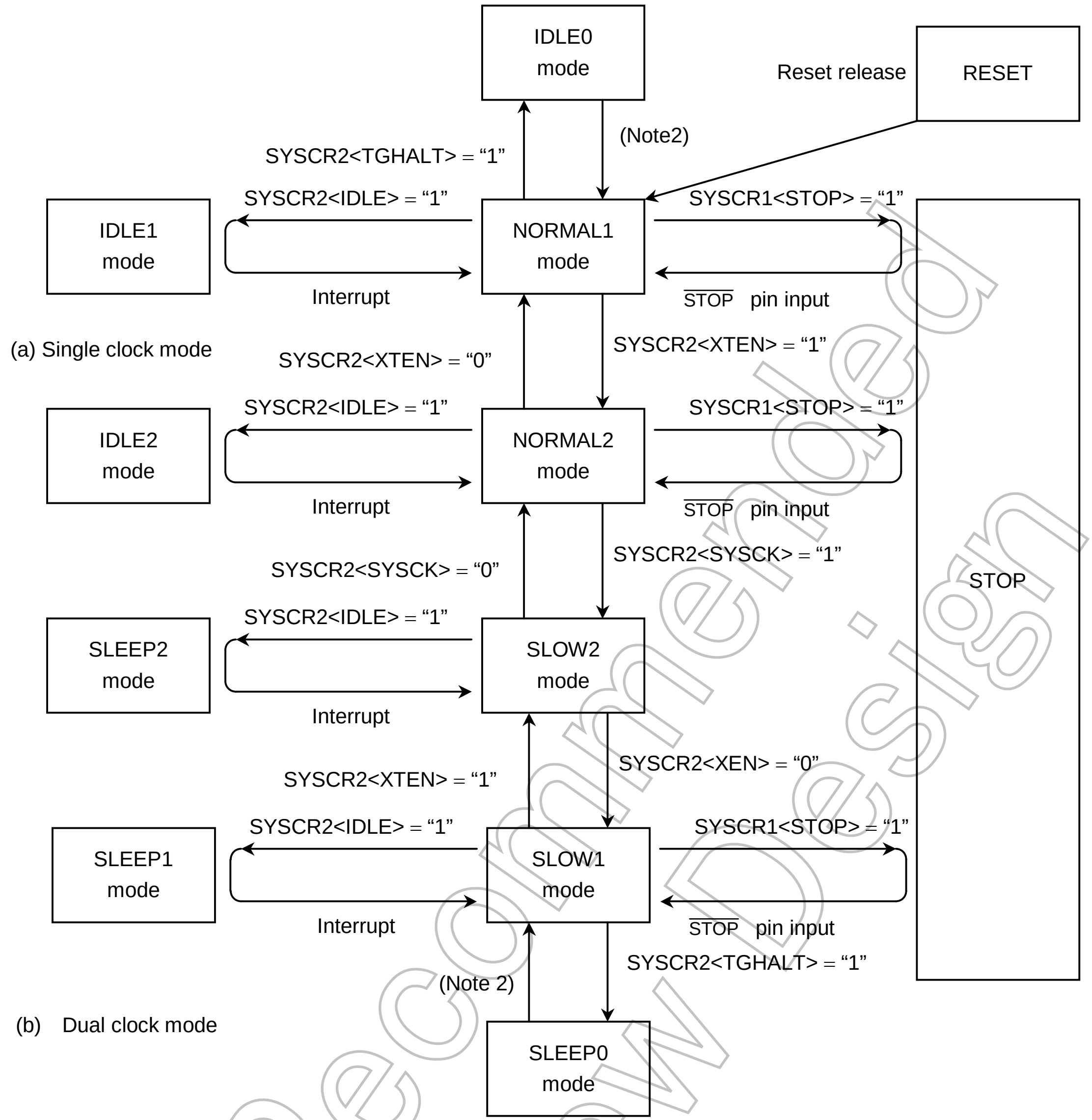
(3) STOP mode

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 $\overline{\text{STOP}}$ 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. This CPU wait function is also included in masked ROM “A” version (TMP86CM25AF) for keeping compatibility with Flash product.

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. This function is also included in masked ROM “A” version (TMP86CM25AF) for keeping compatibility with Flash product.



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.

<table>
<tr><th colspan="2" rowspan="2">Operating Mode</th><th colspan="2">Oscillator</th><th rowspan="2">CPU Core</th><th rowspan="2">TBT</th><th rowspan="2">Other Peripherals</th><th rowspan="2">Machine Cycle Time</th></tr>
<tr><th>High frequency</th><th>Low frequency</th></tr>
<tr><td rowspan="5">Single clock</td><td>RESET</td><td rowspan="4">Oscillation</td><td rowspan="5">Stop</td><td>Reset</td><td>Reset</td><td>Reset</td><td rowspan="3">4/fc [s]</td></tr>
<tr><td>NORMAL1</td><td>Operate</td><td rowspan="3">Operate</td><td rowspan="2">Operate</td></tr>
<tr><td>IDLE1</td><td rowspan="3">Halt</td></tr>
<tr><td>IDLE0</td><td rowspan="2">Halt</td><td rowspan="2">–</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Halt</td></tr>
<tr><td rowspan="8">Dual clock</td><td>NORMAL2</td><td rowspan="4">Oscillation</td><td rowspan="7">Oscillation</td><td>Operate with high frequency</td><td rowspan="7">Operate</td><td rowspan="6">Operate</td><td rowspan="2">4/fc [s]</td></tr>
<tr><td>IDLE2</td><td>Halt</td></tr>
<tr><td>SLOW2</td><td>Operate with low frequency</td><td rowspan="4">4/fs [s]</td></tr>
<tr><td>SLEEP2</td><td>Halt</td></tr>
<tr><td>SLOW1</td><td rowspan="4">Stop</td><td>Operate with low frequency</td></tr>
<tr><td>SLEEP1</td><td rowspan="3">Halt</td></tr>
<tr><td>SLEEP0</td><td rowspan="2">Halt</td><td rowspan="2">–</td></tr>
<tr><td>STOP</td><td>Stop</td><td>Halt</td></tr>
</table>

Figure 1.4.6 Operating Mode Transition Diagram

System Control Register 1

SYSCR1	7	6	5	4	3	2	1	0	
(0038$_H$)	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			R/W
RETM	Operating mode after STOP mode	0: Return to NORMAL1/2 mode 1: Return to SLOW1 mode			R/W
OUTEN	Port output during STOP mode	0: High impedance 1: Output kept			R/W
WUT	Warm-up time at releasing STOP mode (Note 8)		Return to NORMAL mode	Return to SLOW mode	R/W
		00	3×2^{16}/fc + (2^{10}/fc)	3×2^{13}/fs + (2^{3}/fs)	
		01	2^{16}/fc + (2^{10}/fc)	2^{13}/fs + (2^{3}/fs)	
		10	3×2^{14}/fc + (2^{10}/fc)	3×2^{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 $\overline{\text{RESET}}$ 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 (STOP2 to STOP5) is used, RELM should be set to “1”.

Note 7: 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 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. This CPU wait function is included in masked ROM “A” version (TMP86CM25AF) for keeping compatibility with Flash product.
(The CPU wait period for FLASH is shown in parentheses)

System Control Register 2

SYSCR2	7	6	5	4	3	2	1	0	
(0039$_H$)	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	R/W
SYSCK	Main system clock select (write)/main system clock monitor (read)	0: High-frequency clock 1: Low-frequency clock	R/W
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)	R/W
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)	R/W

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 (STOP2 to STOP5) 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. This CPU wait function is also included in masked ROM "A" version (TMP86CM25A) for keeping compatibility with Flash product.

Note 1: The STOP mode can be released by either the STOP or key-on wakeup pin (STOP2 to STOP5). 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. <u>Level-sensitive release mode (RELM = "1")</u>

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high or setting the STOPx (x: 2 to 5) 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.

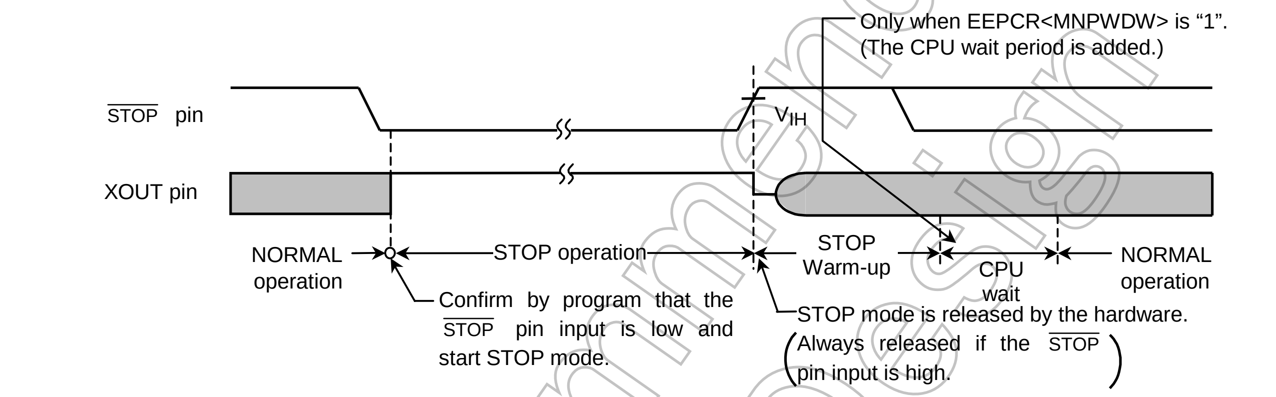
b. Using an external interrupt input $\overline{\text{INT5}}$ ($\overline{\text{INT5}}$ is a falling edge-sensitive input).

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

```
         LD     (SYSCR1), 01010000B   ; Setup the level-sensitive release mode.
SSTOPH:  TEST   (P2PRD). 0            ; Wait until the 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   ; Setup the level-sensitive release mode.
         SET    (SYSCR1). 7           ; Starts STOP mode.
SINT5:   RETI
```



Note : When the STOP mode is started with the EEPCR<MNPWDW> = "1", the CPU wait for the power supply of Flash control circuit is executed after the STOP warm-up time. This function is included in masked ROM "A" version (TMP86CM25AF) for keeping compatibility with Flash product (TMP86FM25F).

Figure 1.4.8 Level-sensitive Release Mode

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

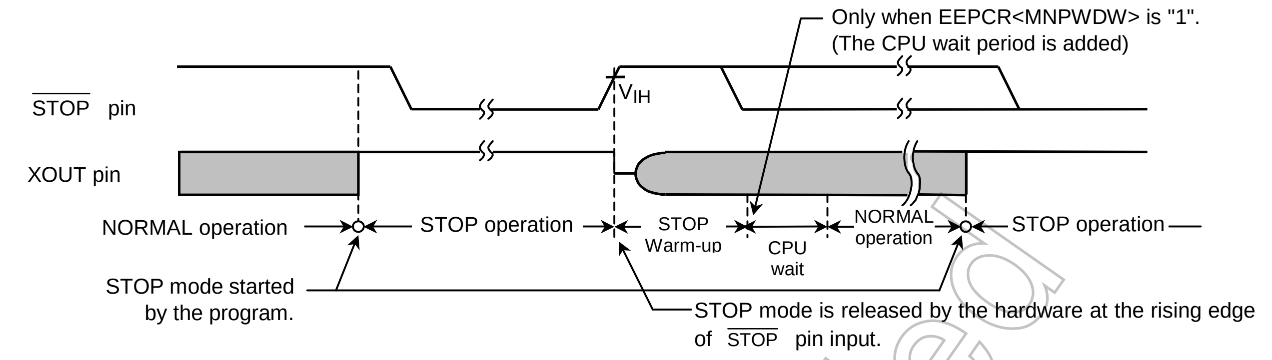
Note 2: In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the $\overline{\text{STOP}}$ pin input is detected.

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

In this mode, STOP mode is released by a rising edge of the $\overline{\text{STOP}}$ pin input. This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the $\overline{\text{STOP}}$ pin. In the edge-sensitive release mode, STOP mode is started even when the $\overline{\text{STOP}}$ pin input is high level. Do not use any STOPx (x: 2 to 5) 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.
```



Note : When the STOP mode is started with the EEPCR<MNPWDW> = "1", the CPU wait for the power supply of Flash control circuit is executed after the STOP warm-up time. This function is included in masked ROM "A" version (TMP86CM25AF) for keeping compatibility with Flash product (TMP86FM25F).

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 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 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 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 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.

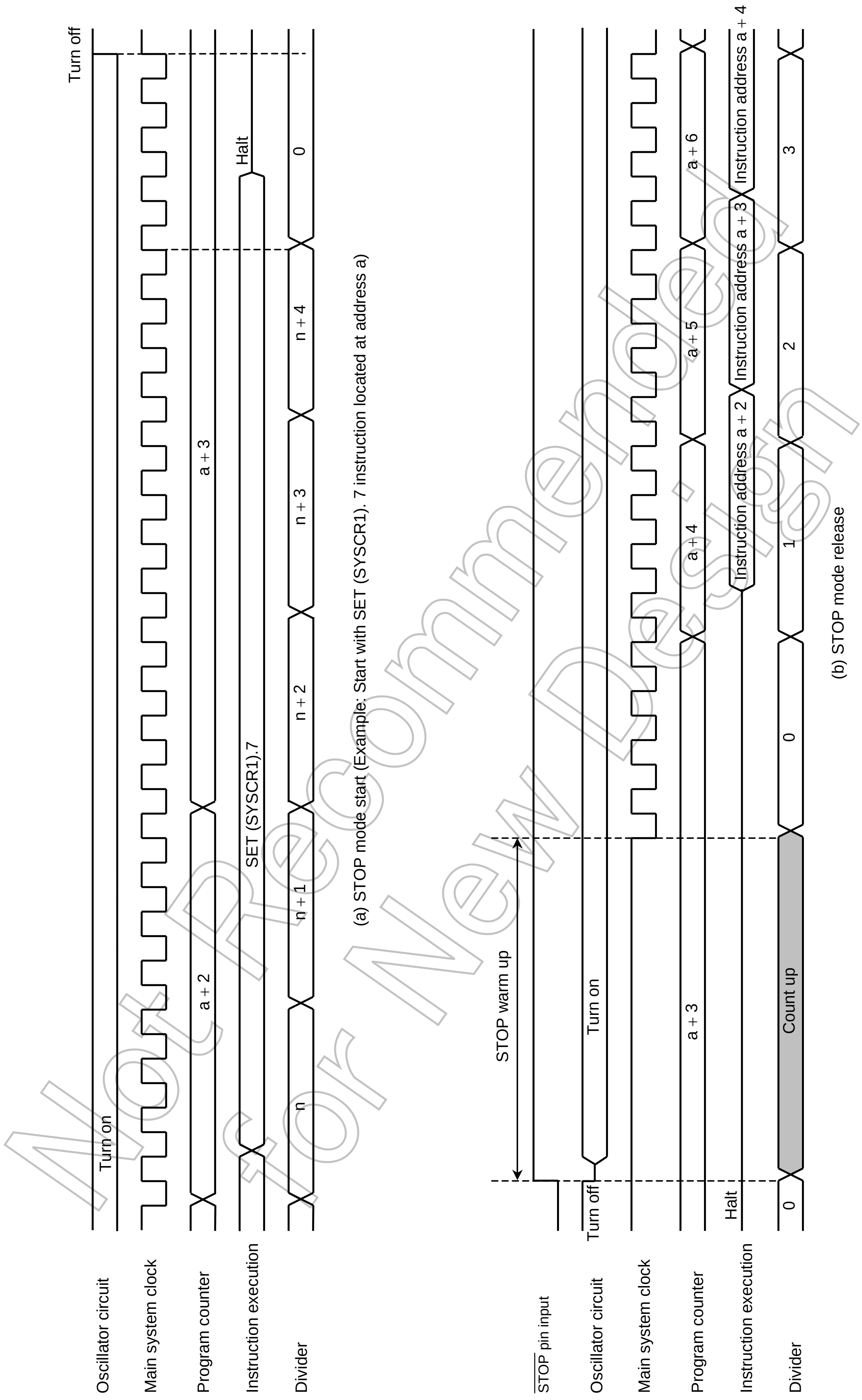


Figure 1.4.10 STOP Mode Start/Release (when EEPCR<MNPWDW> = “0”)

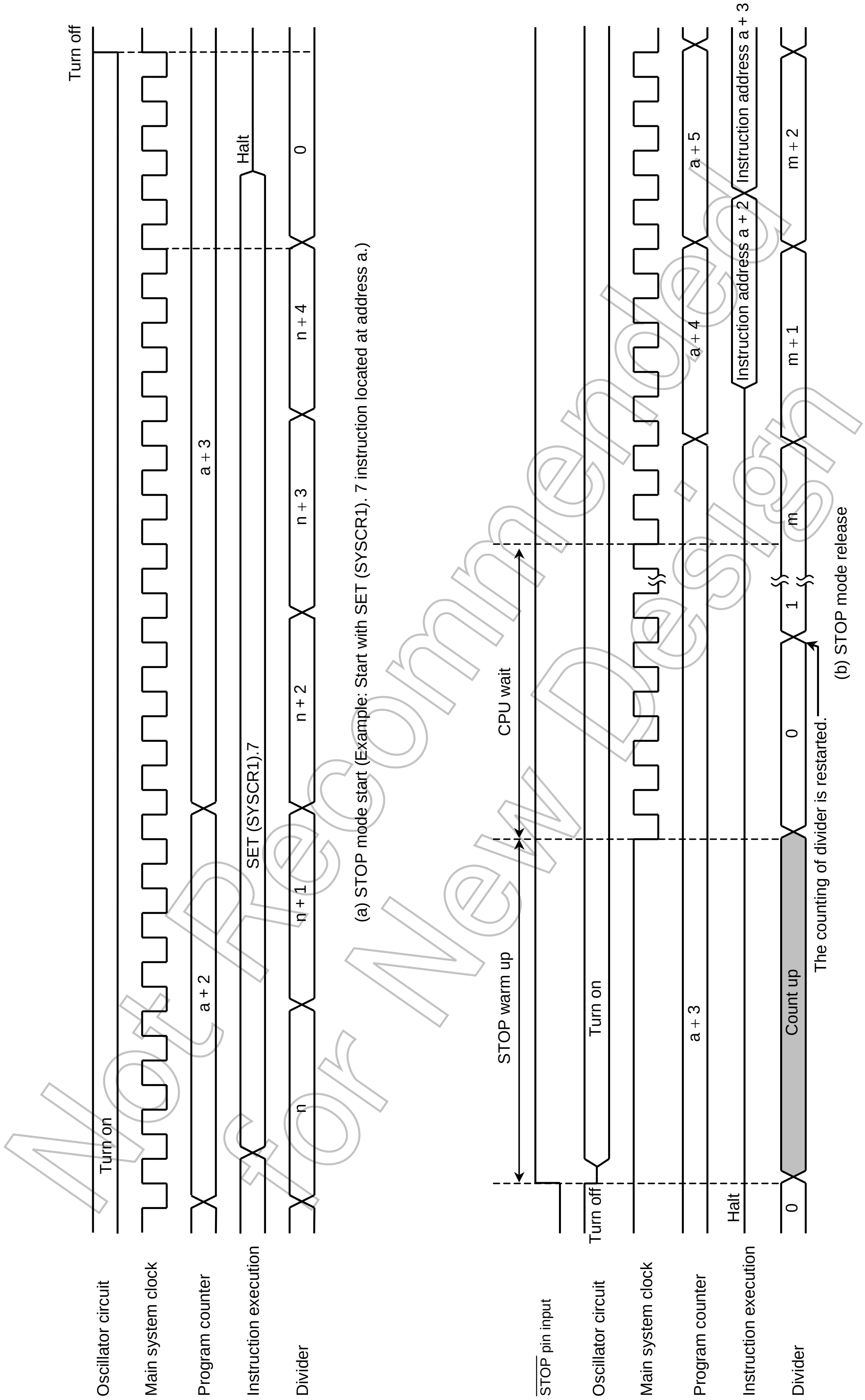


Figure 1.4.11 STOP Mode Start/Release (when EEPCR<MNPWDW> = “1”)

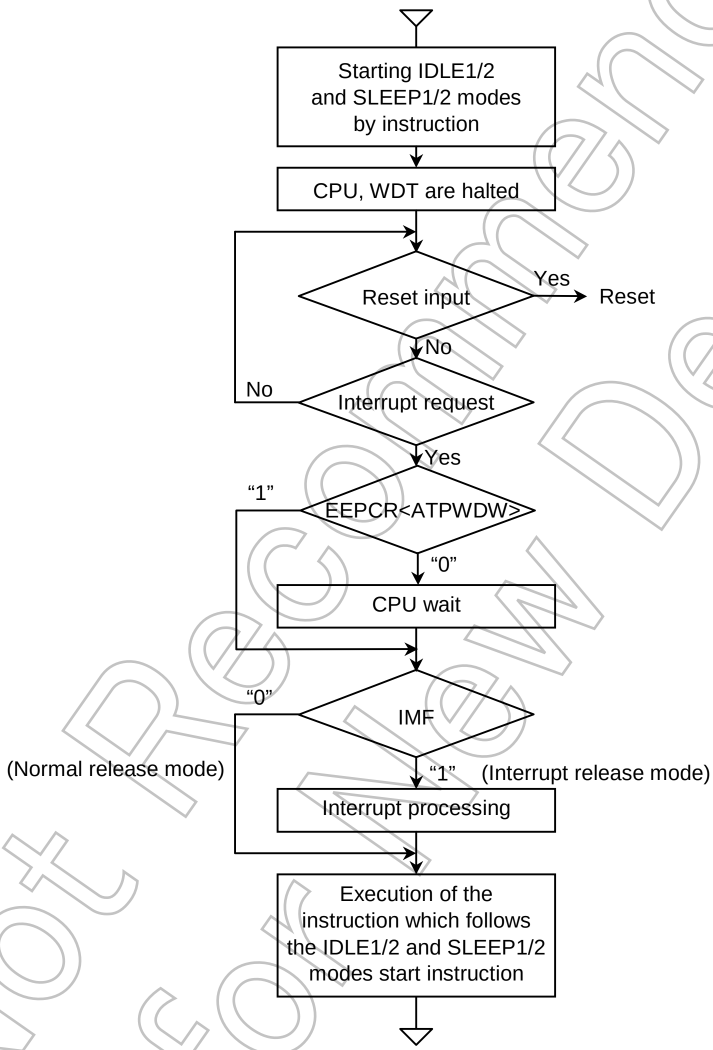
(2) IDLE1/2 mode, SLEEP1/2 mode

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].

  The CPU wait function is also included in masked ROM "A" version (TMP86CM25A) for keeping compatibility with Flash product (TMP86FM25F).

  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 mode are started, the watchdog timer interrupt will be processed but IDLE1/2 and SLEEP1/2 mode will not be started.

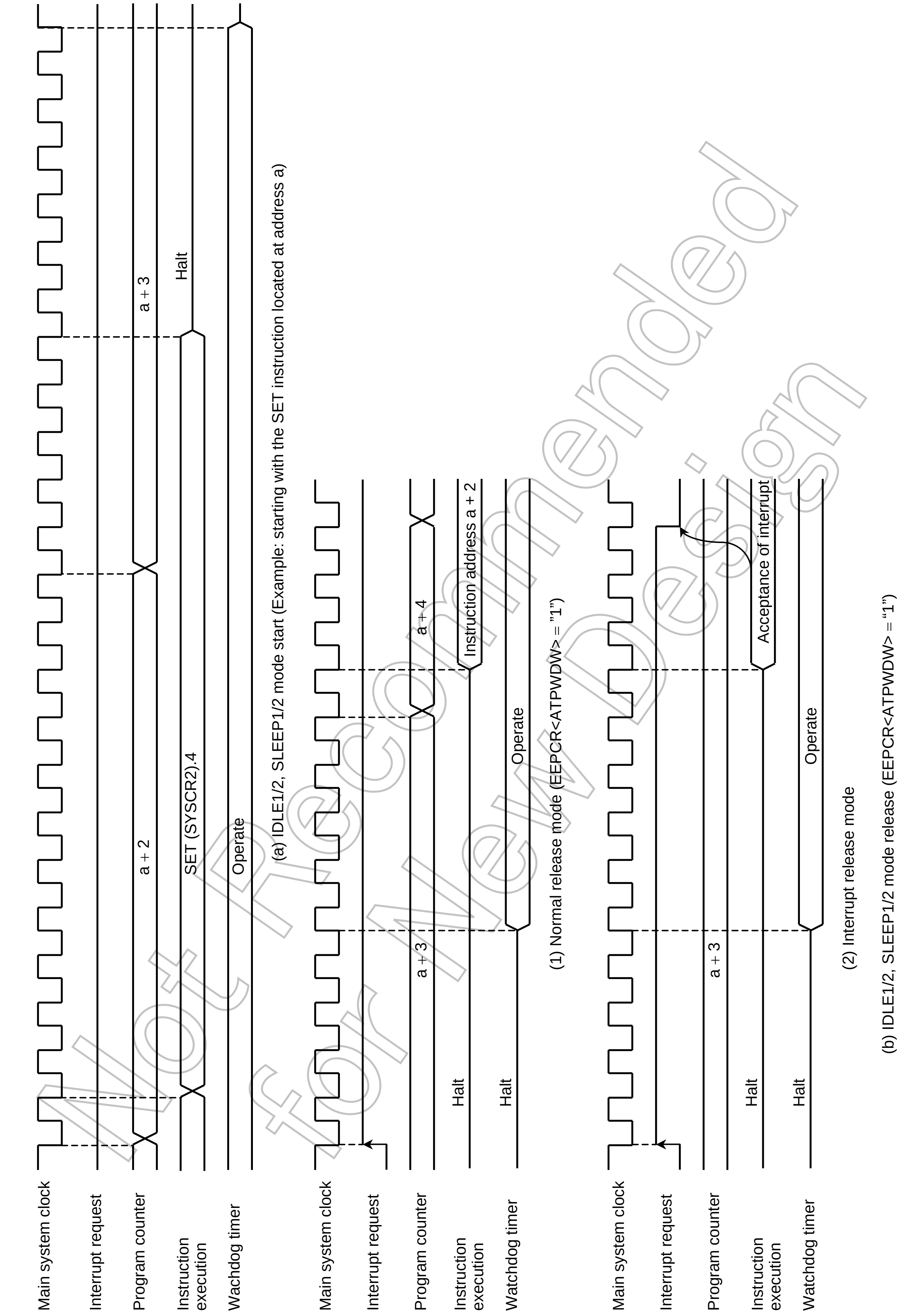


Figure 1.4.13 IDLE1/2, SLEEP1/2 Mode Start/Release

(3) IDLE0, SLEEP0 mode (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.

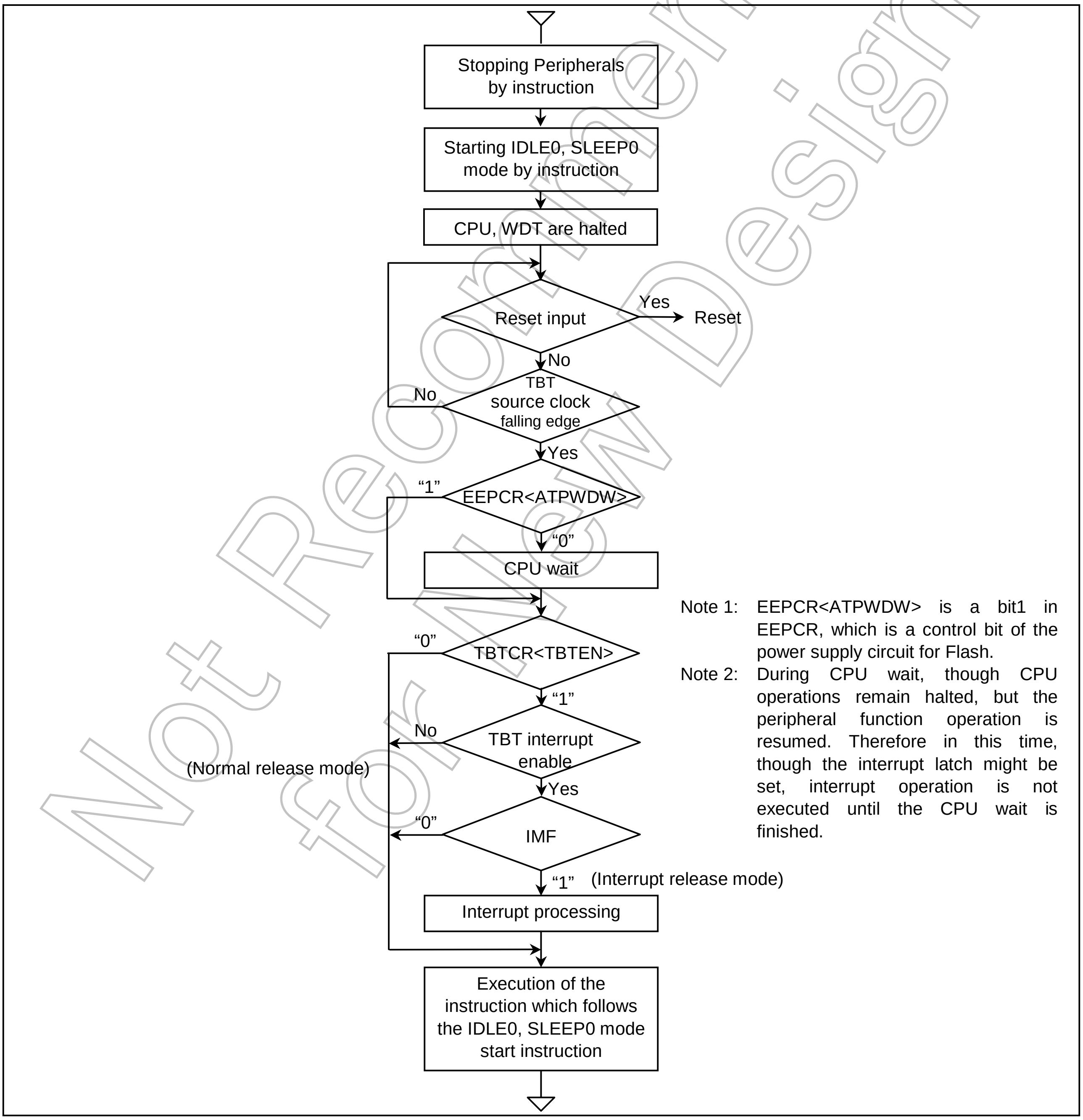


Figure 1.4.14 IDLE0, SLEEP0 Mode

- 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 (EF6) 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].

  The CPU wait function is also included in masked ROM “A” version (TMP86CM25A) for keeping compatibility with Flash product (TMP86FM25F).

  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・EF6・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・EF6・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.

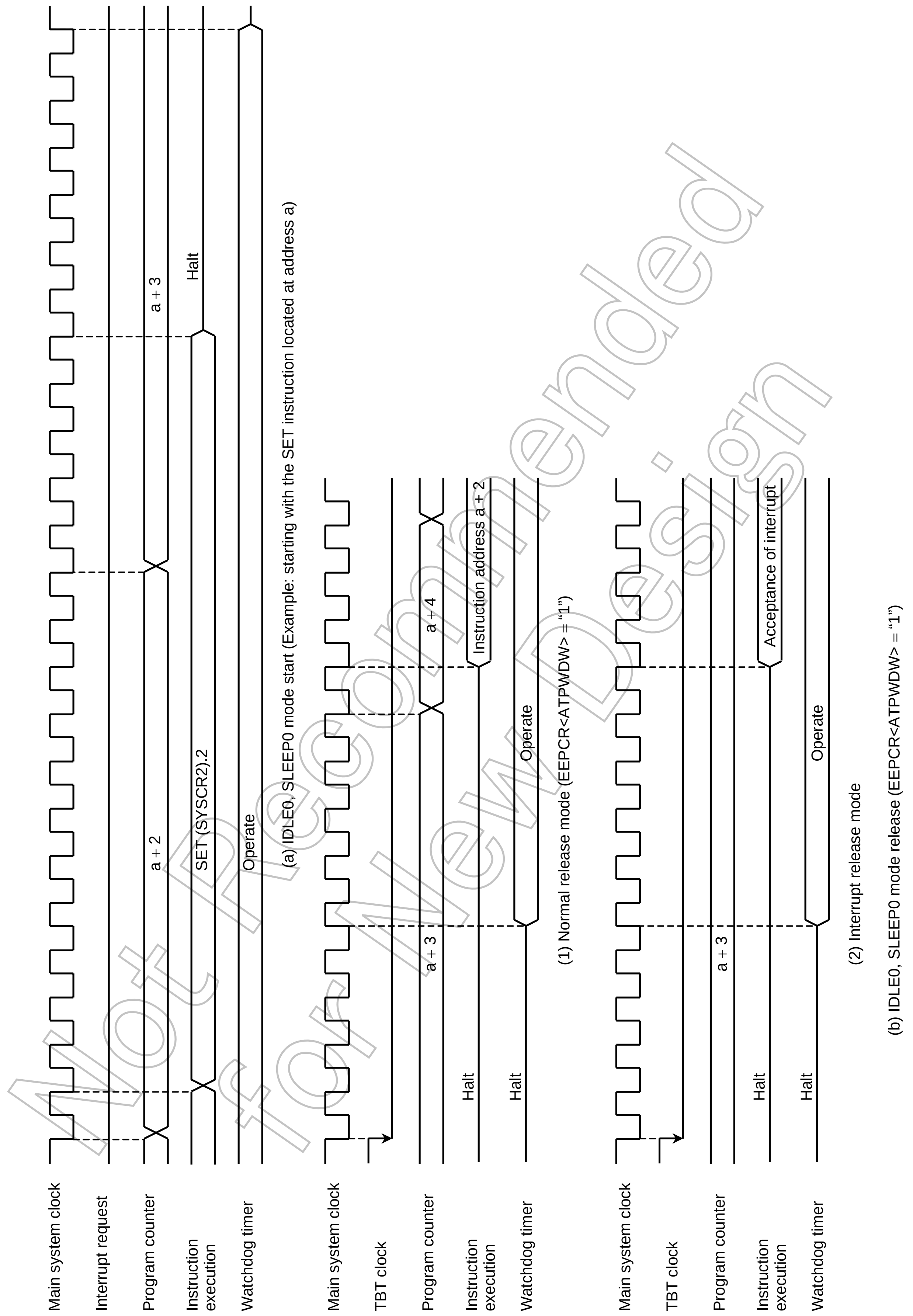


Figure 1.4.15 IDLE0, SLEEP0 Mode Start/Release

(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 (TC4, TC3).

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 4, 3 (TC4, TC3) 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.)
```

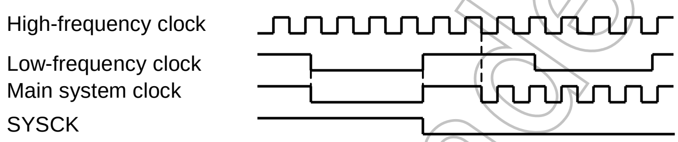
Example 2: Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
          SET    (SYSCR2). 6          ; SYSCR2<XTEN> ← 1
          LD     (TC3CR), 43H         ; Sets mode for TC4, TC3
                                      ; (16-bit TC, fs for source clock)
          LD     (TC4CR), 05H
          LDW    (TTREG3), 8000H      ; Sets warm-up time
                                      ; (Depend on oscillator accompanied.)
          DI                          ; IMF ← 0
          SET    (EIRH). 3            ; Enables INTTC4
          EI                          ; IMF ← 1
          SET    (TC4CR). 3           ; Starts TC4, 3
           :
PINTTC4:  CLR    (TC4CR). 3           ; Stops TC4, 3
          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
           :
VINTTC4:  DW     PINTTC4              ; INTTC4 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 (TC4, TC3), 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.



