TOSHIBA

8 Bit Microcontroller
870C Series

TMP86FS49UG

TOSHIBA CORPORATION

The information contained herein is subject to change without notice.

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 patent or patent rights of TOSHIBA or others.

TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress.
It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications.
Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..

The Toshiba products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.).
These Toshiba products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of Toshiba products listed in this document shall be made at the customer's own risk.

The products described in this document may include products subject to the foreign exchange and foreign trade laws.

TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law 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.

Revision History

Date	Revision	
2005/5/10	1A	First Release

Table of Contents

TMP86FS49UG

2. Operational Description

3. Interrupt Control Circuit

9. 16-Bit Timer/Counter 2

10. 8-Bit Timer/Counter (TC3, TC4)

11. 8-Bit Timer/Counter (TC5, TC6)

12. UART 1 (Asynchronous serial interface 1)

13. UART 2 (Asynchronous serial interface 2)

14. HSIO1 (High-Speed Synchronous Serial Interface)

15. HSIO2 (High-Speed Synchronous Serial Interface)

16. Serial Bus Interface (SBI)

17. 10-Bit AD Converter (ADC)

18. Key-on Wakeup (KWU)

19. Flash Memory

20. Serial PROM Mode

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

CMOS 8-Bit Microcontroller

TMP86FS49UG

Product No.	FLASH	RAM	PACKAGE	Mask MCU
TMP86FS49UG	60K bytes	2K bytes	P-LQFP64-1010-0.50D	TMP86CH49/CM49

1.1 Features

1. 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 & 731 basic instructions
2. 24interrupt sources (External : 5 Internal : 19)
3. Input / Output ports (56pins)
4. Time Base Timer
5. Watchdog Timer
6. 16-bit timer counter: 1 ch
   - Timer, External trigger, Window, Pulse width measurement, Event counter, PPG(Programmable Pulse Generator) output modes
7. 16-bit timer counter : 1ch
   - Timer, Event counter, Window modes
8. 8-bit timer counter : 4ch

   Timer, Event counter, PWM(Pulse width modulation) output, PDO(Programmable Divider Output) output and PPG(Programmable Pulse Generator) modes
9. 8-bit UART: 2ch
10. High-Speed SIO: 2ch
11. Serial Bus Interface(I^2C Bus): 1ch

This product uses the Super Flash® technology under the licence of Sillicon Storage Technology,Inc. Super Flash® is registered trademark of Sillicon Storage Technology,Inc.

Purchase of TOSHIBA I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

030619EBP1

- The information contained herein is subject to change without notice.
- 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 patent or patent rights of TOSHIBA or others.
- TOSHIBA is continually working to improve the quality and reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to comply with the standards of safety in making a safe design for the entire system, and to avoid situations in which a malfunction or failure of such TOSHIBA products could cause loss of human life, bodily injury or damage to property.
  In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent TOSHIBA products specifications. Also, please keep in mind the precautions and conditions set forth in the "Handling Guide for Semiconductor Devices," or "TOSHIBA Semiconductor Reliability Handbook" etc..
- The TOSHIBA products listed in this document are intended for usage in general electronics applications (computer, personal equipment, office equipment, measuring equipment, industrial robotics, domestic appliances, etc.). These TOSHIBA products are neither intended nor warranted for usage in equipment that requires extraordinarily high quality and/or reliability or a malfunction or failure of which may cause loss of human life or bodily injury ("Unintended Usage"). Unintended Usage include atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, medical instruments, all types of safety devices, etc.. Unintended Usage of TOSHIBA products listed in this document shall be made at the customer' s own risk.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- TOSHIBA products should not be embedded to the downstream products which are prohibited to be produced and sold, under any law 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.

12. 10-bit successive approximation type AD converter

    Analog inputs: 16ch

13. Key On Wake Up : 4ch

14. Clock operation

    Single clock mode

    Dual clock mode

15. Low power consumption operation

    STOP mode: Oscillation stops. (Battery/Capacitor back-up.)

    SLOW1 mode: Low power consumption operation using low-frequency clock.(High-frequency clock stop.)

    SLOW2 mode: Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)

    IDLE0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Release by falling edge of the source clock which is set by TBTCR<TBTCK>.

    IDLE1 mode: CPU stops and peripherals operate using high frequency clock. Release by interruputs(CPU restarts).

    IDLE2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interruputs. (CPU restarts).

    SLEEP0 mode: CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock.Release by falling edge of the source clock which is set by TBTCR<TBTCK>.

    SLEEP1 mode: CPU stops, and peripherals operate using low frequency clock. Release by interruput.(CPU restarts).

    SLEEP2 mode: CPU stops and peripherals operate using high and low frequency clock. Release by interruput.