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 TMP86CM25A 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    (TC3CR), 63H         ;  Sets mode for TC4, TC3
                                     ;  (16-bit TC, fc for source clock)
          LD    LD   (TC4CR), 05H
          LD    (TTREG4), 0F8H       ;  Sets warm-up time
                                     ;  (Depend on oscillator accompanied)
          DI                         ;  IMF ← 0
          SET   (EIRH). 3            ;  Enables INTTC4
          EI                         ;  IMF ← 1
          SET   (TC4CR). 3           ;  Starts TC4, 3
           :
PINTTC4:  CLR   (TC4CR). 3           ;  Stops TC4, 3
          CLR   (SYSCR2). 5          ;  SYSCR2<SYSCK> ← 0
                                     :  (Switches the main system clock to the
                                     :  high-frequency clock)
          RETI
           :
VINTTC4:  DW    PINTTC4              ;  INTTC4 vector table
```

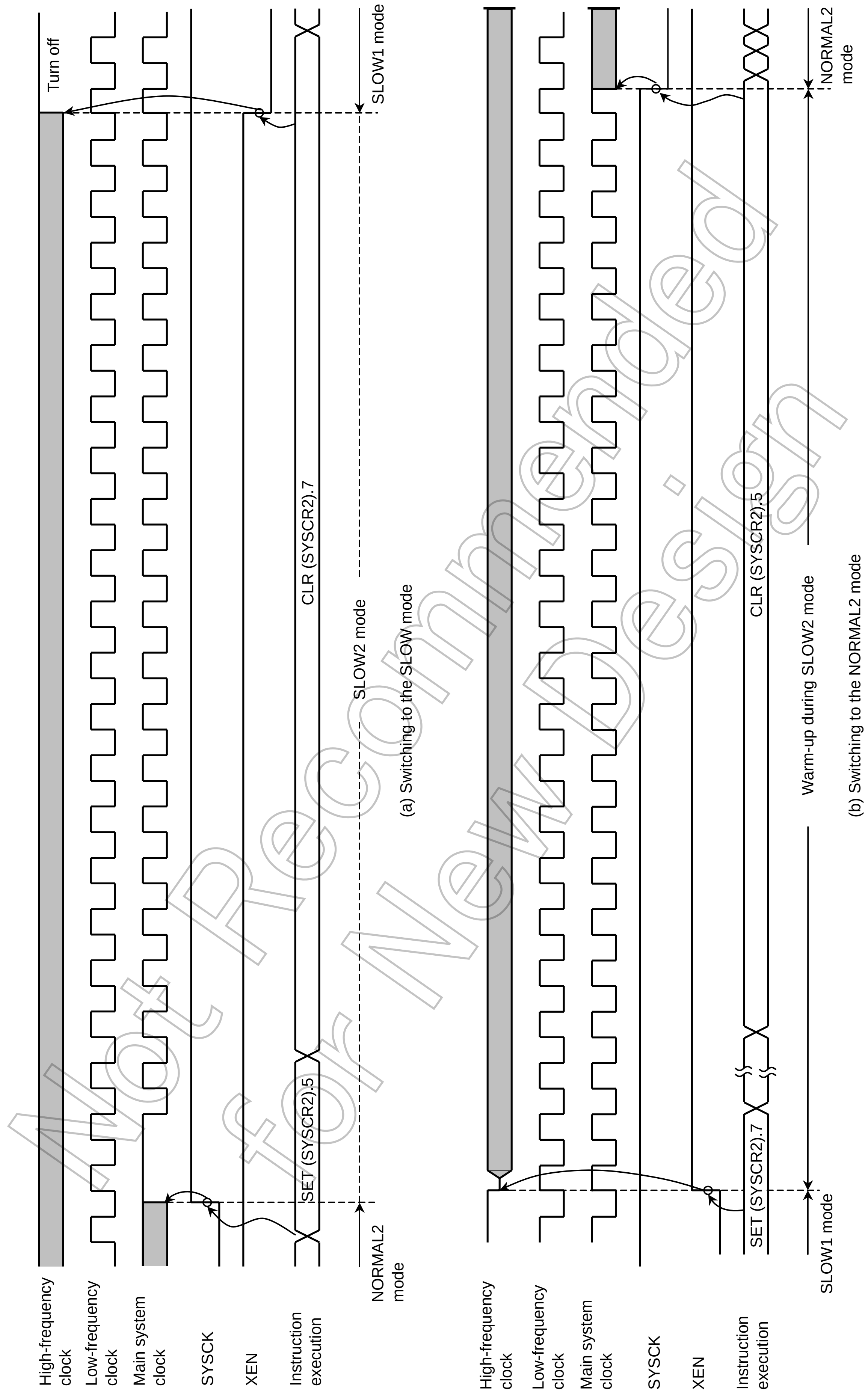


Figure 1.4.16 Switching between the NORMAL2 and SLOW Modes

1.5 Interrupt Control Circuit

The TMP86CM25A has a total (Reset is excluded) of 15 interrupt sources for 20 interrupt factors; 4 of the sources are multiplexed. Multiple interrupt with priorities is available. 4 of the internal factors 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	–	$FFFE_H$	High 1
Internal	INTSWI	(Software interrupt)	Non-maskable	–	$FFFC_H$	2
Internal	INTUNDEF	(Executed the Undefined Instruction interrupt)	Non-maskable	–	$FFFC_H$	2
Internal	INTATRAP	(Address Trap interrupt)	Non-maskable	IL_2	$FFFA_H$	2
Internal	INTWDT	(Watchdog Timer interrupt)	Non-maskable	IL_3	$FFF8_H$	2
External	$\overline{INT0}$	(External interrupt 0)	$IMF \cdot EF_4 = 1$	IL_4	$FFF6_H$	5
External	INT1	(External interrupt 1)	$IMF \cdot EF_5 = 1$	IL_5	$FFF4_H$	6
Internal	INTTBT	(Time base timer interrupt)	$IMF \cdot EF_6 = 1$	IL_6	$FFF2_H$	7
External	INT2	(External interrupt 2)	$IMF \cdot EF_7 = 1$	IL_7	$FFF0_H$	8
Internal	INTTC1	(TC1 interrupt)	$IMF \cdot EF_8 = 1$	IL_8	$FFEE_H$	9
Internal	INTRXD	(UART received interrupt)	$IMF \cdot EF_9 = 1$	IL_9	$FFEC_H$	10
Internal	INTSIO0	(Serial interface 0 interrupt)				
Internal	INTTXD	(UART transmitted interrupt)	$IMF \cdot EF_{10} = 1$	IL_{10}	$FFEA_H$	11
Internal	INTSIO1	(Serial interface 1 interrupt)				
Internal	INTTC4	(TC4 interrupt)	$IMF \cdot EF_{11} = 1$	IL_{11}	$FFE8_H$	12
Internal	INTTC6	(TC6 interrupt)	$IMF \cdot EF_{12} = 1$	IL_{12}	$FFE6_H$	13
Internal	INTADC	(AD converter interrupt)	$IMF \cdot EF_{13} = 1$	IL_{13}	$FFE4_H$	14
External	INT3	(External interrupt 3)	$IMF \cdot EF_{14} = 1$	IL_{14}	$FFE2_H$	15
Internal	INTTC3	(TC3 interrupt)				
External	$\overline{INT5}$	(External interrupt 5)	$IMF \cdot EF_{15} = 1$	IL_{15}	$FFE0_H$	Low 16
Internal	INTTC5	(TC5 interrupt)				

Note 1: The following interrupt factors share their interrupt source; the factor is selected on the register INTSEL.

1. INTRXD and INTSIO0 share the source whose priority is 10.
2. INTTXD and INTSIO1 share the source whose priority is 11.
3. INT3 and INTTC3 share the source whose priority is 15.
4. $\overline{INT5}$ and INTTC5 share the source whose priority is 16.

Note 2: 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 2.5 “Watchdog Timer”.

Note 3: 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 2.5.5 “Address Trap”.

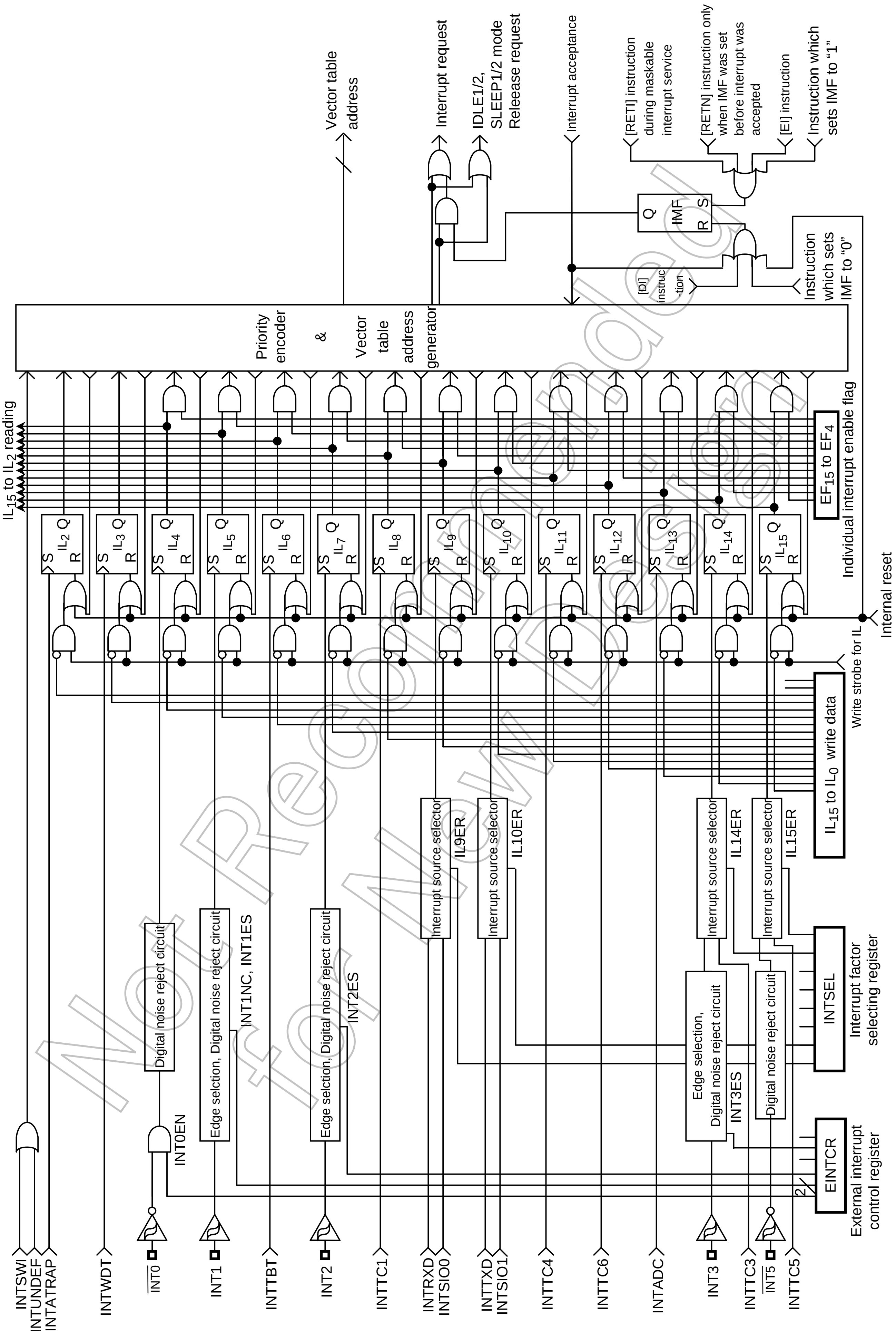


Figure 1.5.1 Interrupt Controller Block Diagram

(1) Interrupt latches (IL_{15} 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 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
            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 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 003AH and 003BH in SFR area, and they can be read and written by 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_{15} 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_{15} to EF_4) are located on EIRL to EIRH (Address: 003AH to 003BH in SFR), and can be read and written by an instruction. During reset, all the individual interrupt enable flags (EF_{15} 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"
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) (bits 15–8), ILL (003CH) (bits 7–0)

(Initial value: 00000000 000000**)

IL_{15} 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) (bits 15–8), EIRL (003AH) (bits 7–0)

(Initial value: 00000000 00000***0)

EF_{15} 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.	R/W

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)

(3) Selecting interrupt factor (INTSEL)

Each interrupt factor, that shares its interrupt source with other factors, enables its interrupt latch (IL) only if it is selected on INTSEL. The interrupt controller does not hold the interrupt request, while the factor generates the interrupt request is not selected on INTSEL. Therefore, set INTSEL appropriately before interrupt factors arises.

Interrupt source selector

INTSEL ($003E_H$)

7	6	5	4	3	2	1	0
–	IL9ER	IL10ER	–	–	–	IL14ER	IL15ER

(Initial value: *00* **00)

IL9ER	Alternative of INTRXD or INTSIO0	0: INTRXD 1: INTSIO0	
IL10ER	Alternative of INTTXD or INTSIO1	0: INTTXD 1: INTSIO1	R/W
IL14ER	Alternative of INT3 or INTTC3	0: INT3 1: INTTC3	R/W
IL15ER	Alternative of $\overline{INT5}$ or INTTC5	0: $\overline{INT5}$ 1: INTTC5	R/W

Figure 1.5.3 Interrupt Source Selector

1.5.1 Interrupt Sequence

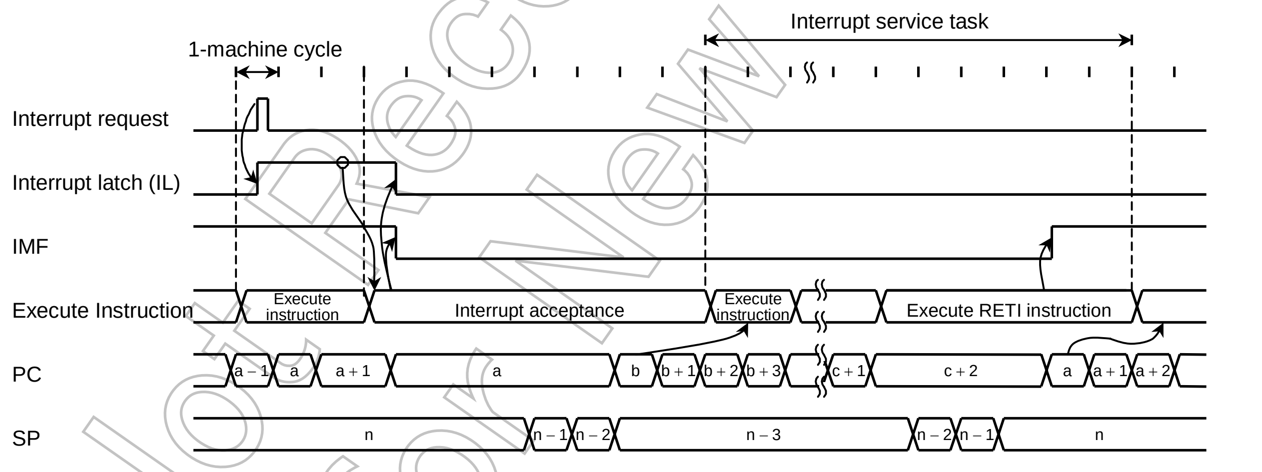
An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8 machine cycles (4 μs at 8.0 MHz) after the completion of the current instruction. The interrupt service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

Figure 1.5.4 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.

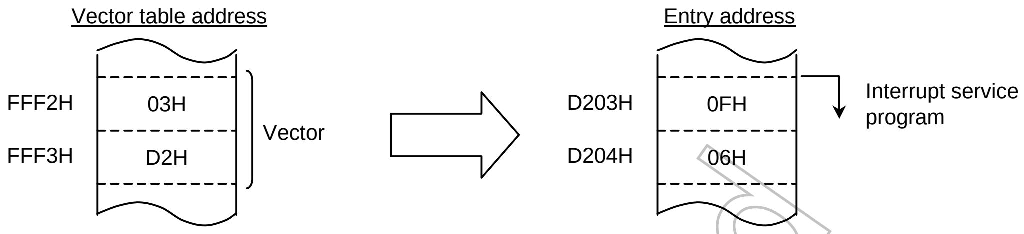


Note 1: a: Return address entry address, b:Entry address, c: Address which RETI instructrion is stored

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 1.5.4 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



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

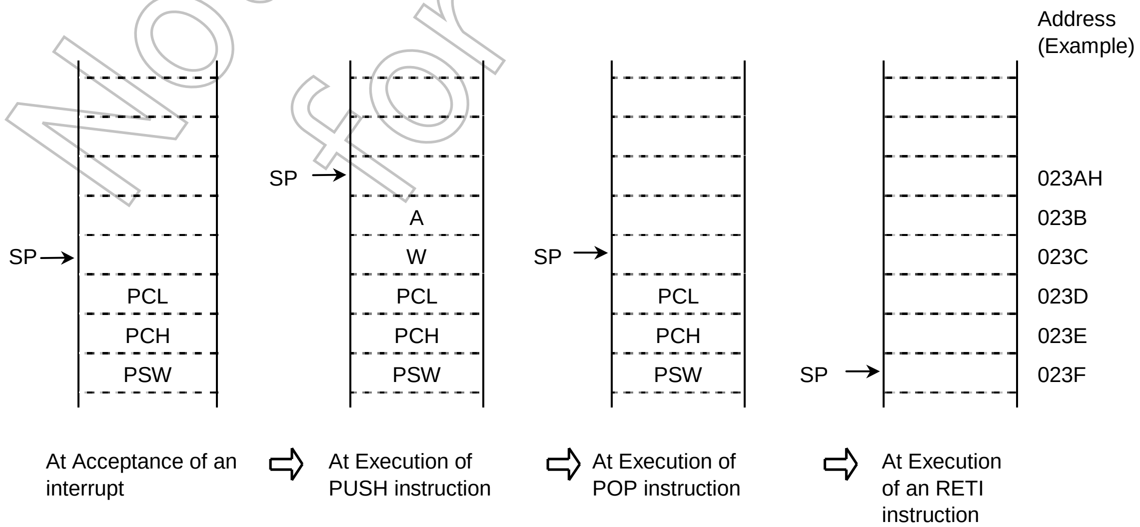
During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary 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
```

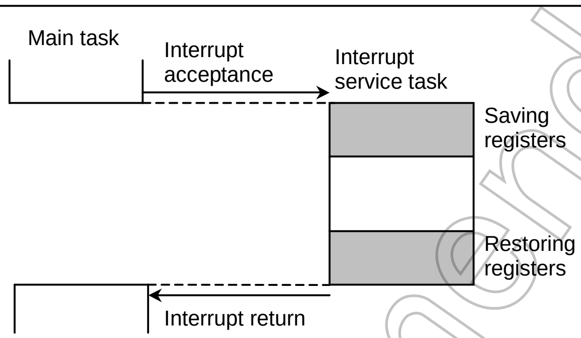


b. Using data transfer instructions

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 1.5.5 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

FF_H 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 FF_H is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FF_H to unused areas of the program memory. Address-trap reset is generated 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 TMP86CM25A 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. $\overline{\text{INT0}}$/P63 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 $\overline{\text{INT0}}$/P63 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	$\overline{\text{INT0}}$	P63/AIN3	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	MUL4 (Note 4)	P12/SEG54 or P34/COM5	IMF・EF_5 = 1		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	MUL5 (Note 4)	P13/SEG55 or P35/COM6	IMF・EF_7 = 1	Falling edge or Rising edge	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	MUL6 (Note 4)	P14/SEG56 or P36/COM7	IMF・EF_{14} = 1 IL14ER=0		
INT5	$\overline{\text{INT5}}$	P20/ $\overline{\text{STOP}}$	IMF・EF_{15} = 1 IL15ER=0	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 $\overline{\text{INT0}}$ 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: MUL4 to MUL6 it interrupt input can be changed by the MULSEL register.

Note 5: The maximum time from modifying INT1NC until a noise reject time is changed is 2^6/fc

External Interrupt Control Register

EINTCR (0037H)	7	6	5	4	3	2	1	0	
	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	R/W
INT0EN	P63/ $\overline{\text{INT0}}$ pin configuration	0: P63 input/output port 1: $\overline{\text{INT0}}$ pin (Port P63 should be set to an input mode)	R/W
INT3ES INT2ES INT1ES	INT3 to INT1 edge select	0: Rising edge 1: Falling edge	R/W

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 1.5.6 External Interrupt Control Register

1.6 Reset Circuit

The TMP86CM25A 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).

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) when power is turned on.

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.

Note: The Flash reset function is included in mask-ROM "A" version (TMP86CM25A) for keeping the compatibility with Flash products (TMP86FM25).

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 $\overline{RESET}$ pin contains a Schmitt trigger (Hysteresis) with an internal pull-up resistor. When the $\overline{RESET}$ 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 $\overline{RESET}$ pin input goes high, the reset operation is released and the program execution starts at the vector address stored at addresses FFFEH to FFFFH.

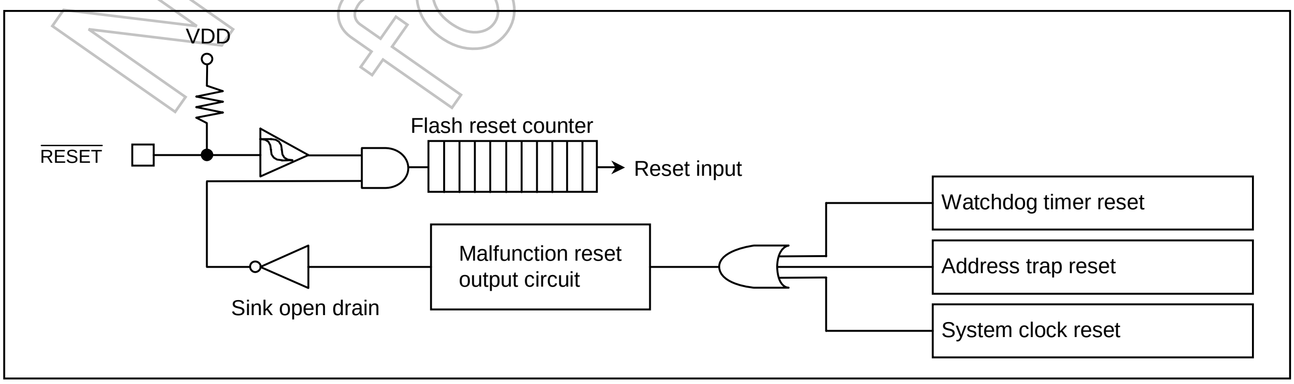
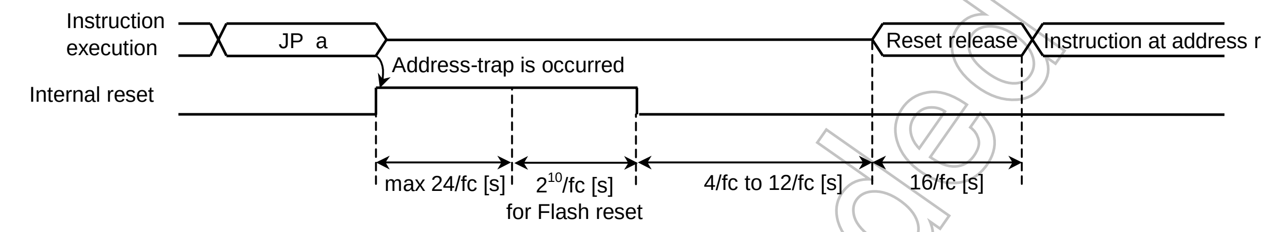


Figure 1.6.1 Reset Circuit

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”), DBR 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).



Note 1: Address “a” is in the SFR or on-chip RAM (WDTCR1<ATAS> = “1”) space.

Note 2: During reset release, reset vector “r” is read out, and an instruction at address “r” is fetched and decoded.

Note 3: The Flash reset function is included in mask-ROM “A” version (TMP86CM25AF) for keeping the compatibility with Flash products.

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 2.5 “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 TMP86CM25A adopts the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function register (SFR) or the data buffer register (DBR). The SFR is mapped on address 0000_H to $003F_H$, DBR is mapped on address $0F00_H$ to $0FFF_H$.

Figure 2.1.1 to Figure 2.1.2 indicate the special function register (SFR) and data buffer register (DBR) for TMP86CM25A.

Address	Read	Write	Address	Read	Write
0000_H	Reserved		0020_H	ADCDR1 (AD result register 1)	–
01	P1DR (P1 port output latch)		21	ADCDR2 (AD result register 2)	–
02	P2DR (P2 port output latch)		22	Reserved	
03	P3DR (P3 port output latch)		23	Reserved	
04	P3LCR (P3 segment/common output control)		24	Reserved	
05	P5DR (P5 port output latch)		25	UARTSR (UART status register)	UARTCR1 (UART control register 1)
06	P6DR (P6 port output latch)		26	–	UARTCR2 (UART control register 2)
07	P7DR (P7 port output latch)		27	LCDCTL1 (LCD control register 1)	
08	P1PRD (P1 terminal input)	–	28	LCDCTL2 (LCD control register 2)	
09	P2PRD (P2 terminal input)	–	29	P1LCR (P1 segment output control)	
0A	P3PRD (P3 terminal input)	–	2A	P5LCR (P5 segment output control)	
0B	P5PRD (P5 terminal input)	–	2B	P7LCR (P7 segment output control)	
0C	P6CR (P6 port input/output control)		2C	PWREG3 (Timer register 3)	
0D	P7PRD (P7 terminal input)	–	2D	PWREG4 (Timer register 4)	
0E	ADCCR1 (AD control register 1)		2E	PWREG5 (Timer register 5)	
0F	ADCCR2 (AD control register 2)		2F	PWREG6 (Timer register 6)	
10	TREG1AL (Timer register 1A low)		30	Reserved	
11	TREG1AM (Timer register 1A middle)		31	Reserved	
12	TREG1AH (Timer register 1A high)		32	Reserved	
13	TREG1B (Timer register 1B)		33	Reserved	
14	TC1CR1 (Timer counter 1 control 1)		34	–	WDTCR1 (Watchdog timer control)
15	TC1CR2 (Timer counter 1 control 2)		35	–	WDTCR2 (Watchdog timer control)
16	TC1SR (TC1 status)	–	36	TBTCR (TBT/TG/DVO control)	
17	Reserved		37	EINTCR (External interrupt control)	
18	TC3CR (Timer counter 3 control)		38	SYSCR1 (System control 1)	
19	TC4CR (Timer counter 4 control)		39	SYSCR2 (System control 2)	
1A	TC5CR (Timer counter 5 control)		3A	EIR_L (Interrupt enable register)	
1B	TC6CR (Timer counter 6 control)		3B	EIR_H (Interrupt enable register)	
1C	TTREG3 (Timer register 3)		3C	IL_L (Interrupt latch)	
1D	TTREG4 (Timer register 4)		3D	IL_H (Interrupt latch)	
1E	TTREG5 (Timer register 5)		3E	INTSEL (Interrupt sorce selector)	
1F	TTREG6 (Timer register 6)		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 TMP86CM25A

LCD Data Buffer (Write/Read)

0F00H	0F10H	0F20H	0F30H	0F40H	0F50H	0F60H	0F70H	COM0
0F01H	0F11H	0F21H	0F31H	0F41H	0F51H	0F61H	0F71H	COM1
0F02H	0F12H	0F22H	0F32H	0F42H	0F52H	0F62H	0F72H	COM2
0F03H	0F13H	0F23H	0F33H	0F43H	0F53H	0F63H	0F73H	COM3
0F04H	0F14H	0F24H	0F34H	0F44H	0F54H	0F64H	0F74H	COM4
0F05H	0F15H	0F25H	0F35H	0F45H	0F55H	0F65H	0F75H	COM5
0F06H	0F16H	0F26H	0F36H	0F46H	0F56H	0F66H	0F76H	COM6
0F07H	0F17H	0F27H	0F37H	0F47H	0F57H	0F67H	0F77H	COM7
0F08H	0F18H	0F28H	0F38H	0F48H	0F58H	0F68H	0F78H	COM8
0F09H	0F19H	0F29H	0F39H	0F49H	0F59H	0F69H	0F79H	COM9
0F0AH	0F1AH	0F2AH	0F3AH	0F4AH	0F5AH	0F6AH	0F7AH	COM10
0F0BH	0F1BH	0F2BH	0F3BH	0F4BH	0F5BH	0F6BH	0F7BH	COM11
0F0CH	0F1CH	0F2CH	0F3CH	0F4CH	0F5CH	0F6CH	0F7CH	COM12
0F0DH	0F1DH	0F2DH	0F3DH	0F4DH	0F5DH	0F6DH	0F7DH	COM13
0F0EH	0F1EH	0F2EH	0F3EH	0F4EH	0F5EH	0F6EH	0F7EH	COM14
0F0FH	0F1FH	0F2FH	0F3FH	0F4FH	0F5FH	0F6FH	0F7FH	COM15
SEG7 to SEG0	SEG15 to SEG8	SEG23 to SEG16	SEG31 to SEG24	SEG39 to SEG32	SEG47 to SEG40	SEG55 to SEG48	SEG59 to SEG56	

<table>
<tr><th>Address</th><th>Read</th><th>Write</th></tr>
<tr><td>0F90H</td><td colspan="2">SIO0BR0 (SIO0 buffer 0)</td></tr>
<tr><td>91</td><td colspan="2">SIO0BR1 (SIO0 buffer 1)</td></tr>
<tr><td>92</td><td colspan="2">SIO0BR2 (SIO0 buffer 2)</td></tr>
<tr><td>93</td><td colspan="2">SIO0BR3 (SIO0 buffer 3)</td></tr>
<tr><td>94</td><td colspan="2">SIO0BR4 (SIO0 buffer 4)</td></tr>
<tr><td>95</td><td colspan="2">SIO0BR5 (SIO0 buffer 5)</td></tr>
<tr><td>96</td><td colspan="2">SIO0BR6 (SIO0 buffer 6)</td></tr>
<tr><td>97</td><td colspan="2">SIO0BR7 (SIO0 buffer 7)</td></tr>
<tr><td>98</td><td>–</td><td>SIO0CR1 (SIO0 control register 1)</td></tr>
<tr><td>99</td><td>SIO0SR (SIO0 status register)</td><td>SIO0CR2 (SIO0 control register 2)</td></tr>
<tr><td>9A</td><td>–</td><td>STOPCR (Key-on wakeup control register)</td></tr>
<tr><td>9B</td><td>RDBUF (UART received data buffer)</td><td>TDBUF (UART transmit data buffer)</td></tr>
<tr><td>9C</td><td colspan="2">Reserved</td></tr>
<tr><td>:</td><td colspan="2">:</td></tr>
<tr><td>9F</td><td colspan="2">Reserved</td></tr>
<tr><td>A0</td><td colspan="2">SIO1BR0 (SIO1 buffer 0)</td></tr>
<tr><td>A1</td><td colspan="2">SIO1BR1 (SIO1 buffer 1)</td></tr>
<tr><td>A2</td><td colspan="2">SIO1BR2 (SIO1 buffer 2)</td></tr>
<tr><td>A3</td><td colspan="2">SIO1BR3 (SIO1 buffer 3)</td></tr>
<tr><td>A4</td><td colspan="2">SIO1BR4 (SIO1 buffer 4)</td></tr>
<tr><td>A5</td><td colspan="2">SIO1BR5 (SIO1 buffer 5)</td></tr>
<tr><td>A6</td><td colspan="2">SIO1BR6 (SIO1 buffer 6)</td></tr>
<tr><td>A7</td><td colspan="2">SIO1BR7 (SIO1 buffer 7)</td></tr>
<tr><td>A8</td><td>–</td><td>SIO1CR1 (SIO1 control register 1)</td></tr>
<tr><td>A9</td><td>SIO1SR (SIO1 status register)</td><td>SIO1CR2 (SIO1 control register 2)</td></tr>
<tr><td>AA</td><td colspan="2">Reserved</td></tr>
<tr><td>:</td><td colspan="2">:</td></tr>
<tr><td>BF</td><td colspan="2">Reserved</td></tr>
<tr><td>C0</td><td colspan="2">MULSEL (Multiplexed function select register)</td></tr>
<tr><td>C1</td><td colspan="2">Reserved</td></tr>
<tr><td>:</td><td colspan="2">:</td></tr>
<tr><td>DF</td><td colspan="2">Reserved</td></tr>
<tr><td>E0</td><td colspan="2">EEPCR (Flash memory control) (Note 4)</td></tr>
<tr><td>E1</td><td>EEPSR (Flash memory status) (Note 4)</td><td>–</td></tr>
<tr><td>E2</td><td colspan="2">Reserved</td></tr>
<tr><td>:</td><td colspan="2">:</td></tr>
<tr><td>FF</td><td colspan="2">Reserved</td></tr>
</table>

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.).

Note 4: EEPCR and EEPSR are registers for Flash product. For details refer to 1.1 “Flash Memory” in TMP86FM25F data sheet.

Figure 2.1.2 The Data Buffer Register (DBR) for TMP86CM25A

2.2 I/O Ports

The TMP86CM25A has 6 parallel input/output ports (42 pins) as follows.

	Primary Function	Secondary Functions
Port P1	8-bit I/O port	External interrupt input, serial interface input/output, UART input/output and segment output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	7-bit I/O port	Timer/counter input/output, divider output and segment/common output.
Port P5	8-bit I/O port	Segment output.
Port P6	8-bit I/O port	Analog input, external interrupt input, timer/counter input and STOP mode release signal input.
Port P7	8-bit I/O port	Timer/counter input/output, divider output and common 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.

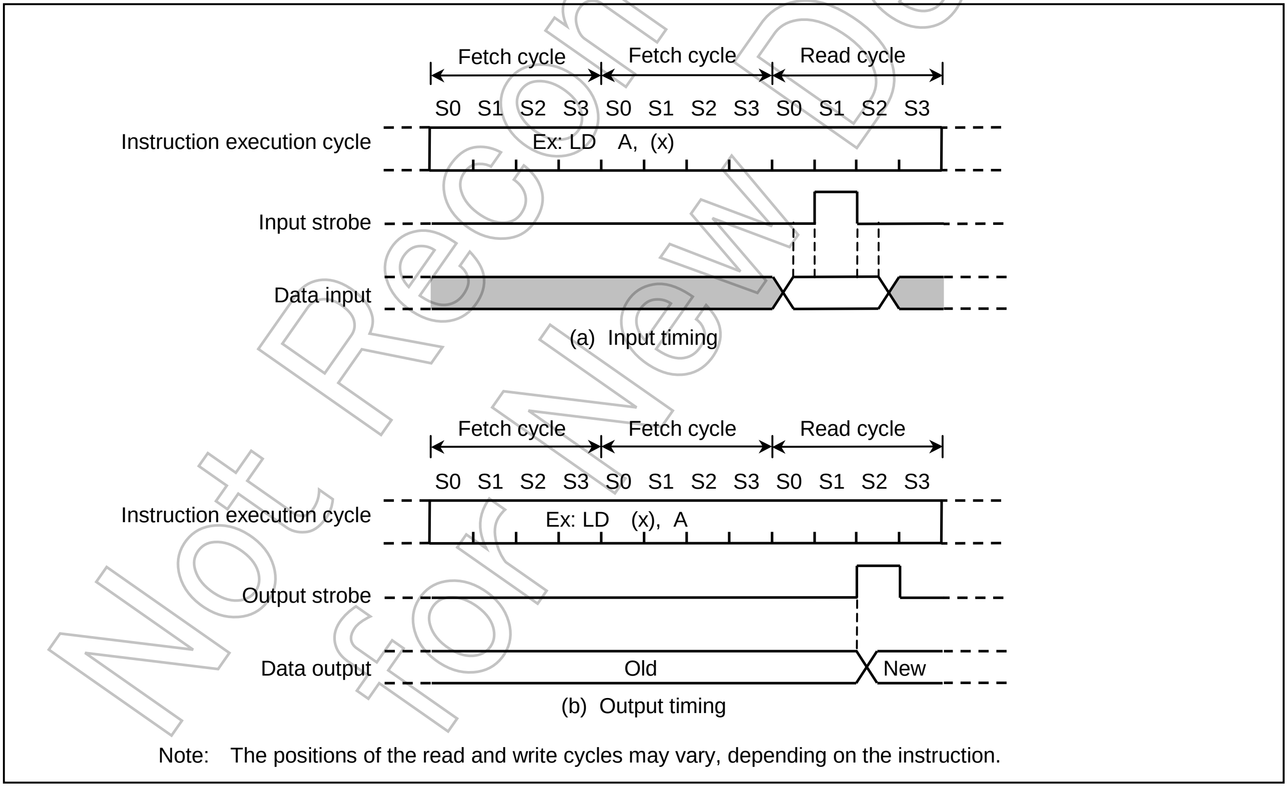


Figure 2.2.1 Input/Output Timing (Example)

2.2.1 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port which is also used as an external interrupt input, serial interface input/output, UART input/output and segment output of LCD.

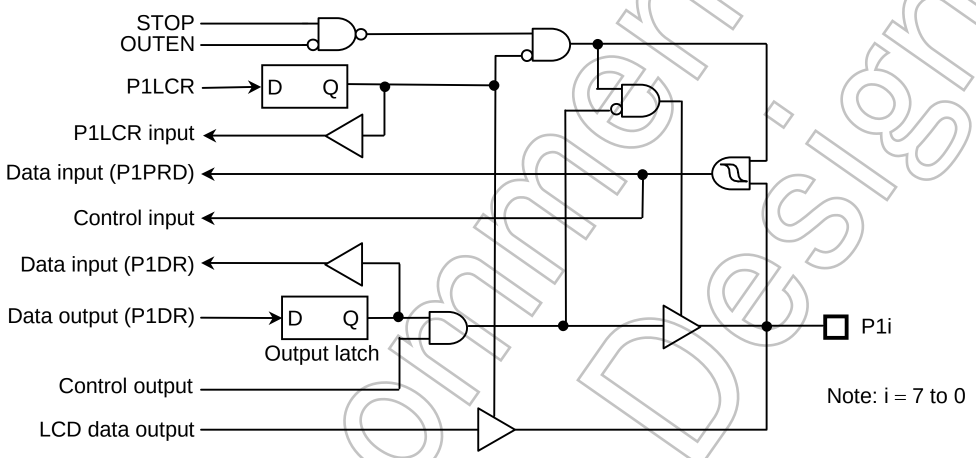
When used as segment pins of LCD, the respective bit of P1LCR should be set to “1”.

When used as an input port or a secondary function pin (external interrupt input, serial interface input/output or UART input/output) except for segment, the respective output latch (P1DR) should be set to “1” and its corresponding P1LCR bit should be cleared to “0”.

When used as an output port, the corresponding P1LCR bit should be cleared to “0”.

During reset, the P1DR is initialized to “1” and P1LCR 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. The P1PRD data of pin which is set as a segment output pin is always “0”.



P1DR (0001H) R/W

7	6	5	4	3	2	1	0
P17 SEG59 $\overline{SCK0}$	P16 SEG58 TXD SO0	P15 SEG57 RXD SI0	P14 SEG56 MUL6	P13 SEG55 MUL5	P12 SEG54 MUL4	P11 SEG53	P10 SEG52

(Initial value: 1111 1111)

P1LCR (0029H)

(Initial value: 0000 0000)

P1LCR	Port P1/segment output select (Set for each bit individually)	0: P1 input/output port or secondary function (except for segment) 1: Segment output	R/W

P1PRD (0008H) Read only

P17	P16	P15	P14	P13	P12	P11	P10

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are as follow:

MUL0: $\overline{DVO}$
MUL1: $\overline{PWM3}$, $\overline{PDO3}$, TC3
MUL2: $\overline{PPG4}$, $\overline{PWM4}$, $\overline{PDO4}$, TC4
MUL3: $\overline{PPG6}$, $\overline{PWM6}$, $\overline{PDO6}$, TC6
MUL4: INT1
MUL5: INT2
MUL6: INT3

Figure 2.2.2 Port 1

2.2.2 Port P2 (P22 to P20)

Port P2 is a 3-bit input/output port.

It is also used as an external interrupt, a STOP mode release signal input, and low-frequency crystal oscillator connection pins. When used as an input port or a secondary function pins, respective output latch (P2DR) should be set to “1”.

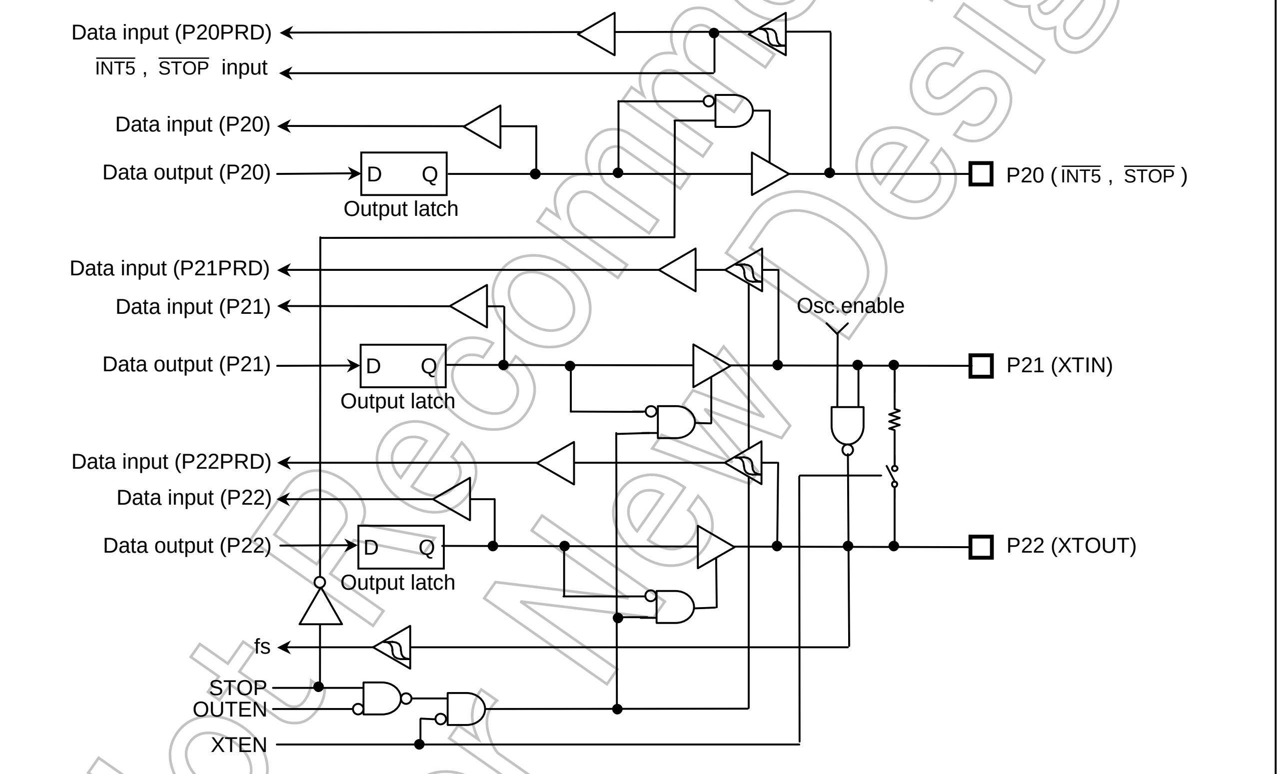
During reset, the P2DR is initialized to “1”.

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 P2, read data of bits 7 to 3 are unstable.



	7	6	5	4	3	2	1	0	
P2DR (0002_H) R/W						P22 XTOUT	P21 XTIN	P20 INT5 STOP	(Initial value: **** *111) *: Don’t care

	7	6	5	4	3	2	1	0
P2PRD (0009_H) Read only						P22	P21	P20

Note 1: When XTEN<SYSCR2> is set to “1”, P21 and P22 become a high impedance state.

Note 2: Port P20 is used as STOP pin. Therefore, when stop mode is started, OUTEN does not affect to P20, and P20 becomes High-Z state.

Figure 2.2.3 Port 2

2.2.3 Port P3 (P36 to P30)

Port P3 is a 7-bit input/output port which is also used as an external interrupt input, timer/counter input/output, divider output and segment/common output of LCD.

When used as segment/common pins of LCD, the respective bit of P3LCR should be set to "1".

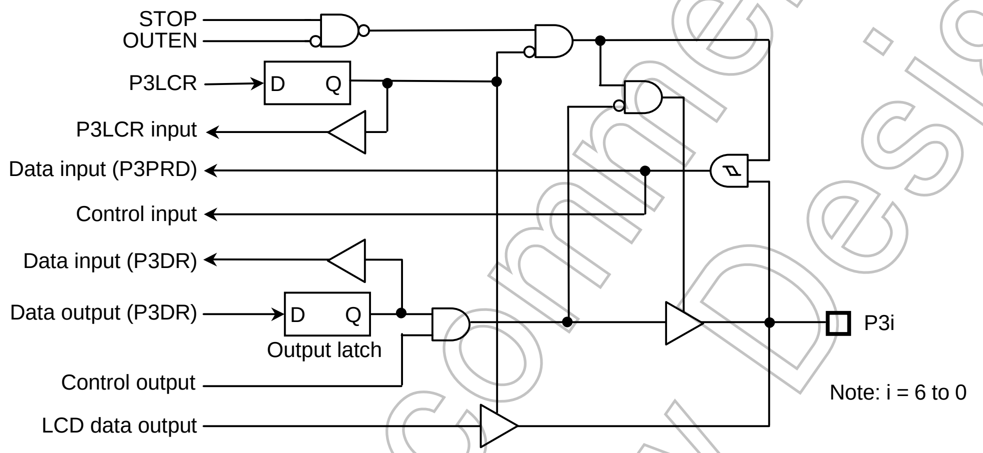
When used as an input port or a secondary function pin (External interrupt input, timer/counter input/output) except for segment/common, the respective output latch (P3DR) should be set to "1" and its corresponding P3LCR bit should be cleared to "0".

When used as an output port, the corresponding P3LCR bit should be cleared to "0".

During reset, the P3DR is initialized to "1" and P3LCR 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. The P3PRD data of pin which is set as a segment output pin is always "0".

If a read instruction is executed for port P3DR and P3LCR, read data of bit7 is unstable.



	7	6	5	4	3	2	1	0	
P3DR (0003_H) R/W		P36 COM7 MUL6	P35 COM6 MUL5	P34 COM5 MUL4	P33 SEG51 MUL3	P32 SEG50 MUL2	P31 SEG49 MUL1	P30 SEG48 MUL0	(Initial value: *111 1111)
P3LCR (0004_H)									(Initial value: *000 0000)

P3LCR	Port P3/segment output select (Set for each bit individually)	0: P3 input/output port or secondary function (except for segment) 1: Segment output/common output	R/W

	7	6	5	4	3	2	1	0
P3PRD ($000A_H$) Read only		P36	P35	P34	P33	P32	P31	P30

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are as follow:

MUL0: $\overline{\text{DVO}}$
MUL1: $\overline{\text{PWM3}}$, $\overline{\text{PDO3}}$, TC3
MUL2: $\overline{\text{PPG4}}$, $\overline{\text{PWM4}}$, $\overline{\text{PDO4}}$, TC4
MUL3: $\overline{\text{PPG6}}$, $\overline{\text{PWM6}}$, $\overline{\text{PDO6}}$, TC6
MUL4: INT1
MUL5: INT2
MUL6: INT3

Figure 2.2.4 Port 3

2.2.4 Port P5 (P57 to P50)

Port P5 is an 8-bit input/output port which is also used as a segment output of LCD.

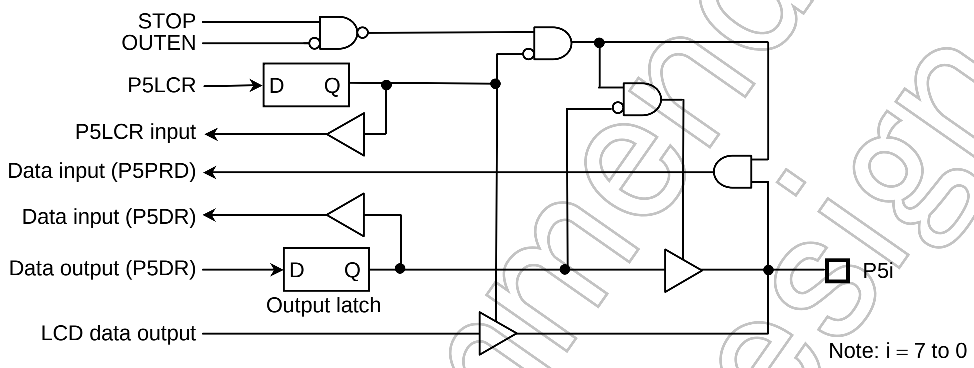
When used as segment pins of LCD, the respective bit of P5LCR should be set to “1”.

When used as an input port, the respective output latch (P5DR) should be set to “1” and its corresponding P5LCR bit should be cleared to “0”.

When used as an output port, the corresponding P5LCR bit should be cleared to “0”.

During reset, the P5DR is initialized to “1” and P5LCR 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. The P5PRD data of pin which is set as a segment output pin is always “0”.



P5DR (0005_H) R/W

7	6	5	4	3	2	1	0	
P57 SEG47	P56 SEG46	P55 SEG45	P54 SEG44	P53 SEG43	P52 SEG42	P51 SEG41	P50 SEG40	(Initial value: 1111 1111)

P5LCR ($002A_H$)

								(Initial value: 0000 0000)

P5LCR	Port P5/segment output select (Set for each bit individually)	0: P5 input/output port or secondary function (except for segment) 1: Segment output	R/W

P5PRD ($000B_H$) Read only

P57	P56	P55	P54	P53	P52	P51	P50

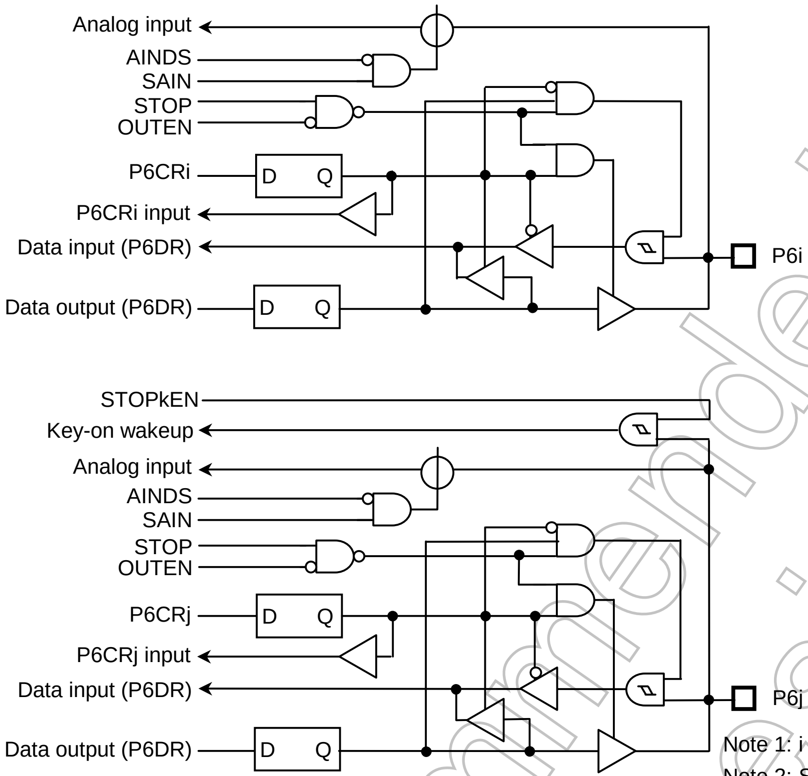
Figure 2.2.5 Port 5

2.2.5 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, key-on wakeup input, timer/counter input and external interrupt input. Input/output mode is specified by the P6 control register (P6CR) and the P6 output latch (P6DR). During reset, P6CR and P6DR are initialized to “0”. At the same time, the input data of pins P67 to P60 are fixed to “0”. To use port P6 as an input port, external interrupt input, timer/counter input or key-on wakeup input, set data of P6DR to “1” and clear P6CR to “0”. To use it as an output port, set data of P6CR to “1”. To use it as an analog input, clear data of P6DR to “0” and P6CR to “0”, and start the AD. It is the penetration electric current measures by the analog voltage.

Pins not used for analog input can be used as I/O ports. During AD conversion, output instructions should not be executed to keep a precision. In addition, a variable signal should not be input to a port adjacent to the analog input during AD conversion.

When the AD converter is in use (P6DR = 0), bits mentioned above are read as “0” by executing input instructions.



Note 1: i = 0 to 3, j = 7 to 4, k = 5 to 2.
Note 2: SAIN is bits 0 to 3 in ADCCR1.
Note 3: STOPkEN is bits 7 to 4 in STOPCR.

	7	6	5	4	3	2	1	0	
P6DR (0006$_H$) R/W	P67 AIN7 STOP5	P66 AIN6 STOP4	P65 AIN5 STOP3	P64 AIN4 STOP2	P63 AIN3 $\overline{\text{INT0}}$	P62 AIN2 ECNT	P61 AIN1 ECIN	P60 AIN0	(Initial value: 0000 0000)
P6CR (000C$_H$)									(Initial value: 0000 0000)

P6CR	Port P6 I/O control (Set for each bit individually)		AINDS = 1 (AD unused)		AINDS = 0 (AD used)		R/W
			P6DR = "0"	P6DR = "1"	P6DR = "0"	P6DR = "1"	
		0:	Input "0" fixed	Input mode	AD input	Input mode	
		1:	Output mode		Output mode		

Note 1: Do not set output mode to pin which is used for an analog input.

Note 2: When used as an $\overline{\text{INT0}}$, ECNT and ECIN pins of a secondary function, the respective bit of P6CR should be cleared to "0" and the P6DR should set to "1".

Note 3: When used as STOP2 to STOP5 pins of key-on wakeup, the respective bit of P6CR should be cleared to "0".

Note 4: When a read instruction for port P6 is executed, the bit of analog input mode becomes read data "0".

Note 5: Although P6DR is a read/write register, because it is also used as an input mode control function, read-modify-write instructions such as bit manipulate instructions cannot be used. Read-modify-write instruction writes the all data of 8-bit after data is read and modified. Because a bit setting input mode read of terminal, the output latch is changed by these instruction. So P6 port can not input data.

Figure 2.2.6 Port 6 and P6CR

2.2.6 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which is also used as an external interrupt input, serial interface input/output, timer/counter input/output, divider output and common output of LCD.

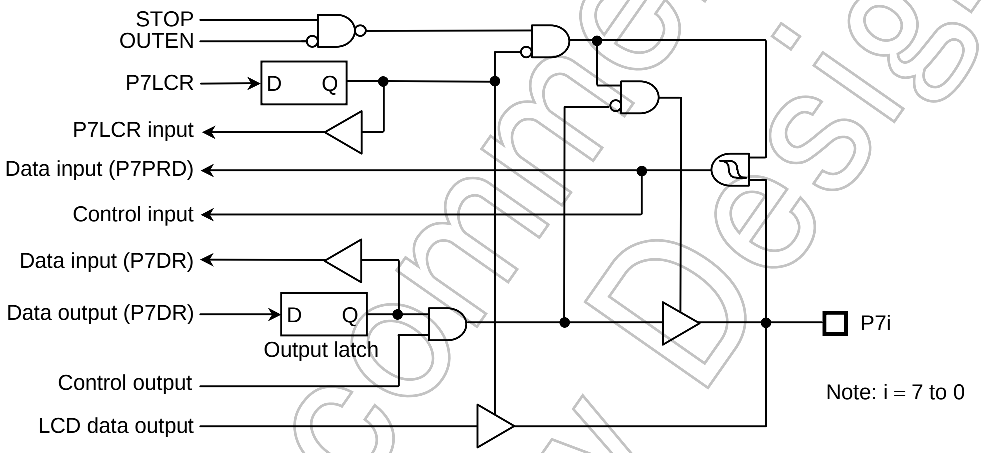
When used as common pins of LCD, the respective bit of P7LCR should be set to “1”.

When used as an input port or a secondary function pin (external interrupt input, serial interface input/output, timer/counter input/output or divider output) except for common, the respective output latch (P7DR) should be set to “1” and its corresponding P7LCR bit should be cleared to “0”.

When used as an output port, the corresponding P7LCR bit should be cleared to “0”.

During reset, the P7DR is initialized to “1” and P7LCR is initialized to “0”.

P7 port output latch (P7DR) and P7 port terminal input (P7PRD) are located on their respective address. When read the output latch data, the P7DR should be read and when read the terminal input data, the P7PRD register should be read. The P7PRD data of pin which is set as a segment output pin is always “0”.



P7DR (0007$_H$) R/W

7	6	5	4	3	2	1	0
P77 COM15 $\overline{\text{SCK1}}$	P76 COM14 SO1	P75 COM13 SI1	P74 COM12 MUL3	P73 COM11 MUL2	P72 COM10 MUL1	P71 COM9 MUL0	P70 COM8

(Initial value: 1111 1111)

P7LCR (002B$_H$)

(Initial value: 0000 0000)

P7LCR	Port P7/common output select (Set for each bit individually)	0: P7 input/output port or secondary function (except for common) 1: Common output	R/W

P7PRD (000D$_H$) Read only

P77	P76	P75	P74	P73	P72	P71	P70

Note: With ports assigned as MUL6 to MUL0, assigned pins can be switched by the multi function register (MULSEL). The assigned functions are as follow:

MUL0: $\overline{\text{DVO}}$
MUL1: $\overline{\text{PWM3}}$, $\overline{\text{PDO3}}$, TC3
MUL2: $\overline{\text{PPG4}}$, $\overline{\text{PWM4}}$, $\overline{\text{PDO4}}$, TC4
MUL3: $\overline{\text{PPG6}}$, $\overline{\text{PWM6}}$, $\overline{\text{PDO6}}$, TC6
MUL4: INT1
MUL5: INT2
MUL6: INT3

Figure 2.2.7 Port 7

2.3 Multi Function Register

With function pins assigned as MUL6 to MUL0, the port to be used can be switched by MULSEL.

MULSEL ($0FC0_H$)

	MUL6	MUL5	MUL4	MUL3	MUL2	MUL1	MUL0

(Initial value: 0000 0000)

MUL6	INT3 function pin select	0: P14 1: P36	R/W
MUL5	INT2 function pin select	0: P13 1: P35	
MUL4	INT1 function pin select	0: P12 1: P34	
MUL3	$\overline{PPG6}$, $\overline{PWM6}$, $\overline{PDO6}$, TC6 function pin select	0: P33 1: P74	
MUL2	$\overline{PPG4}$, $\overline{PWM4}$, $\overline{PDO4}$, TC4 function pin select	0: P32 1: P73	
MUL1	$\overline{PPG3}$, $\overline{PWM3}$, TC3 function pin select	0: P31 1: P72	
MUL0	$\overline{DVO}$ function pin select	0: P30 1: P71	

Figure 2.3.1 Multi Function Register

2.4 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 2.4.1 (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.

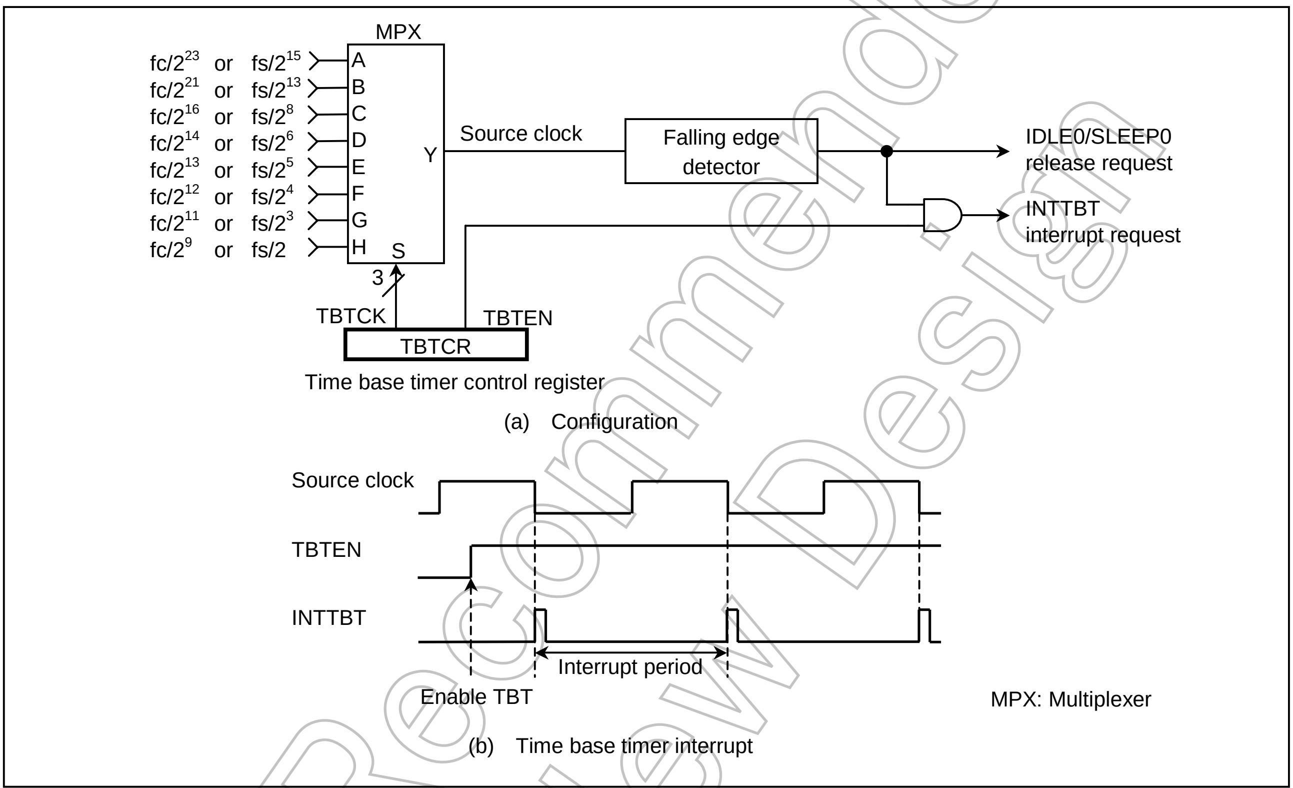


Figure 2.4.1 Time Base Timer

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 (0036H)

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

(Initial value: 0000 0000)

TBTEN	Time base timer enable/disable	0: Disable 1: Enable				R/W
TBTCK	Time base timer interrupt frequency select [Hz]		NORMAL1/2, IDLE1/2 mode		SLOW, SLEEP mode	
			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 2.4.2 Time Base Timer Control Register

Table 2.4.1 Time Base Timer Interrupt Frequency (Example: fc = 16 MHz, fs = 32.768 kHz)

TBTCK	Time Base Timer Interrupt Frequency [Hz]		
	NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode
	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	–

2.5 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 generate” or a non-maskable “interrupt request”. However, selection is possible only once after reset. At first the “reset generate” 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.

2.5.1 Watchdog Timer Configuration

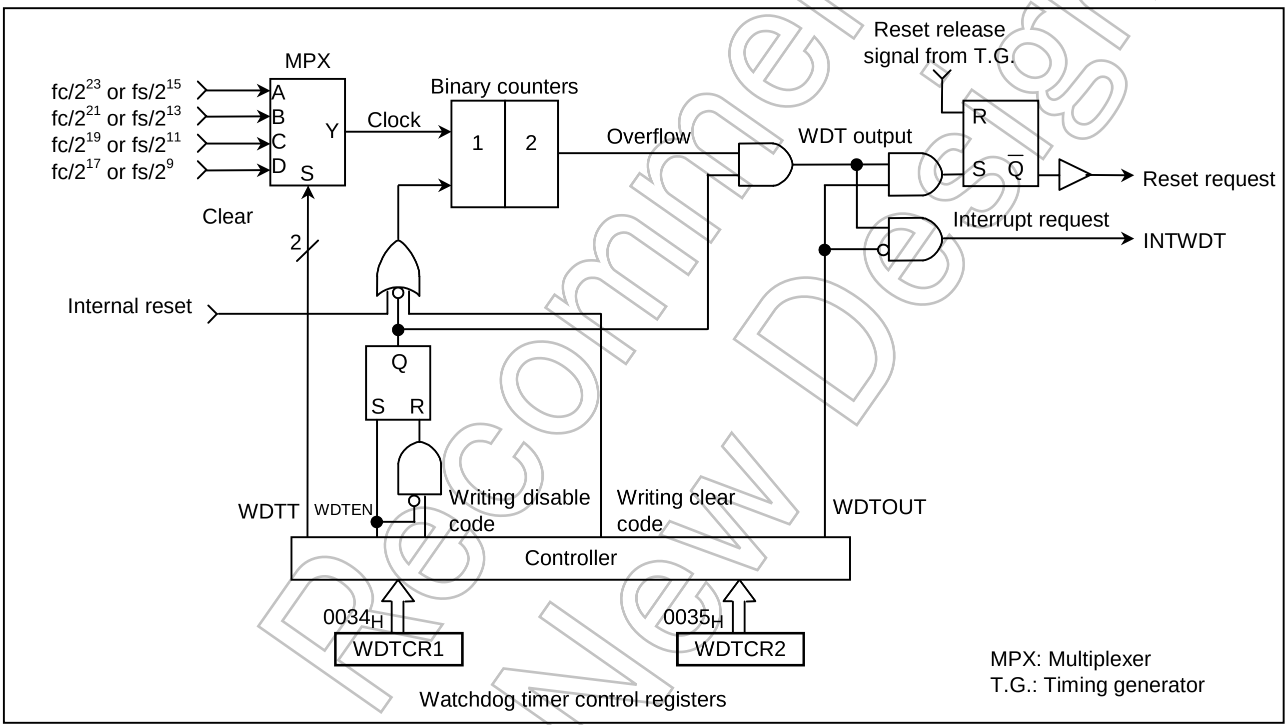


Figure 2.5.1 Watchdog Timer Configuration

2.5.2 Watchdog Timer Control

Figure 2.5.2 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 $4E_H$ is written, only the binary counter is cleared, not the internal divider. Depending on the timing at which clear code $4E_H$ is written on the WDTCR2 register, the overflow time of the binary counter may be at minimum 3/4 of the time set in WDTCR1<WDTT>. Thus, write the clear code using a shorter cycle than 3/4 of the time set in WDTCR1<WDTT>.

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

```
                      LD   (WDTCR2), 4EH          ; Clears the binary counters.
                      LD   (WDTCR1), 00001101B    ; WDTT ← 10, WDTOUT ← 1
Within 3/4 of    ┌    LD   (WDTCR2), 4EH          ; Clears the binary counters.
WDT              │     :                            (Always clear immediately before and after
detection time   │     :                            changing WDTT.)
                 └    LD   (WDTCR2), 4EH          ; Clears the binary counters.
Within 3/4 of    ┌     :
WDT              │     :
detection time   │
                 └    LD   (WDTCR2), 4EH          ; Clears the binary counters.
                       :
```

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 DV7CK = 0	NORMAL1/2 mode DV7CK = 1	SLOW mode	Write only
		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 generate				Write only

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 “$4E_H$” (Clear code) to WDTCR2 for clearing the binary counter before writing “0” to WDTEN, and then write “$B1_H$” (Disable code) to WDTCR2.
Also, immediately before these procedures, 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 $4E_H$ within 3/4 of the time set in WDTCR1<WDTT>.

Figure 2.5.2 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 “$4E_H$” (Clear code) to WDTCR2 for clearing the binary counter before writing “0” to WDTCR1<WDTEN>, and then write “$B1_H$” (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 procedures, 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 2.5.1 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

2.5.3 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
```

2.5.4 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 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).

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.

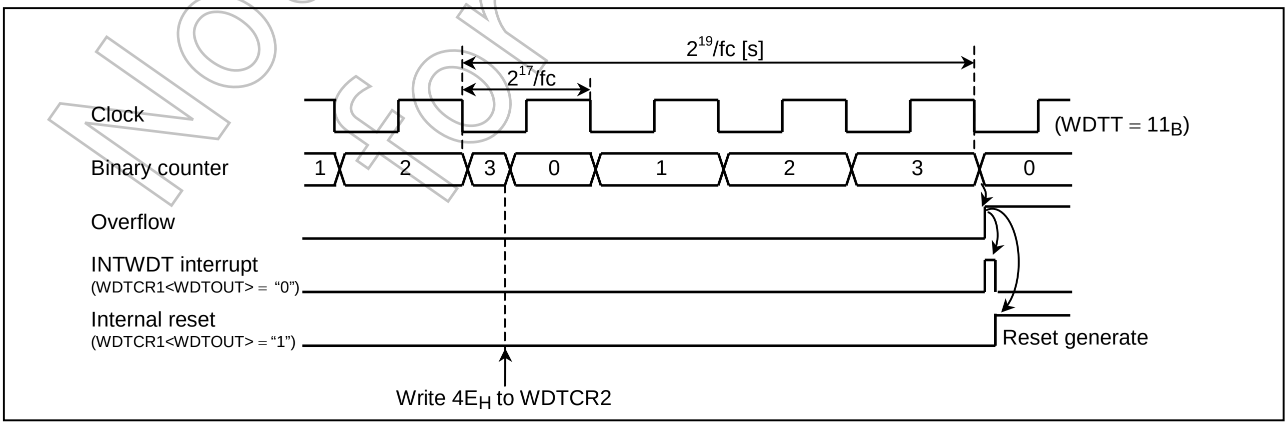


Figure 2.5.3 Watchdog Timer Interrupt/Reset

2.5.5 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 2.5.4.

Watchdog Timer Control Register 1

WDTCR1	7	6	5	4	3	2	1	0	
(0034_H)	–	–	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 $D2_H$ to WDTCR2)	Write only
ATOUT	Selection of operation at address trap	0: Interrupt 1: Reset generate	Write only

Watchdog Timer Control Register 2

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

WDTCR2	Watchdog timer control code and Address trapped area control code	$D2_H$: Address trapped area valid to set (ATRAP control code) $4E_H$: Watchdog timer binary counter clear (WDT clear code) $B1_H$: Watchdog timer disable (WDT disable code) Others: Invalid	Write only

Figure 2.5.4 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 generation or reset generate can be selected by WDTCR1<ATOUT>.

2.6 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 P30 or P71 pin. The selection of P30 or P71 is controlled by MULSEL<MUL0>. To output $\overline{\text{DVO}}$, the corresponding bit of output latch (P3DR or P7DR) 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 (0036_H)

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

(Initial value: 0000 0000)

			NORMAL1/2 Mode DV7CK = 0	NORMAL1/2 Mode DV7CK = 1	SLOW, SLEEP Mode	
DVOEN	Divider output enable/disable	0: Disable 1: Enable				R/W
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]	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

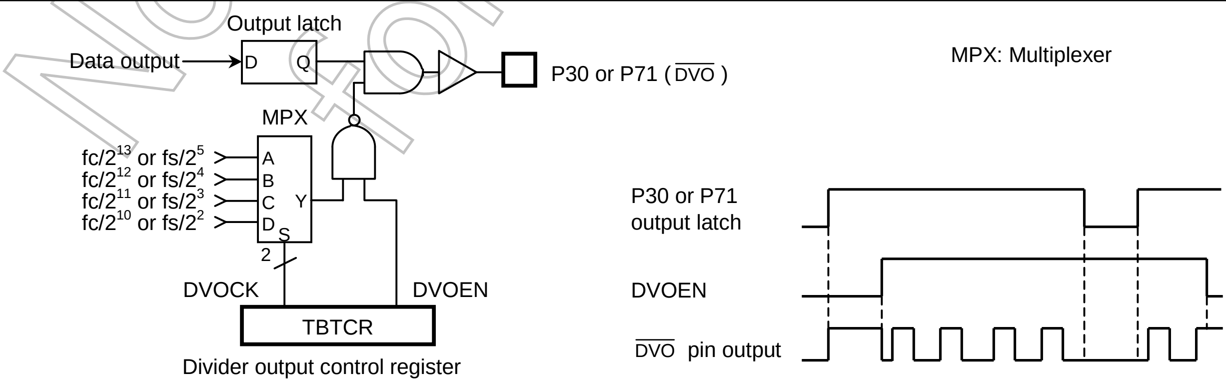
Figure 2.6.1 Divider Output Control Register

Example: 1.95 kHz pulse output (at fc = 16.0 MHz)

```
LD    (MULSEL), 00000000B   ;  DVO output from P30
SET   (P.DR).1              ;  P30 output latch ← "1"
LD    (TBTCR), 00000000B    ;  DVOCK ← "00"
LD    (TBTCR), 10000000B    ;  DVOEN ← "1"
```

Table 2.6.1 Divider Output Frequency (Example: at fc = 16.0 MHz, fs = 32.768 kHz)

DVOCK	Divider Output Frequency [Hz] NORMAL1/2, IDLE1/2 Mode DV7CK = 0	DV7CK = 1	SLOW, SLEEP Mode
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



(a) Configuration (b) Timing Chart

Figure 2.6.2 Divider Output

2.7 18-Bit Timer/Counter (TC1)

2.7.1 Configuration

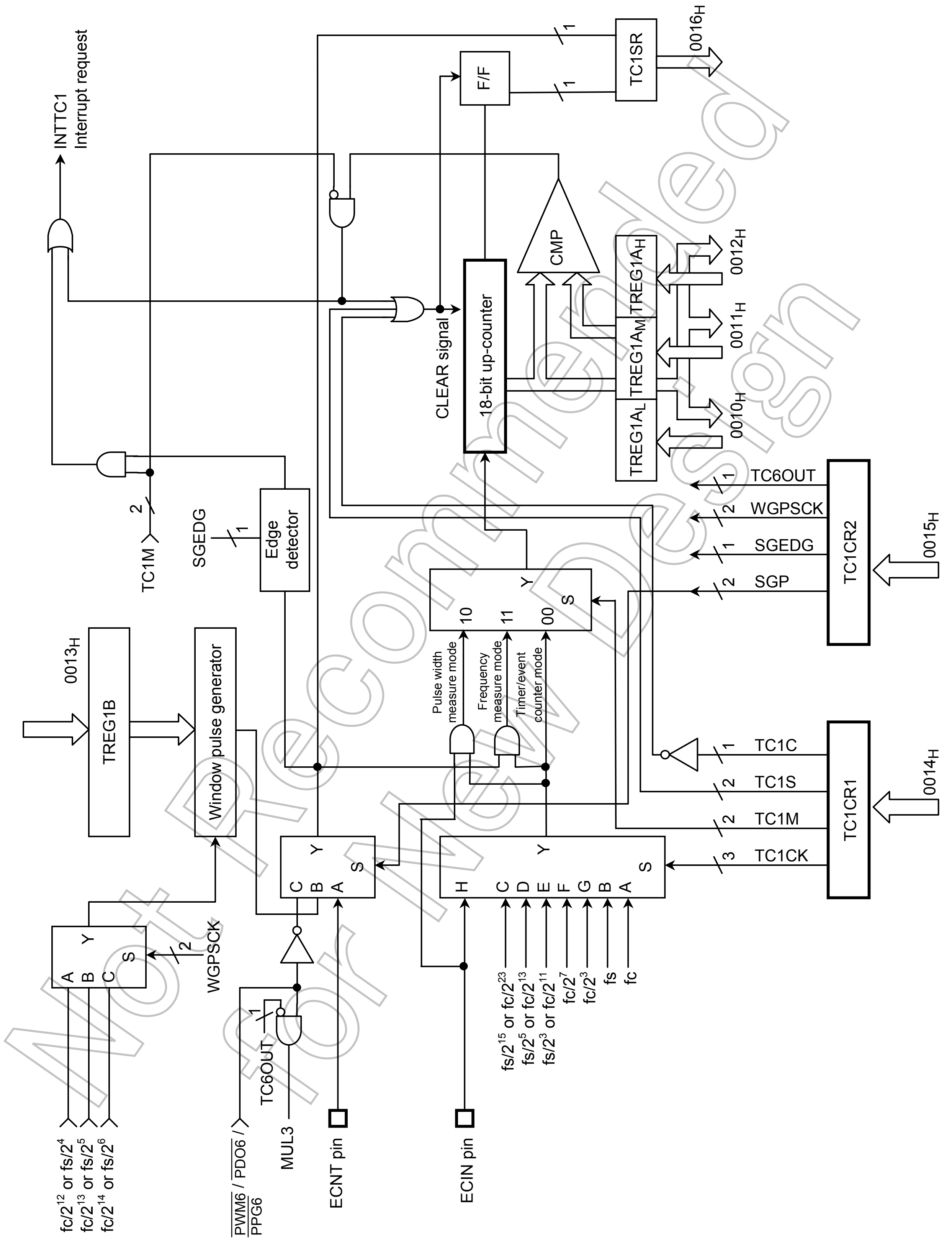


Figure 2.7.1 Timer/Counter 1

2.7.2 Control

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

Timer Register (TREG1A)

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

Read/Write (Initial value: 00 0000 0000 0000 0000)

Internal Window Gate Pulse Setting Register (TREG1B)

TREG1B (0013_H)	7	6	5	4	3	2	1	0	
	Ta				Tb				(Initial value: 0000 0000)

		WGPSCK	NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	
			DV7CK = 0	DV7CK = 1		
Ta	Setting "H" level period of the window gate pulse [s]	00	$(16 - Ta) \times 2^{12}/fc$		$(16 - Ta) \times 2^{4}/fs$	R/W
		01	$(16 - Ta) \times 2^{13}/fc$		$(16 - Ta) \times 2^{5}/fs$	
		10	$(16 - Ta) \times 2^{14}/fc$		$(16 - Ta) \times 2^{6}/fs$	
Tb	Setting "L" level period of the window gate pulse [s]	00	$(16 - Tb) \times 2^{12}/fc$		$(16 - Tb) \times 2^{4}/fs$	
		01	$(16 - Tb) \times 2^{13}/fc$		$(16 - Tb) \times 2^{5}/fs$	
		10	$(16 - Tb) \times 2^{14}/fc$		$(16 - Tb) \times 2^{6}/fs$	

Note: WGPSCK is bit3 and bit2 in TC1CR2.

Timer/Counter 1 Control Register 1

TC1CR1 (0014_H)	7	6	5	4	3	2	1	0	
	TC1C	TC1S		TC1CK			TC1M		(Initial value: 1000 1000)

TC1C		0: Counter/overflow flag clear request ("1" is automatically set after clearing.)					R/W
TC1S	TC1 start control	00: Stop and counter clear and overflow flag clear 10: Start *1: Reserved					
TC1CK	TC1 source clock select [Hz]		NORMAL1/2, IDLE1/2 Mode DV7CK = 0	NORMAL1/2, IDLE1/2 Mode DV7CK = 1	SLOW Mode	SLEEP Mode	
		000:	fc	fc	fc (Note4)	–	
		001:	fs	fs	–	–	
		010:	$fc/2^{23}$	$fs/2^{15}$	$fs/2^{15}$	$fs/2^{15}$	
		011:	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$	
		100:	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	$fs/2^{3}$	
		101:	$fc/2^{7}$	$fc/2^{7}$	–	–	
		110:	$fc/2^{3}$	$fc/2^{3}$	–	–	
		111:	External clock (ECIN pin input)				
TC1M	TC1 mode select	00: Timer/Event counter mode 01: Reserved 10: Pulse width measurement mode 11: Frequency measurement mode					

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

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

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

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

Note 5: When a read instruction is executed to the timer register (TREG1A), the counter immediate value, not the register set value, is read out. Therefore it is impossible to read out the written value of TREG1A. To read the counter value, the read instruction should be executed when the counter stops to avoid reading unstable value.

Note 6: Set the timer register (TREG1A) to ≥ 1.

Note 7: When using the timer mode and pulse width measurement mode, set TC1CK (TC1 source clock select) to internal clock.

Note 8: When using the event counter mode, set TC1CK (TC1 source clock select) to external clock.

Note 9: Because the read value is different from the written value, do not use read-modify-write instructions to TREG1A.

Note 10: $fc/2^7$ and $fc/2^3$ can not be used as source clock in SLOW/SLEEP mode.

Note 11: The read data of bits 7 to 2 in TREG1AH are always "0". (Data "1" can not be written.)

Figure 2.7.2 Timer Register/Window Gate Pulse Setting Register/Control Register of the TC1

Timer/Counter 1 Control Register 2

TC1CR2	7	6	5	4	3	2	1	0	
(0015_H)	"0"	SGP		SGEDG	WGPSCK		TC6OUT	"0"	(Initial value: *000 000*)

SGP	Window gate pulse select	00: ECNT input 01: Internal window gate pulse (TREG1B) 10: $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$ (TC6) output 11: Reserved				R/W
SGEDG	Window gate pulse interrupt edge select	0: Interrupts at the falling edge 1: Interrupts at the falling/rising edges				
WGPSCK	Window gate pulse source clock select		NORMAL1/2, IDLE1/2 Mode		SLOW Mode	SLEEP Mode
			DV7CK = 0	DV7CK = 1		
		00:	2^{12}/fc	2^{4}/fs	2^{4}/fs	2^{4}/fs
		01:	2^{13}/fc	2^{5}/fs	2^{5}/fs	2^{5}/fs
		10:	2^{14}/fc	2^{6}/fs	2^{6}/fs	2^{6}/fs
		11:	Reserved			
TC6OUT	TC6 output ($\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$) external output select	0: Output to MUL3 pin (Either P33 or P74 output can be selected by MULSEL<MUL3>) 1: No output to MUL3 pin				

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

Note 2: Set the mode, source clock, and edge (selection) when the TC1 stops (TC1S = 00).

Note 3: If there is no need to use $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$ as window gate pulse of TC1, always write "0" to TC6OUT.

Note 4: Make sure to write "0" to bit7 and bit0 in TC1CR2.

TC1 Status Register

TC1SR	7	6	5	4	3	2	1	0	
(0016_H)	HECF	HEOVF	"0"	"0"	"0"	"0"	"0"	"0"	(Initial value: 0000 0000)

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

Figure 2.7.3 Control Register of the TC1/Status Register

2.7.3 Function

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

(1) Timer mode

In this mode, counting up is performed using the internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes after the counter is cleared.

Table 2.7.1 Source Clock (Internal clock) of Timer/Counter 1

Source Clock				Resolution		Maximum Time Setting	
NORMAL1/2, IDLE1/2 Mode		SLOW Mode	SLEEP Mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1						
$fc/2^{23}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	$fs/2^{15}$ [Hz]	0.52 [s]	1 [s]	38.2 [h]	72.8 [h]
$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	$fs/2^{5}$	512 [μs]	0.98 [ms]	2.2 [min]	4.3 [min]
$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	$fs/2^{3}$	128 [μs]	244 [μs]	0.6 [min]	1.07 [min]
$fc/2^{7}$	$fc/2^{7}$	–	–	8 [μs]	–	2.1 [s]	–
$fc/2^{3}$	$fc/2^{3}$	–	–	0.5 [μs]	–	131.1 [ms]	–
fc	fc	fc (Note)	–	62.5 [ns]	–	16.4 [ms]	–
fs	fs	–	–	–	30.5 [μs]	–	8 [s]

Note: When fc is selected for the source clock in SLOW mode, the lower bits 11 of TREG1A is invalid, and a match of the upper bits 7 makes interrupts.

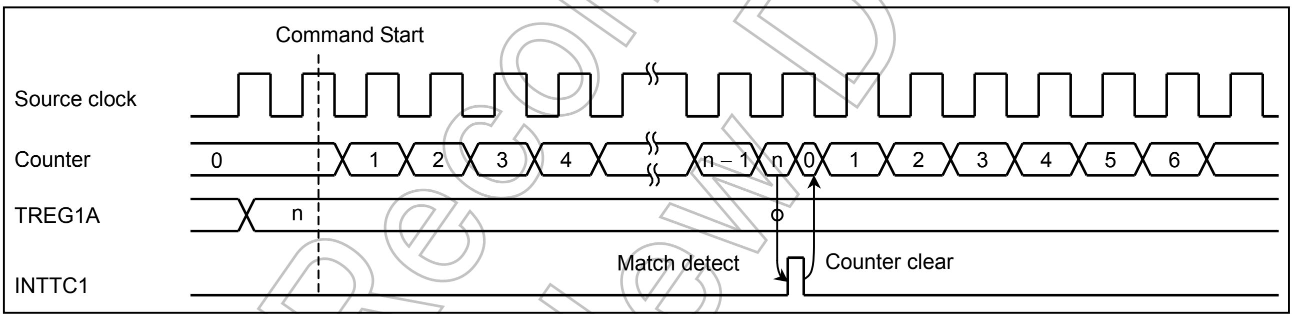


Figure 2.7.4 Timing Chart for Timer Mode

(2) Event counter mode

It is a mode to count up at the falling edge of the ECIN pin input. Both edges can not be used. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Counting up resumes for ECIN pin input edge each after the counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL 1/2 or IDLE 1/2 mode and $fs/2^4$ [Hz] in SLOW or SLEEP mode. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width.

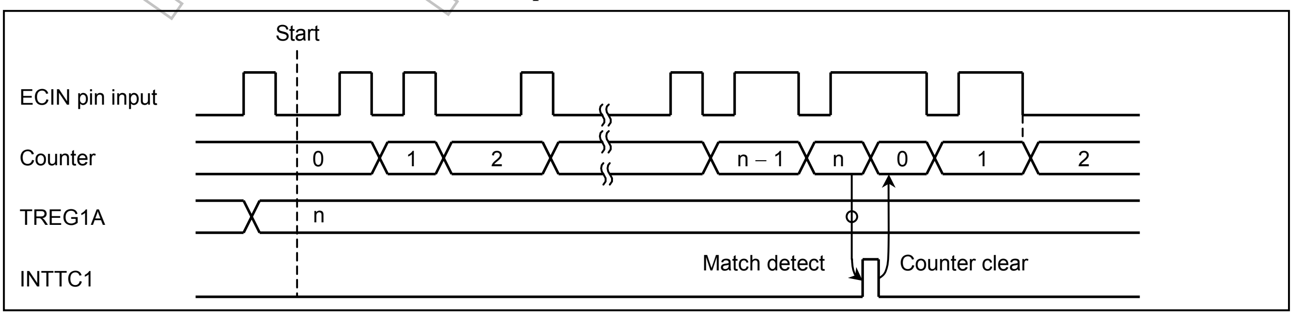


Figure 2.7.5 Pulse Width Measurement Mode Timing Chart

(3) Pulse width measurement mode

In this mode, pulse widths are counted on the falling edge of logical AND-ed product between ECIN pin input (Window pulse) and the internal clock. The internal clock is selected by TC1CK (Bit2, 3 and 4 in TC1CR1). An INTTC1 interrupt is generated at the falling edge of the window pulse or both rising and falling edges of the window pulse, that can be selected by SGEDG (Bit4 in TC1CR2). In the interrupt service program, read the contents of TREG1A while the count is stopped (ECIN pin is low), then clear the counter using TC1C (Bit7 in TC1CR1). When the counter is not cleared, counting up resumes by starting count-up. When TREG1A is counted up from 3FFFFH to 00000H, an overflow occurs. HEOVF (Bit6 in TC1SR) of the status register can monitor whether the overflows or not. HEOVF remains the old data until the counter is required to be cleared by TC1C.

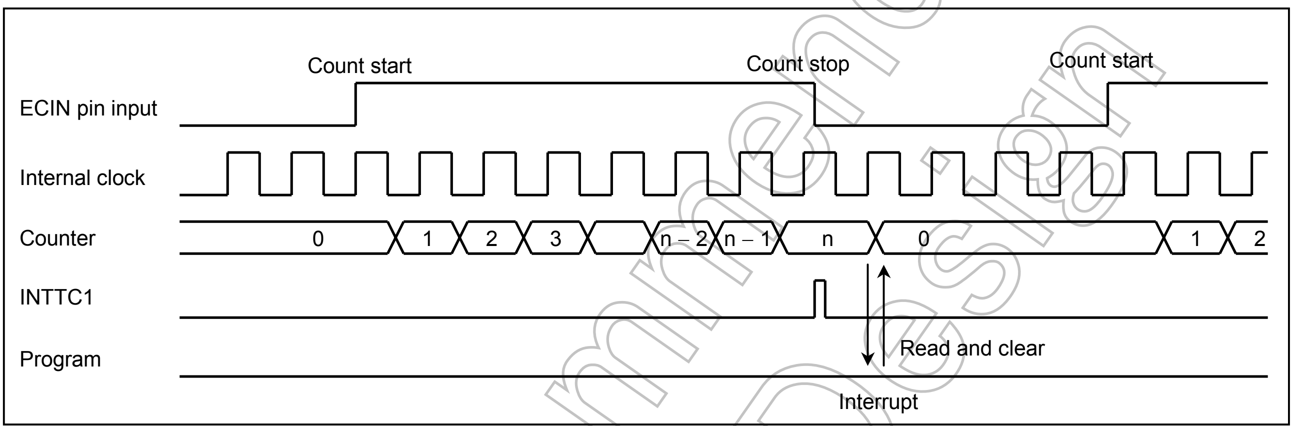


Figure 2.7.6 Pulse Width Measurement Mode Timing Chart (TC1CR2<SGEDG> = "0")

Note 1: INTTC1 interrupt occurs when ECIN input is "1" and TC1S of TC1CR1 is written to "00". According to the following step, when timer counter is stopped, INTTC1 interrupt latch should be cleared to "0".