16. Wide operation voltage:

    4.5 V~5.5 V at 16.0MHz /32.768 kHz

    3.0 V~3.6 V at 8 MHz /32.768 kHz

1.2 Pin Assignment

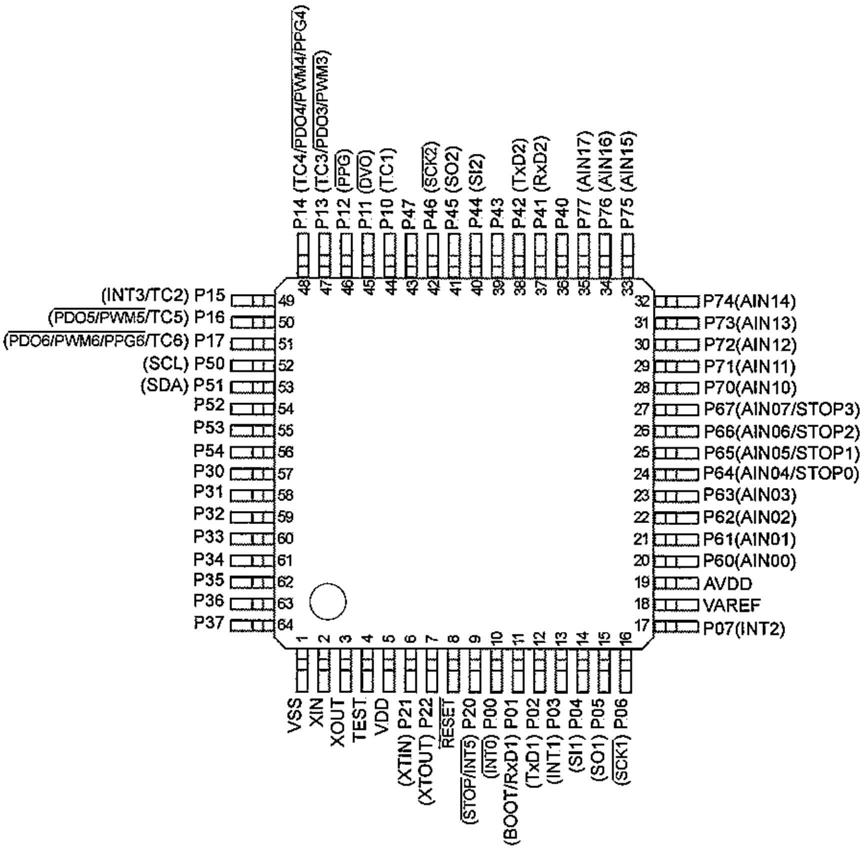


Figure 1-1 Pin Assignment

1.3 Block Diagram

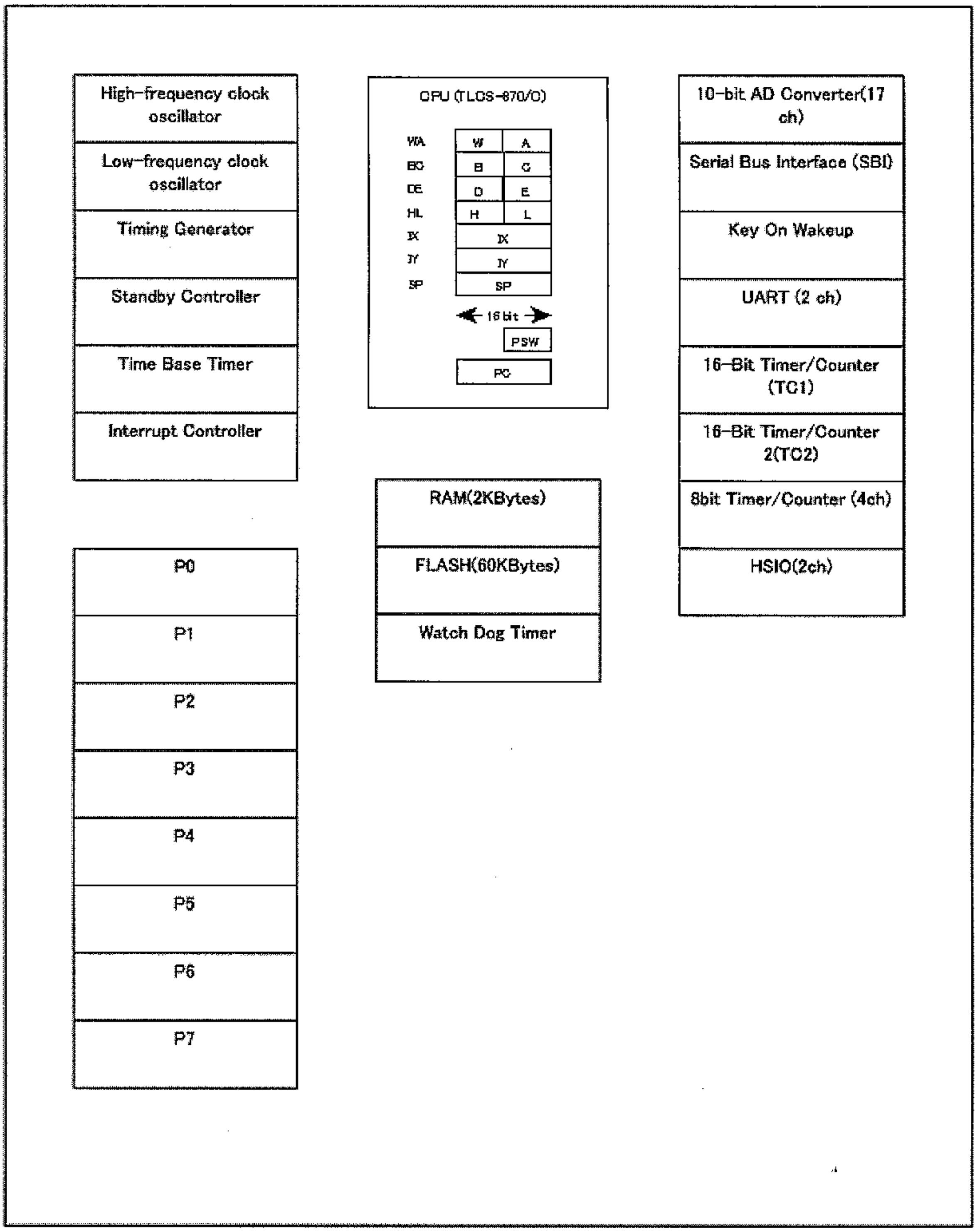


Figure 1-2 Block Diagram

1.4 Pin Names and Functions

Table 1-1 Pin Names and Functions (1/3)