```
TC1STOP:
        ¦
        DI                              ; Clear IMF
        CLR     (EIRH). EF8             ; Clear EF8
        LD      (TC1CR1), 00011010B     ; Stop timer counter 1
        LD      (ILH), 11111110B        ; Clear IL8
        SET     (EIRH). EF8             ; Set EF8
        EI                              ; Set IMF
        ¦
```

Note 2: When SGEDG (Window gate pulse interrupt edge select) is set to both edges and ECIN pin input is "1" in the pulse width measurement mode, an INTTC1 interrupt is generated by setting TC1S (TC1 start control) to "10" (Start).

Note 3: In the pulse width measurement mode, HECF (Operating status monitor) cannot used.

(4) Frequency measurement mode

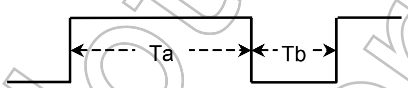
In this mode, the frequency of ECIN pin input pulse is measured. TC1CK is required to be set to the external clock (TC1CK = “111”). The edge of the input pulse is counted during “H” level of the window gate pulse selected by SGP (Bit5 and 6 in TC1CR2). Whether the input pulse is counted on the falling edge. An INTTC1 interrupt is generated on the falling edge or both the rising/falling edges of the window gate pulse, that can be selected by SGEDG (Bit4 in TC1CR2). To use ECNT terminal input as a window gate pulse, SGP (Bit5 and 6 in TC1CR2) should be set to “00”. In the interrupt service program, read the contents of TREG1A while the count is stopped (Window gate pulse is low), then clear the counter using TC1C. When the counter is not cleared, counting up resumes by stating count-up. The window pulse status can be monitored by HECF of the status register. HEOVF of the status register can monitor whether the binary counter overflows or not. In the overflow flag status, a new data is not input until the counter clear requests.

- Using TC6 output ($\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$) for the window gate pulse, external output of $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$ to MUL3 pin (Either P33 or P74 output can be selected by MULSEL<MUL3>.) can be controlled using TC6OUT (Bit1 in TC1CR2). Zero-clearing TC6OUT outputs $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$ to MUL3 pin; setting 1 in TC6OUT does not output $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$ to MUL3 pin. (TC6OUT is used to control output to MUL3 pin only. Thus, use the timer counter 6 control register to operate/stop $\overline{PWM6}$ / $\overline{PDO6}$ / $\overline{PPG6}$.)

- When the internal window gate pulse is selected, the window gate pulse is set as follows. The internal window gate pulse consists of “H” level period (Ta) that is counting time and “L” level period (Tb) that is counting stop time. Ta or Tb can be individually set by TREG1B. One cycle contains Ta + Tb.

  Note 1: Because the internal window gate pulse is generated in synchronization with the internal divider, it may be delayed for a maximum of one cycle of the source clock (WGPSCK) immediately after start of the timer.

  Note 2: Set the internal window gate pulse when the timer counter is not operating or during the Tb period. When Tb is overwritten during the Tb period, the update is valid from the next Tb period.



The setting time of Ta or Tb is shown as the following format.
at (WGPSCK = 10)
$(16 - n) \times 2^{14}/fc$ [s] or
$(16 - n) \times 2^{6}/fs$ [s]
The setting time at fs = 32.768 kHz is listed in a table mentioned right.
Ta is set by the upper side (Bits 7 to 4), and Tb is set by the lower side (Bits 3 to 0) in TREG1B.

Figure 2.7.7 Window Gate Pulse Format

Table 2.7.2 Setting Ta and Tb
(WGPSCK = 10, fs = 32.768 kHz)

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

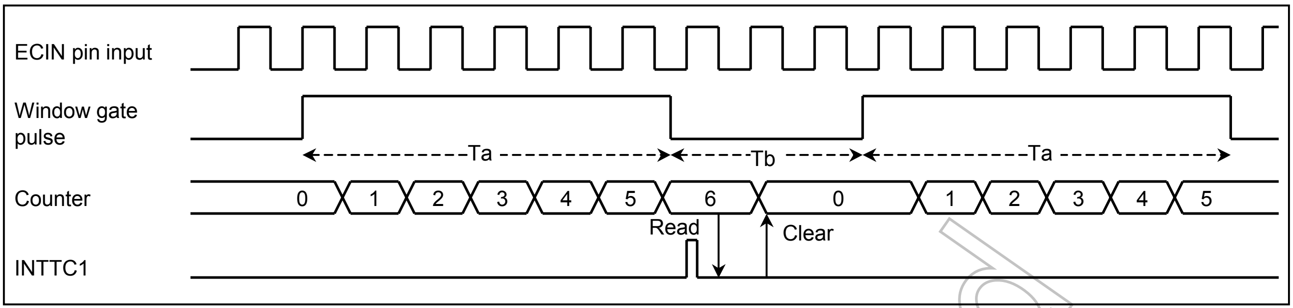


Figure 2.7.8 Timing Chart for the Frequency Measurement Mode
(ECIN falling edge count, window gate pulse falling interrupt at TC1CR2<SGEDG> = "0")

2.8 8-Bit Timer/Counter (TC3, TC4, TC5, TC6)

The TMP86CM25A has four channels of 8-bit timer/counter (TC3, TC4, TC5, TC6). These timer/counter are used as timer, event counter, PWM, PPG and PDO. These are also available as a 16-bit timer/counter by cascade connection.

2.8.1 Configuration

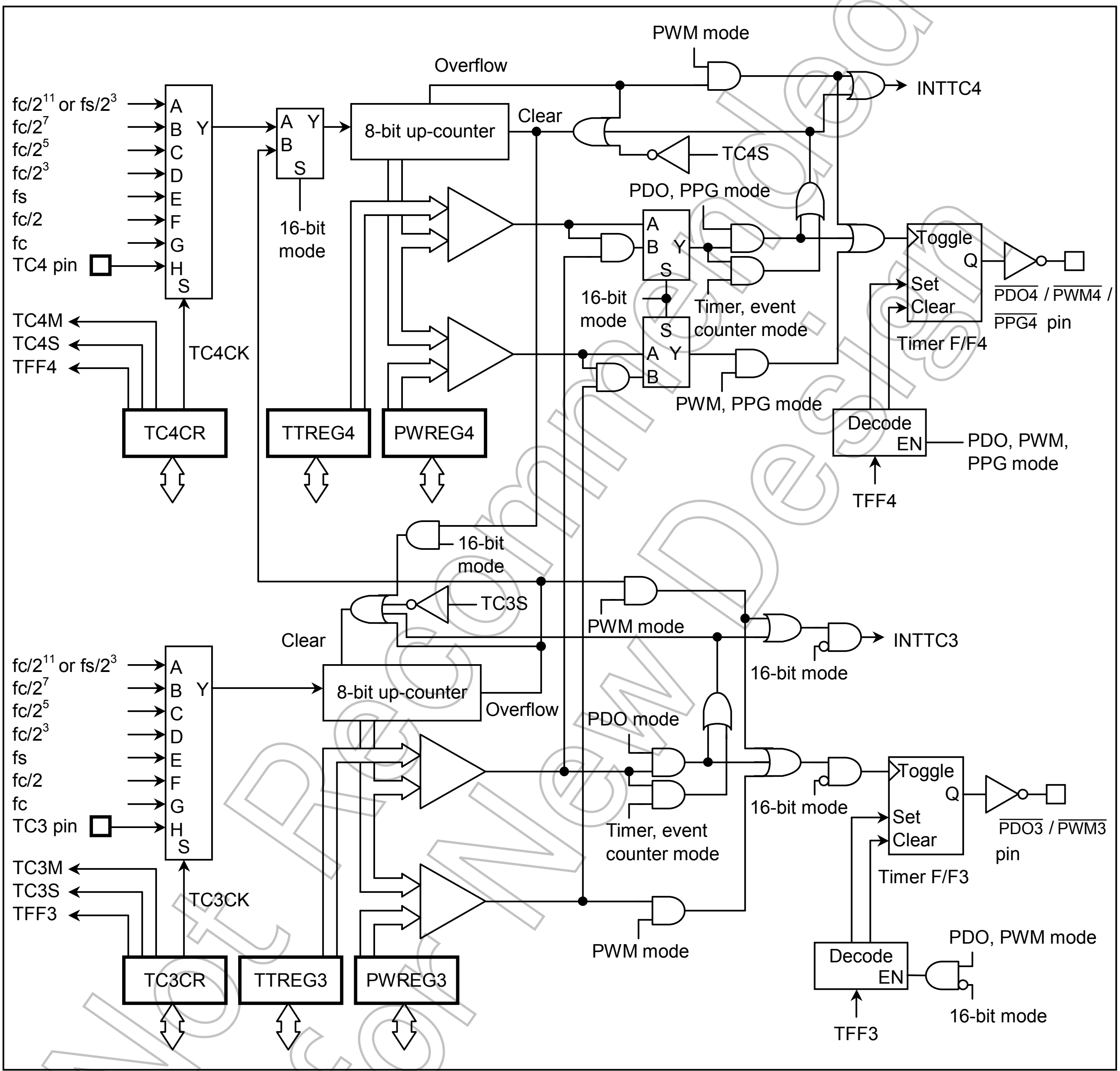


Figure 2.8.1 8-Bit Timer 3, 4

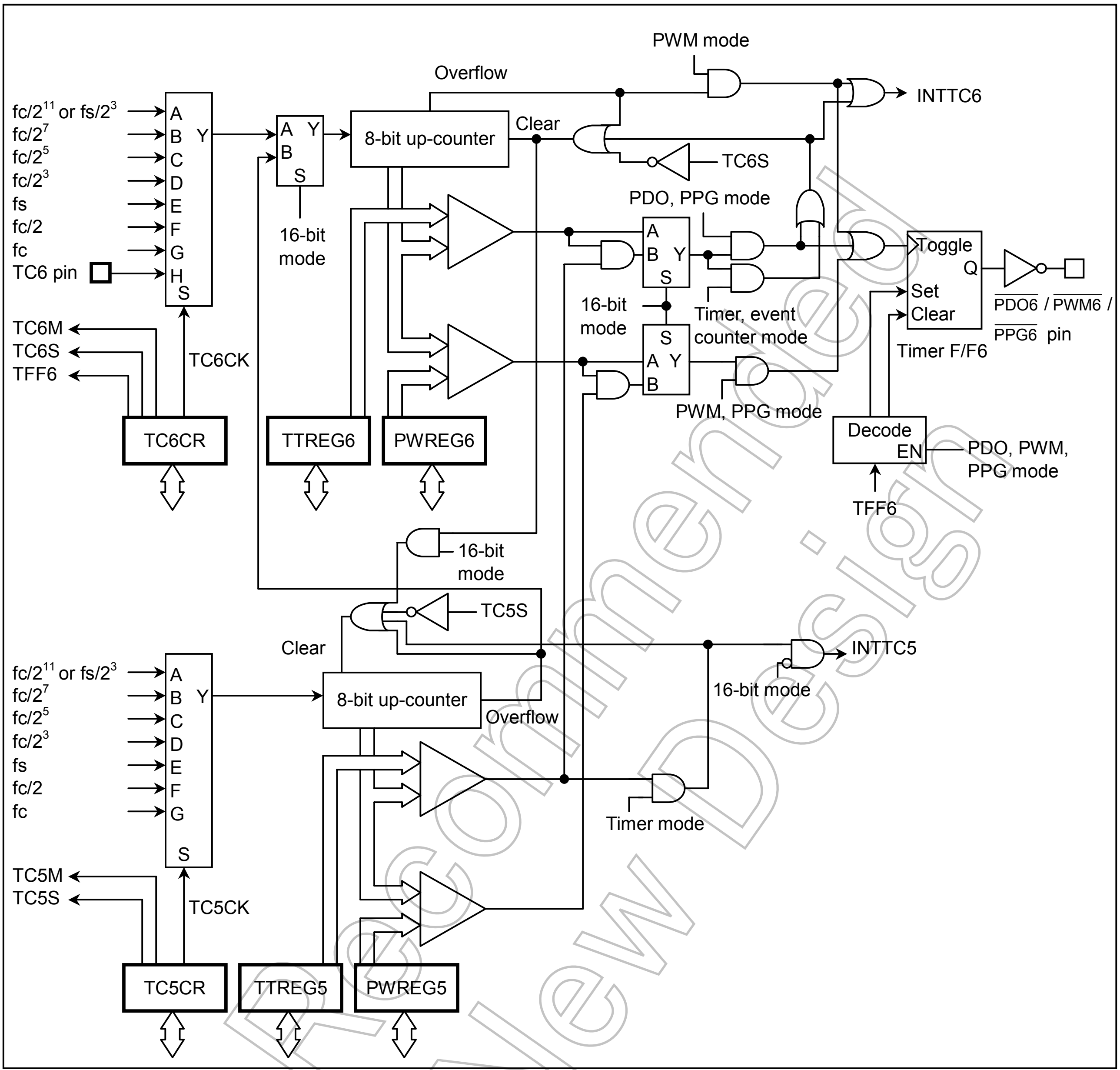


Figure 2.8.2 8-Bit Timer 5, 6

2.8.2 Control

The timer/counter 3 is controlled by a timer/counter 3 control register (TC3CR) and two 8-bit timer registers (TTREG3 and PWREG3).

Timer Register

TTREG3 (001C$_H$) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG3 (002C$_H$) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG3) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG3) while timer/counter is operating, except 8-bit PWM and 16-bit PWM mode.

Timer/Counter 3 Control Register

TC3CR (0018$_H$)	7	6	5	4	3	2	1	0	
	TFF3	TC3CK			TC3S	TC3M			(Initial value: 0000 0000)

TFF3	Timer F/F3 control	0: Clear 1: Set				R/W
TC3CK	TC3 source clock select [Hz]		NORMAL1/2, IDLE1/2 Mode DV7CK = 0	NORMAL1/2, IDLE1/2 Mode DV7CK = 1	SLOW1/2 SLEEP1/2 Mode	R/W
		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	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	fc(Note 8)	
		111	TC3 pin (MUL1) input			
TC3S	TC3 start control	0: Stop and counter clear 1: Command start				R/W
TC3M	TC3 operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) mode 011: 16-bit mode (Mode selection is controlled by TC4M) 1**: Reserved				R/W

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]
Note 2: During TC3 operation, do not change TC3M, TC3CK and TFF3.
Note 3: When TC3 operation is stopped (TC3S = "1" → "0"), do not change TC3M, TC3CK and TFF3. But it is possible to change TC3M, TC3CK and TFF3 at the start timing (TC3S = "0" → "1").
Note 4: When used as 16-bit mode, the operating mode is selected by TC4CR<TC4M>, and TC3M should be set to "011".
Note 5: When used as 16-bit mode, only the source clock is selected by TC3CK, and start of operation and control of F/F are controlled by TC4CR<TC4S> and TC4CR<TFF4>.
Note 6: Selecting source clock depends on the operating mode, refer to Table 2.8.1 and Table 2.8.2 for details.
Note 7: Value of timer register depends on the operating mode, refer to Table 2.8.3 for details.
Note 8: When used as the SLOW and SLEEP modes, the "fs" of TC3 source clock can use only "fc warm-up counter" mode.

Figure 2.8.3 Timer 3 Register and Timer/Counter 3 Control Register

The timer/counter 4 is controlled by a timer/counter 4 control register (TC4CR) and two 8-bit timer registers (TTREG4 and PWREG4).

Timer Register

Register	7	6	5	4	3	2	1	0	
TTREG4 ($001D_H$) R/W									(Initial value: 1111 1111)
PWREG4 ($002D_H$) R/W									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG4) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG4) while timer/counter is operating, except 8-bit PWM and 16-bit PWM mode.

Timer/Counter 4 Control Register

TC4CR (0019_H)	7	6	5	4	3	2	1	0	
	TFF4	TC4CK			TC4S	TC4M			(Initial value: 0000 0000)

<table>
<tr><td>TFF4</td><td>Timer F/F4 control</td><td colspan="4">0: Clear
1: Set</td><td rowspan="5">R/W</td></tr>
<tr><td rowspan="2">TC4CK</td><td rowspan="2">TC4 source clock select [Hz]</td><td></td><td colspan="2">NORMAL1/2, IDLE1/2 Mode</td><td rowspan="1">SLOW1/2
SLEEP1/2
Mode</td></tr>
<tr><td>000
001
010
011
100
101
110
111</td><td>DV7CK = 0
fc/2^11
fc/2^7
fc/2^5
fc/2^3
fs
fc/2
fc
TC4 (MUL2) pin input</td><td>DV7CK = 1
fs/2^3
fc/2^7
fc/2^5
fc/2^3
fs
fc/2
fc
</td><td>
fs/2^3
–
–
–
fs
–
–
</td></tr>
<tr><td>TC4S</td><td>TC4 start control</td><td colspan="4">0: Stop and counter clear
1: Command start</td></tr>
<tr><td>TC4M</td><td>TC4 operating mode select</td><td colspan="4">000: 8-bit timer/event counter mode
001: 8-bit programmable divider output (PDO) mode
010: 8-bit pulse width modulation (PWM) mode
011: Reserved
100: 16-bit timer/event counter mode
101: Warm-up counter mode
110: 16-bit programmable divider output (PDO) mode
111: 16-bit programmable pulse generate (PPG) output mode</td></tr>
</table>

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]
Note 2: During TC4 operation, do not change TC4M, TC4CK and TFF4.
Note 3: When TC4 operation is stopped (TC4S = “1” → “0”) , do not change TC4M, TC4CK and TFF4. But it is possible to change TC4M, TC4CK and TFF4 at the start timing (TC4S = “0” → “1”).
Note 4: When TC4M is selected to “1**” (16-bit mode), the source clock is automatically selected to the over-flowing signal of TC3 counter.
Note 5: When used as 16-bit mode, the operating mode is selected by TC4M, and TC3CR<TC3M> should be set to “011”.
Note 6: When used as 16-bit mode, only the source clock is selected by TC3CR<TC3CK>, and start of operation and control of F/F are controlled by TC4S and TFF4.
Note 7: Selecting source clock depends on the operating mode, refer to Table 2.8.1 and Table 2.8.2 for details.
Note 8: Value of timer register depends on the operating mode, refer to Table 2.8.3 for details.

Figure 2.8.4 Timer 4 Register and Timer/Counter 4 Control Register

The timer/counter 5 is controlled by a timer/counter 5 control register (TC5CR) and two 8-bit timer registers (TTREG5 and PWREG5).

Timer Register

TTREG5 (001E$_H$) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG5 (002E$_H$) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG5) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG5) while timer/counter is operating, except 16-bit PWM mode.

Timer/Counter 5 Control Register

TC5CR (001A$_H$)	7	6	5	4	3	2	1	0	
		TC5CK			TC5S	TC5M			(Initial value: ∗000 0000)

			NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	
			DV7CK = 0	DV7CK = 1		
TC5CK	TC5 source clock select [Hz]	000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	R/W
		001	$fc/2^7$	$fc/2^7$	–	
		010	$fc/2^5$	$fc/2^5$	–	
		011	$fc/2^3$	$fc/2^3$	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	fc (Note 8)	
		111	Reserved			
TC5S	TC5 start control	0: Stop and counter clear 1: Command start				
TC5M	TC5 operating mode select	000: 8-bit timer mode 001: Reserved 010: Reserved 011: 16-bit mode (Mode selection is controlled by TC6M) 1∗∗: Reserved				

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]
Note 2: During TC5 operation, do not change TC5M and TC5CK.
Note 3: When TC5 operation is stopped (TC5S = "1" → "0"), do not change TC5M and TC5CK. But it is possible to change TC5M and TC5CK at the start timing (TC5S = "0" → "1").
Note 4: When used as 16-bit mode, the operating mode is selected by TC6CR<TC6M>, and TC5M should be set to "011".
Note 5: When used as 16-bit mode, only the source clock is selected by TC5CK, and start of operation and control of F/F are controlled by TC6CR<TC6S> and TC6CR<TFF6>.
Note 6: Selecting source clock depends on the operating mode, refer to Table 2.8.1 and Table 2.8.2 for details.
Note 7: Value of timer register depends on the operating mode, refer to Table 2.8.3 for details.
Note 8: When used as the SLOW and SLEEP modes, the "fs" of TC5 source clock can use only "fc warm-up counter" mode.

Figure 2.8.5 Timer 5 Register and Timer/Counter 5 Control Register

The timer/counter 6 is controlled by a timer/counter 6 control register (TC6CR) and two 8-bit timer registers (TTREG6 and PWREG6).

Timer Register

	7	6	5	4	3	2	1	0	
TTREG6 ($001F_H$) R/W									(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
PWREG4 ($002F_H$) R/W									(Initial value: 1111 1111)

Note 1: Do not change the timer register (TTREG6) while timer/counter is operating.
Note 2: Do not change the timer register (PWREG6) while timer/counter is operating, except 8-bit PWM and 16-bit PWM mode.

Timer/Counter 6 Control Register

	7	6	5	4	3	2	1	0	
TC6CR ($001B_H$)	TFF6	TC6CK			TC6S	TC6M			(Initial value: 0000 0000)

TFF6	Timer F/F6 control	0: Clear 1: Set				R/W
TC6CK	TC6 source clock select [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW1/2 SLEEP1/2 Mode	R/W
			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	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	–	
		111	TC6(MUL3) pin input			
TC6S	TC6 start control	0: Stop and counter clear 1: Command start				R/W
TC6M	TC6 operating mode select	000: 8-bit timer/event counter mode 001: 8-bit programmable divider output (PDO) mode 010: 8-bit pulse width modulation (PWM) mode 011: Reserved 100: 16-bit timer/event counter mode 101: Warm-up counter mode 110: 16-bit programmable divider output (PDO) mode 111: 16-bit programmable pulse generate (PPG) output mode				R/W

Note 1: fc: High-frequency clock [Hz], fs: Low-frequency clock [Hz]
Note 2: During TC6 operation, do not change TC6M, TC6CK and TFF6.
Note 3: When TC6 operation is stopped (TC6S = "1" → "0") , do not change TC6M, TC6CK and TFF6. But it is possible to change TC6M, TC6CK and TFF6 at the start timing (TC6S = "0" → "1").
Note 4: When TC6M is selected to "1**" (16-bit mode), the source clock is automatically selected to the over-flowing signal of TC5 counter.
Note 5: When used as 16-bit mode, the operating mode is selected by TC6M, and TC5CR<TC5M> should be set to "011".
Note 6: When used as 16-bit mode, only the source clock is selected by TC5CR<TC5CK>, and start of operation and control of F/F are controlled by TC6S and TFF6.
Note 7: Selecting source clock depends on the operating mode, refer to Table 2.8.1 and Table 2.8.2 for details.
Note 8: Value of timer register depends on the operating mode, refer to Table 2.8.3 for details.
Note 9: If there is no need to use $\overline{PDO6}$ / $\overline{PWM6}$ / $\overline{PPG6}$ as window gate pulse of TC1, always write "0" to TC6OUT.

Figure 2.8.6 Timer 6 Register And Timer/Counter 6 Control Register

Table 2.8.1 Operating Mode and Available Source Clock
(NORMAL1/2, IDLE1/2 mode)

Operating Mode	$fc/2^{11}$ or $fc/2^{3}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{3}$	fs	fc/2	fc	TCi pin input
8-bit timer	○	○	○	○	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○
8-bit PDO	○	○	○	○	–	–	–	–
8-bit PWM	○	○	○	○	○	○	○	–
16-bit timer	○	○	○	○	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○
Warm-up counter	–	–	–	–	○	–	–	–
16-bit PWM	○	○	○	○	○	○	○	–
16-bit PPG	○	○	○	○	–	–	–	–

Note 1: For 16-bit operation (16-bit timer/event counter, Warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bits (TC3CK, TC5CK).

Note 2: i = 3, 4, 6 (8-bit mode)
i = 3 (16-bit mode)

Table 2.8.2 Operating Mode and Available Source Clock
(under SLOW1/2 mode, SLEEP1/2 mode)

Operating Mode	$fc/2^{11}$ or $fc/2^{3}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{3}$	fs	fc/2	fc	TCi pin input
8-bit timer	○	–	–	–	–	–	–	–
8-bit event counter	–	–	–	–	–	–	–	○
8-bit PDO	○	–	–	–	–	–	–	–
8-bit PWM	○	–	–	–	○	–	–	–
16-bit timer	○	–	–	–	–	–	–	–
16-bit event counter	–	–	–	–	–	–	–	○
Warm-up counter	–	–	–	–	–	–	○	–
16-bit PWM	○	–	–	–	○	–	–	–
16-bit PPG	○	–	–	–	–	–	–	–

Note 1: For 16-bit operation (16-bit timer/event counter, Warm-up counter, 16-bit PWM and 16-bit PPG), set its source clock on lower bits (TC3CK, TC5CK).

Note 2: i = 3, 4, 6 (8-bit mode)
i = 3 (16-bit mode)

Table 2.8.3 Restriction against the Rate for Comparing Registers

Operating Mode	Authorized Rate for Register
8-bit timer/event counter	1 ≤ (TTREGn) ≤ 255
8-bit PDO	1 ≤ (TTREGn) ≤ 255
8-bit PWM	2 ≤ (PWREGn) ≤ 254
16-bit timer/event counter	1 ≤ (TTREG4, TTREG3) ≤ 65535, 1 ≤ (TTREG6, TTREG5) ≤ 65535
fc warm-up counter	256 ≤ (TTREG4, TTREG3) ≤ 65535, 256 ≤ (TTREG6, TTREG5) ≤ 65535
16-bit PWM	2 ≤ (PWREG4, PWREG3) ≤ 65534, 2 ≤ (PWREG6, PWREG5) ≤ 65534
16-bit PPG	1 ≤ (PWREG4, PWREG3) < (TTREG4, TTREG3) ≤ 65535 and (PWREG4, PWREG3) + 1 < (TTREG4, TTREG3) 1 ≤ (PWREG6, PWREG5) < (TTREG6, TTREG5) ≤ 65535 and (PWREG6, PWREG5) + 1 < (TTREG6, TTREG5)

Note: n = 3 to 6

2.8.3 Function

Timer/counter 3, 4, 5 and 6 have eight operating modes: 8-bit timer, 8-bit external trigger timer, 8-bit programmable divider output mode, 8-bit pulse width modulation output mode, 16-bit timer, 16-bit external trigger timer, 16-bit pulse width modulation output mode, 16-bit programmable pulse generator output mode.

16-bit timer mode can use Timer counter 3 and 4 (5, 6) by cascade connection.

(1) 8-bit timer mode (Timer/counter 3, 4, 5 and 6)

In this mode, counting up is performed using the internal clock. The contents of TTREGi are compared with the contents of up-counter. If a match is found, an INTTCi interrupt is generated, and the counter is cleared to “0”. Counting up resumes after the counter is cleared.

Note 1: In the timer mode, always write TCjCR<TFFj> to “0”. If TFFj is set to “1”, unexpected pulse may be output from $\overline{PDOj}$ / $\overline{PWMj}$ / $\overline{PPGj}$ pin.

Note 2: In the timer mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 3: j = 3, 4, 6 i = 3 to 6

Table 2.8.4 Timer/Counter 3, 4, 5, 6 Source Clock (Internal Clock)

Source Clock			Resolution		Maximum Time Setting	
NORMAL1/2, IDLE1/2 Modes		SLOW1/2, SLEEP1/2 Modes	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At fs = 32.768 kHz
DV7CK = 0	DV7CK = 0					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fc/2^3$ [Hz]	128 [μs]	244.14 [μs]	32.6 [ms]	62.3 [ms]
$fc/2^7$	$fs/2^7$	–	8 [μs]	–	2.0 [ms]	–
$fc/2^5$	$fs/2^5$	–	2 [μs]	–	510 [μs]	–
$fc/2^3$	$fc/2^3$	–	500 [ns]	–	127.5 [μs]	–

Example: Sets the timer mode with source clock $fc/2^7$ [Hz] and generates an interrupt 80 μs later (at fc = 16 MHz).

```
LD    (TTREG4), 0AH           ; Sets the timer register (80 µs ÷ 2^7/fc = 0AH)
DI
SET   (EIRH). EF11            ; Enables INTTC4 interrupt
EI
LD    (TC4CR), 00010000B      ; Sets the 8-bit timer mode and source clock (fc/2^7)
LD    (TC4CR), 00011000B      ; Starts TC4.
```

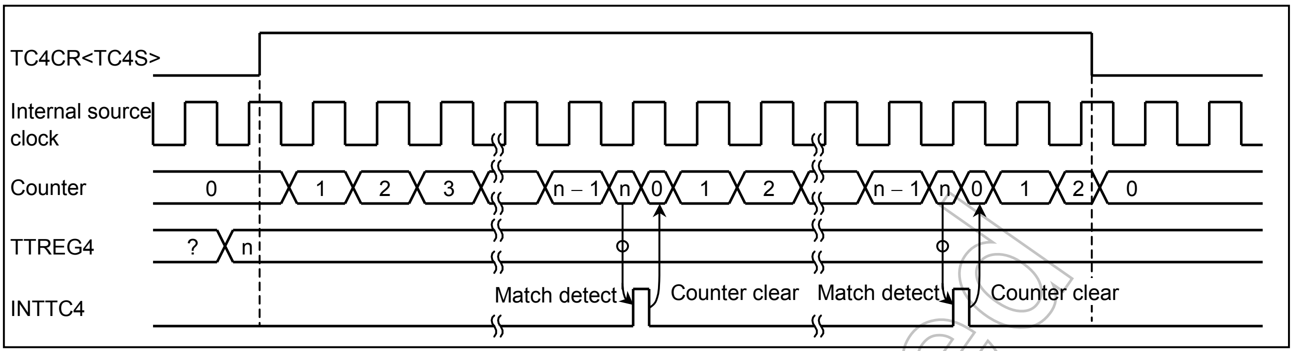


Figure 2.8.7 8-Bit Timer Mode Timing Chart (in case of timer/counter 4)

(2) 8-bit event counter mode (Timer/counter 3, 4 and 6)

In this mode, events are counted on the falling edge of TCj pin input. The contents of TTREGj are compared with the contents of up-counter. If a match is found, an INTTCj interrupt is generated, and the counter is cleared. The maximum applied frequency is $fc/2^4$ [Hz] in NORMAL1/2 or IDLE1/2 mode and $fs/2^4$ [Hz] in SLOW1/2 or SLEEP1/2 mode. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width.

Note 1: In the event counter mode, always write TCjCR<TFFj> to “0”. If TFFj is set to “1”, unexpected pulse may be output from $\overline{\text{PDOj}}$ / $\overline{\text{PWMj}}$ / $\overline{\text{PPGj}}$ pin.

Note 2: In the event counter mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 3: j = 3, 4, 6

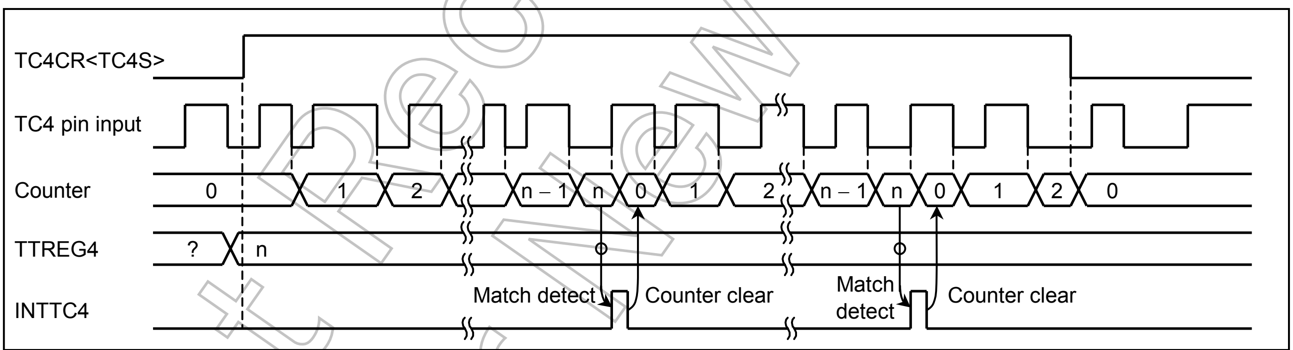


Figure 2.8.8 Event Counter Mode Timing Chart (in case of timer/counter 4)

(3) 8-bit programmable divider output (PDO) mode (Timer/counter 3, 4 and 6)

The internal clock is used for counting up. The contents of TTREGj are compared with the contents of the up-counter. Timer F/Fj output is toggled and the counter is cleared each time a match is found. Timer F/Fj output is inverted and output to the $\overline{\text{PDOj}}$ pin. When used as this mode, respective output latch should be set to “1”. This mode can be used for 50% duty pulse output. Timer F/Fj can be initialized by program, and it is initialized to “0” during reset. An INTTCj interrupt is generated each time the PDOj output is toggled.

Example: Output a 1024 Hz pulse (at fc = 16.0 MHz, MULSEL<MUL2> = “0”, in case of TC4)

```
SET    (P3DR). 2              ;  P32 output latch ← 1
LD     (TTREG4), 3DH          ;  (1/1024 ÷ 2^7/fc) ÷ 2 = 3DH
LD     (TC4CR), 00010001B     ;  Set the 8-bit PDO mode and source clock
                              ;  (fc/2^7)
LD     (TC4CR), 00011001B     ;  Starts TC4.
```

Note 1: In the programmable divider output(PDO) mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 2: If PDO output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{PDOj}}$ pin, modify TCjCR<TTFj> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

Example: Fixes $\overline{\text{PDOj}}$ output at high level after timer/counter is stopped

```
CLR    (TCjCR).3     ;  Stops timer/counter.
CLR    (TCjCR).7     ;  Sets PDOj output to high level output
```

Note 3: i = 3, 4, 6

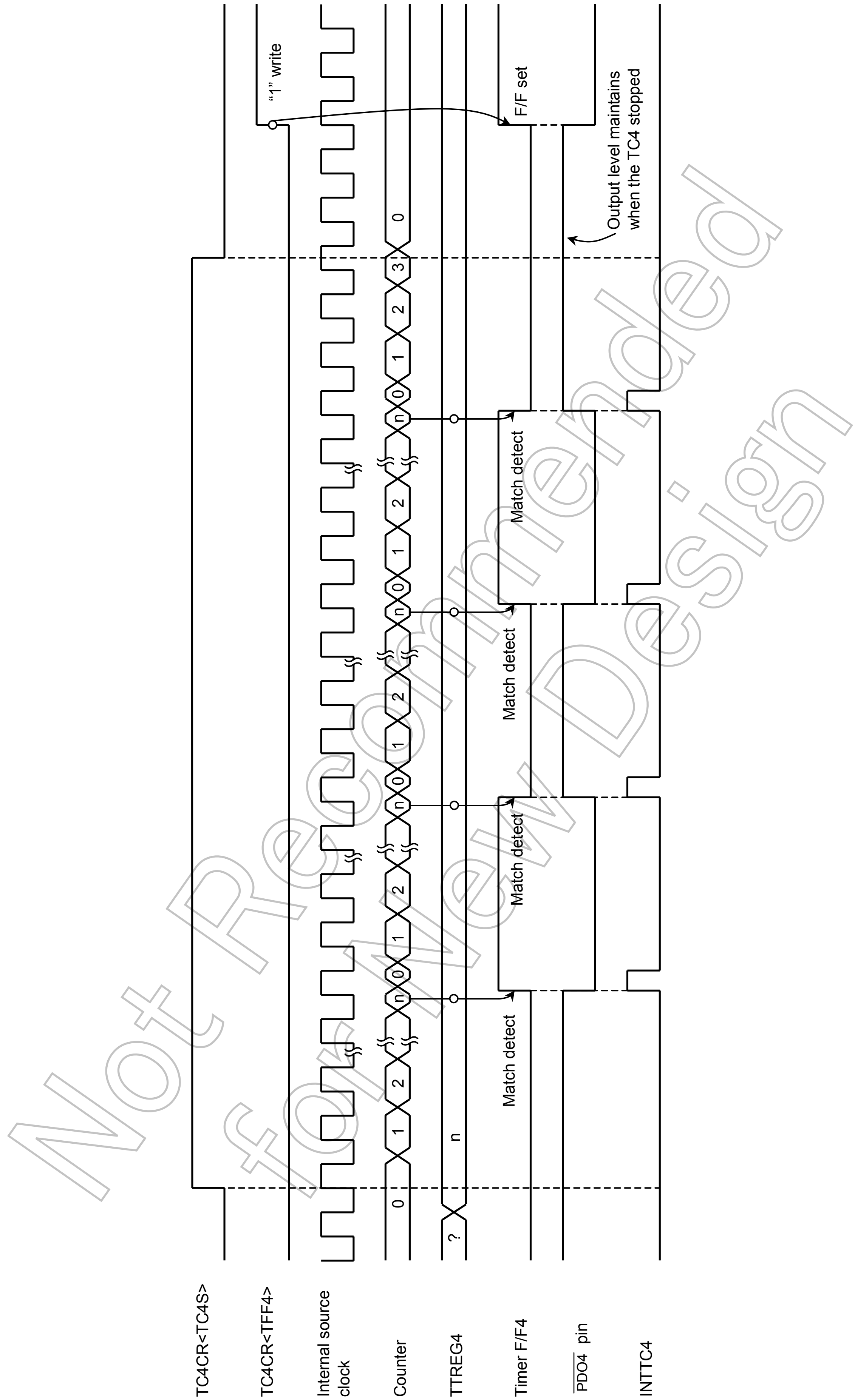


Figure 2.8.9 8-Bit PDO Mode Timing Chart (in case of timer/counter 4)

(4) 8-bit pulse width modulation (PWM) output mode (Timer/counter 3, 4 and 6)

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of PWREGi are compared with the contents of up counter. If a match is found, the timer F/Fi output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/Fi output is again toggled and the counter is cleared. Timer F/Fi output is inverted and output to the $\overline{\text{PWMi}}$ pin. An INTTCi interrupt is generated when an overflow occurs.

In PWM mode, because PWREGi becomes a 2-stage registers with shift register, it is possible to change the setting value of PWREGi while timer/counter is operating. Therefore, output can be altered continuously. The shift operation of PWREGi to shift register is executed at the INTTCi timing. While timer/counter is operating, the data by read instruction is not a setting value of PWREGi but a value of shift register. Thereofre, after writing to PWREGi, the reading data of PWREGi is previous value till INTTCi is generated.

While timer/counter stops, written value to PWREGi is shifted to shift register immediately.

Note 1: In PWM mode, write to the timer register PWREGi immediately after an INTTCi interrupt is generated (Normally during the INTTCi interrupt service routine). If writing to PWREGi and INTTCi interrupt occur at the same time, the unstable value being written is shifted. This may cause pulses different from the set value to be output until the next INTTCi interrupt is generated.

Note 2: If PWM output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{PWMi}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{PWMi}}$ output at high level after timer/counter is stopped

```
CLR    (TCiCR).3    ;    Stops timer/counter.
CLR    (TCiCR).7    ;    Sets PWMi output to high level output
```

Note 3: Before starting STOP mode, disable PWM output. When the timer/counter is enabled and fc, fc/2 or fs is selected as the source clock, pulse is output from PWM pin during warm-up after releasing STOP mode.

Note 4: i = 3, 4, 6

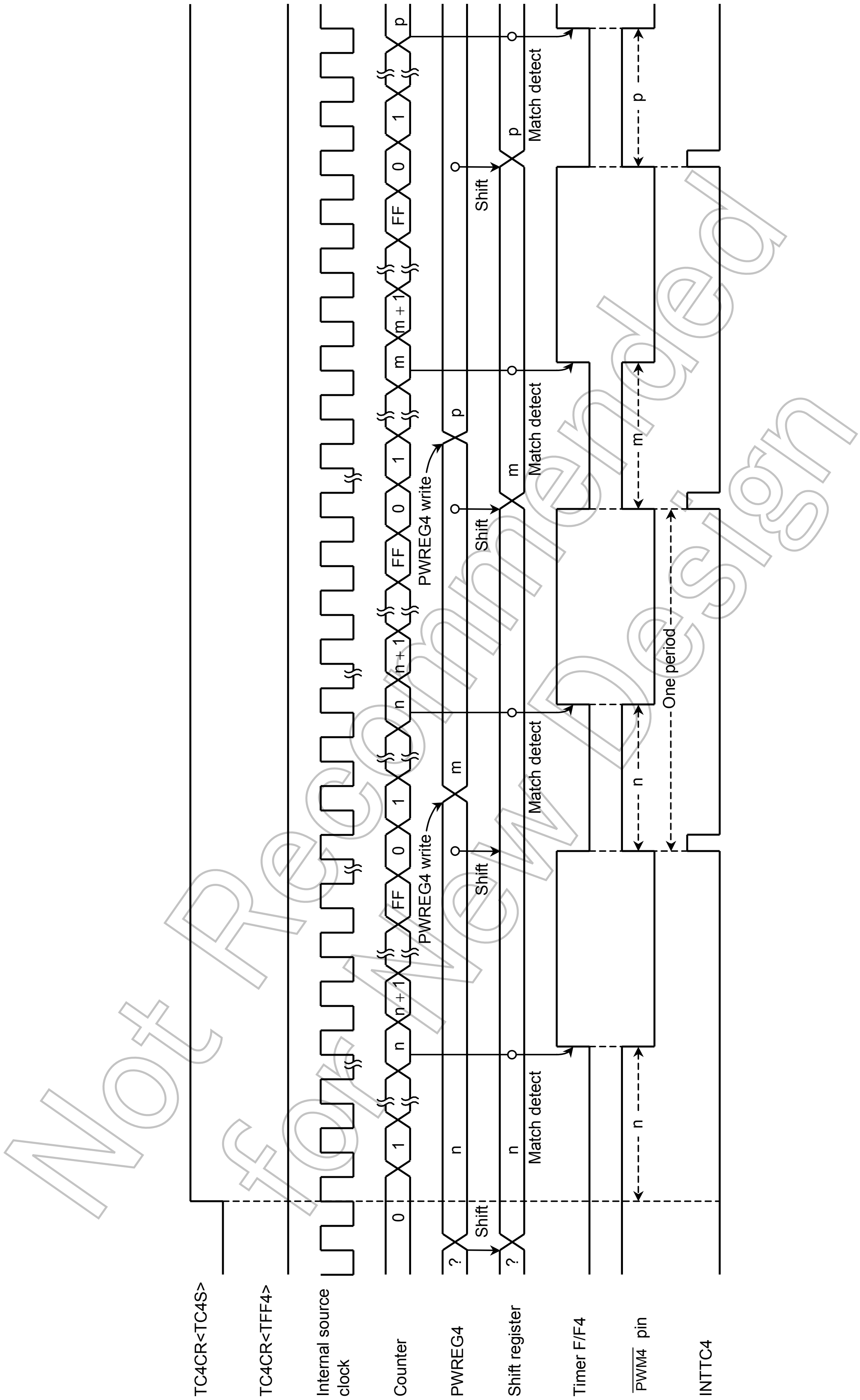


Figure 2.8.10 8-Bit PWM Mode Timing Chart (in case of timer/counter 4)

Table 2.8.5 PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$ [Hz]	128 [μs]	244.14 [μs]	32.8 [ms]	62.5 [ms]
$fc/2^7$	$fs/2^7$	–	8 [μs]	–	2.05 [ms]	–
$fc/2^5$	$fs/2^5$	–	2 [μs]	–	512 [μs]	–
$fc/2^3$	$fc/2^3$	–	500 [ns]	–	128 [μs]	–
fs	fs	fs	30.5 [μs]	30.5 [μs]	7.81 [ms]	78.1 [ms]
fc/2	fc/2	–	125 [ns]	–	32 [μs]	–
fc	fc	–	62.5 [ns]	–	16 [μs]	–

(5) 16-bit timer mode (Timer/counter 3 and 4, Timer/counter 5 and 6)

In this mode, counting up is performed using the internal clock.

Timer/counter 3 and 4 (5 and 6) are also available as a 16-bit timer mode by cascade connection.

a. 16-bit timer mode of timer/counter 3 and 4

If a match is found, the INTTC4 interrupt is generated and the counter is cleared to "0". Counting up resumes after the counter is cleared. The timer register should write to the TTREG3 more first than TTREG4. The timer register must not write only either TTREG3 or TTREG4.

b. 16-bit timer mode of timer/counter 5 and 6

If a match is found, the INTTC6 interrupt is generated and the counter is cleared to "0". Counting up resumes after the counter is cleared. The timer register should write to the TTREG5 more first than TTREG6. The timer register must not write only either TTREG5 or TTREG6.

Note 1: In the timer mode, always write TCjCR<TFFj> to "0". If TFFj is set to "1", unexpected pulse may be output from $\overline{PDOj}$ / $\overline{PWMj}$ / $\overline{PPGj}$ pin.

Note 2: In the timer mode, do not change the setting of timer registers (TTREGi) while timer/counter is operating. Since TTREGi is configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 3: j = 3, 4, 6, i = 3 to 6

Table 2.8.6 Source Clock of 16-Bit Timer Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 Mode		SLOW1/2, SLEEP1/2 Mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$ [Hz]	$fs/2^3$ [Hz]	$fs/2^3$	128 [μs]	244.14 [μs]	8.39 [s]	16 [s]
$fc/2^7$	$fs/2^7$	–	8 [μs]	–	524.3 [ms]	–
$fc/2^5$	$fs/2^5$	–	2 [μs]	–	131.1 [μs]	–
$fc/2^3$	$fc/2^3$	–	500 [ns]	–	32.8 [μs]	–

Example: Set the 16-bit timer mode with source clock fc/2^7 [Hz] and generates an interrupt 300 [ms] later (at fc = 16 [MHz])

```
LDW   (TTREG3), 927CH   ;  Sets the timer register
                           (300 ms ÷ 2^7/fc = 927CH)
DI
SET   (EIRH). EF11      ;  Enable INTTC4 interrupt
EI
LD    (TC3CR), 13H      ;  Sets the 16-bit timer mode (lower) and
                           source clock
LD    (TC4CR), 04H      ;  Sets the 16-bit timer mode (upper)
LD    (TC4CR), 0CH      ;  Starts timer/counter
```

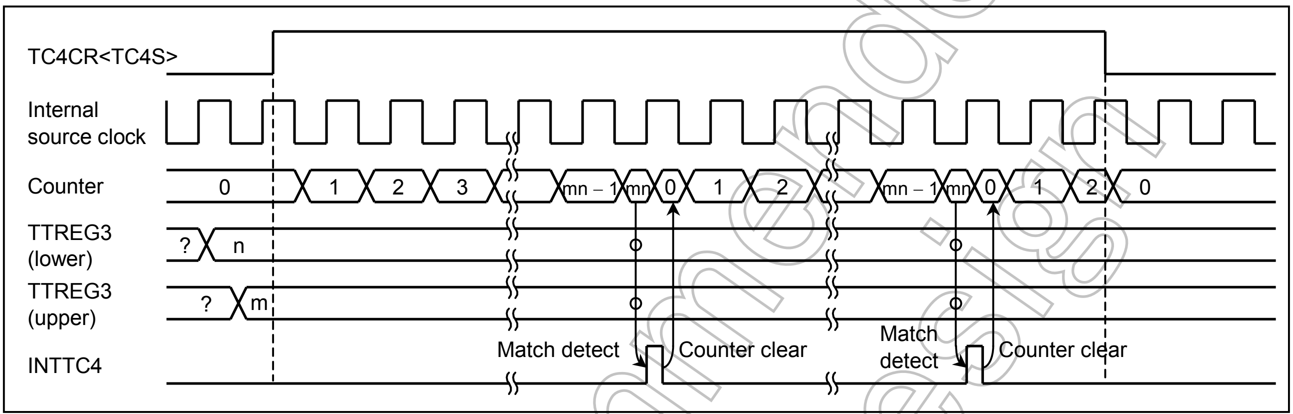


Figure 2.8.11 16-Bit Timer Mode Timing Chart (in case of timer/counter 3 and 4)

(6) 16-bit event counter mode (Timer/counter 3 and 4)

In this mode, events are counted on the falling edge of the TC3 pin input. Timer/counter 5 and 6 can not use a 16-bit event counter mode. Timer/counter 3 and 4 are also available as a 16-bit event counter mode by cascade connection.

a. 16-bit event counter mode of timer/counter 3 and 4

If a match is found, the INTTC4 interrupt is generated and the counter is cleared to “0”. After the counter is cleared, counting up resumes every falling edge of TC3 input. The maximum applied frequency is fc/2^4 [Hz] in NORMAL1/2 or IDLE1/2 mode and fs/2^4 [Hz] in SLOW1/2 or SLEEP1/2 mode. Two or more machine cycles are required for both the “H” and “L” levels of the pulse width. The timer register should write to the TTREG3 more first than TTREG4. The timer register must not write only either TTREG3 or TTREG4.

Note 1: In the event counter mode, always write TCjCR<TFFj> to “0”. If TFFj is set to “1”, unexpected pulse may be output from $\overline{\text{PDOj}}$ / $\overline{\text{PWMj}}$ / $\overline{\text{PPGj}}$ pin.

Note 2: In the event counter mode, do not change the setting of timer registers (TTREGj) while timer/counter is operating. Since TTREGj is configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 3: j = 3, 4

(7) 16-bit pulse width modulation (PWM) output mode
(Timer/counter 3 and 4, Timer/counter 5 and 6)

PWM output with a resolution of 16 bits is possible. Timer/counter 3 and 4 (5 and 6) are also available as a 16-bit PWM output mode by cascade connection.

a. 16-bit PWM output mode of timer/counter 3 and 4

The contents of PWREG3/4 are compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F4 output is again toggled and the counter is cleared. Timer F/F4 output is inverted and output to the $\overline{\text{PWM4}}$ pin. An INTTC4 interrupt is generated when an overflow occurs. When used as $\overline{\text{PWM4}}$ pin, respective output latch should be set to “1”. In PWM mode, because PWREG4/3 each becomes a 2-stage registers with shift register, it is possible to change the setting value of PWREG4/3 while timer/counter is operating. Therefore, output can be altered continuously. The shift operation of PWREG4/3 to shift register is executed at the INTTC4 timing. While timer/counter is operating, the data by read instruction is not a setting value of PWREG4/3 but a value of shift register. Therefore, after writing to PWREG4/3, the reading data of these registers is previous value till INTTC4 is generated.

While timer/counter stops, written value to PWREG4/3 is shifted to shift register immediately. When writing to PWREG4/3, always write to the lower side (PWREG3) and then the upper side (PWREG4) in that order. Writing to only lower side (PWREG3) or the upper side (PWREG4) has no effect.

b. 16-bit PWM output mode of timer/counter 5 and 6

The contents of PWREG5/6 are compared with the contents of up counter. If a match is found, the timer F/F6 output is toggled. The counter continues counting. And, when an overflow occurs, the timer F/F6 output is again toggled and the counter is cleared. Timer F/F6 output is inverted and output to the $\overline{\text{PWM6}}$ pin. An INTTC6 interrupt is generated when an overflow occurs. When used as $\overline{\text{PWM6}}$ pin, respective output latch should be set to “1”. In PWM mode, because PWREG6/5 each becomes a 2-stage registers with shift register, it is possible to change the setting value of PWREG6/5 while timer/counter is operating. Therefore, output can be altered continuously. The shift operation of PWREG6/5 to shift register is executed at the INTTC6 timing. While timer/counter is operating, the data by read instruction is not a setting value of PWREG6/5 but a value of shift register. Therefore, after writing to PWREG6/5, the reading data of these registers is previous value till INTTC6 is generated.

While timer/counter stops, written value to PWREG6/5 is shifted to shift register immediately. When writing to PWREG6/5, always write to the lower side (PWREG5) and then the upper side (PWREG6) in that order. Writing to only lower side (PWREG5) or the upper side (PWREG6) has no effect.

Note 1: In PWM mode, write to the timer register PWREGm,n immediately after an INTTCm interrupt is generated (Normally during the INTTCm interrupt service routine). If writing to PWREGm, n and INTTCm interrupt occur at the same time, the unstable value being written is shifted. This may cause pulses different from the set value to be output until the next INTTCm interrupt is generated.

Note 2: If PWM output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{PWMi}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFi simultaneously.

Example: Fixes $\overline{\text{PWMi}}$ output at high level after timer/counter is stopped

```
CLR    (TCiCR).3    ; Stops timer/counter
CLR    (TCiCR).7    ; Sets PWMi output to high level output
```

Note 3: Before starting STOP mode, disable PWM output. When the timer/counter is enabled and fc, fc/2 or fs is selected as the source clock, pulse is output from PWM pin during warm-up after releasing STOP mode.

Note 4: m = 4 and n = 3, or m = 6 and n = 5. i = 4, 6.

Table 2.8.7 16-Bit PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode	fc = 16 MHz	fs = 32.768 kHz	fc = 16 MHz	fs = 32.768 kHz
DV7CK = 0	DV7CK = 1					
$fc/2^{11}$	$fs/2^3$[Hz]	$fs/2^3$	128 [μs]	244.14 [μs]	8.39 [s]	16 [s]
$fc/2^7$	$fs/2^7$	–	8 [μs]	–	524.3 [ms]	–
$fc/2^5$	$fs/2^5$	–	2 [μs]	–	131.1 [ms]	–
$fc/2^3$	$fc/2^3$	–	500 [ns]	–	32.8 [ms]	–
fs	fs	fs	30.5 [μs]	30.5 [μs]	2 [s]	2[s]
fc/2	fc/2	–	125 [ns]	–	8.2 [ms]	–
fc	fc	–	62.5 [ns]	–	4.1 [ms]	–

Example: Extract the pulse, whose term and "high" width is 32.768 ms and 1 ms respectively, from P32 width 16-bit PWM mode
(at fc = 16 MHz, MULSEL<MUL2> = "0", DV7CK = "0")

```
SET    (P3DR).2            ; Sets P32 output data latch to "1"
LDW    (PWREG3), 07D0H     ; Sets pulse width
LD     (TC3CR), 33H        ; Sets the 16-bit PWM mode (lower) and source clock
                           ; (fc/2^3)
LD     (TC4CR), 0D6H       ; Sets the TFF4 to "1" and sets the 16-bit PWM mode
                           ; (upper)
LD     (TC4CR), 0DEH       ; Starts timer/counter
```

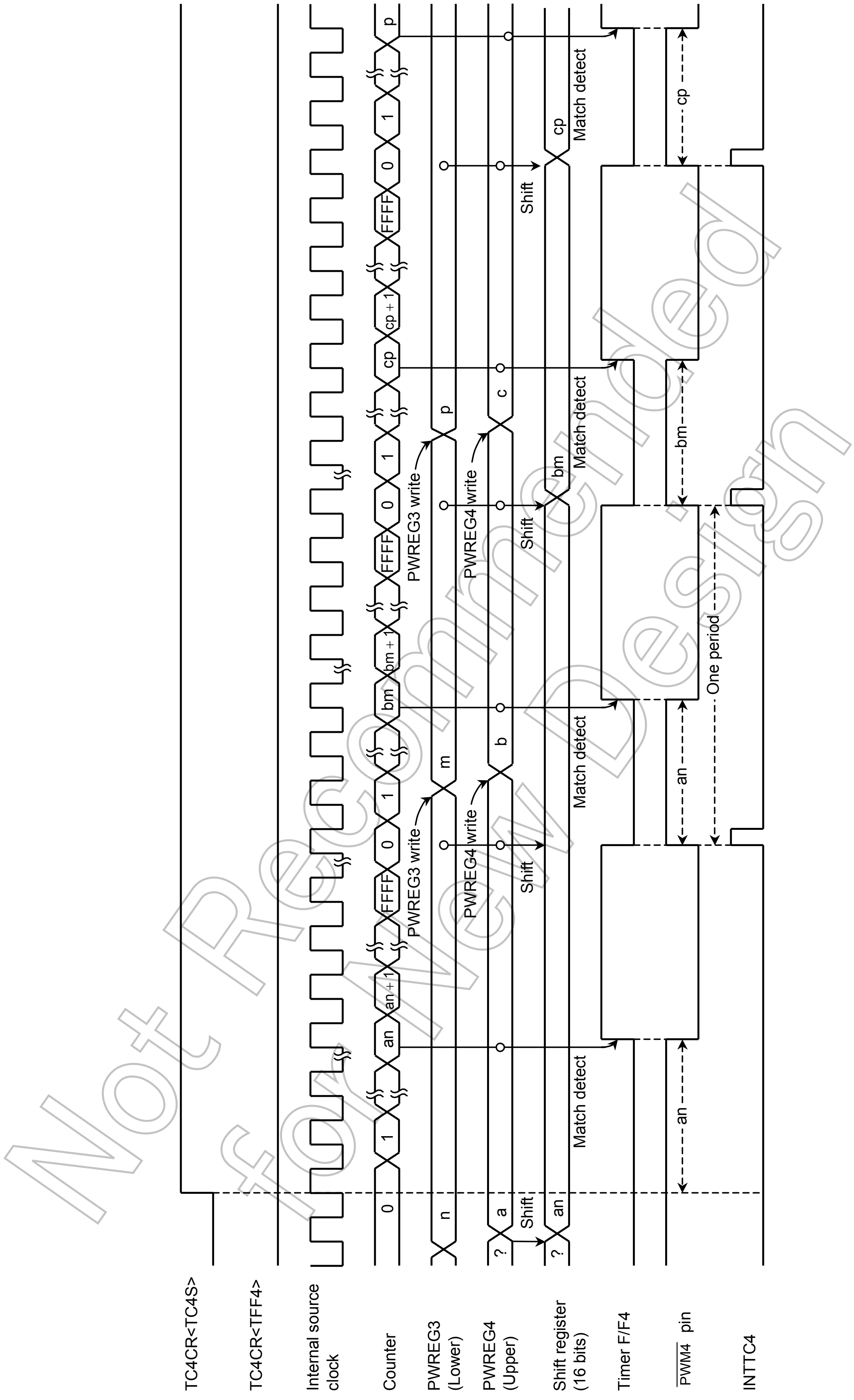


Figure 2.8.12 16-Bit PWM Mode Timing Chart (in case of timer/counter 3 and 4)

(8) 16-bit programmable pulse generate (PPG) output mode
(Timer/counter 3 and 4, timer/counter 5 and 6)

PPG output with a resolution of 16 bits is possible. Timer/counter 3 and 4 (5 and 6) are also available as a 16-bit PPG output mode by cascade connection.

a. 16-bit PPG output mode of timer/counter 3 and 4

First, the contents of PWREG3/4 are compared with the contents of up counter. If a match is found, the timer F/F4 output is toggled. Next, timer F/F4 is again toggled and the counter is cleared by matching with TTREG3/4. The INTTC4 interrupt is generated at this time.

When used as $\overline{\text{PPG4}}$ pin, respective output latch should be set to “1”. During reset, the F/F4 is initialized to “0”.

The F/F4 output is configured by TC4CR<TFF4>. Therefore, the $\overline{\text{PPG4}}$ can output either output high or output low at first time. The timer register should write to the PWREG3/TTREG3 more first than PWREG4/TTREG4. The timer register must not write only either PWREG3/TTREG3 or PWREG4/TTREG4.

b. 16-bit PPG output mode of Timer/counter 5 and 6

First, the contents of PWREG5/6 are compared with the contents of up counter. If a match is found, the timer F/F6 output is toggled. Next, timer F/F6 is again toggled and the counter is cleared by matching with TTREG5/6. The INTTC6 interrupt is generated at this time.

When used as $\overline{\text{PPG6}}$ pin, respective output latch should be set to “1”. During reset, the F/F6 is initialized to “0”.

The F/F6 output is configured by TC6CR<TFF6>. Therefore, the $\overline{\text{PPG6}}$ can output either output high or output low at first time. The timer register should write to the PWREG5/TTREG5 more first than PWREG6/TTREG6. The timer register must not write only either PWREG5/TTREG5 or PWREG6/TTREG6.

Example: Extract the pulse, whose term and “high” width is 16.385 ms and 1 ms respectively, from P32 with 16-bit PPG mode (at fc = 16.0 MHz, DV7CK = 0)

SET	(P3DR).2	;	Sets P32 output data latch to “1”
LDW	(PWREG3), 07D0H	;	Sets pulse width
LDW	(TTREG3), 8002H	;	Sets pulse term
LD	(TTREG3), 8002H	;	Sets the 16-bit PPG mode (Lower) and source clock
		;	(fc/2^3)
LD	(TC4CR), 0D7H	;	Sets the TFF4 to “1” and sets the 16-bit PPG mode
		;	(Upper)
LD	(TC4CR), 0DFH	;	Starts timer/counter

Note 1: In the programmable pulse generate (PPG) mode, do not change the setting of timer registers (PWREGi, TTREGi) while timer/counter is operating. Since PWREGi, TTREGi are configured as one-stage register, a newly set value is immediately reflected on the timer register.

Note 2: If PPG output is stopped during output operation, the output state is maintained at the state immediately before timer/counter is stopped. For changing the level of $\overline{\text{PPGj}}$, modify TCiCR<TTFi> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

Example: Fixes $\overline{\text{PPGj}}$ output at high level after timer/counter is stopped

CLR	(TCjCR).3	;	Stops timer/counter.
CLR	(TCjCR).7		Sets $\overline{\text{PPGj}}$ output to high level output

Note 3: j = 4, 6 i = 3 to 6

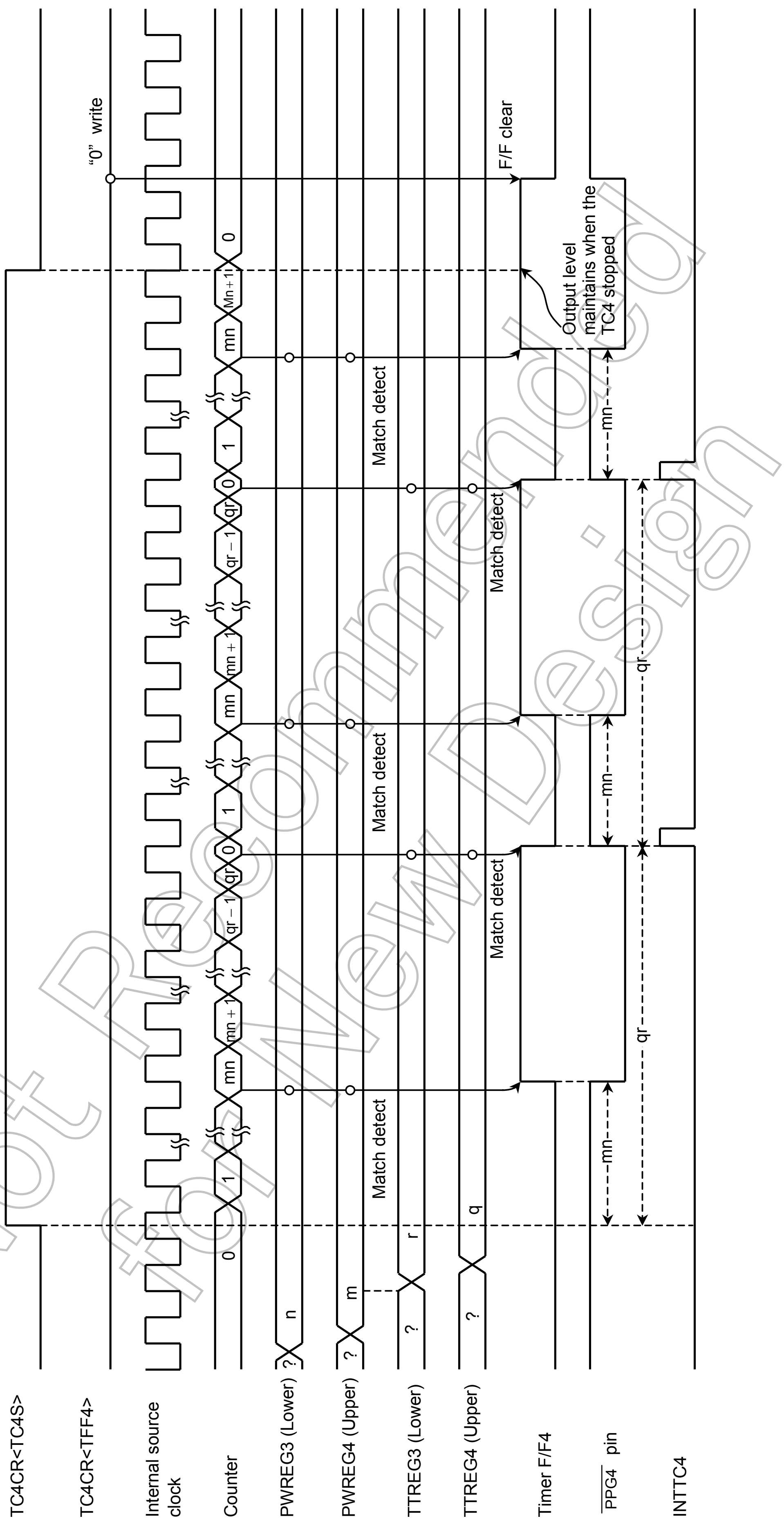


Figure 2.8.13 16-Bit PPG Mode Timing Chart (in case of timer/counter 3 and 4)

(9) Warm-up counter mode

In this mode, the warm-up period for switching the main system clock can be generated. Timer/counter 3 and 4 (5 and 6) are used as a 16-bit timer by cascade connection.

There are 2 modes in warm-up counter mode, one is a mode from NORMAL to SLOW and the other is a mode from SLOW to NORMAL.

Note 1: In the warm-up mode, always write TCiCR<TFFi> to “0”. If TFFi is set to “1”, unexpected pulse may be output from $\overline{\text{PDOi}}$ / $\overline{\text{PWMi}}$ / $\overline{\text{PPGi}}$ pin.

Note 2: In the warm-up mode, the lower 11 bits of TTREGm,n are ignored and an interrupt is generated by matching the upper 5 bits.

Note 3: i = 3, 4, 6 m = 4 and n = 3, or m = 6 and n = 5

a. Warm-up counter mode for low frequency
(NORMAL1 → NORMAL2 → SLOW2 → SLOW1)

In this mode, it can obtain the warm-up period till the oscillation for low-frequency (fs) is stabilized.

Before timer/counter is started, turn on low-frequency oscillation by setting SYSCR2<XTEN> to “1”.

After timer/counter is started by setting TCmCR<TCmS>, the contents of TTREGm, n are compared with the contents of up-counter. If a match is found, an INTTCm interrupt is generated, and the counter is cleared to “0”.

In the interrupt service program, stop the timer/counter and change the system clock to low-frequency clock by setting SYSCR2<SYSCK> to “1”.

After that, halt the high-frequency oscillation by clearing SYSCR2<XEN> to “0”.

Table 2.8.8 Warm-up Period for Low-frequency Oscillation (at fs = 32.768 kHz)

Min(at TTREGm, n = 0800H)	Max (at TTREGm, n = F800H)
62.5 ms	1.94 s

Example: Switching to the SLOW1 mode after low-frequency clock has stabilized by using TC4, 3.

```
         SET   (SYSCR2).6         ; SYSCR2<XTEN> ← "1"
         LD    (TC3CR),43H        ; TFF3 = "0", fs for source clock, sets 16-bit mode
         LD    (TC4CR),05H        ; TFF4 = "0", sets warm-up counter mode
         LD    (TTREG3),8000H     ; Sets warm-up time (depend on oscillator
                                  ; characteristics)
         DI                       ; IMF ← "0"
         SET   (EIRH).3           ; Enables INTTC4
         EI                       ; IMF ← "1"
         SET   (TC4CR).3          ; Starts TC4, 3
          :      :
PINTTC4  CLR   (TC4CR).3          ; Stops TC4, 3
         SET   (SYSCR2).5         ; SYSCR2<SYSCK> ← "1" (Switches the main
                                  ; system clock to the low-frequency clock)
         CLR   (SYSCR2).7         ; SYSCR2<XEN> ← "0" (Turns off low-frequency
                                  ; oscillation)
         RETI
          :      :
VINTTC   DW    PINTTC4            ; INTTC4 vector table
```

b. Warm-up counter mode for high-frequency
(SLOW1 → SLOW2 → NORMAL2 → NORMAL1)

In this mode, it can obtain the warm-up period till the oscillation for high frequency (fc) is stabilized.

Before timer/counter is started, turn on high-frequency oscillation by setting SYSCR2<XEN> to “1”.

After timer/counter is started by setting TCmCR<TCmS>, the contents of TTREGm, TTREGn are compared with the contents of up counter. If a match is found, an INTTCm interrupt is generated, and the counter is cleared to “0”.

In the interrupt service program, stop the timer/counter and change the system clock to high-frequency clock by clearing SYSCR2<SYSCK> to “0”.

After that, halt the low-frequency oscillation by clearing SYSCR2<XTEN> to “0”.

Table 2.8.9 Warm-up Period for High-frequency (at fc = 16 MHz)

Min (at TTREGm, n = 0800H)	Max (at TTREGm, n = F800H)
128 μs	3.97 ms

Example: Switching to the NORMAL1 mode after high-frequency clock has stabilized by using TC4, 3.