Pin Name	Pin Number	Input/Output	Functions
P07 INT2	17	IO I	PORT07 External interrupt 2 input
P06 $\overline{SCK1}$	16	IO IO	PORT06 Serial clock input/output 1
P05 SO1	15	IO O	PORT05 Serial data output 1
P04 SI1	14	IO I	PORT04 Serial data input 1
P03 INT1	13	IO I	PORT03 External interrupt 1 input
P02 TxD1	12	IO O	PORT02 UART data output 1
P01 RxD1 BOOT	11	IO I I	PORT01 UART data input 1 Serial PROM mode control input
P00 $\overline{INT0}$	10	IO I	PORT00 External interrupt 0 input
P17 TC6 $\overline{PDO6/PWM6/PPG6}$	51	IO I O	PORT17 Timer counter 6 input PDO6/PWM6/PPG6 output
P16 TC5 $\overline{PDO5/PWM5}$	50	IO I O	PORT16 Timer counter 5 input PDO5/PWM5 output
P15 TC2 INT3	49	IO I I	PORT15 Timer counter 2 input External interrupt 3 input
P14 TC4 $\overline{PDO4/PWM4/PPG4}$	48	IO I O	PORT14 Timer counter 4 input PDO4/PWM4/PPG4 output
P13 TC3 $\overline{PDO3/PWM3}$	47	IO I O	PORT13 Timer counter 3 input PDO3/PWM3 output
P12 $\overline{PPG}$	46	IO I	PORT12 PPG Output
P11 $\overline{DVO}$	45	IO O	PORT11 Divider output
P10 TC1	44	IO I	PORT10 Timer counter 1 input
P22 XTOUT	7	IO O	PORT22 Resonator connecting pins(32.768kHz) for inputting external clock
P21 XTIN	6	IO I	PORT21 Resonator connecting pins(32.768kHz) for inputting external clock
P20 $\overline{INT5}$ $\overline{STOP}$	9	IO I I	PORT20 External interrupt 5 input STOP mode release signal input

Table 1-1 Pin Names and Functions (2/3)

Pin Name	Pin Number	Input/Output	Functions
P37	64	IO	PORT37
P36	63	IO	PORT36
P35	62	IO	PORT35
P34	61	IO	PORT34
P33	60	IO	PORT33
P32	59	IO	PORT32
P31	58	IO	PORT31
P30	57	IO	PORT30
P47	43	IO	PORT47
P46 $\overline{\text{SCK2}}$	42	IO IO	PORT46 Serial clock input/output 2
P45 SO2	41	IO O	PORT45 Serial data output 2
P44 SI2	40	IO I	PORT44 Serial data input 2
P43	39	IO	PORT43
P42 TxD2	38	IO O	PORT42 UART data output 2
P41 RxD2	37	IO I	PORT41 UART data input 2
P40	36	IO	PORT40
P54	56	IO	PORT54
P53	55	IO	PORT53
P52	54	IO	PORT52
P51 SDA	53	IO IO	PORT51 I2C bus serial data input/output
P50 SCL	52	IO IO	PORT50 I2C bus serial clock input/output
P67 AIN07 STOP3	27	IO I I	PORT67 AD converter analog input 7 STOP3 input
P66 AIN06 STOP2	26	IO I I	PORT66 AD converter analog input 6 STOP2 input
P65 AIN05 STOP1	25	IO I I	PORT65 AD converter analog input 5 STOP1 input
P64 AIN04 STOP0	24	IO I I	PORT64 AD converter analog input 4 STOP0 input
P63 AIN03	23	IO I	PORT63 AD converter analog input 3
P62 AIN02	22	IO I	PORT62 AD converter analog input 2

Table 1-1 Pin Names and Functions (3/3)

Pin Name	Pin Number	Input/Output	Functions
P61 AIN01	21	IO I	PORT61 AD converter analog input 1
P60 AIN00	20	IO I	PORT60 AD converter analog input 0
P77 AIN17	35	IO I	PORT77 AD converter analog input 17
P76 AIN16	34	IO I	PORT76 AD converter analog input 16
P75 AIN15	33	IO I	PORT75 AD converter analog input 15
P74 AIN14	32	IO I	PORT74 AD converter analog input 14
P73 AIN13	31	IO I	PORT73 AD converter analog input 13
P72 AIN12	30	IO I	PORT72 AD converter analog input 12
P71 AIN11	29	IO I	PORT71 AD converter analog input 11
P70 AIN10	28	IO I	PORT70 AD converter analog input 10
XIN	2	I	Resonator connecting pins for high-frequency clock
XOUT	3	O	Resonator connecting pins for high-frequency clock
$\overline{\text{RESET}}$	8	I	Reset signal input
TEST	4	I	Test pin for out-going test and the Serial PROM mode control pin. Usually fix to low level. Fix to high level when the Serial PROM mode starts.
VAREF	18	I	Analog reference voltage input (High)
AVDD	19	I	AD circuit power supply
VDD	5	I	+5V
VSS	1	I	0(GND)

2. Operational Description

2.1 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.

2.1.1 Memory Address Map

The TMP86FS49UG memory consists of 4 blocks: FLASH, RAM, DBR (Data buffer register) and SFR (Special function register). They are all mapped in 64-Kbyte address space. Figure 2-1 shows the TMP86FS49UG memory address map. The general-purpose registers are not assigned to the RAM address space.