```
          SET    (SYSCR2).7           ;  SYSCR2<XEN> ← “1”
          LD     (TC3CR),63H          ;  TFF3 = “0”, fc for source clock, sets 16-bit mode
          LD     (TC4CR),05H          ;  TFF4 = “0”, sets warm-up counter mode
          LD     (TTREG3),0F800H      ;  Sets warm-up time  (depend on oscillator
                                      ;  characteristics)
          DI                          ;  IMF ← “0”
          SET    (EIRH).3             ;  Enables INTTC4
          EI                          ;  IMF ← “1”
          SET    (TC4CR).3            ;  Starts TC4, 3
           |       |
PINTTC4   CLR    (TC4CR).3            ;  Stops TC4, 3
          CLR    (SYSCR2).5           ;  SYSCR2<SYSCK> ← “0” (Switches the main
                                      ;  system clock to the high-frequency clock)
          CLR    (SYSCR2).6           ;  SYSCR2<XTEN> ← “0” (Turns off high-frequency
                                      ;  oscillation)
          RETI
           |       |
VINTTC4   DW     PINTTC4              ;  INTTC4 vector table
```

2.9 UART (Asynchronous serial interface)

The TMP86CM25A has 1 channel of UART (Asynchronous serial interface).

The UART is connected to external devices via RXD and TXD. RXD is also used as P15; TXD, as P16. To use P15 or P16 as the RXD or TXD pin, set P1 port output latches to "1".

2.9.1 Configuration

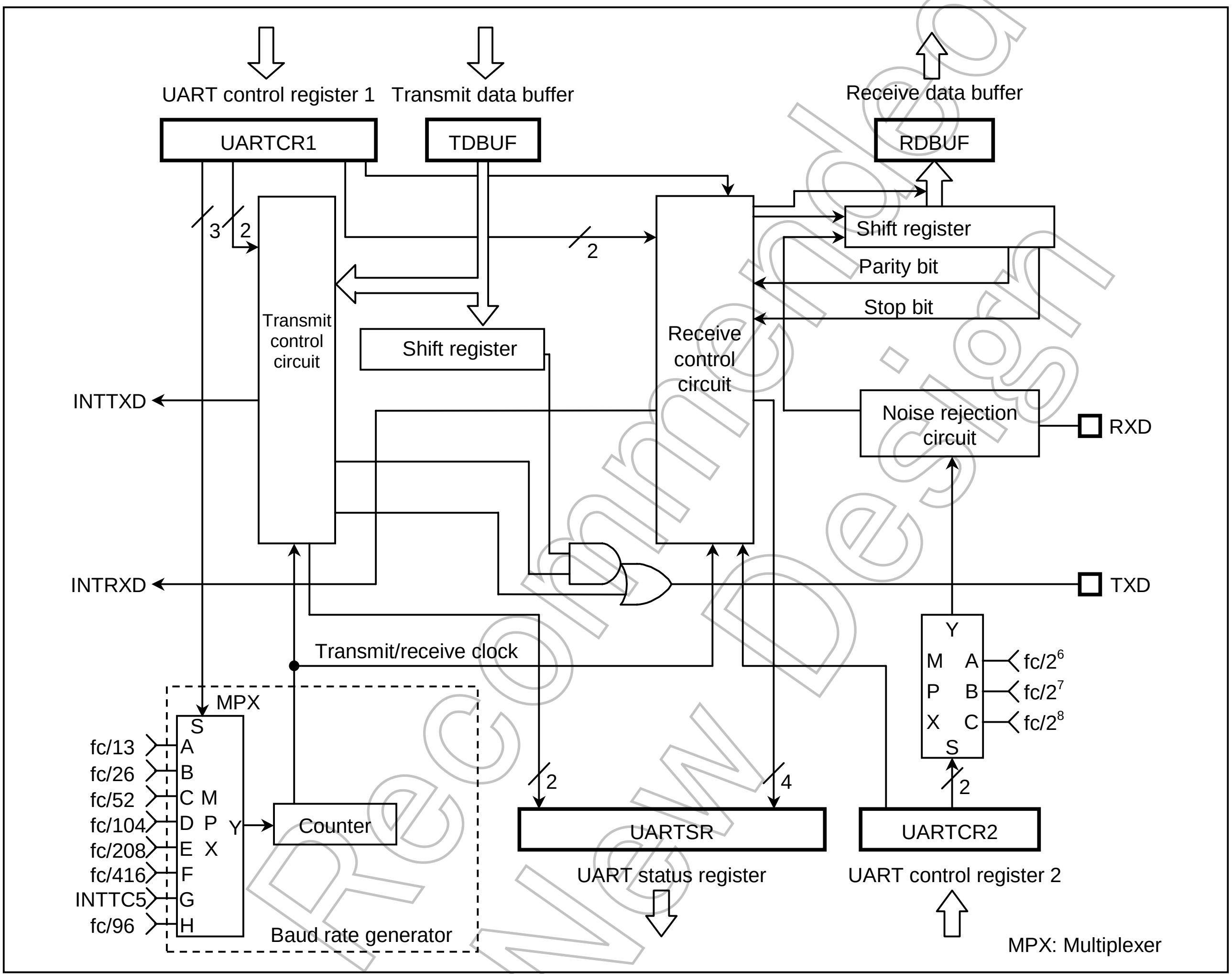


Figure 2.9.1 UART

2.9.2 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 (0025_H)

	7	6	5	4	3	2	1	0
UARTCR1 (0025_H)	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.

	7	6	5	4	3	2	1	0
UARTCR2 (0026_H)						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 2.9.2 UART Control Register

UARTSR (0025_H)

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

(Initial value: 0000 11**)

TBEP	Transmit data buffer empty flag	0: Transmit data buffer full 1: Transmit data buffer empty	Read only
TEND	Transmit end flag	0: Transmitting 1: Transmit end	
RBFL	Receive data buffer full flag	0: Receive data buffer empty 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 ($0F9B_H$)

7	6	5	4	3	2	1	0

Read only (Initial value: 0000 0000)

UART Transmit Data Buffer

TDBUF ($0F9B_H$)

7	6	5	4	3	2	1	0

Write only (Initial value: 0000 0000)

Figure 2.9.3 UART Status Register and Data Buffer Registers

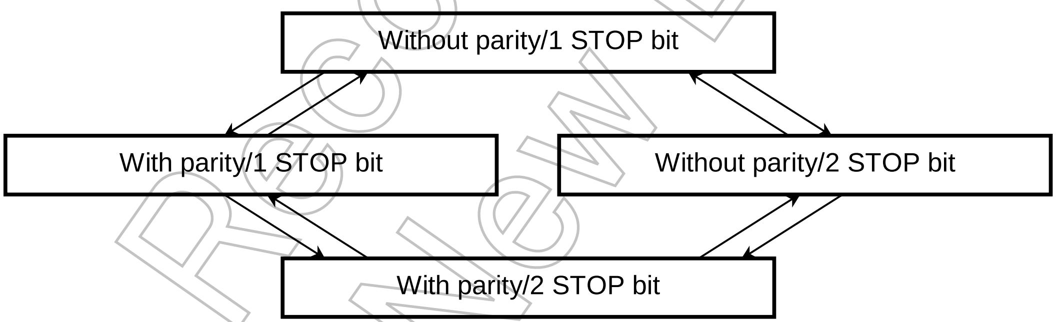
2.9.3 Transfer Data Format

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 2.9.1 Transfer Data Format

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

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



2.9.4 Transfer Rate

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

Table 2.9.2 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:

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

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

2.9.5 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).

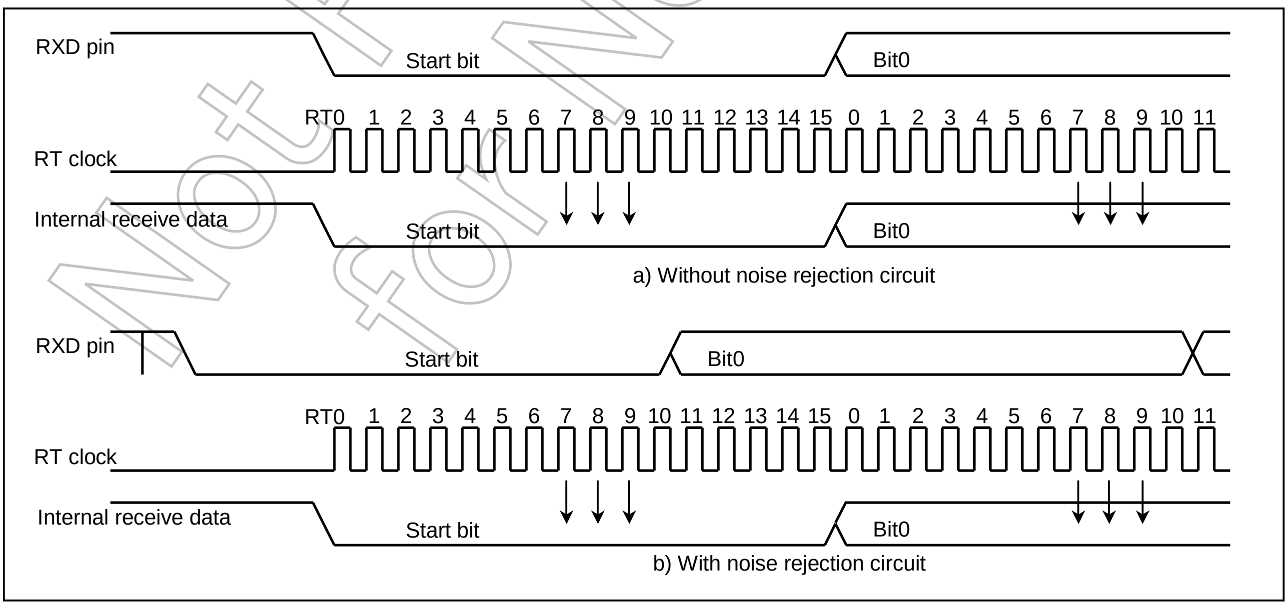


Figure 2.9.4 Data Sampling

2.9.6 STOP Bit Length

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

2.9.7 Parity

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

2.9.8 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.

2.9.9 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.

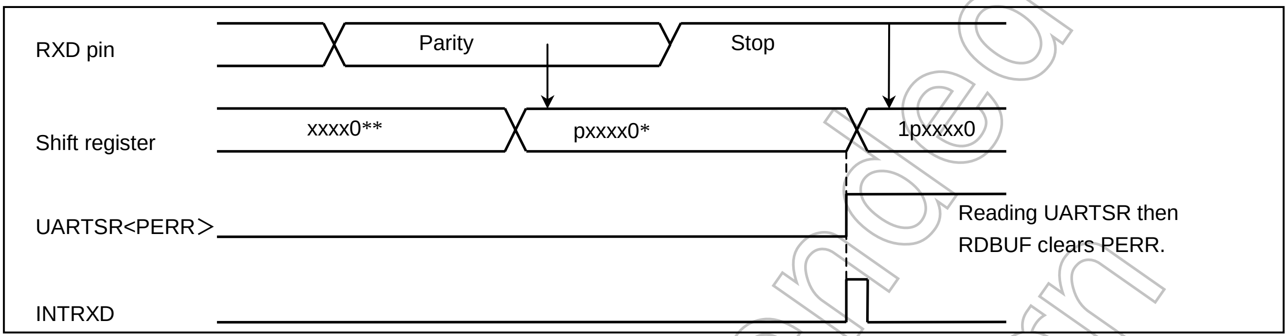


Figure 2.9.5 Generation of Parity Error

(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.

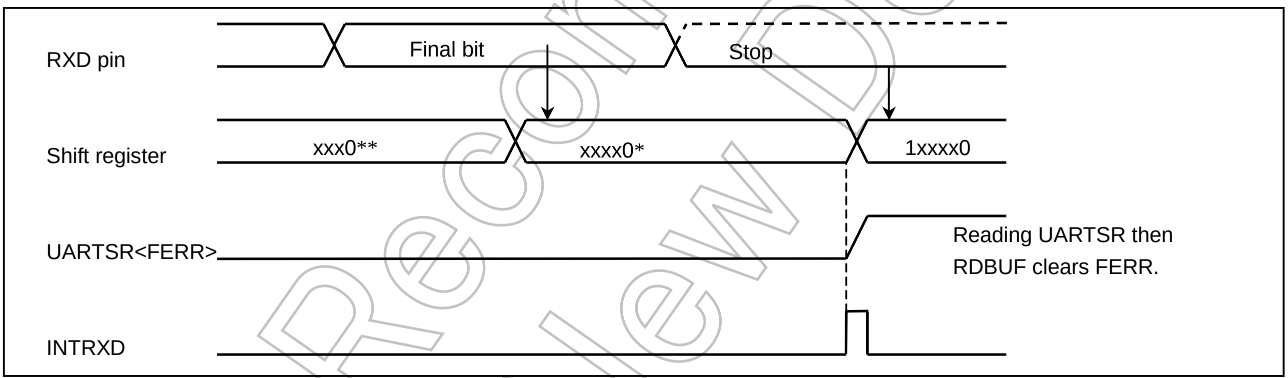


Figure 2.9.6 Generation of Framing Error

(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.

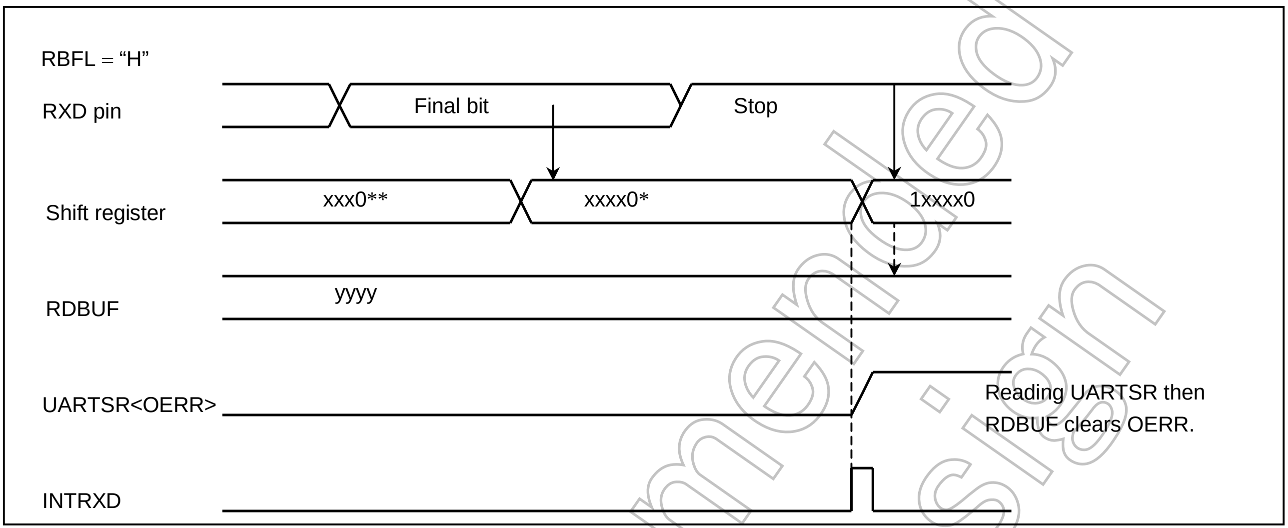


Figure 2.9.7 Generation of Overrun Error

(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.

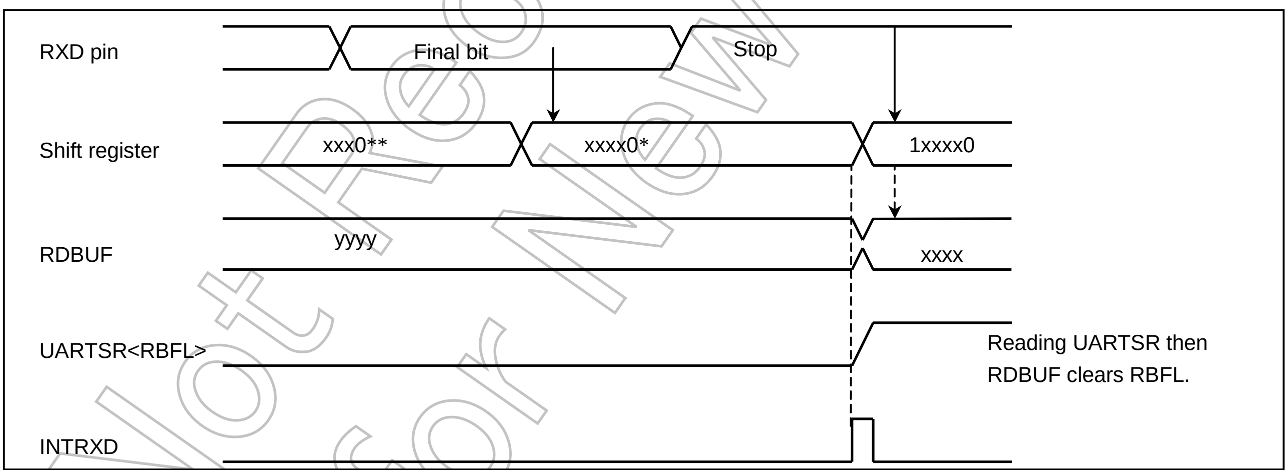


Figure 2.9.8 Generation of Receive Buffer Full

(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.

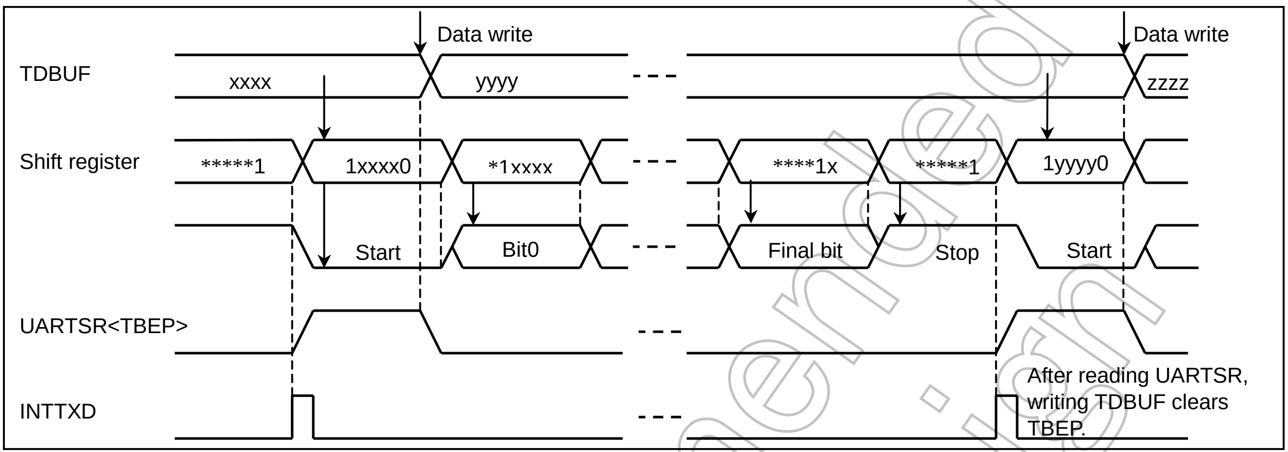


Figure 2.9.9 Generation of Transmit Buffer Empty

(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.

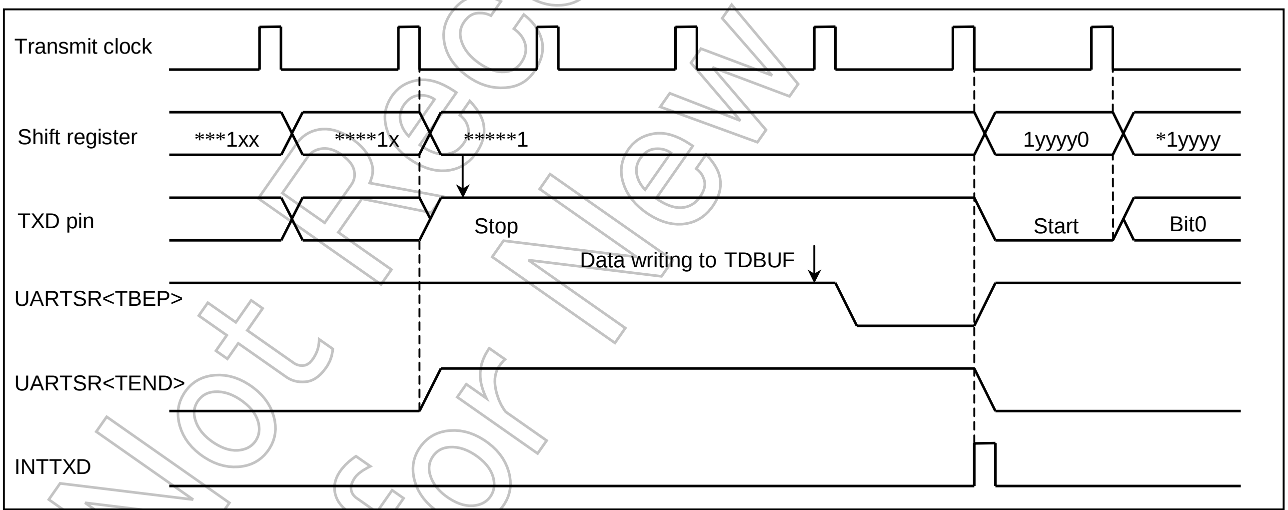


Figure 2.9.10 Generation of Transmit Buffer Empty

2.10 Serial Interface (SIO0)

The TMP86CM25A has two channels of clocked-synchronous 8-bit serial interface. (SIO0,SIO1) Serial interface has an 8-byte transmit and receive data buffer that can automatically and continuously transfer up to 64 bits of data.

The serial interface 0 is connected to external devices via pins P16 (SO0), P15 (SI0) and P17 ($\overline{\text{SCK0}}$). The serial interface pins are also used as port P1. When these pins are used as serial interface pins, the correspondence output latch should be set to “1”. In the transmit mode, pin P15 can be used as normal I/O port, and in the receive mode, the pin P16 can be used as normal I/O ports.

2.10.1 Configuration

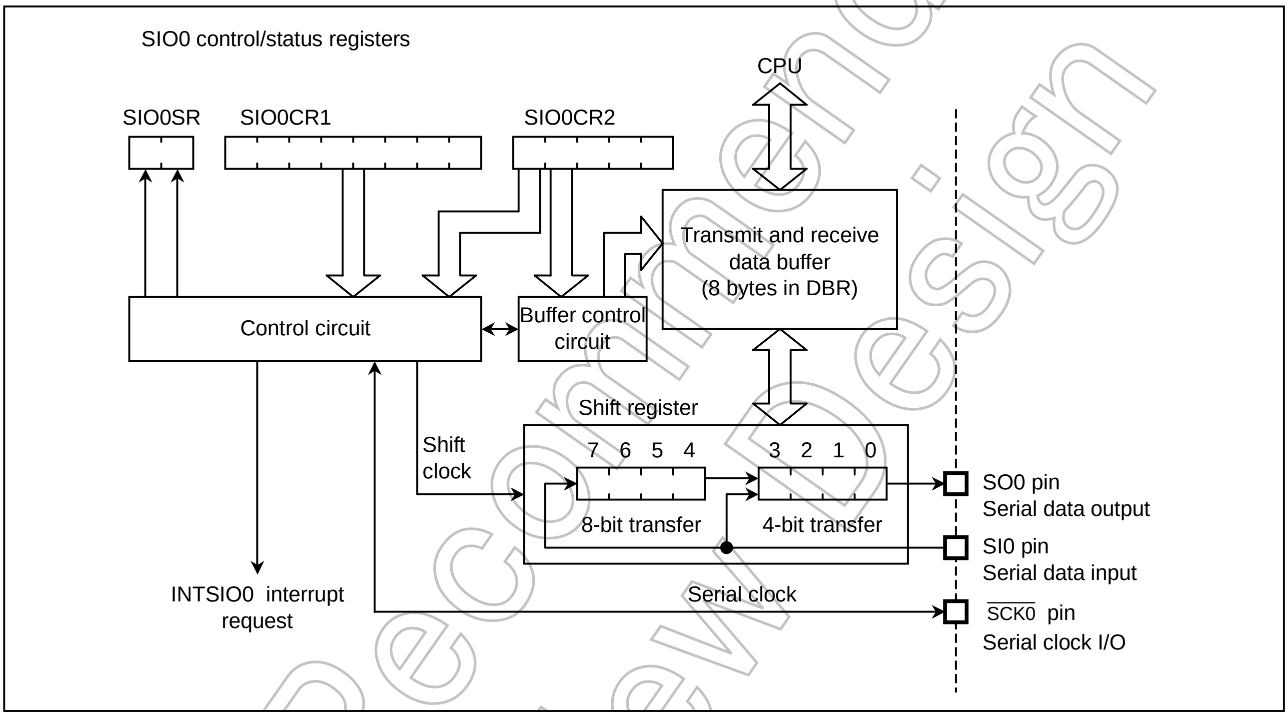


Figure 2.10.1 Serial Interfaces 0 (SIO0)

2.10.2 Control

The serial interface is controlled by SIO control registers (SIO0CR1/SIO0CR2). The serial interface status can be determined by reading SIO status register (SIO0SR).

The transmit and receive data buffer is controlled by the BUF (Bits 2 to 0 in SIO0CR2). The data buffer is assigned to address $0F90_H$ to $0F97_H$ for SIO in the DBR area, and can continuously transfer up to 8 words (Bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO0) is generated.

When the internal clock is used as the serial clock in the 8-bit receive mode and the 8-bit transmit/receive mode, a fixed interval wait can be applied to the serial clock for each word transferred. Four different wait times can be selected with WAIT (Bits 4 and 3 in SIO0CR2).

SIO0 Control Register 1

SIO0CR1 ($0F99_H$)	7	6	5	4	3	2	1	0
	SIOS	SIOINH	SIOM			SCK		

(Initial value: 0000 0000)

SIOS	Indicate transfer start/stop	0: Stop 1: Start	Write only
SIOINH	Continue/Abort transfer	0: Continue transfer 1: Abort transfer (Automatically cleared after abort)	Write only
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit/receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved	Write only
SCK	Serial clock select	(see table below)	Write only

SCK (Serial clock select):

	NORMAL1/2, IDLE1/2 Mode DV7CK = 0	NORMAL1/2, IDLE1/2 Mode DV7CK = 1	SLOW, SLEEP Mode
000	$fc/2^{13}$	$fs/2^5$	$fs/2^5$
001	$fc/2^8$	$fc/2^8$	–
010	$fc/2^7$	$fc/2^7$	–
011	$fc/2^6$	$fc/2^6$	–
100	$fc/2^5$	$fc/2^5$	–
101	$fc/2^4$	$fc/2^4$	–
110	Reserved		–
111	External clock (input from $\overline{SCK1}$ (P17) pin)		

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

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

Note 3: SIO0CR1 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

SIO0 Status Register

SIO0SR ($0F99_H$)	7	6	5	4	3	2	1	0
	SIOF	SEF						

(Initial value: 00** ****)

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

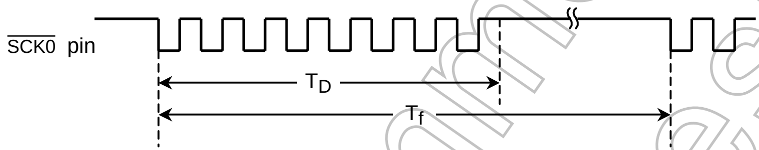
Figure 2.10.2 SIO0 Control Register and Status Register (1/2)

SIO0 Control Register 2

SIO0CR2 (0F99H)	7	6	5	4	3	2	1	0	
				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets “00” except 8-bit transmit/receive mode. 00: $T_f = T_D$ (Non wait) 01: $T_f = 2T_D$ (Wait) 10: $T_f = 4T_D$ (Wait) 11: $T_f = 8T_D$ (Wait)		Write only
BUF	Number of transfer words		SIO0 Buffer	Write only
		000: 1 word transfer	0F90H	
		001: 2 words transfer	0F90H to 0F91H	
		010: 3 words transfer	0F90H to 0F92H	
		011: 4 words transfer	0F90H to 0F93H	
		100: 5 words transfer	0F90H to 0F94H	
		101: 6 words transfer	0F90H to 0F95H	
		110: 7 words transfer	0F90H to 0F96H	
		111: 8 words transfer	0F90H to 0F97H	

Note :1 T_f: Frame time, T_D: Data transfer time



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

Note 3: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO, the first buffer address transmitted is $0F90_H$.

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

Note 5: SIO0CR2 must be set when the serial interface is stopped (SIOF = 0).

Note 6: SIO0CR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Note 7: *: Don’t care

Figure 2.10.3 SIO Control Register and Status Register (2/2)

(1) Serial clock

a. Clock source

SIO0CR1<SCK> is able to select the following:

1. Internal clock

Any of seven frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK0}$ pin. The $\overline{SCK0}$ pin goes high when transfer starts.

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

Table 2.10.1 Serial Clock Rate

	NORMAL1/2, IDLE1/2 Modes				SLOW, SLEEP Modes	
	DV7CK = 0		DV7CK = 1			
SCK	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate
000	$fc/2^{13}$	1.91 Kbps	$fs/2^5$	1024 bps	$fs/2^5$	1024 bps
001	$fc/2^8$	61.04 Kbps	$fc/2^8$	61.04 Kbps	–	–
010	$fc/2^7$	122.07 Kbps	$fc/2^7$	122.07 Kbps	–	–
011	$fc/2^6$	244.14 Kbps	$fc/2^6$	244.14 Kbps	–	–
100	$fc/2^5$	488.28 Kbps	$fc/2^5$	488.28 Kbps	–	–
101	$fc/2^4$	976.56 Kbps	$fc/2^4$	976.56 Kbps	–	–
110	–	–	–	–	–	–
111	External		External		External	

1 Kbit = 1024 bits
(fc = 16 MHz, fs = 32.768 kHz)

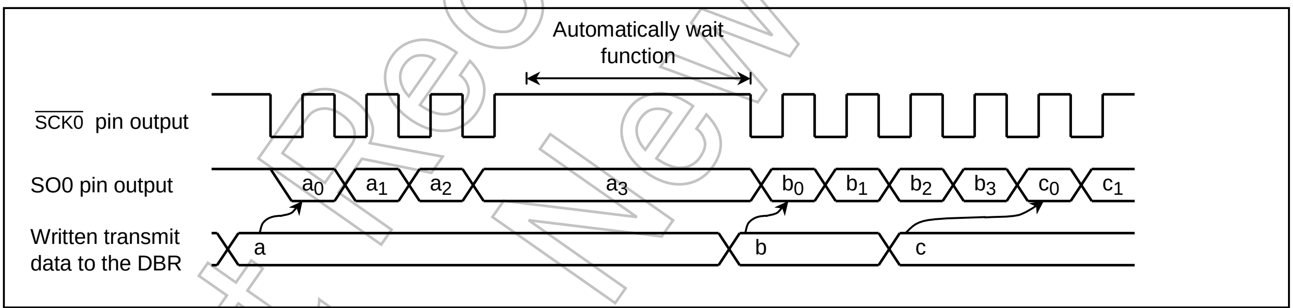
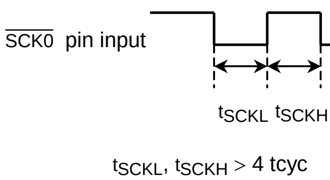


Figure 2.10.4 Clock Source (Internal clock)

2. External clock

An external clock connected to the $\overline{SCK0}$ pin is used as the serial clock. In this case, the P17 ($\overline{SCK0}$) must be set to the input mode. To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program.



Note: t_{cyc} = 4/fc (in NORAML1/2, IDLE1/2 modes)
4/fs (in SLOW, SLEEP modes)

b. Shift edge

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

1. Leading edge

   Transmitted data are shifted on the leading edge of the serial clock (Falling edge of the $\overline{\text{SCK0}}$ pin input/output).

2. Trailing edge

   Received data are shifted on the trailing edge of the serial clock (Rising edge of the $\overline{\text{SCK0}}$ pin input/output).

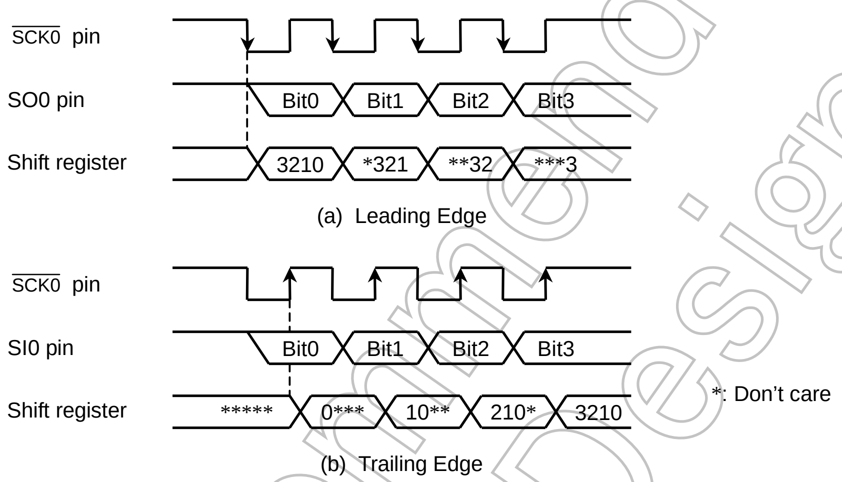


Figure 2.10.5 Shift Edge

(2) Number of bits to transfer

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

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

(3) Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred is loaded to SIO0CR2<BUF>.

An INTSIO0 interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

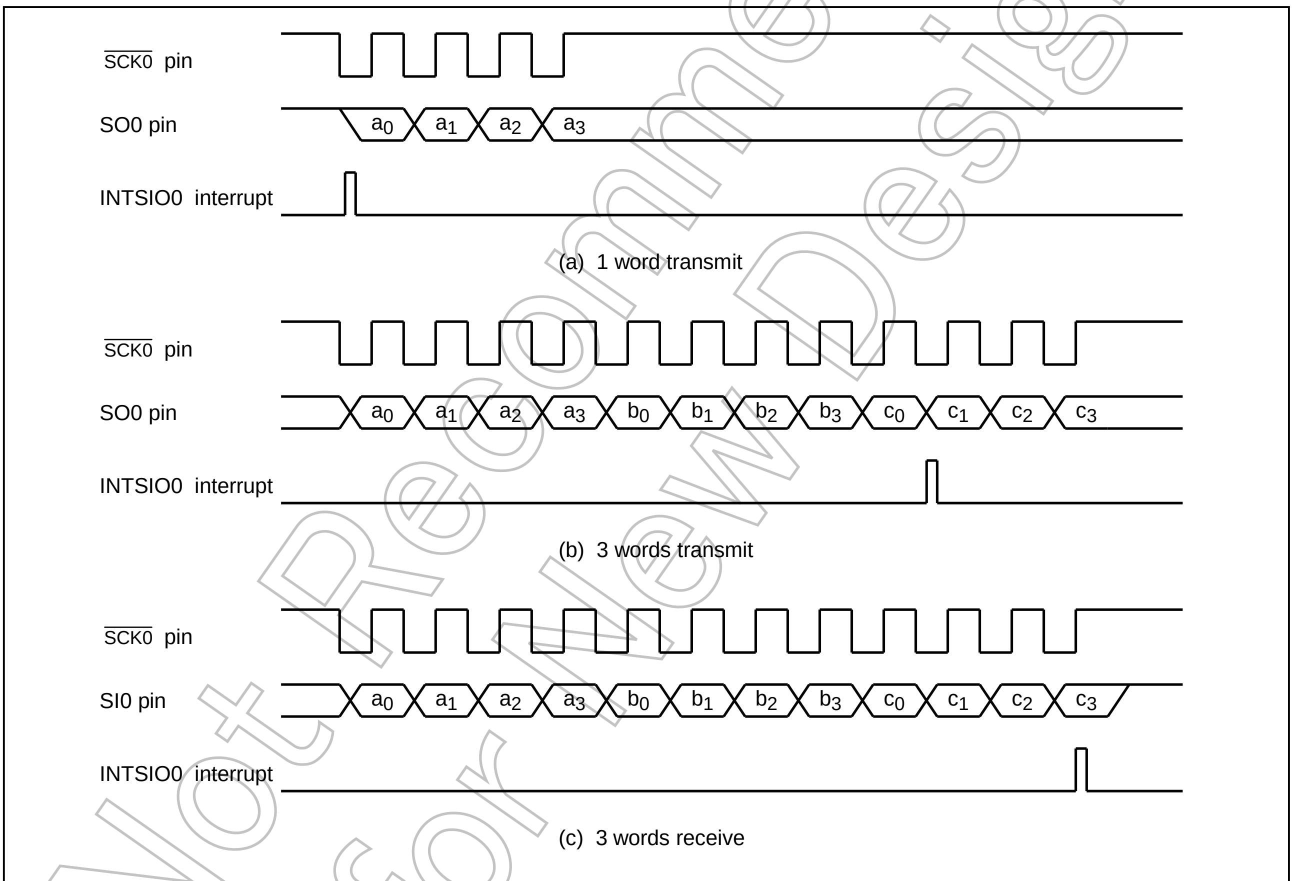


Figure 2.10.6 Number of Bits to Transfer (Example: 4-bit serial transfer)

2.10.3 Transfer Mode

SIO0CR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

(1) 4-bit and 8-bit transmit modes

In these modes, the SIO0CR1 is set to the transmit mode and then the data to be transmitted first are written to the data buffer registers (DBR). After the data are written, the transmission is started by setting SIO0CR1<SIOS> to “1”. The data are then output sequentially to the SO0 pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO0 (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIO0CR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

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

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIO0CR1<SIOS> to “0” or setting SIO0CR1<SIOINH> to “1” in buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIO0SR<SIOF> because SIO0SR<SIOF> is cleared to “0” when a transfer is completed.

When SIO0CR1<SIOINH> is set, the transmission is immediately ended and SIO0SR<SIOF> is cleared to “0”.

When an external clock is used, it is also necessary to clear SIO0CR1<SIOS> to “0” before shifting the next data; otherwise, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIO0CR1<SIOS> should be cleared to “0”, then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<SIOF> has been cleared to “0”.

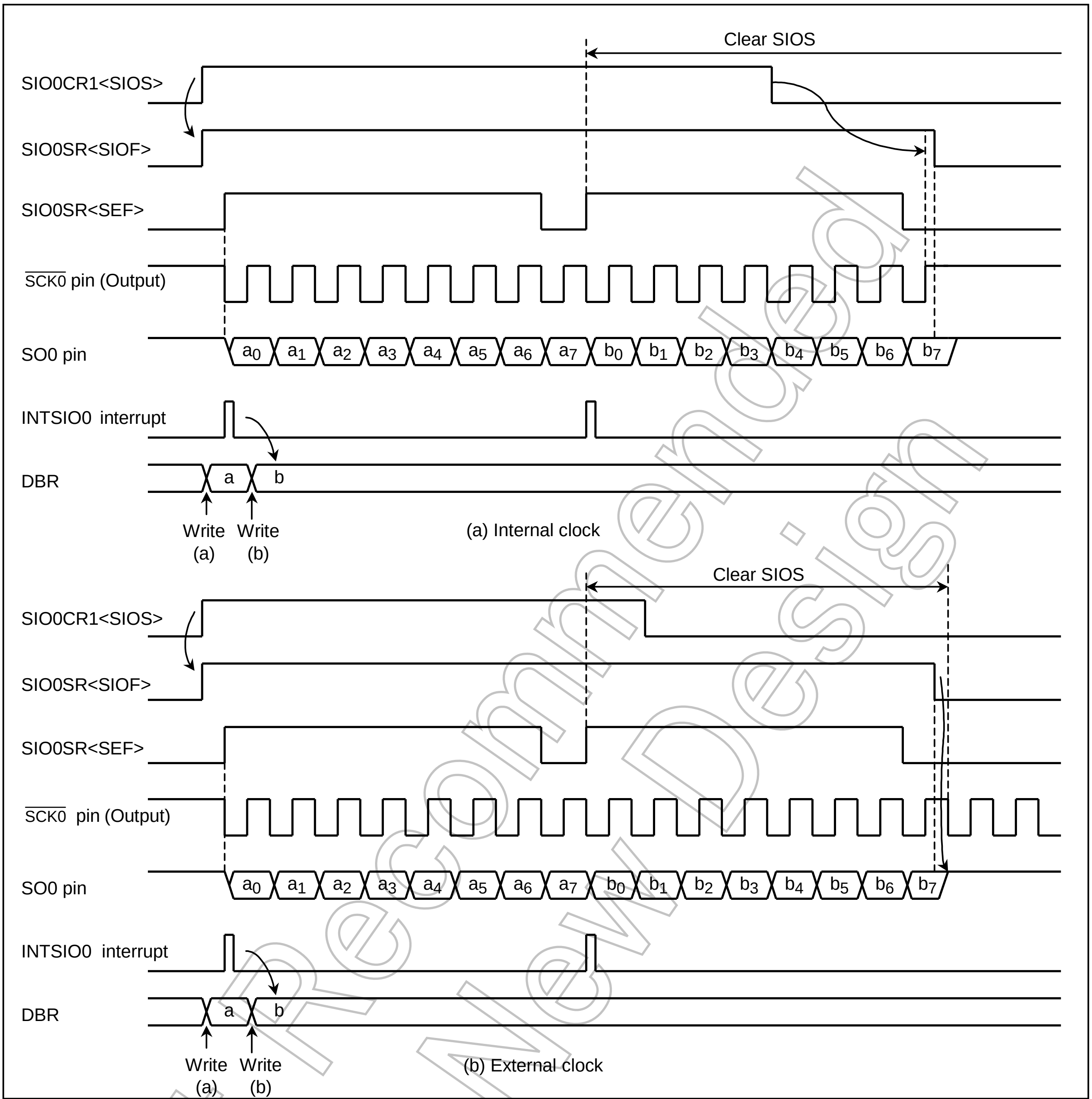


Figure 2.10.7 Transfer Mode (Example: 8 bits, 1 word transfer)

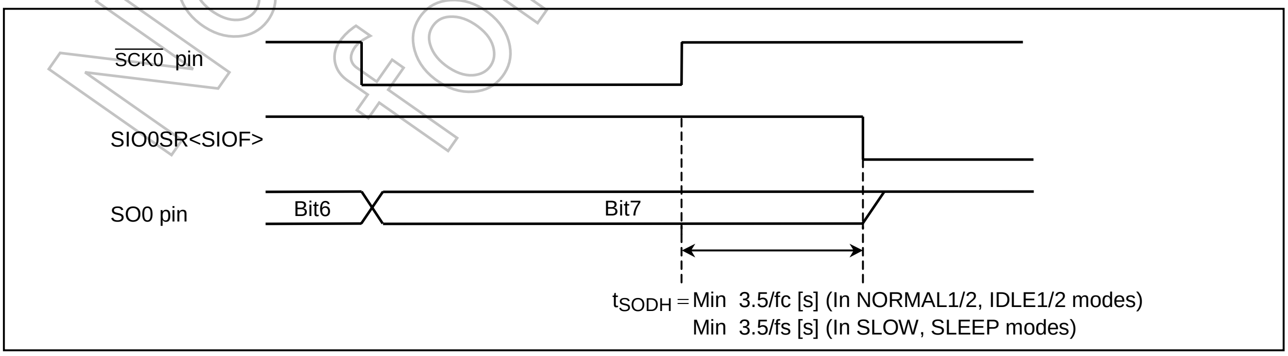


Figure 2.10.8 Transmitted Data Hold Time at End of Transmit

(2) 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIO0CR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIO0CR2<BUF> has been received, an INTSIO0 (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

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

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

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

The receiving is ended by clearing SIO0CR1<SIOS> to “0” or setting SIO0CR1<SIOINH> to “1” in buffer full interrupt service program. When SIO0CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO0CR1<SIOS >cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIO0SR<SIOF>. SIO0SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO0CR1<SIOINH> is set, the receiving is immediately ended and SIO0SR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIO0CR1<SIOS> should be cleared to “0” then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIO0CR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO0CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

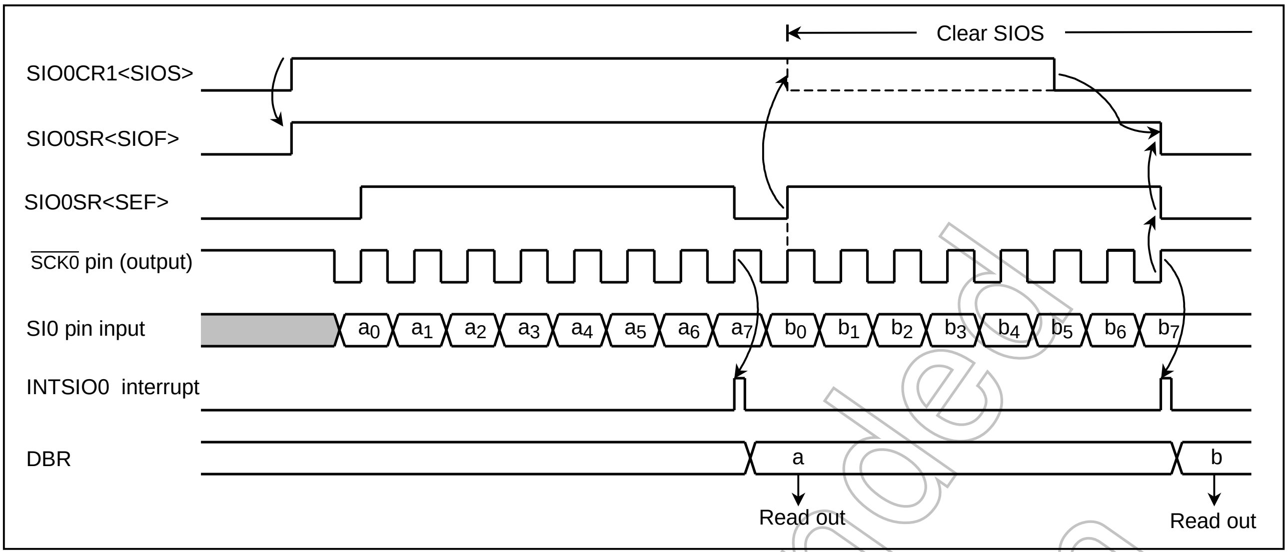


Figure 2.10.9 Receive Mode (Example: 8 bits, 1 word, internal clock)

(3) 8-bit transmit/receive mode

After setting the control registers to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable transceiving by setting <SIOS> to “1”. When transmitting, the data are output from the SO0 pin at leading edges of the serial clock. When receiving, the data are input to the SI0 pin at the trailing edges of the serial clock. 8-bit data are transferred from the shift register to the data buffer register. An INTSIO0 interrupt is generated when the number of data words specified with the <BUF> has been transferred. The interrupt service program reads the received data from the data buffer register and then writes the data to be transmitted. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the received data.

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

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

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

The transmit/receive operation is ended by clearing SIO0CR1<SIOS> to “0” or setting SIO0SR<SIOINH> to “1” in interrupt service program.

When SIO0CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO0SR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIO0CR1<SIOS> should be cleared to “0”, then SIO0CR2<BUF> must be rewritten after confirming that SIO0SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIO0CR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO0CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

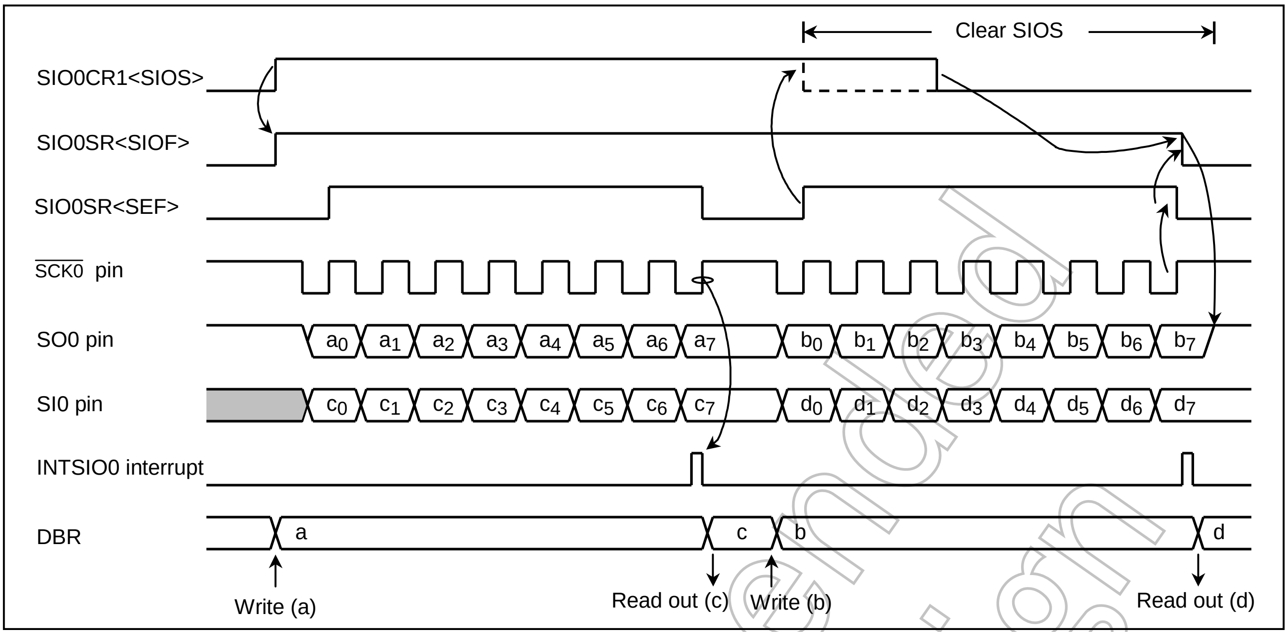


Figure 2.10.10 Transmit/Receive Mode (Example: 8 bits, 1 word, internal clock)

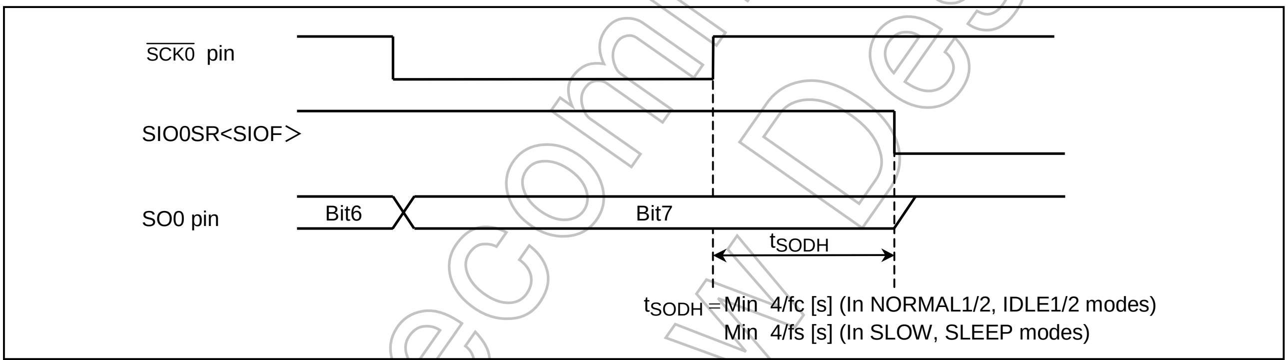


Figure 2.10.11 Transmitted Data Hold Time at End of Transmit/Receive

2.11 Serial Interface (SIO1)

The serial interface 1 is connected to external devices via pins P76 (SO1), P75 (SI1) and P77 ($\overline{\text{SCK1}}$). The serial interface pins are also used as port P7. When these pins are used as serial interface pins, the correspondence output latch should be set to “1”. In the transmit mode, pin P75 can be used as normal I/O port, and in the receive mode, the pin P76 can be used as normal I/O ports.

2.11.4 Configuration

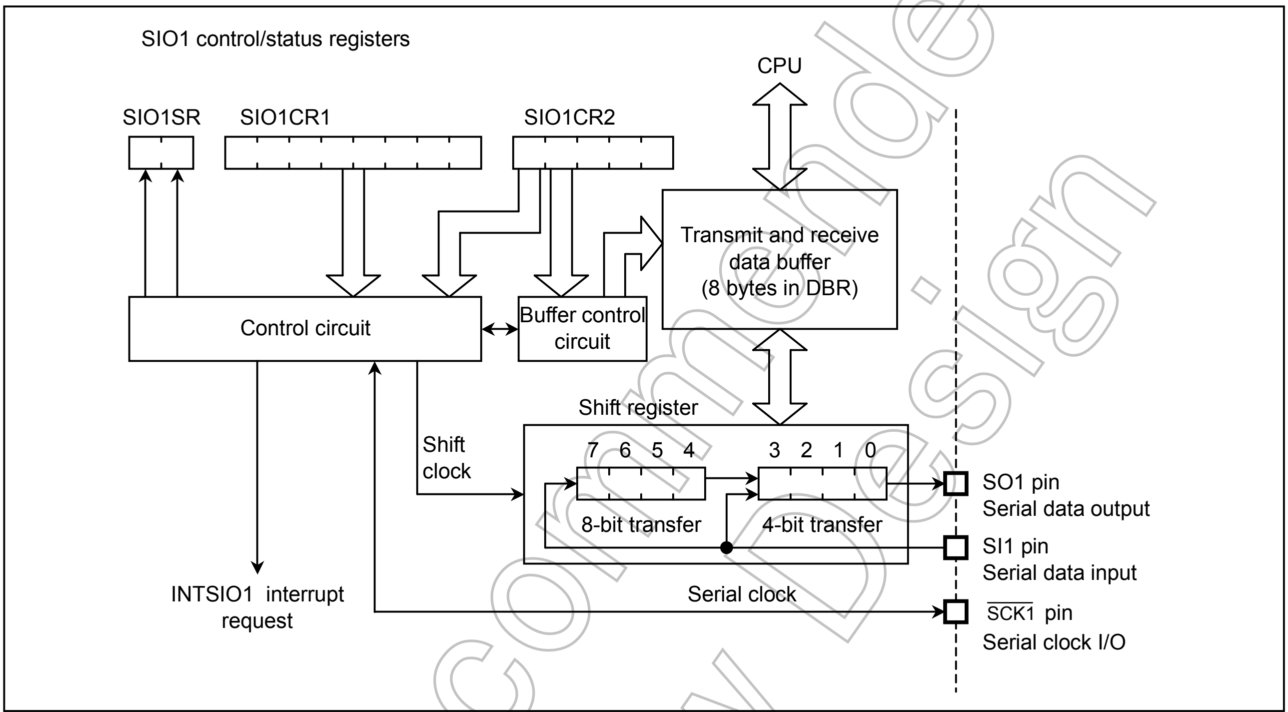


Figure 2.11.1 Serial Interfaces 1 (SIO1)

2.11.5 Control

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

The transmit and receive data buffer is controlled by the BUF (Bits 2 to 0 in SIO1CR2). The data buffer is assigned to address $0F90_H$ to $0F97_H$ for SIO in the DBR area, and can continuously transfer up to 8 words (Bytes or nibbles) at one time. When the specified number of words has been transferred, a buffer empty (in the transmit mode) or a buffer full (in the receive mode or transmit/receive mode) interrupt (INTSIO1) is generated.

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

SIO1 Control Register 1

SIO1CR1 ($0FA8_H$)	7	6	5	4	3	2	1	0	
	SIOS	SIOINH	SIOM			SCK			(Initial value: 0000 0000)

SIOS	Indicate transfer start/stop	0: Stop 1: Start	Write only
SIOINH	Continue/Abort transfer	0: Continue transfer 1: Abort transfer (Automatically cleared after abort)	Write only
SIOM	Transfer mode select	000: 8-bit transmit mode 010: 4-bit transmit mode 100: 8-bit transmit/receive mode 101: 8-bit receive mode 110: 4-bit receive mode Except the above: Reserved	Write only
SCK	Serial clock select	(see table below)	Write only

SCK (Serial clock select):

	NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode
	DV7CK = 0	DV7CK = 1	
000	$fc/2^{13}$	$fs/2^5$	$fs/2^5$
001	$fc/2^8$	$fc/2^8$	–
010	$fc/2^7$	$fc/2^7$	–
011	$fc/2^6$	$fc/2^6$	–
100	$fc/2^5$	$fc/2^5$	–
101	$fc/2^4$	$fc/2^4$	–
110	Reserved		–
111	External clock (Input from $\overline{SCK1}$ (P77) pin)		

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

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

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

SIO1 Status Register

SIO1SR ($0FA9_H$)	7	6	5	4	3	2	1	0	
	SIOF	SEF							(Initial value: 00** ****)

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

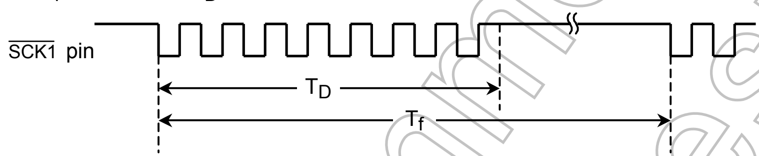
Figure 2.11.2 SIO1 Control Register and Status Register (1/2)

SIO1 Control Register 2

SIO1CR2 (0FA9$_H$)	7	6	5	4	3	2	1	0	
				WAIT		BUF			(Initial value: ***0 0000)

WAIT	Wait control	Always sets "00" except 8-bit transmit/receive mode. 00: $T_f = T_D$ (Non-wait) 01: $T_f = 2T_D$ (wait) 10: $T_f = 4T_D$ (wait) 11: $T_f = 8T_D$ (wait)		Write only
BUF	Number of transfer words		SIO1 Data Buffer	
		000: 1 word transfer	0F90H	
		001: 2 words transfer	0F90H to 0F91H	
		010: 3 words transfer	0F90H to 0F92H	
		011: 4 words transfer	0F90H to 0F93H	
		100: 5 words transfer	0F90H to 0F94H	
		101: 6 words transfer	0F90H to 0F95H	
		110: 7 words transfer	0F90H to 0F96H	
		111: 8 words transfer	0F90H to 0F97H	

Note 1: T_f: Frame time, T_D: Data transfer time



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

Note 3: Transmitting starts at the lowest address. Received data are also stored starting from the lowest address to the highest address. For example, in the case of SIO, the first buffer address transmitted is 0F90H.

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

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

Note 6: SIO1CR2 is write-only register, which cannot access any of in read-modify-write instruction such as bit operate, etc.

Note 7: *: Don't care

Figure 2.11.3 SIO1 Control Register and Status Register (2/2)

(1) Serial clock

a. Clock source

SIO1CR1<SCK> is able to select the following:

1. Internal clock

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

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

Table 2.11.1 Serial Clock Rate

	NORMAL1/2, IDLE1/2 Modes				SLOW, SLEEP Modes	
	DV7CK = 0		DV7CK = 1			
SCK	Clock	Baud Rate	Clock	Baud Rate	Clock	Baud Rate
000	$fc/2^{13}$	1.91 Kbps	$fs/2^5$	1024 bps	$fs/2^5$	1024 bps
001	$fc/2^8$	61.04 Kbps	$fc/2^8$	61.04 Kbps	–	–
010	$fc/2^7$	122.07 Kbps	$fc/2^7$	122.07 Kbps	–	–
011	$fc/2^6$	244.14 Kbps	$fc/2^6$	244.14 Kbps	–	–
100	$fc/2^5$	488.28 Kbps	$fc/2^5$	488.28 Kbps	–	–
101	$fc/2^4$	976.56 Kbps	$fc/2^4$	976.56 Kbps	–	–
110	–	–	–	–	–	–
111	External		External		External	

1 Kbit = 1024 bits
(fc = 16 MHz, fs = 32.768 kHz)

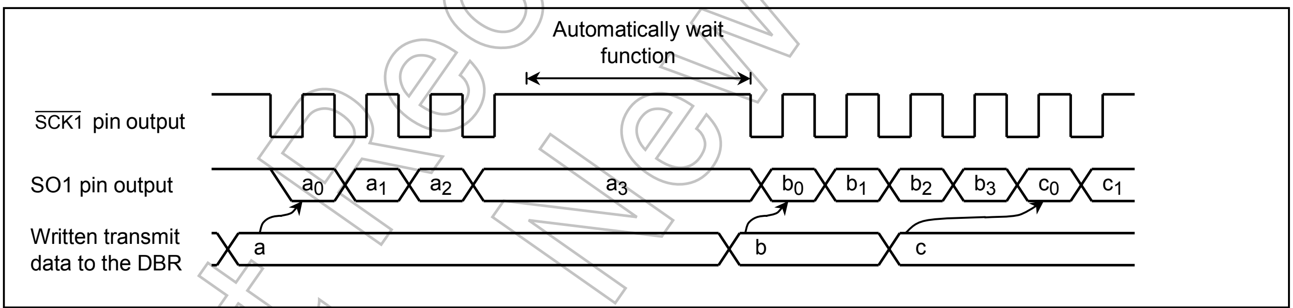
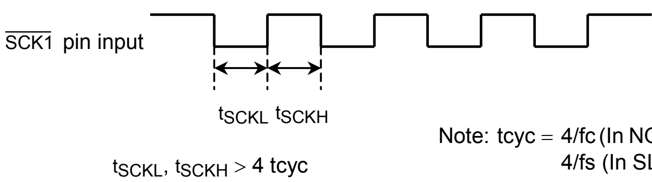


Figure 2.11.4 Clock Source (Internal clock)

2. External clock

An external clock connected to the $\overline{SCK1}$ pin is used as the serial clock. In this case, the P77 ($\overline{SCK1}$) must be set to the input mode. To ensure shifting, a pulse width of at least 4 machine cycles is required. This pulse is needed for the shift operation to execute certainly. Actually, there is necessary processing time for interrupting, writing, and reading. The minimum pulse is determined by setting the mode and the program.



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

b. Shift edge

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

1. Leading edge

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

2. Trailing edge

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

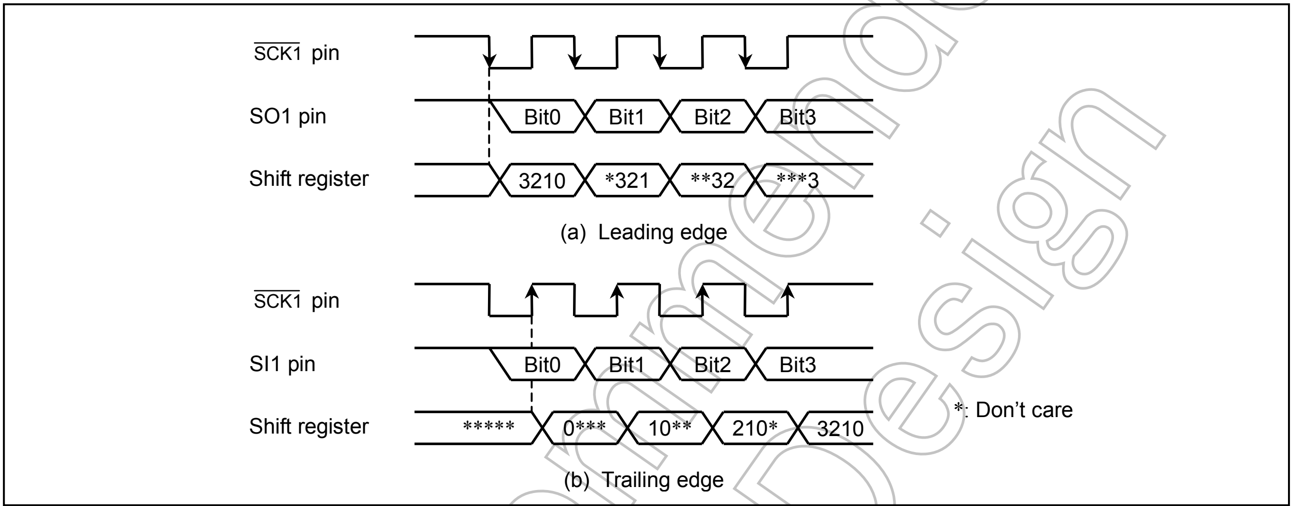


Figure 2.11.5 Shift Edge

(2) Number of bits to transfer

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

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

(3) Number of words to transfer

Up to 8 words consisting of 4 bits of data (4-bit serial transfer) or 8 bits (8-bit serial transfer) of data can be transferred continuously. The number of words to be transferred is loaded to SIO1CR2<BUF>.

An INTSIO1 interrupt is generated when the specified number of words has been transferred. If the number of words is to be changed during transfer, the serial interface must be stopped before making the change. The number of words can be changed during automatic-wait operation of an internal clock. In this case, the serial interface is not required to be stopped.

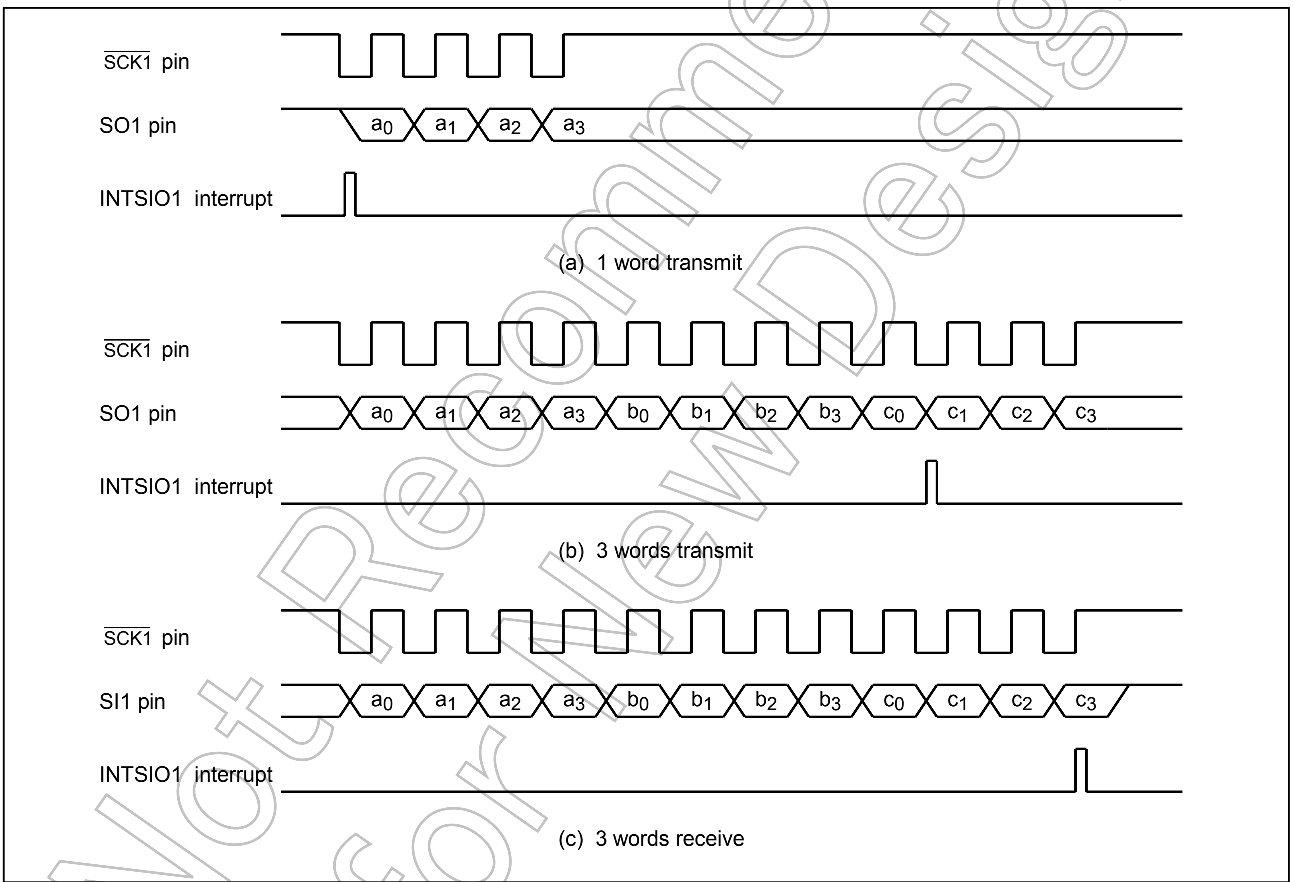


Figure 2.11.6 Number of Bits to Transfer (Example: 4-bit serial transfer)

2.11.6 Transfer Mode

SIO1CR1<SIOM> is used to select the transmit, receive, or transmit/receive mode.

(1) 4-bit and 8-bit transmit modes

In these modes, the SIO1CR1 is set to the transmit mode and then the data to be transmitted first are written to the data buffer registers (DBR). After the data are written, the transmission is started by setting SIO1CR1<SIOS> to "1". The data are then output sequentially to the SO0 pin in synchronous with the serial clock, starting with the least significant bit (LSB). As soon as the LSB has been output, the data are transferred from the data buffer register to the shift register. When the final data bit has been transferred and the data buffer register is empty, an INTSIO1 (Buffer empty) interrupt is generated to request the next transmitted data.

When the internal clock is used, the serial clock will stop and an automatic-wait will be initiated if the next transmitted data are not loaded to the data buffer register by the time the number of data words specified with the SIO1CR2<BUF> has been transmitted. Writing even one word of data cancels the automatic-wait; therefore, when transmitting two or more words, always write the next word before transmission of the previous word is completed.

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

When an external clock is used, the data must be written to the data buffer register before shifting next data. Thus, the transfer speed is determined by the maximum delay time from the generation of the interrupt request to writing of the data to the data buffer register by the interrupt service program.

The transmission is ended by clearing SIO1CR1<SIOS> to "0" or setting SIO1CR1<SIOINH> to "1" in buffer empty interrupt service program. That the transmission has ended can be determined from the status of SIO1SR<SIOF> because SIO1SR<SIOF> is cleared to "0" when a transfer is completed.

When SIO1CR1<SIOINH> is set, the transmission is immediately ended and SIO1SR<SIOF> is cleared to "0".

When an external clock is used, it is also necessary to clear SIO1CR1<SIOS> to "0" before shifting the next data; otherwise, dummy data will be transmitted and the operation will end.

If it is necessary to change the number of words, SIO1CR1<SIOS> should be cleared to "0", then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to "0".

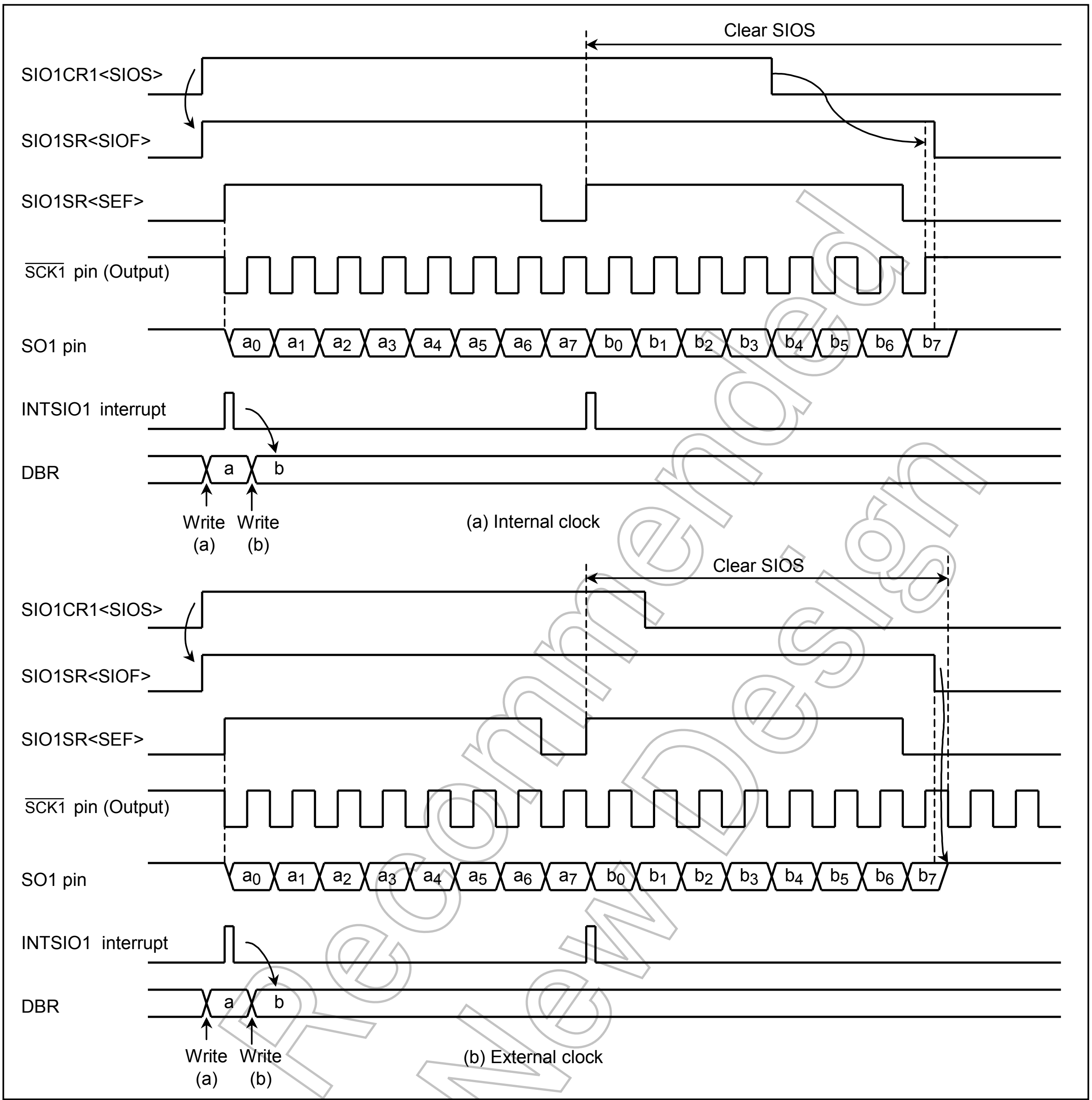


Figure 2.11.7 Transfer Mode (Example: 8 bits, 1 word transfer)

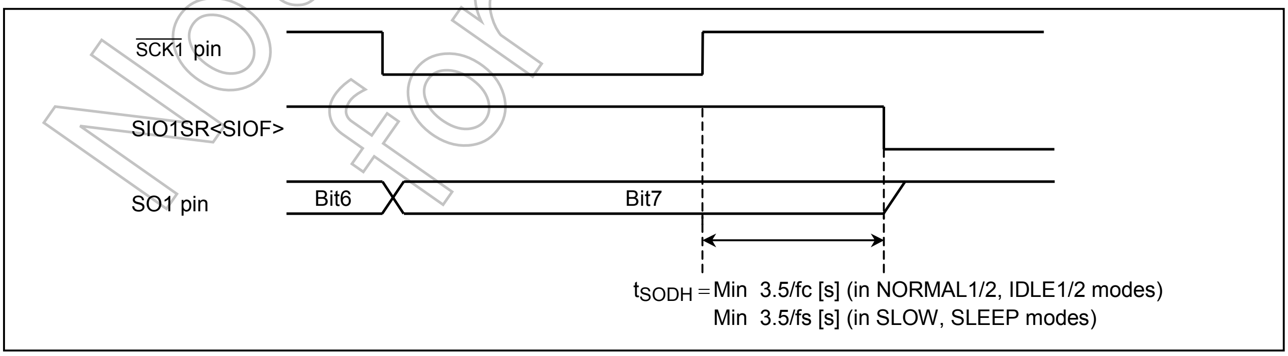


Figure 2.11.8 Transmitted Data Hold Time at End of Transmit

(2) 4-bit and 8-bit receive modes

After setting the control registers to the receive mode, set SIO1CR1<SIOS> to “1” to enable receiving. The data are then transferred to the shift register via the SI pin in synchronous with the serial clock. When one word of data has been received, it is transferred from the shift register to the data buffer register (DBR). When the number of words specified with the SIO1CR2<BUF> has been received, an INTSIO1 (Buffer full) interrupt is generated to request that these data be read out. The data are then read from the data buffer registers by the interrupt service program.

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

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

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

The receiving is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1CR1<SIOINH> to “1” in buffer full interrupt service program. When SIO1CR1<SIOS> is cleared, the current data are transferred to the buffer. After SIO1CR1<SIOS >cleared, the receiving is ended at the time that the final bit of the data has been received. That the receiving has ended can be determined from the status of SIO1SR<SIOF>. SIO1SR<SIOF> is cleared to “0” when the receiving is ended. After confirmed the receiving termination, the final receiving data is read. When SIO1CR1<SIOINH> is set, the receiving is immediately ended and SIO1SR<SIOF> is cleared to “0”. (The received data is ignored, and it is not required to be read out.)

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to “0” then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of data receiving, SIO1CR2<BUF> must be rewritten before the received data is read out.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

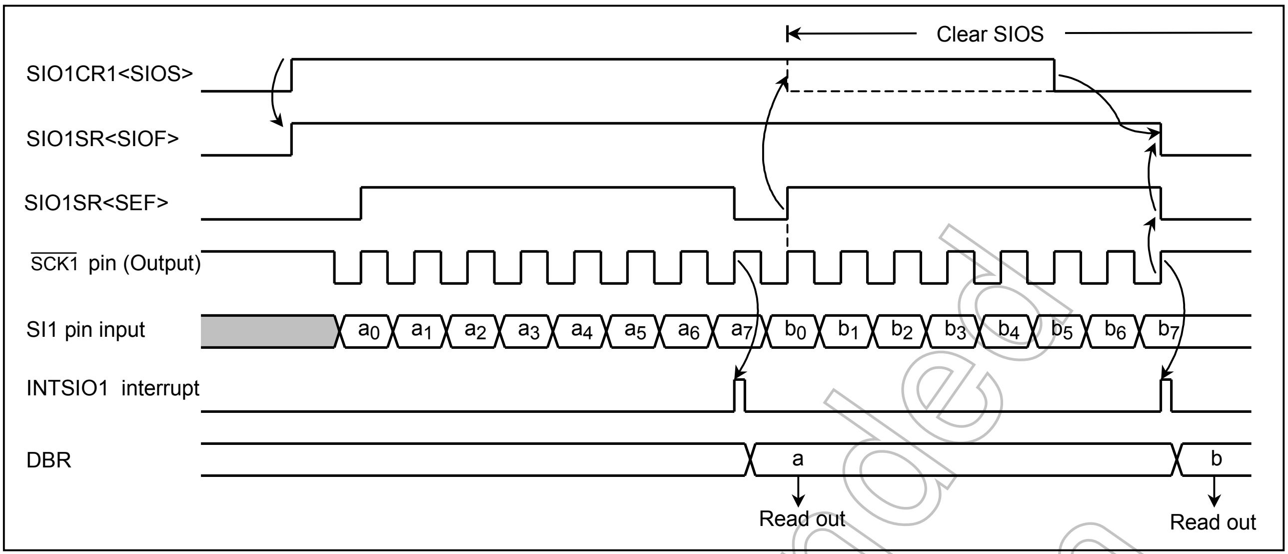


Figure 2.11.9 Receive Mode (Example: 8 bits, 1 word, internal clock)

(3) 8-bit transmit/receive mode

After setting the control registers to the 8-bit transmit/receive mode, write the data to be transmitted first to the data buffer registers (DBR). After that, enable transceiving by setting <SIOS> to “1”. When transmitting, the data are output from the SO0 pin at leading edges of the serial clock. When receiving, the data are input to the SI0 pin at the trailing edges of the serial clock. 8-bit data are transferred from the shift register to the data buffer register. An INTSIO1 interrupt is generated when the number of data words specified with the <BUF> has been transferred. The interrupt service program reads the received data from the data buffer register and then writes the data to be transmitted. The data buffer register is used for both transmitting and receiving; therefore, always write the data to be transmitted after reading the received data.

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

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

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

The transmit/receive operation is ended by clearing SIO1CR1<SIOS> to “0” or setting SIO1SR<SIOINH> to “1” in interrupt service program.

When SIO1CR1<SIOINH> is set, the transmit/receive operation is immediately ended and SIO1SR<SIOF> is cleared to “0”.

If it is necessary to change the number of words in external clock operation, SIO1CR1<SIOS> should be cleared to “0”, then SIO1CR2<BUF> must be rewritten after confirming that SIO1SR<SIOF> has been cleared to “0”.

If it is necessary to change the number of words in internal clock, during automatic-wait operation which occurs after completion of transmit/receive operation, SIO1CR2<BUF> must be rewritten before reading and writing of the receive/transmit data.

Note: The buffer contents are lost when the transfer mode is switched. If it should become necessary to switch the transfer mode, end receiving by clearing SIO1CR1<SIOS> to “0”, read the last data and then switch the transfer mode.

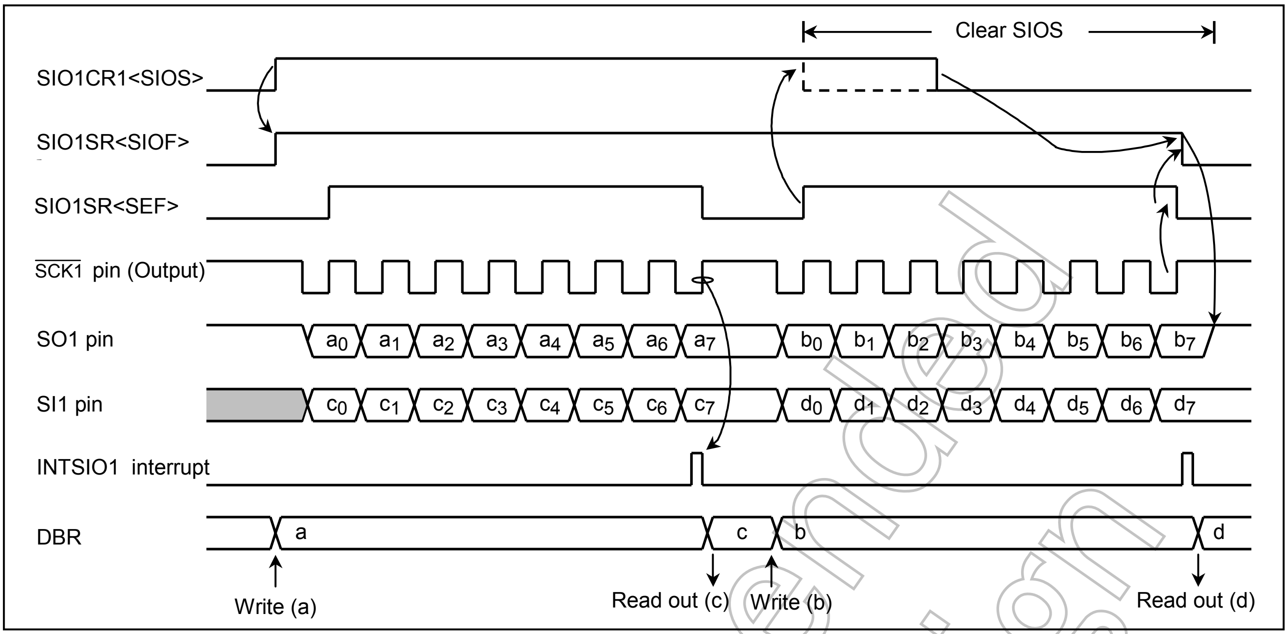


Figure 2.11.10 Transmit/Receive Mode (Example: 8 bits, 1 word, internal clock)

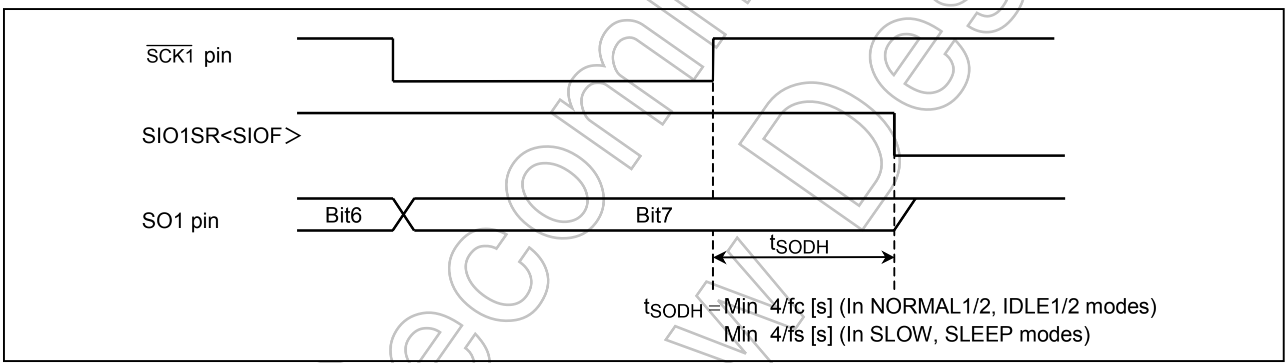


Figure 2.11.11 Transmitted Data Hold Time at End of Transmit/Receive

2.12 8-Bit AD Converter (ADC)

The TMP86CM25A has a 8-bit successive approximation type AD converter.

2.12.1 Configuration

The circuit configuration of the 8-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.

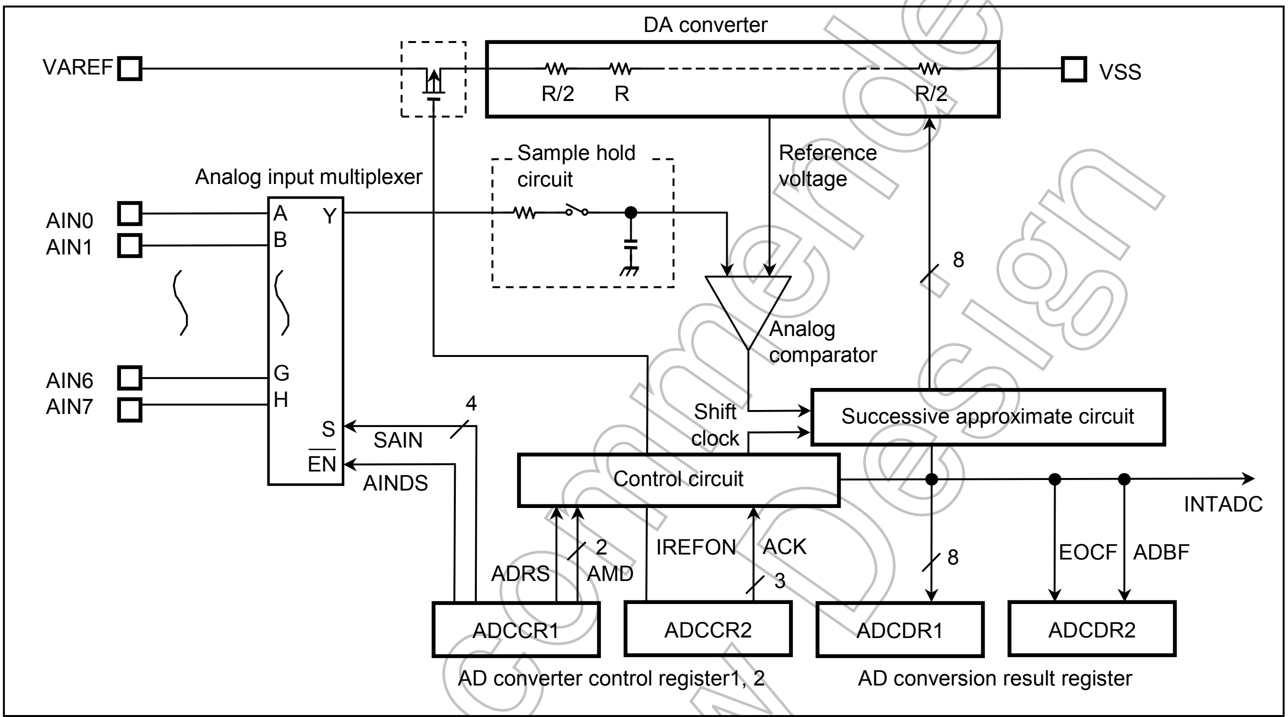


Figure 2.12.1 AD Converter (ADC)

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 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 converted value register (ADCDR1)

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

(4) AD converted value register (ADCDR2)

This register is 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 (000E_H)	7	6	5	4	3	2	1	0	
	ADRS	AMD		AINDS	SAIN				(Initial value: 0001 0000)

ADRS	AD conversion start	0: – 1: Start	R/W
AMD	AD Operating mode	00: AD operation disable 01: Software start mode 10: Reserved 11: Reserved	
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 (000F_H)	7	6	5	4	3	2	1	0	
			IREFON	"1"	ACK			"0"	(Initial value: **00 0000)

IREFON	DA converter (Ladder resistor) connection control	Inputting current to the ladder resistor 0: Connected only during AD conversion 1: Always connected	R/W
ACK	AD conversion time select	(see table below)	

ACK	Conversion time	fc = 16 MHz	fc = 8 MHz	fc = 4 MHz	fc = 1 MHz
000	Reserved				
001	Reserved				
010	76/fc	–	–	–	76.0 μs
011	152/fc	–	–	38.0 μs	152.0 μs
100	304/fc	–	38.0 μs	76.0 μs	–
101	608/fc	38.0 μs	76.0 μs	152.0 μs	–
110	1216/fc	76.0 μs	152.0 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 (38.0 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, bits 6 to 7 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	
(0020_H)	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00	(Initial value: 0000 0000)

ADCDR2	7	6	5	4	3	2	1	0	
(0021_H)			EOCF	ADBF					(Initial value: **00 ****)

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 bits 7, 6 and 3 to 0 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.

(2) Set up 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), 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

(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) 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).

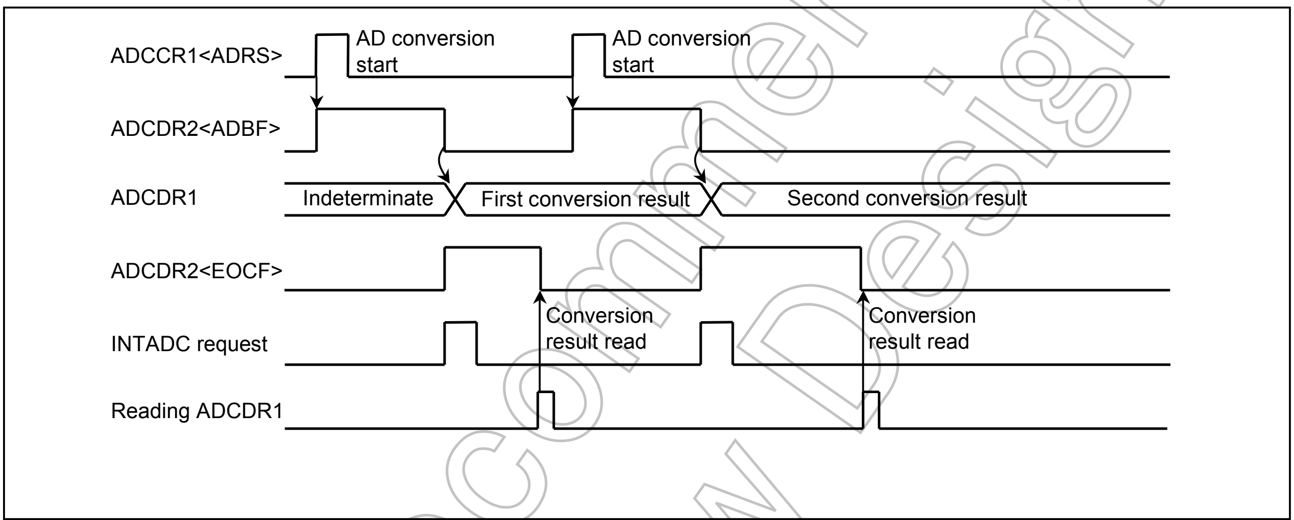


Figure 2.12.4 Operation in Software Start Mode

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.

Example: After selecting the conversion time of 38.0 μs at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store 8 bits data in address 009FH on RAM. The operation mode is software start mode.