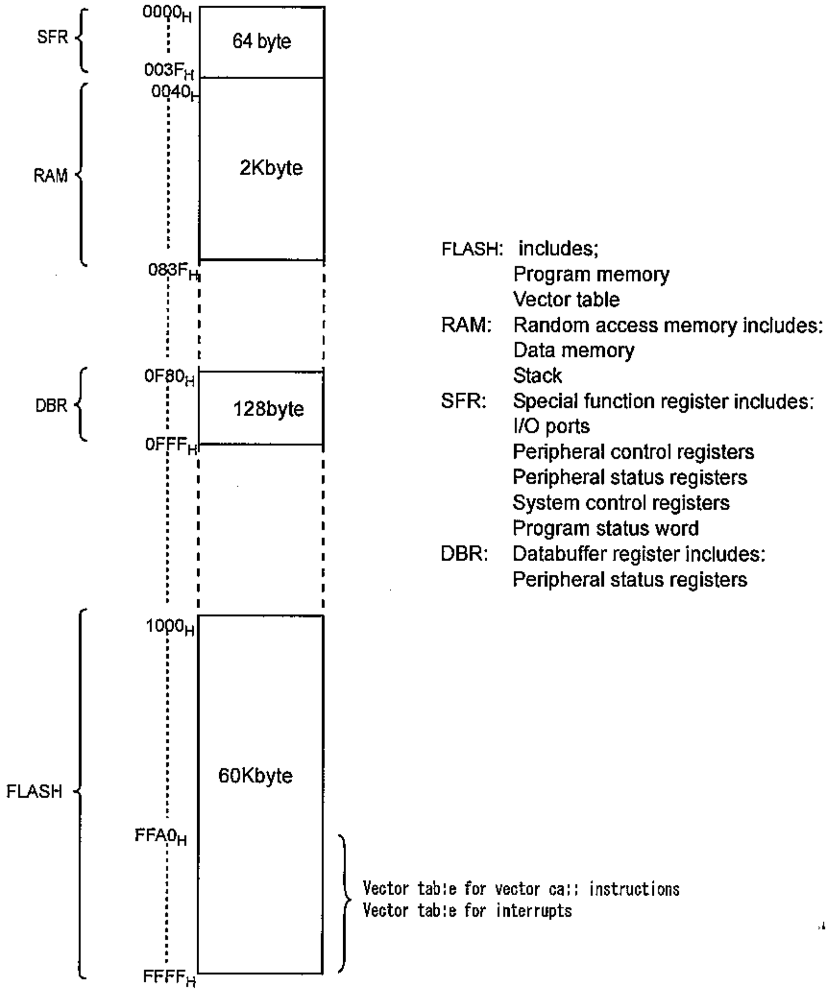

SFR
0000H
64 byte
003FH
0040H
RAM
2Kbyte
083FH
DBR
0F80H
128byte
0FFFH
FLASH
1000H
60Kbyte
FFA0H
FFFFH
Vector tab:e for vector ca:: instructions
Vector tab:e for interrupts
FLASH: includes;
Program memory
Vector table
RAM: Random access memory includes:
Data memory
Stack
SFR: Special function register includes:
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Program status word
DBR: Databuffer register includes:
Peripheral status registers

Figure 2-1 Memory Address Map

2.1.2 Program Memory (FLASH)

The TMP86FS49UG has a 60K × 8 bits (Address 1000H to FFFFH) of program memory (FLASH). A program code placed on the internal RAM can be excutable when a certain procedure is executed (See "2.3.2 Address trap reset ").

2.1.3 Data Memory (RAM)

Data memory consists of internal data memory (Internal FLASH or RAM). The TMP86FS49UG has 2Kbyte (Address 0040H to 083FH) of internal RAM. The first 192 bytes (0040H to 00FFH) of the internal RAM are located in the direct area; instructions with shorten operations are available against such an area.

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

Example :Clears RAM to “00H”. (TMP86FS49UG)

```
            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
```

2.2 System Clock Controller

The system clock controller consists of a clock generator, a timing generator, and a standby controller.

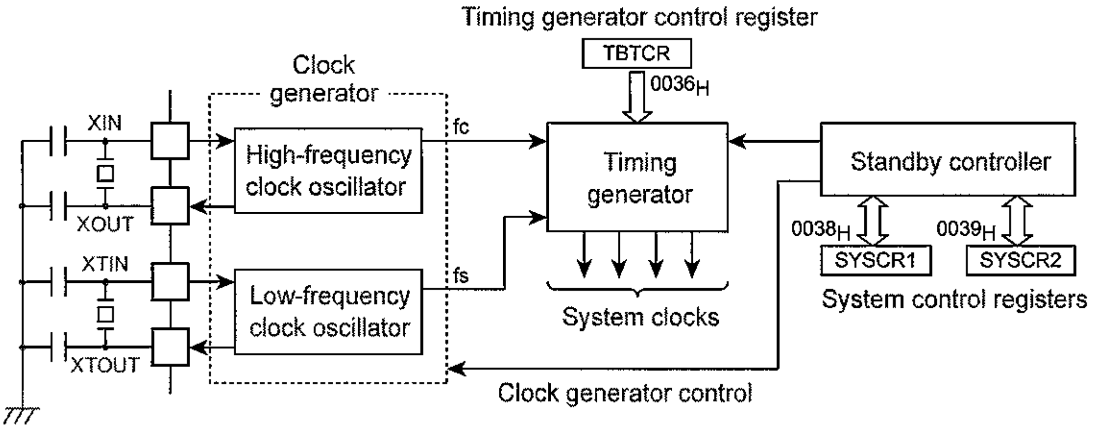


Figure 2-2 System Colck Control

2.2.1 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) clocks and low-frequency (fs) clock 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.

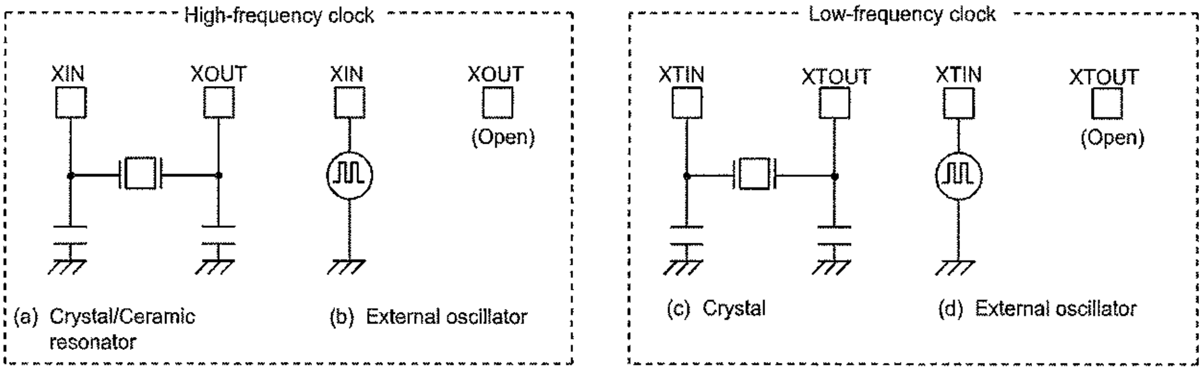


Figure 2-3 Examples of Resonator Connection

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.

2.2.2 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.

1. Generation of main system clock
2. Generation of divider output ($\overline{\text{DVO}}$) pulses
3. Generation of source clocks for time base timer
4. Generation of source clocks for watchdog timer
5. Generation of internal source clocks for timer/counters
6. Generation of warm-up clocks for releasing STOP mode

2.2.2.1 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, DV7CK (Bit4 in TBTCR), that is shown in Figure 1-5. As reset and STOP mode started/canceled, the prescaler and the divider are cleared to “0”.

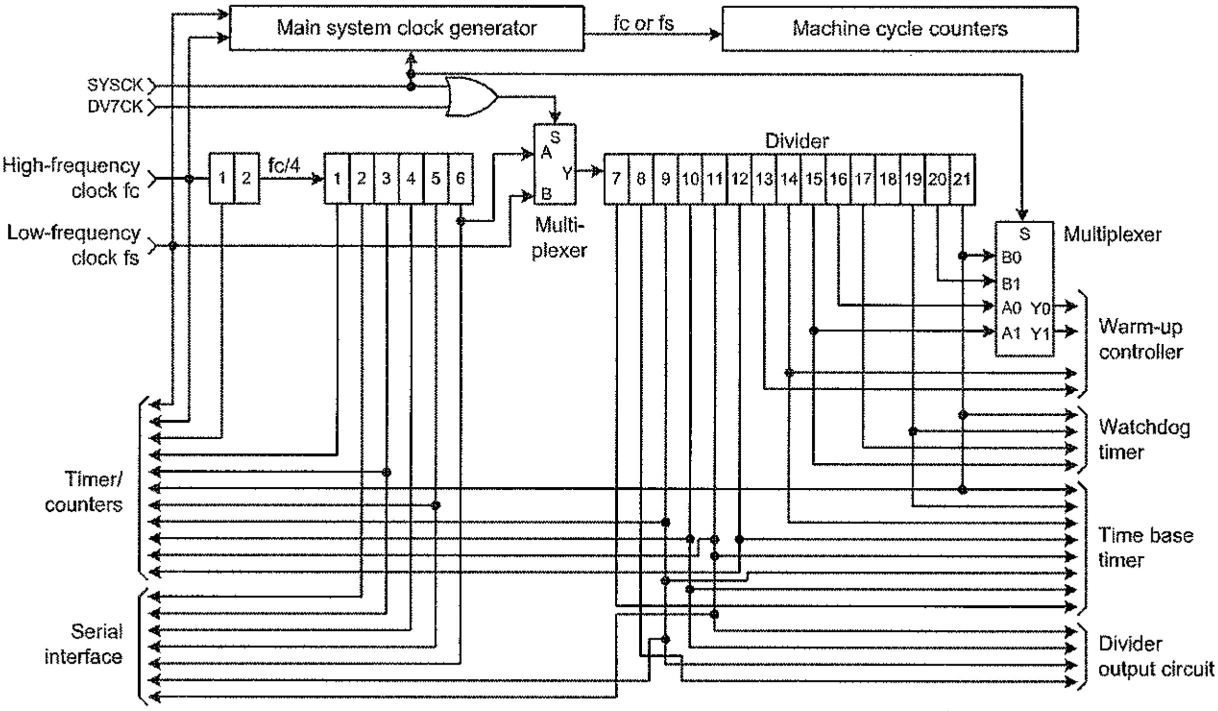


Figure 2-4 Configuration of Timing Generator

Timing Generator Control Register

TBTCR	7	6	5	4	3	2	1	0	
(0036H)	(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 NORMAL1/2 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.

2.2.2.2 Machine cycle

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

The minimum instruction execution unit is called an "machine cycle". There are a total of 10 different types of instructions for the TLCS-870/C Series: Ranging from 1-cycle instructions which require one machine cycle for execution to 10-cycle instructions which require 10 machine cycles for execution. A machine cycle consists of 4 states (S0 to S3), and each state consists of one main system clock.

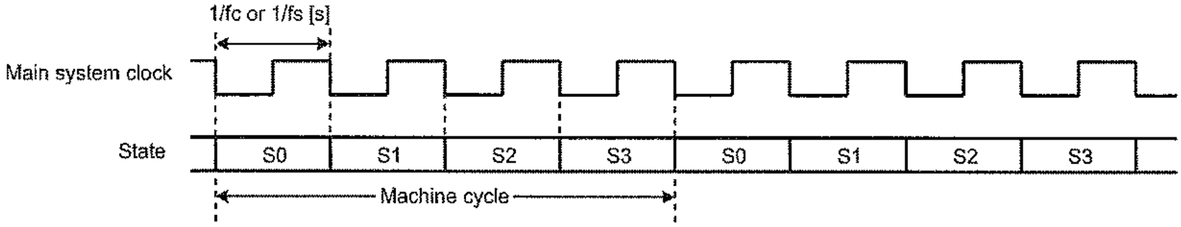


Figure 2-5 Machine Cycle

2.2.3 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 2-6 shows the operating mode transition diagram.

2.2.3.1 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].

(1) NORMAL1 mode

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

The TMP86FS49UG is placed in this mode after reset.