```
        ; AIN SELECT
        LD      (P6CR), 00000000B     ;  P6CR bit3 = 0.
        LD      (P6DR), 00000000B     ;  P6DR bit3 = 0.
        LD      (ADCCR1), 00100011B   ;  Select AIN3.
        LD      (ADCCR2), 11011010B   ;  Select conversion time (608/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, (ADCDR1)
        LD      (9FH), A
```

2.12.6 Analog Input Voltage and AD Conversion Result

The analog input voltage is corresponded to the 8-bit digital value converted by the AD as shown in Figure 2.12.5.

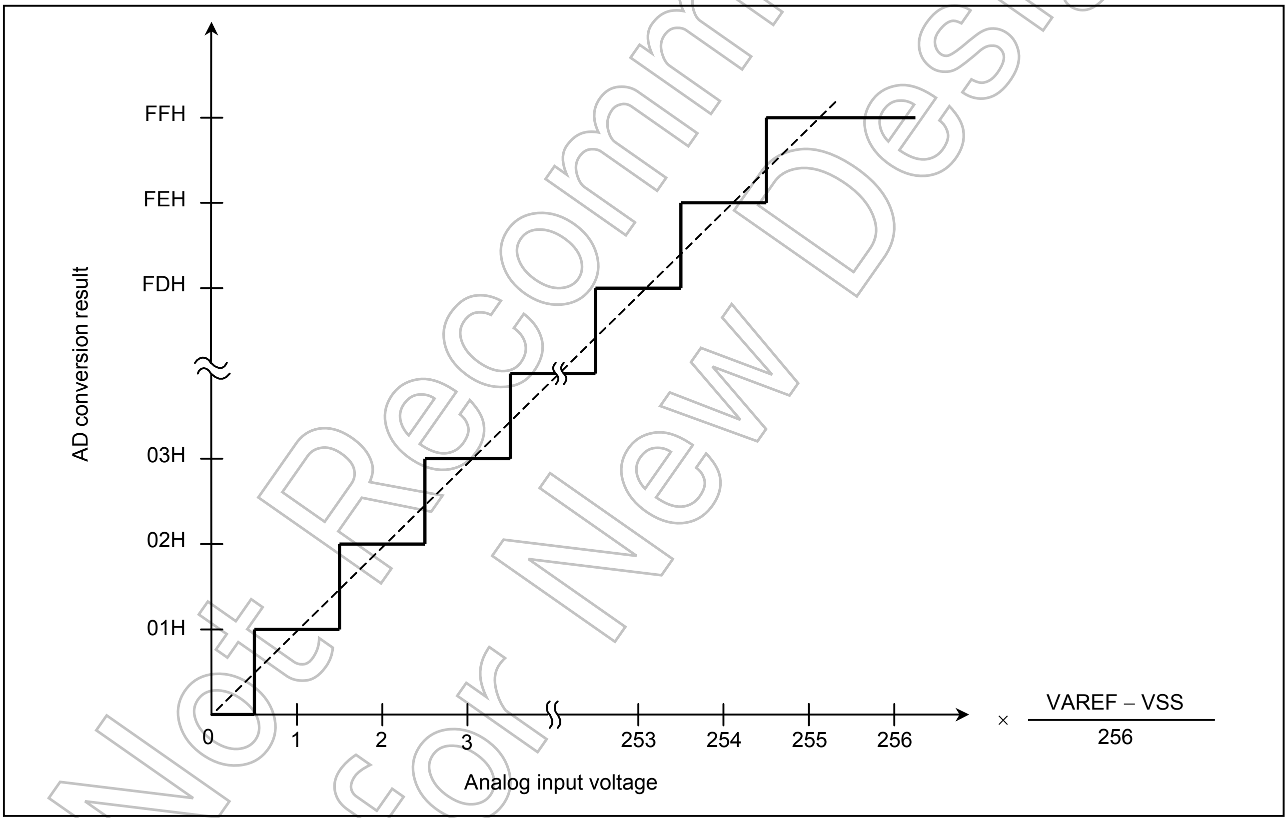


Figure 2.12.5 Analog Input Voltage and AD Conversion Result (Typ.)

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 2.12.6. 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.

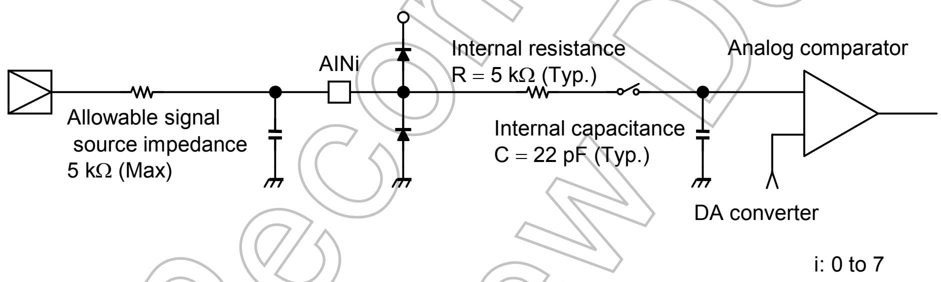


Figure 2.12.6 Analog Input Equivalent Circuit and Example of Input Pin Processing

2.13 Key-on Wakeup (KWU)

In the TMP86CM25A, the STOP mode must be released by not only P20 ($\overline{\text{INT5}}$ / $\overline{\text{STOP}}$) pin but also P64 to P67 pins.

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

2.13.1 Configuration

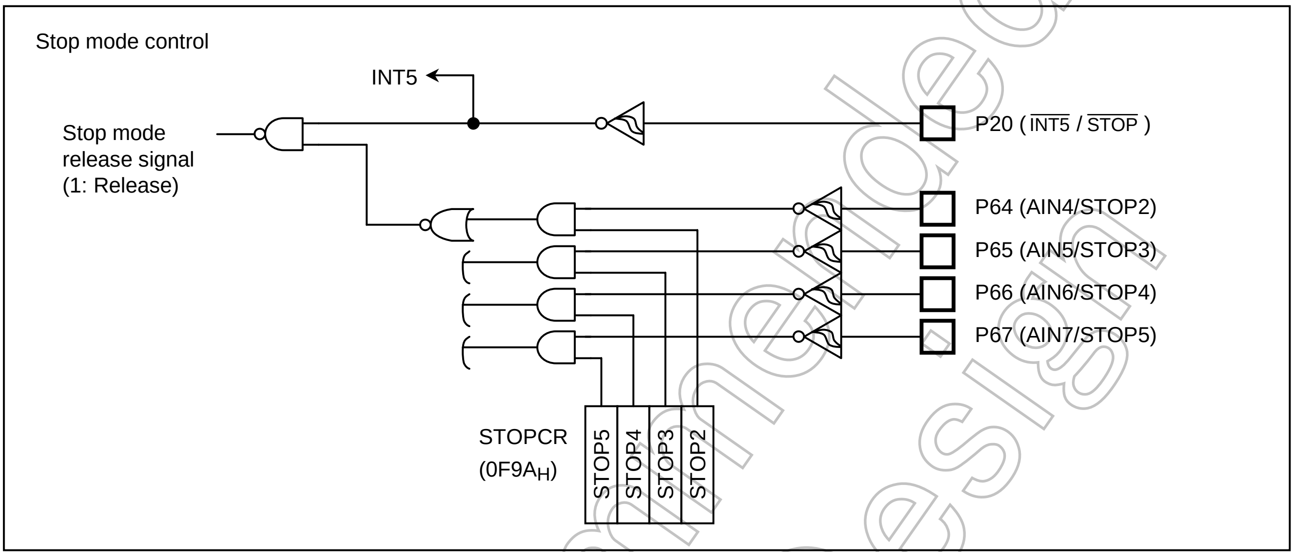
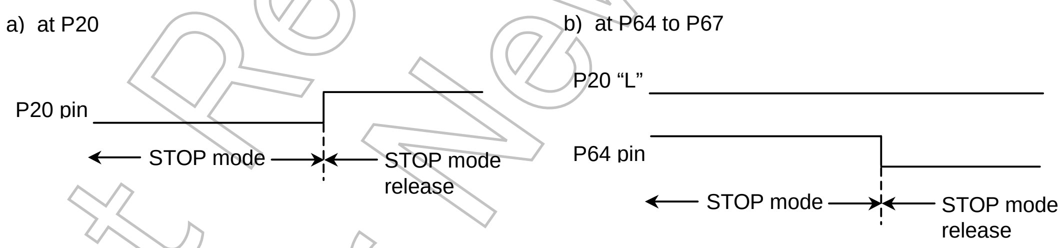


Figure 2.13.1 Stop Mode Control Circuit

Note: $\overline{\text{STOP}}$ pin doesn't have the control register such as STOPCR, so when STOP mode is relesed by STOPx (x: 2 to 5), $\overline{\text{STOP}}$ pin should be used as STOP function. (The P20 must be input to "0" level.)



2.13.2 Control

P64 to P67 (STOP2 to STOP5) pin can controlled by key-on wakeup control register (STOPCR). It can be configured as enable/disable in one-bit unit. When those pins are used for releasing STOP mode, those pins must be set input mode.

STOP mode can be entered by setting up the system control register1 (SYSCR1), and can be exited by detecting low level of STOP2 to STOP5 pins, which are enabled by STOPCR, for releasing STOP mode (Note 1). Also, because each level of the STOP2 to STOP5 can be confirmed by reading P6DR, check all STOP2 to STOP5 pins that is enabled by STOPCR before the STOP mode is started.

Note 1: When the STOP mode is used by edge-sensitive mode (SYSCR1<RELM> = “0”), all bit of STOPCR (STOP2 to STOP5) should be cleared to “0”.

Note 2: When the $\overline{STOP}$ pin input is high or STOP2 to STOP5 pin input which is enabled by STOPCR is low, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (Warm-up).

Table 2.13.1 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 (Note2)	Rising edge
STOP2	P64/AIN4	“L” level (Note 2)	Do not use key-on wakeup function (Note 1)
STOP3	P65/AIN5		
STOP4	P66/AIN6		
STOP5	P67/AIN7		

Key-on Wakeup Control Register

STOPCR (0F9A$_H$)

	7	6	5	4	3	2	1	0
STOPCR (0F9A$_H$)	STOP5	STOP4	STOP3	STOP2	–	–	–	–

(Initial value: 0000 ****)

STOP5	Stop mode released by P67 port	0: Disable 1: Enable	Write only
STOP4	Stop mode released by P66 port	0: Disable 1: Enable	
STOP3	Stop mode released by P65 port	0: Disable 1: Enable	
STOP2	Stop mode released by P64 port	0: Disable 1: Enable	

Figure 2.13.2 Key-on Wakeup Control Register

2.14 LCD Driver

The TMP86CM25A incorporates a driver to directly drive the liquid crystal display (LCD) and its control circuit. The connecting pins with the LCD are as shown below:

(1)	Segment output pin	40 pins (SEG39 to SEG0)
(2)	Segment output/I/O port pin (shared)	20 pins (SEG59 to SEG40)
(3)	Common output pin	5 pins (COM4 to COM0)
(4)	Common output I/O port pin (shared)	11 pins (COM15 to COM5)

In addition, C0, C1, V1, V2, V3 and V4 are provided as the LCD drive booster circuit pins. The following three types of LCD can be driven directly:

(1) 1/4 duty LCD: Maximum 240 pixels (60 segments × 4 digits)

(2) 1/8 duty LCD: Maximum 480 pixels (60 segments × 8 digits)

(3) 1/16 duty LCD: Maximum 960 pixels (60 segments × 16 digits)

2.14.1 Configuration of LCD Driver

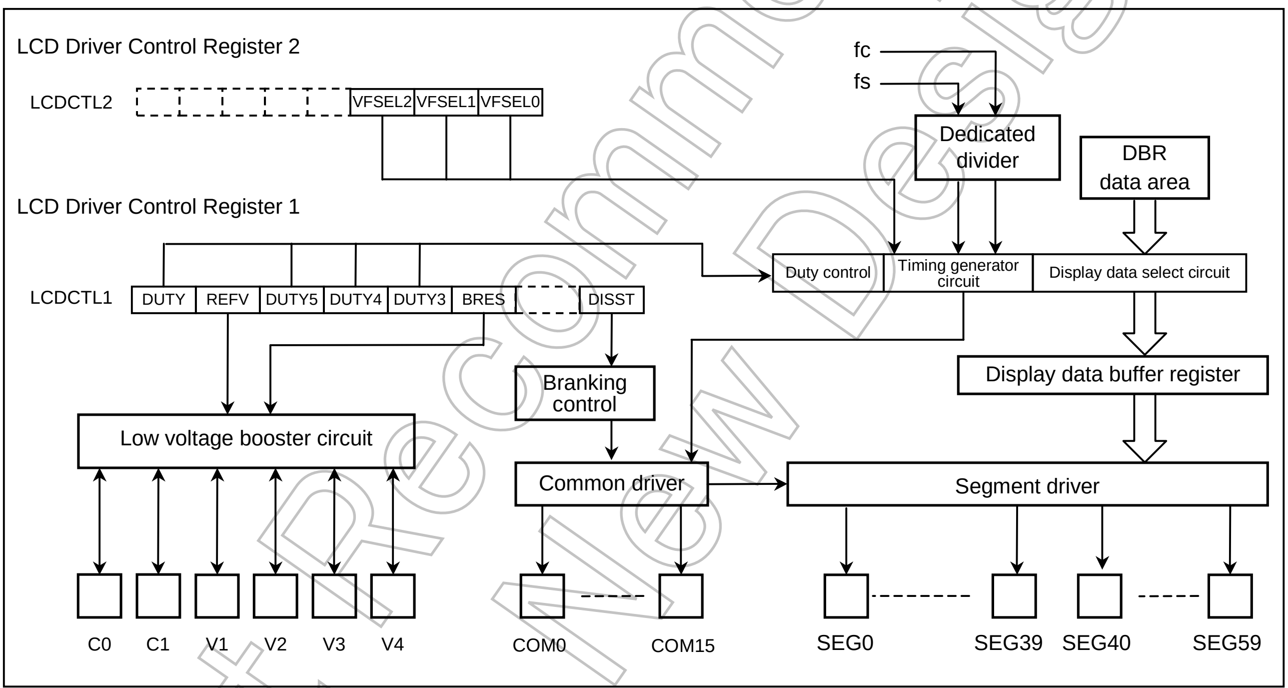


Figure 2.14.1 LCD Driver Block Diagram

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.

2.14.2 Controlling LCD Driver

The LCD driver is controlled by the LCD control register 1 (LCDCTL1) and the LCD control register 2 (LCDCTL2). The display of the LCD driver is enabled by DISST.

LCD Control Register 1

LCDCTL1 (0027_H)	7	6	5	4	3	2	1	0	
	DUTY7	REFV	DUTY5	DUTY4	DUTY3	BRES		DISST	(Initial value: 0000 00*0)

DUTY7 DUTY5 DUTY4 DUTY3	Select duty.	0***: Reserved 1000: 1/4 duty 1001: Reserved 1010: 1/8 duty 1011: Reserved 1100: Reserved 1101: Reserved 1110: 1/16 duty 1111: Reserved	R/W
REFV	Sets LCD reference voltage.	0: V4 ≤ VDD 1: VDD < V4 ≤ 3.6 V	R/W
BRES	Sets booster circuit.	0: Booster circuit disable 1: Booster circuit enable (Note 4)	R/W
DISST	Controls LCD display.	0: LCD display blanking 1: LCD display enable	R/W

Note 1: After reset, <DUTYs> are set to “0000” (Initial value: Reserved). Set the duty as appropriate for LCD panel.

Note 2: Switch <REFV> according to VDD. If it is not set appropriately, an overcurrent may flow causing damage to the device. Caution is especially required when VDD is battery-driven.

Note 3: If <DISST> is set to “0” (LCD display blanking), all SEG/COM pins become VSS level.

Note 4: When <REFV> for the LCD reference voltage <REFV> is set to “0”, always make sure the reference power supply is entered from the V4 pin.

Note 5: Reserved: Not to be set.

LCD Control Register 2

LCDCTL2 (0028_H)	7	6	5	4	3	2	1	0	
						VFSEL2	VFSEL1	VFSEL0	(Initial value: **** *011)

			NORMAL1, IDLE1 Mode	NORMAL2, IDLE2, SLOW2, SLEEP2 Mode	SLOW1, SLEEP1 Mode	
VFSEL	Selects base frequency for frame frequency.	000:	$fc/2^9$	$fc/2^9$	–	R/W
		001:	$fc/2^8$	$fc/2^8$	–	
		010:	$fc/2^7$	$fc/2^7$	–	
		011:	$fc/2^6$	$fc/2^6$	–	
		1**:	–	fs	fs	

Note: Set the LCD control register 2 according to operating frequency. For details of the actual frame frequency, see Table 2.14.1.

Figure 2.14.2 LCD Driver Control Register

(1) Frame frequency

The frame frequency is set depending on the driving method and the base frequency as shown in Table 2.14.1.

The base frequency is selected with LCDCTL2<VFSEL> depending on the basic clock frequencies fc and fs to be used.

Table 2.14.1 Frame Frequency Settings

VFSEL	Base Frequency [Hz]	Frame Frequency [Hz]		
		1/4 duty	1/8 duty	1/16 duty
000	$\frac{fc}{2^9}$	$\frac{fc}{2^9 \cdot 84 \cdot 4}$	$\frac{fc}{2^9 \cdot 42 \cdot 8}$	$\frac{fc}{2^9 \cdot 21 \cdot 16}$
	(fc = 16 MHz)	93	93	93
001	$\frac{fc}{2^8}$	$\frac{fc}{2^8 \cdot 84 \cdot 4}$	$\frac{fc}{2^8 \cdot 42 \cdot 8}$	$\frac{fc}{2^8 \cdot 21 \cdot 16}$
	(fc = 8 MHz)	93	93	93
010	$\frac{fc}{2^7}$	$\frac{fc}{2^7 \cdot 84 \cdot 4}$	$\frac{fc}{2^7 \cdot 42 \cdot 8}$	$\frac{fc}{2^7 \cdot 21 \cdot 16}$
	(fc = 4 MHz)	93	93	93
011	$\frac{fc}{2^6}$	$\frac{fc}{2^6 \cdot 84 \cdot 4}$	$\frac{fc}{2^6 \cdot 42 \cdot 8}$	$\frac{fc}{2^6 \cdot 21 \cdot 16}$
	(fc = 2 MHz)	93	93	93
1**	fs	$\frac{fs}{84 \cdot 4}$	$\frac{fs}{42 \cdot 8}$	$\frac{fs}{21 \cdot 16}$
	(fs = 32.768 kHz)	97.5	97.5	97.5

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

Note 2: Although this product is guaranteed to operate at fc = 1.32 [MHz] or less is not recommended for LCD display as the frame frequency becomes 61 [Hz] or less.

2.14.3 LCD Booster Circuit

(1) LCD booster circuit

The TMP86CM25A can boost (Divide) the externally-supplied reference voltage using the built-in booster circuit as a power supply for driving the LCD. When V2 pin is the reference voltage, the inputted reference voltage is divided/boosted by 1/2 time (V1), 3/2 times (V3) and two times (V4). Likewise, when V3 pin or V4 pin is the reference, the inputted reference voltage is boosted/divided and the voltage ratio is $V1 \times 4 = V2 \times 2 = V3 \times (4/3) = V4$. As this circuit uses a 4-times boosting method, the bias ratio is 1/4 only.

2.14.4 Methods of Connecting LCD Booster Circuit

(1) Method of connecting booster circuit by using a regulator

If VDD is not stable because it is battery-driven, etc., we recommend a connection method using a regulator as shown below in order to preserve the quality of display.

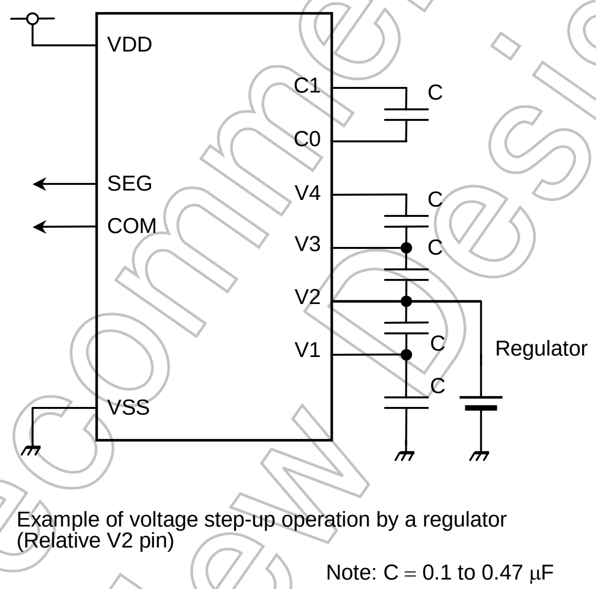
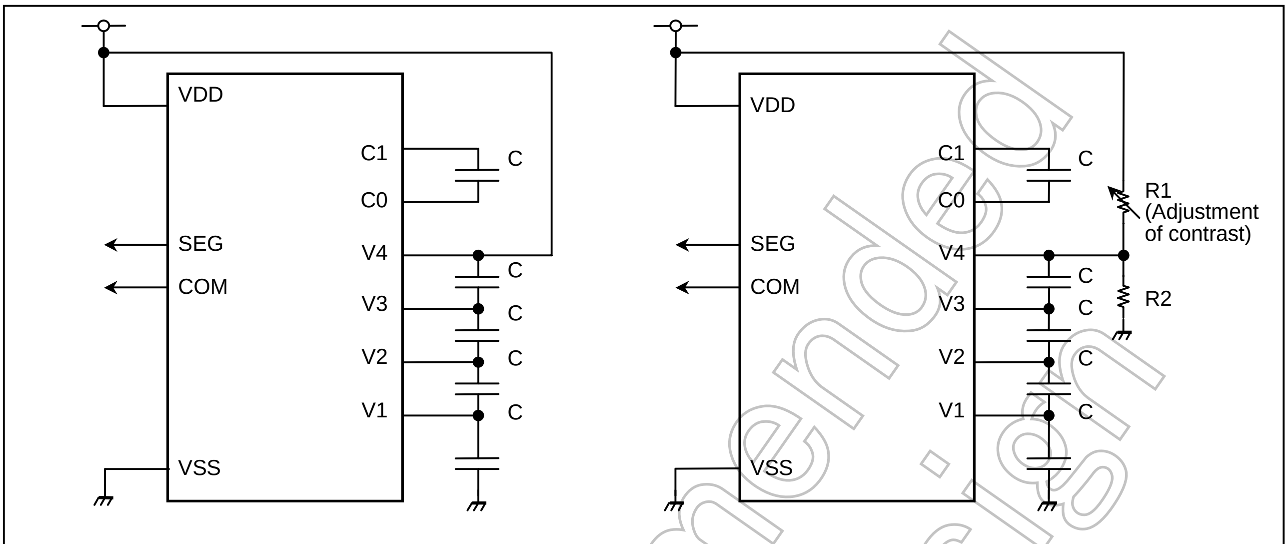


Figure 2.14.3 Method of Connecting Booster Circuit by Using a Regulator

Note: For use with VDD ≥ V4 (LCDCTL1<REFV> = 0), always make sure the reference power supply is entered from V4.

(2) Method of connecting booster circuit without using a regulator

If stable VDD supply is achieved (VDD ≥ V4), the booster circuit can be connected without using a regulator as shown below. In this case, set LCDCTL1<REFV> to "0" and make sure the reference power supply is entered from the V4 pin.



(c) Example of voltage division from VDD (Relative to V4 pin) (d) Example of voltage division from VDD (Relative to V4 pin)

Note: C = 0.1 to 0.47 μF

Figure 2.14.4 Method of Connecting Booster Circuit without using a Regulator

2.14.5 LCD Display Operation

(1) Setting display data

Display data is stored in display data area (128 bytes in addresses 0F00H to 0F7FH) provided in DBR.

Display data stored in the display data area is automatically read by hardware and sent to the LCD driver. The LCD driver generates segment and common signals according to display data and the driving method. Thus, display patterns can be changed simply by rewriting the contents of display data area in the program.

Figure 2.14.5 shows the correspondence between display data areas and SEG/COM pins. The light comes on when display data is "1" and it goes out when "0". Because the number of pixels that can be driven varies with the method of driving the LCD, the number of bytes in the display data area used to store display data also varies. Thus, bytes not used to store display data and data memory corresponding to ddresses not connected to the LCD can be used for storing generally processed data. (See Table 2.14.2)

Note: Because the contents of display data area become unstable at powering on, execute the initialize routine for the initial setting.

0F00H	0F10H	0F20H	0F30H	0F40H	0F50H	0F60H	0F70H	COM0
0F01H	0F11H	0F21H	0F31H	0F41H	0F51H	0F61H	0F71H	COM1
0F02H	0F12H	0F22H	0F32H	0F42H	0F52H	0F62H	0F72H	COM2
0F03H	0F13H	0F23H	0F33H	0F43H	0F53H	0F63H	0F73H	COM3
0F04H	0F14H	0F24H	0F34H	0F44H	0F54H	0F64H	0F74H	COM4
0F05H	0F15H	0F25H	0F35H	0F45H	0F55H	0F65H	0F75H	COM5
0F06H	0F16H	0F26H	0F36H	0F46H	0F56H	0F66H	0F76H	COM6
0F07H	0F17H	0F27H	0F37H	0F47H	0F57H	0F67H	0F77H	COM7
0F08H	0F18H	0F28H	0F38H	0F48H	0F58H	0F68H	0F78H	COM8
0F09H	0F19H	0F29H	0F39H	0F49H	0F59H	0F69H	0F79H	COM9
0F0AH	0F1AH	0F2AH	0F3AH	0F4AH	0F5AH	0F6AH	0F7AH	COM10
0F0BH	0F1BH	0F2BH	0F3BH	0F4BH	0F5BH	0F6BH	0F7BH	COM11
0F0CH	0F1CH	0F2CH	0F3CH	0F4CH	0F5CH	0F6CH	0F7CH	COM12
0F0DH	0F1DH	0F2DH	0F3DH	0F4DH	0F5DH	0F6DH	0F7DH	COM13
0F0EH	0F1EH	0F2EH	0F3EH	0F4EH	0F5EH	0F6EH	0F7EH	COM14
0F0FH	0F1FH	0F2FH	0F3FH	0F4FH	0F5FH	0F6FH	0F7FH	COM15
SEG7 to SEG0	SEG15 to SEG8	SEG23 to SEG16	SEG31 to SEG24	SEG39 to SEG32	SEG47 to SEG40	SEG55 to SEG48	SEG59 to SEG56	

Figure 2.14.5 LCD Display Data Area (DBR)

Table 2.14.2 Areas Used to Store Display Data

Driving Method	COM Number to be Used
1/16 duty	COM15 to COM0
1/8 duty	COM7 to COM0
1/4 duty	COM3 to COM0

(2) Blanking

The LCD display can be blanked by clearing DISST to “0”. Blanking extinguishes the LCD by outputting GND level to COM/SEG pins.

If the STOP mode is entered while the LCD display is on, DISST is cleared to “0” and blanking is performed automatically. If the STOP mode is then reverted, DISST is set to “1” and display is resumed automatically.

Note: At reset, the segment dedicated pins (SEG39 to SEG0) and common output becomes GND level, whereas the I/O port/segment shared pins (P1, P3, P5 ports) output, the I/O port/common shared pins (P3, P7 ports) output become the high-impedance state. Thus, if an external reset input lasts for a significant length of time, it may affect the LCD display such as blurring.

2.14.6 Method of Controlling LCD Driver

(1) Initial setting

The procedure of initial setting is shown below.

Example: When 60 seg × 8 com, 1/8 duty, 3 V-system LCD operates with fc = 8 MHz (at VDD = 3 V).