(2) IDLE1 mode

In this mode, the internal oscillation circuit remains active. The CPU and the watchdog timer are halted; however on-chip peripherals remain active (Operate using the high-frequency clock).

IDLE1 mode is started by 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.

(3) IDLE0 mode

In this mode, all the circuit, except oscillator and the timer-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.

2.2.3.2 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 ms at fs = 32.768 kHz) in the SLOW and SLEEP modes.

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

(1) 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.

(2) 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.

(3) 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 modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

(4) 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.

(5) 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 modes, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

(6) 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.

(7) SLEEP0 mode

In this mode, all the circuit, except oscillator and the timer-base-timer, stops operation. This mode is enabled by setting "1" on bit TGHALT on the system control register 2 (SYSCR2).

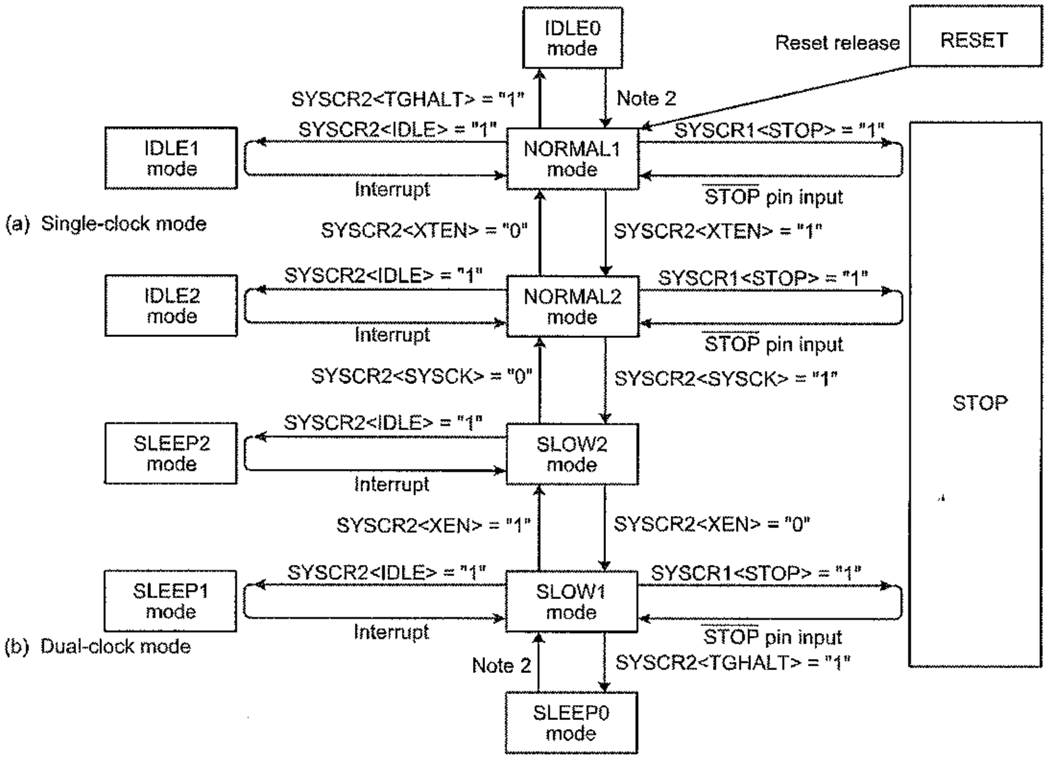
When SLEEP0 mode starts, the CPU stops and the timing generator stops feeding the clock to the peripheral circuits other than TBT. Then, upon detecting the falling edge of the source clock selected with TBTCR<TBTCK>, the timing generator starts feeding the clock to all peripheral circuits.

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.

2.2.3.3 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. After the warm-up period is completed, the execution resumes with the instruction which follows the STOP mode start instruction.



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.

Figure 2-6 Operating Mode Transition Diagram

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

System Control Register 1

SYSCR1 (0038H)	7	6	5	4	3	2	1	0	
	STOP	RELM	RETM	OUTEN	WUT				(Initial value: 0000 00**)

STOP	STOP mode start	0: CPU core and peripherals remain active 1: CPU core and peripherals are halted (Start STOP mode)			R/W
RELM	Release method for STOP mode	0: Edge-sensitive release 1: Level-sensitive release			
RETM	Operating mode after STOP mode	0: Return to NORMAL1/2 mode 1: Return to SLOW1 mode			
OUTEN	Port output during STOP mode	0: High impedance 1: Output kept			
WUT	Warm-up time at releasing STOP mode		Return to NORMAL mode	Return to SLOW mode	
		00	3×2^{16}/fc	3×2^{13}/fs	
		01	2^{16}/fc	2^{13}/fs	
		10	3×2^{14}/fc	3×2^{6}/fs	
		11	2^{14}/fc	2^{6}/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 is used, the edge realease can not function according to some conditions. It is recommended to set the level realease (RELM = "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: The warmig-up time should be set correctly for using oscillator.

System Control Register 2

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

XEN	High-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	R/W
XTEN	Low-frequency oscillator control	0: Turn off oscillation 1: Turn on oscillation	
SYSCK	Main system clock select (Write)/main system clock monitor (Read)	0: High-frequency clock 1: Low-frequency clock	
IDLE	CPU and watchdog timer control (IDLE1/2 and SLEEP1/2 modes)	0: CPU and watchdog timer remain active 1: CPU and watchdog timer are stopped (Start IDLE1/2 and SLEEP1/2 modes)	
TGHALT	TG control (IDLE0 and SLEEP0 modes)	0: Feeding clock to all peripherals from TG 1: Stop feeding clock to peripherals except TBT from TG. (Start IDLE0 and SLEEP0 modes)	

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.

2.2.4 Operating Mode Control

2.2.4.1 STOP mode

STOP mode is controlled by the system control register 1, the $\overline{\text{STOP}}$ pin input. The $\overline{\text{STOP}}$ pin is also used both as a port P20 and an $\overline{\text{INT5}}$ (external interrupt input 5) pin.

STOP mode is started by setting STOP (Bit7 in SYSCR1) to "1". During STOP mode, the following status is maintained.

1. Oscillations are turned off, and all internal operations are halted.
2. The data memory, registers, the program status word and port output latches are all held in the status in effect before STOP mode was entered.
3. The prescaler and the divider of the timing generator are cleared to "0".
4. 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 RELM (Bit6 in SYSCR1).

Note 1: 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.

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

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high. This mode is used for capacitor 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.

1. Testing a port P20.
2. Using an external interrupt input $\overline{\text{INT5}}$ ($\overline{\text{INT5}}$ is a falling edge-sensitive input).

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

```
            LD      (SYSCR1), 01010000B    ; Sets up the level-sensitive release mode
SSTOPH:     TEST    (P2PRD). 0             ; Wait until the STOP pin input goes low level
            JRS     F, SSTOPH
            DI                             ; IMF ← 0
            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
            JRS     F, SINT5                 port P20 is at high
            LD      (SYSCR1), 01010000B    ; Sets up the level-sensitive release mode.
            DI                             ; IMF ← 0
            SET     (SYSCR1). 7            ; Starts STOP mode
SINT5:      RETI
```

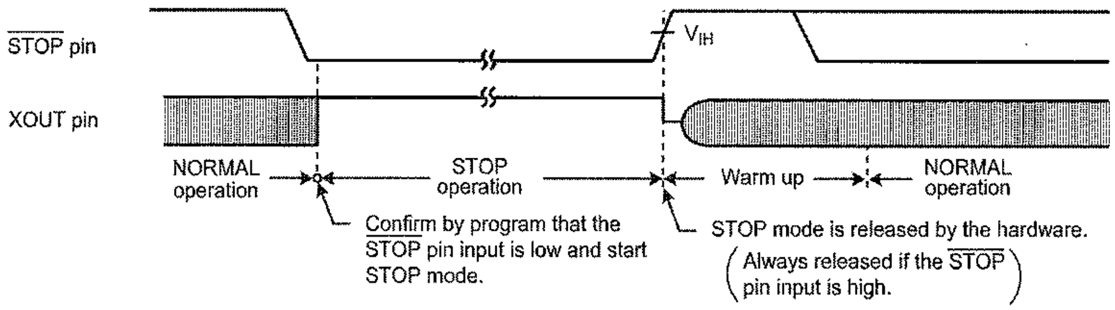

STOP pin
XOUT pin
VIH
NORMAL operation
STOP operation
Warm up
NORMAL operation
Confirm by program that the STOP pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the STOP pin input is high.)

Figure 2-7 Level-sensitive Release Mode

Note 1: 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.

(2) 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.

Example :Starting STOP mode from NORMAL mode

```
            DI                             ; IMF ← 0
            LD      (SYSCR1), 10010000B    ; Starts after specified to the edge-sensitive release mode
```

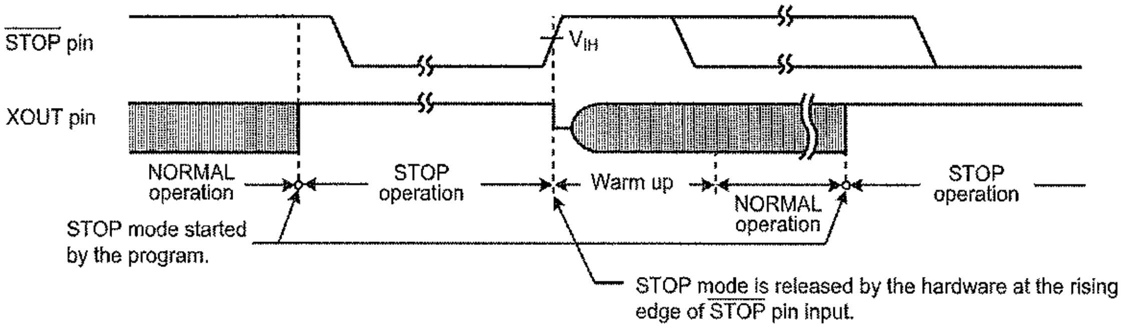

$\overline{\text{STOP}}$ pin
XOUT pin
V_{IH}
NORMAL operation
STOP operation
Warm up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of $\overline{\text{STOP}}$ pin input.

Figure 2-8 Edge-sensitive Release Mode

STOP mode is released by the following sequence.

1. 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.
2. A warm-up period is inserted to allow oscillation time to stabilize. During warm up, all internal operations remain halted. Four different warm-up times can be selected with the WUT (Bits 2 and 3 in SYSCR1) in accordance with the resonator characteristics.
3. When the warm-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction. The start is made after the prescaler and the divider of the timing generator are cleared to "0".