```
LD   (LCDCTL1), 10010100B   ;  1/8 duty, LCD reference voltage (VDD = V4),
                               booster circuit enable set.
LD   (P1LCR), 0FFH          ;  Set P1 port for segment output.
LD   (P3LCR), 0FFH          ;  Set P3 port for segment/common output.
LD   (P5LCR), 0FFH          ;  Set P5 port for segment output.
LD   (P7LCR), 0FFH          ;  Set P7 port for common output.
 ¦                          ;
 ¦
LD   (LCDCTL1), 10010101B   ;  LCD display enable set.
```

(2) Storing display data

Display data is normally prepared as fixed data in the program memory (ROM) and stored in the display data area by a load instruction.

Example 1: Corresponding to the connection and display using a 1/8 duty LCD shown in Figure 2.14.6, the Table 2.14.3 shows display data and Figure 2.14.7 shows displyay timing.

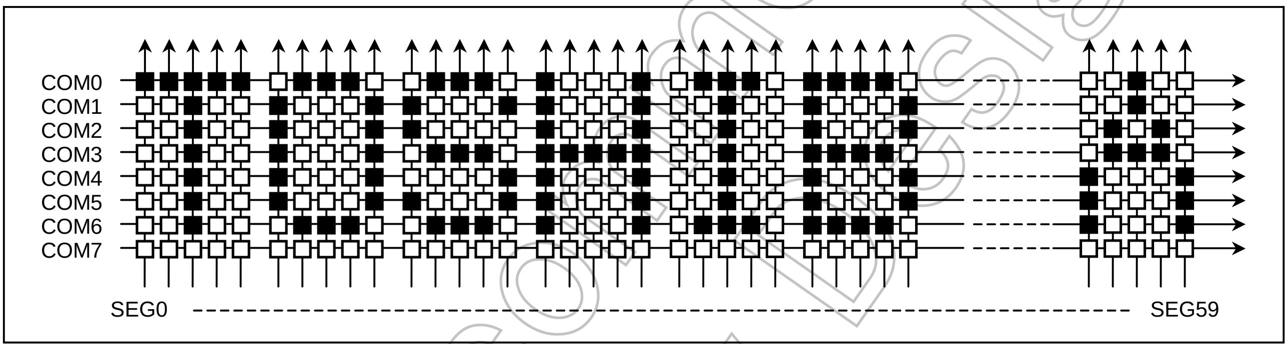


Figure 2.14.6 Example of Display Data (1/8 duty)

Table 2.14.3 Example of Display Data (1/8 duty)

	DBR	SEG0 Bit0	SEG1 Bit1	SEG2 Bit2	SEG3 Bit3	SEG4 Bit4	SEG5 Bit5	SEG6 Bit6	SEG7 Bit7	HEX	DBR	SEG8 Bit0	SEG9 Bit1	SEG10 Bit2	SEG11 Bit3	SEG12 Bit4	SEG13 Bit5	SEG14 Bit6	SEG15 Bit7	HEX	
COM0	0F00H	1	1	1	1	1	0	1	1	DF	0F10H	1	0	0	1	1	1	0	1	B9	…
COM1	0F01H	0	0	1	0	0	1	0	0	24	0F11H	0	1	1	0	0	0	1	1	C6	…
COM2	0F02H	0	0	1	0	0	1	0	0	24	0F12H	0	1	1	0	0	0	0	1	86	…
COM3	0F03H	0	0	1	0	0	1	0	0	24	0F13H	0	1	0	1	1	1	0	1	BA	…
COM4	0F04H	0	0	1	0	0	1	0	0	24	0F14H	0	1	0	0	0	0	1	1	C2	…
COM5	0F05H	0	0	1	0	0	1	0	0	24	0F15H	0	1	1	0	0	0	1	1	C6	…
COM6	0F06H	0	0	1	0	0	0	1	1	C4	0F16H	1	0	0	1	1	1	0	1	B9	…
COM7	0F07H	0	0	0	0	0	0	0	0	00	0F17H	0	0	0	0	0	0	0	0	00	…

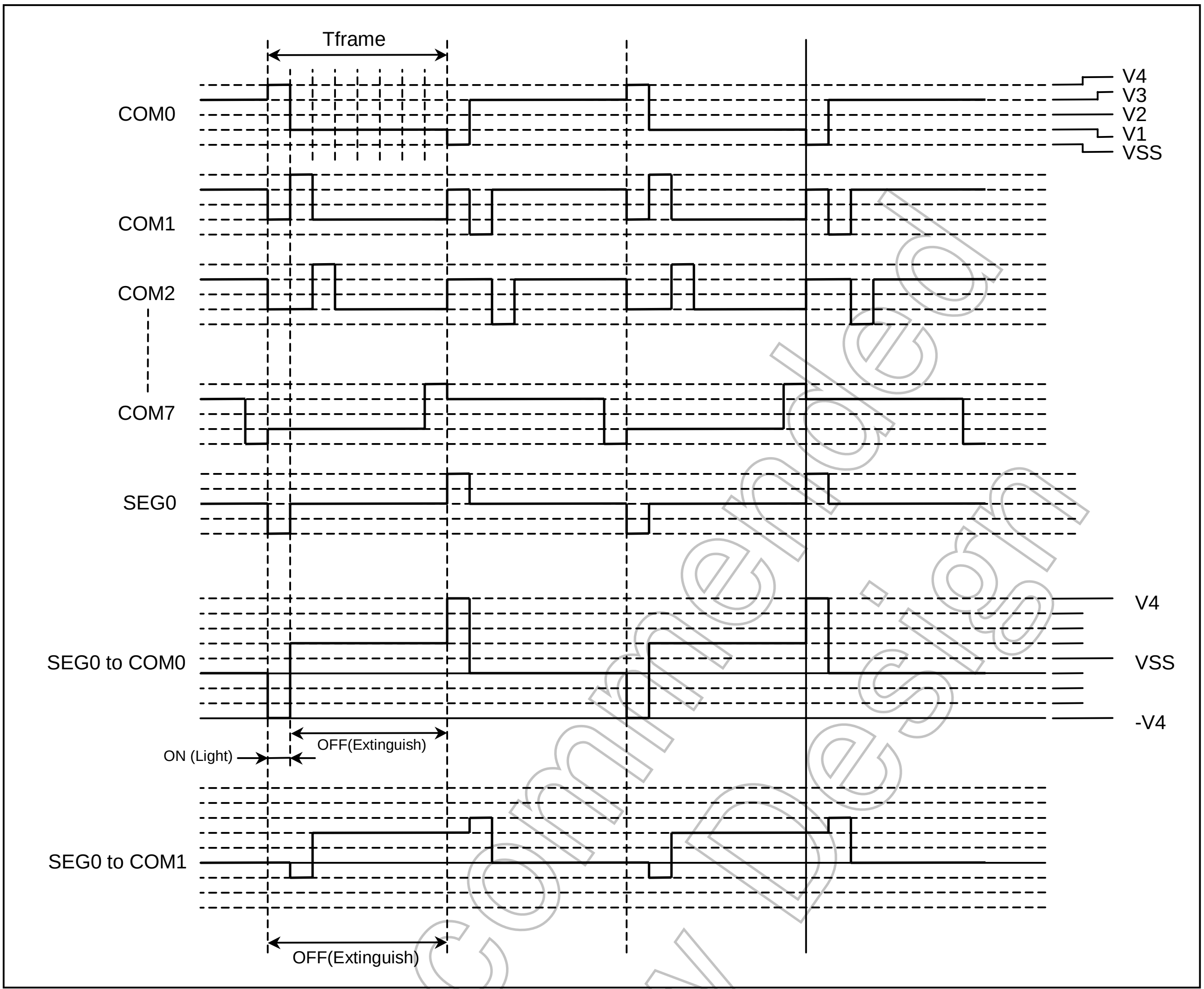


Figure 2.14.7 Example of Display Timing (1/8 duty)

Example 2: Corresponding to the connection and display using a 1/16 duty LCD shown in Figure 2.14.8, Table 2.14.4 shows display data and Figure 2.14.9 shows display timing.

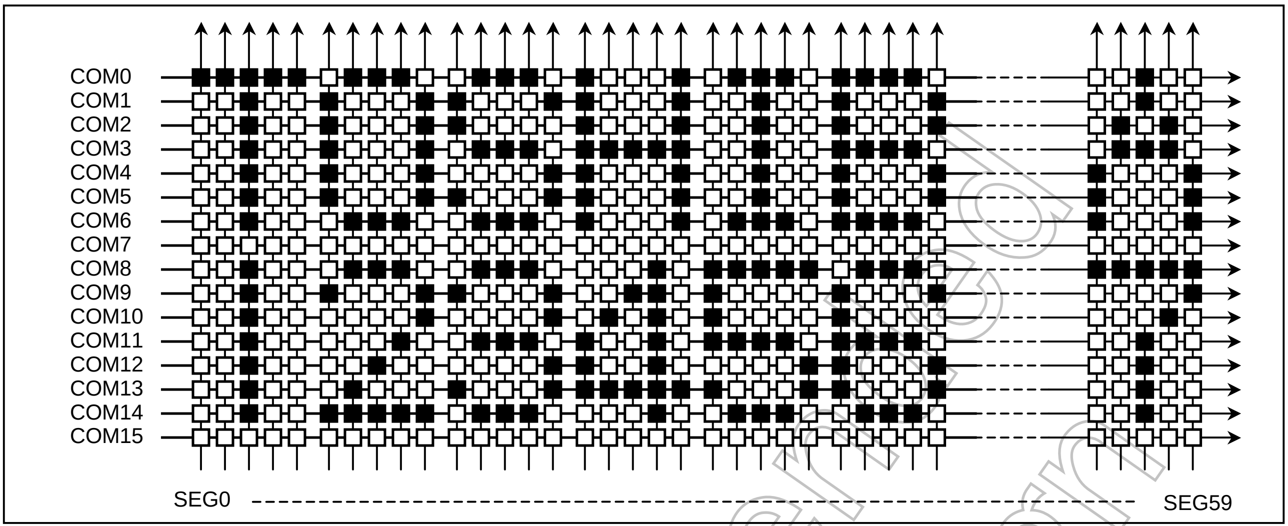


Figure 2.14.8 Example of Display Data (1/16 duty)

Table 2.14.4 Example of Display Data (1/16 duty)

	DBR	SEG0 Bit0	SEG1 Bit1	SEG2 Bit2	SEG3 Bit3	SEG4 Bit4	SEG5 Bit5	SEG6 Bit6	SEG7 Bit7	HEX	DBR	SEG8 Bit0	SEG9 Bit1	SEG10 Bit2	SEG11 Bit3	SEG12 Bit4	SEG13 Bit5	SEG14 Bit6	SEG15 Bit7	HEX	
COM0	0F00H	1	1	1	1	1	0	1	1	DF	0F10H	1	0	0	1	1	1	0	1	B9	···
COM1	0F01H	0	0	1	0	0	1	0	0	24	0F11H	0	1	1	0	0	0	1	1	C6	···
COM2	0F02H	0	0	1	0	0	1	0	0	24	0F12H	0	1	1	0	0	0	0	1	86	···
COM3	0F03H	0	0	1	0	0	1	0	0	24	0F13H	0	1	0	1	1	1	0	1	BA	···
COM4	0F04H	0	0	1	0	0	1	0	0	24	0F14H	0	1	0	0	0	0	1	1	C2	···
COM5	0F05H	0	0	1	0	0	1	0	0	24	0F15H	0	1	1	0	0	0	1	1	C6	···
COM6	0F06H	0	0	1	0	0	0	1	1	C4	0F16H	1	0	0	1	1	1	0	1	B9	···
COM7	0F07H	0	0	0	0	0	0	0	0	00	0F17H	0	0	0	0	0	0	0	0	00	···
COM8	0F08H	0	0	1	0	0	0	1	1	C4	0F18H	1	0	0	1	1	1	0	0	39	···
COM9	0F09H	0	0	1	0	0	1	0	0	24	0F19H	0	1	1	0	0	0	1	0	46	···
COM10	0F0AH	0	0	1	0	0	0	0	0	04	0F1AH	0	1	0	0	0	0	1	0	42	···
COM11	0F0BH	0	0	1	0	0	0	0	0	04	0F1BH	1	0	0	1	1	1	0	1	B9	···
COM12	0F0CH	0	0	1	0	0	0	0	1	84	0F1CH	0	0	0	0	0	0	1	1	C0	···
COM13	0F0DH	0	0	1	0	0	0	1	0	44	0F1DH	0	0	1	0	0	0	1	1	C4	···
COM14	0F0EH	0	0	1	0	0	1	1	1	E4	0F1EH	1	1	0	1	1	1	0	0	3B	···
COM15	0F0FH	0	0	0	0	0	0	0	0	00	0F1FH	0	0	0	0	0	0	0	0	00	···

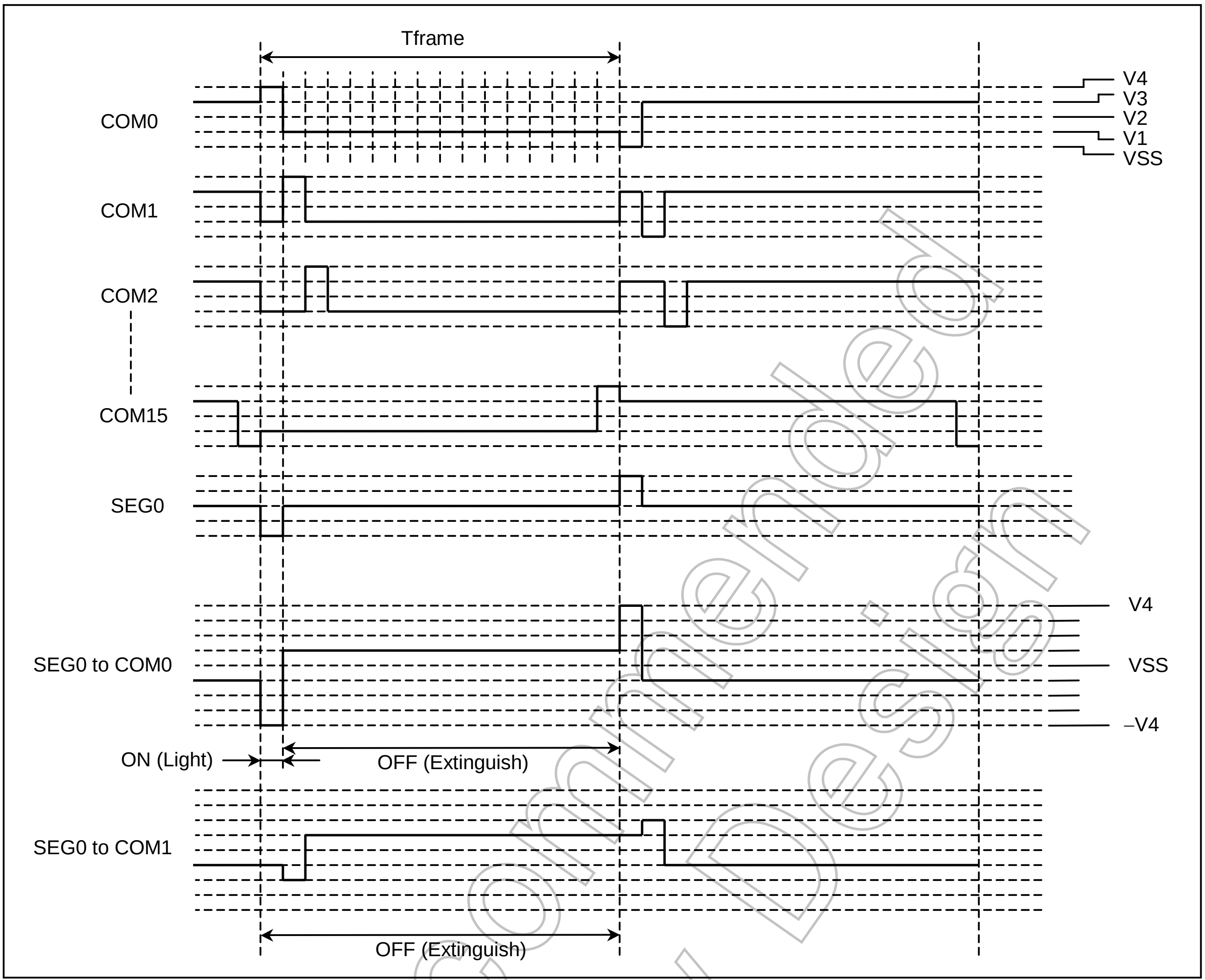
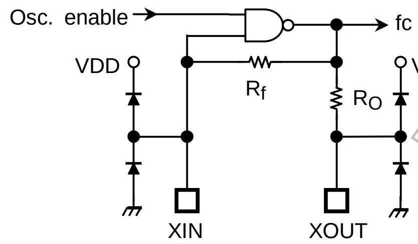
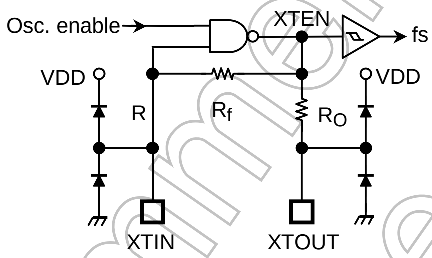
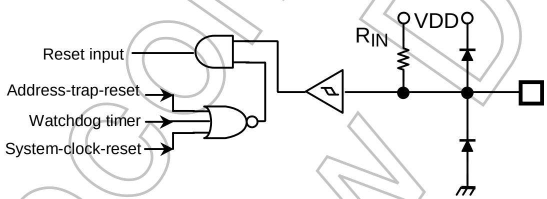
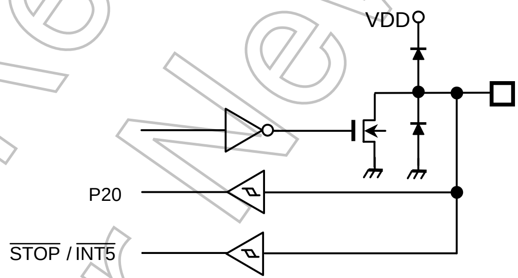
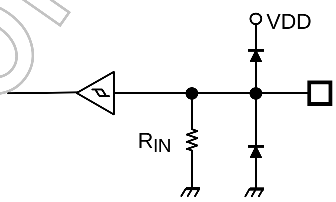


Figure 2.14.9 Example of Display Timing (1/16 duty)

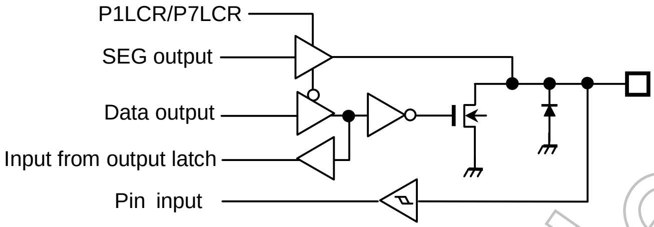
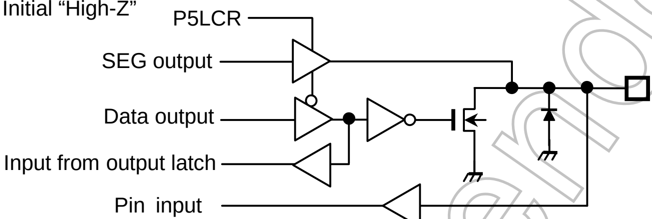
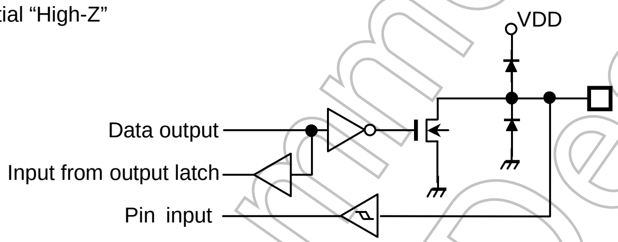
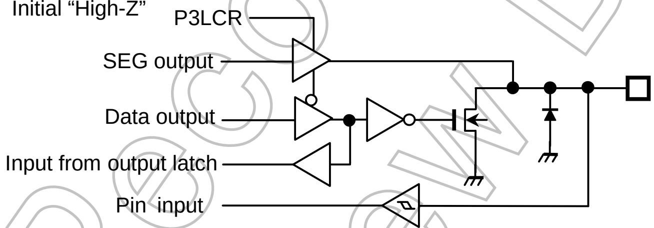
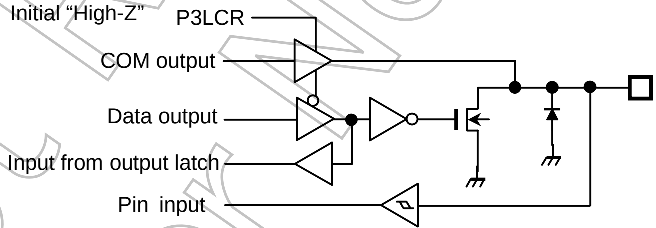
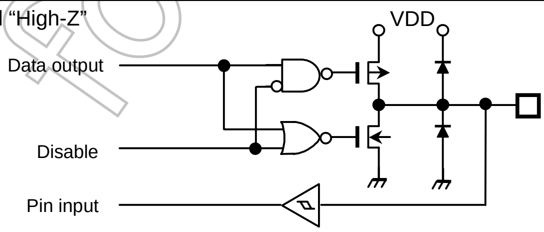
Input/Output Circuitry

(1) Control pins

The input/output circuitries of the TMP86CM25A 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: Refer to port P2 NORMAL2 mode: 	Resonator connecting pins (Low frequency) R_f = 20 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Sink open drain output Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.)
$\overline{\text{STOP}}$ / $\overline{\text{INT5}}$	Input		Hysteresis input
TEST	Input		Hysteresis input Pull-down resistor R_{IN} = 70 kΩ (typ.)

(2) Input/output ports

Port	I/O	Input/Output Circuitry	Remarks
P1 P7	I/O	Initial "High-Z" 	Sink-open-drain output Hysteresis input
P5	I/O	Initial "High-Z" 	Sink-open-drain output
P2	I/O	Initial "High-Z" 	Sink-open-drain output Hysteresis input
P30 P31 P32 P33	I/O	Initial "High-Z" 	Sink-open-drain output Hysteresis input High current output (Nch)
P34 P35 P36	I/O	Initial "High-Z" 	Sink-open-drain output Hysteresis input
P6	I/O	Initial "High-Z" 	Tri-state I/O Hysteresis input

Note: Port P1, P3, P5 and P7 are sink-open-drain output. But they are also used as a segment output of LCD. Therefore,absolute maximum ratings of port input voltage should be used in –0.3 to V_{DD} + 0.3 V.

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 V4 pin	−0.3 to V_{DD} + 0.3	
	V_{OUT2}	V4 pin	-0.3 to 4.0	
Output current (Per 1 pin)	I_{OUT1}	P6 port	−1.8	mA
	I_{OUT2}	P1, P2, P34 to P36, P5, P6, P7 ports	3.2	
	I_{OUT3}	P30 to P33 port	30	
Output current (Total)	ΣI_{OUT1}	P6 port	−30	
	ΣI_{OUT2}	P1, P2, P34 to P36, P5, P6, P7 ports	60	
	ΣI_{OUT3}	P30 to P33 port	80	
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 (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 = 4.2 MHz (in case of external clock)	NORMAL1, 2 mode	1.8		
				IDLE0, 1, 2 mode			
			fc = 8 MHz (in case of connecting a resonator)	NORMAL1, 2 mode	1.8		
				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		$V_{DD} \times 0.90$		
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			$V_{DD} \times 0.10$	
Clock frequency (in case of external clock)	fc	XIN, XOUT	V_{DD} = 1.8 to 3.6 V		1.0	4.2	MHz
			V_{DD} = 2.7 to 3.6 V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 1.8 to 3.6 V		30.0	34.0	kHz
Clock frequency (in case of connecting a resonator)	fc	XIN, XOUT	V_{DD} = 1.8 to 3.6 V		1.0	8.0	MHz
			V_{DD} = 2.7 to 3.6 V			16.0	
	fs	XTIN, XTOUT	V_{DD} = 1.8 to 3.6 V		30.0	34.0	kHz
LCD reference voltage	$V2_{IN}$	V2	LCDCTL1<REFV> = "1" VDD < V4 (Note 2)		1.650	1.800	V
	$V3_{IN}$	V3			2.250	2.700	
	$V4_{IN}$	V4			3.000	3.600	
	$V4_{IN}$	V4 (Note 3)	LCDCTL1<REFV> = "0"		3.000	VDD	
Capacity for LCD booster circuit	C_{LCD}				0.1	0.47	μF

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

Note 2: When LCDCTL1<REFV> is set to "1", always keep the condition of VDD < V4.

Note 3: When LCDCTL1<REFV> is cleared to "0", always supply the reference voltage from V4 pin.

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_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$	V_{DD} = 3.6 V, V_{IN} = 3.6 V	–	–	+5	
Input resistance	R_{IN1}	TEST pull down	V_{DD} = 3.6 V, V_{IN} = 3.6 V	–	70	–	kΩ
	R_{IN2}	$\overline{\text{RESET}}$ pull up	V_{DD} = 3.6 V, V_{IN} = 0 V	100	220	450	
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} = 3.4 V / 0.2 V	–	–	±10	μA
Output high voltage	V_{OH}	C-MOS, Tri-state	V_{DD} = 3.6 V, I_{OH} = –0.6 mA	3.2	–	–	V
Output low voltage	V_{OL}	Except XOUT, P30 to P33 port	V_{DD} = 3.6 V, I_{OL} = 0.9 mA	–	–	0.4	
Output low current	I_{OL}	P30 to P33 ports	V_{DD} = 3.6 V, V_{OL} = 1.0 V	–	6	–	mA
Supply current in NORMAL1, 2 mode	I_{DD}		V_{DD} = 3.6 V V_{IN} = 3.4 V/0.2 V fc = 16 MHz fs = 32.768 kHz	–	3.8	4.6	
Supply current in IDLE0, 1, 2 mode				–	2.4	2.8	
Supply current in SLOW1 mode			V_{DD} = 3.6 V V_{IN} = 3.4 V/0.2 V fs = 32.768 kHz	–	9	20	μA
Supply current in SLEEP1 mode				–	6	16	
Supply current in SLEEP0 mode				–	5	15	
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 SLOW2 and SLEEP2 modes are equivalent to IDLE0, IDLE1, IDLE2.

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}		V_{DD} – 1.0	–	V_{DD}	
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.5	–	–	V
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = V_{AREF} = 3.6 V V_{SS} = 0.0 V	–	0.4	–	mA
Non linearity error		V_{DD} = 2.7 V V_{SS} = 0.0 V V_{AREF} = 2.7 V	–	–	±1	LSB
Zero point error			–	–	±1	
Full scale error			–	–	±1	
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}		V_{DD} – 0.6	–	V_{DD}	
Analog reference voltage range (Note 4)	ΔV_{AREF}		2.0	–	–	V
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = V_{AREF} = 2.0V V_{SS} = 0.0 V	–	0.22	–	mA
Non linearity error		V_{DD} = 2.0 V V_{SS} = 0.0 V V_{AREF} = 2.0 V	–	–	±1	LSB
Zero point error			–	–	±1	
Full scale error			–	–	±1	
Total error			–	–	±2	

(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}		V_{DD} – 0.1	–	V_{DD}	
Analog reference voltage range (Note 4)	ΔV_{AREF}		1.8	–	–	V
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = V_{AREF} = 1.8V V_{SS} = 0.0 V	–	0.2	–	mA
Non linearity error		V_{DD} = 1.8 V V_{SS} = 0.0 V V_{AREF} = 1.8 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
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.

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.

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					

Timer Counter 1 input (ECIN) Characteristics (V_{SS} = 0 V, Topr = –40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
TC1 input (ECIN input)	t_{TC1}	Frequency measurement mode V_{DD} = 2.7 to 3.6 V	–	–	0.5	MHz
		Frequency measurement mode V_{DD} = 1.8 to 2.7 V	–	–	0.25	

Package Dimensions

P-QFP100-1420-0.65A

Unit: mm