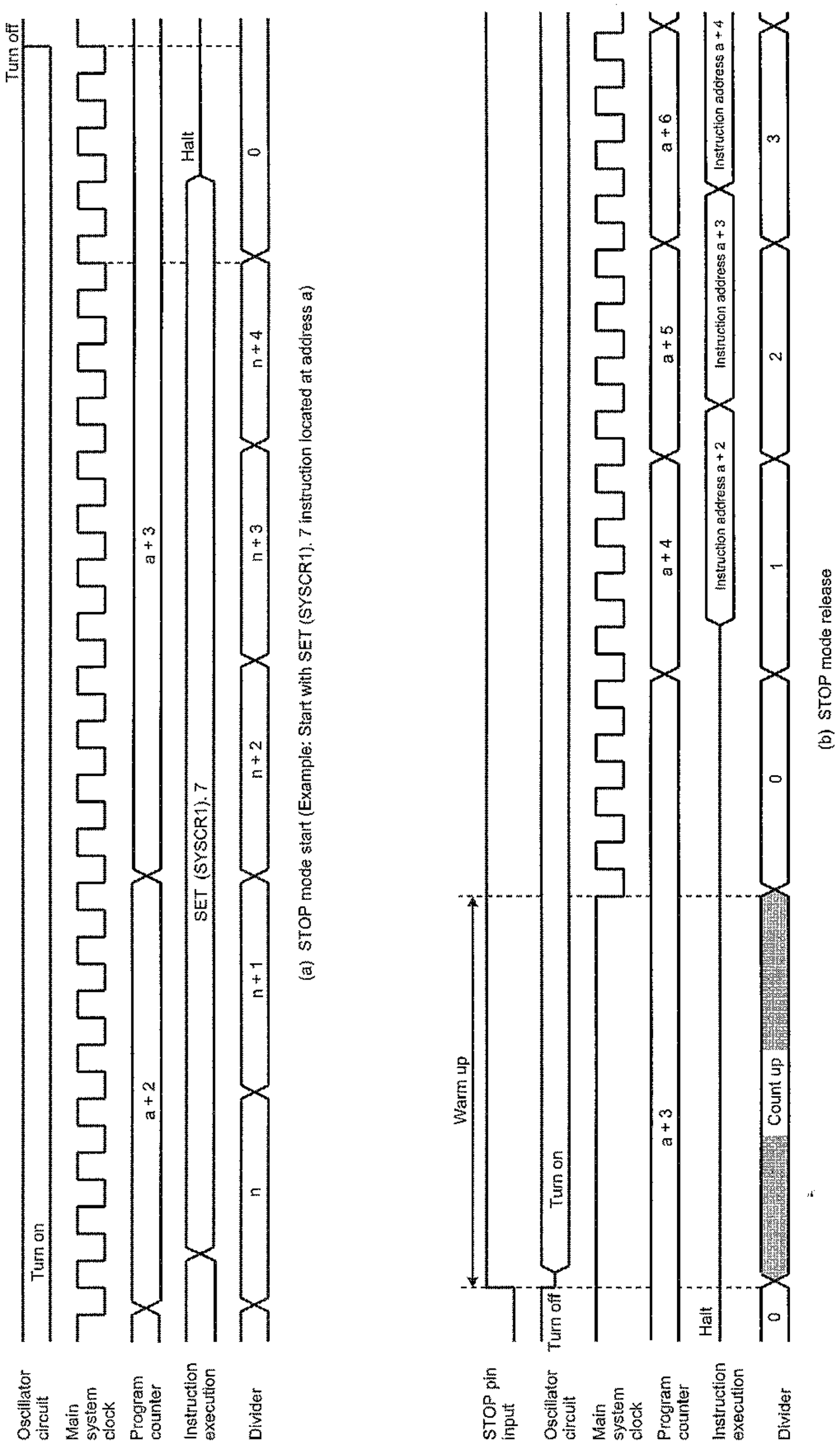
Table 2-1 Warm-up Time Example (at fc = 16.0 MHz, fs = 32.768 kHz)

WUT	Warm-up Time [ms]	
	Return to NORMAL Mode	Return to SLOW Mode
00	12.288	750
01	4.096	250
10	3.072	5.85
11	1.024	1.95

Note: 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 as an approximate value.

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

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



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

(b) STOP mode release

Figure 2-9 STOP Mode Start/Release

2.2.4.2 IDLE1/2 mode and 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.

1. Operation of the CPU and watchdog timer (WDT) is halted. On-chip peripherals continue to operate.
2. 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.
3. The program counter holds the address 2 ahead of the instruction which starts these modes.

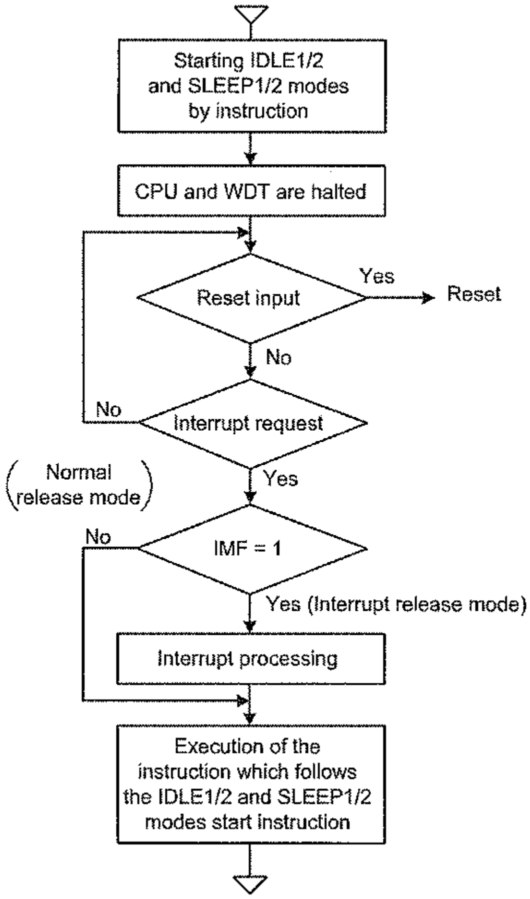

Starting IDLE1/2 and SLEEP1/2 modes by instruction
CPU and WDT are halted
Reset input
Yes
Reset
No
No
Interrupt request
Yes
Normal release mode
No
IMF = 1
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE1/2 and SLEEP1/2 modes start instruction

Figure 2-10 IDLE1/2 and SLEEP1/2 Modes

(1) Start the IDLE1/2 and SLEEP1/2 modes

When IDLE1/2 and SLEEP1/2 modes start, set SYSCR2<IDLE> to “1”. After IMF is set to "0", set the individual interrupt enable flag (EF) which releases IDLE1/2 and SLEEP1/2.

(2) 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.

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.

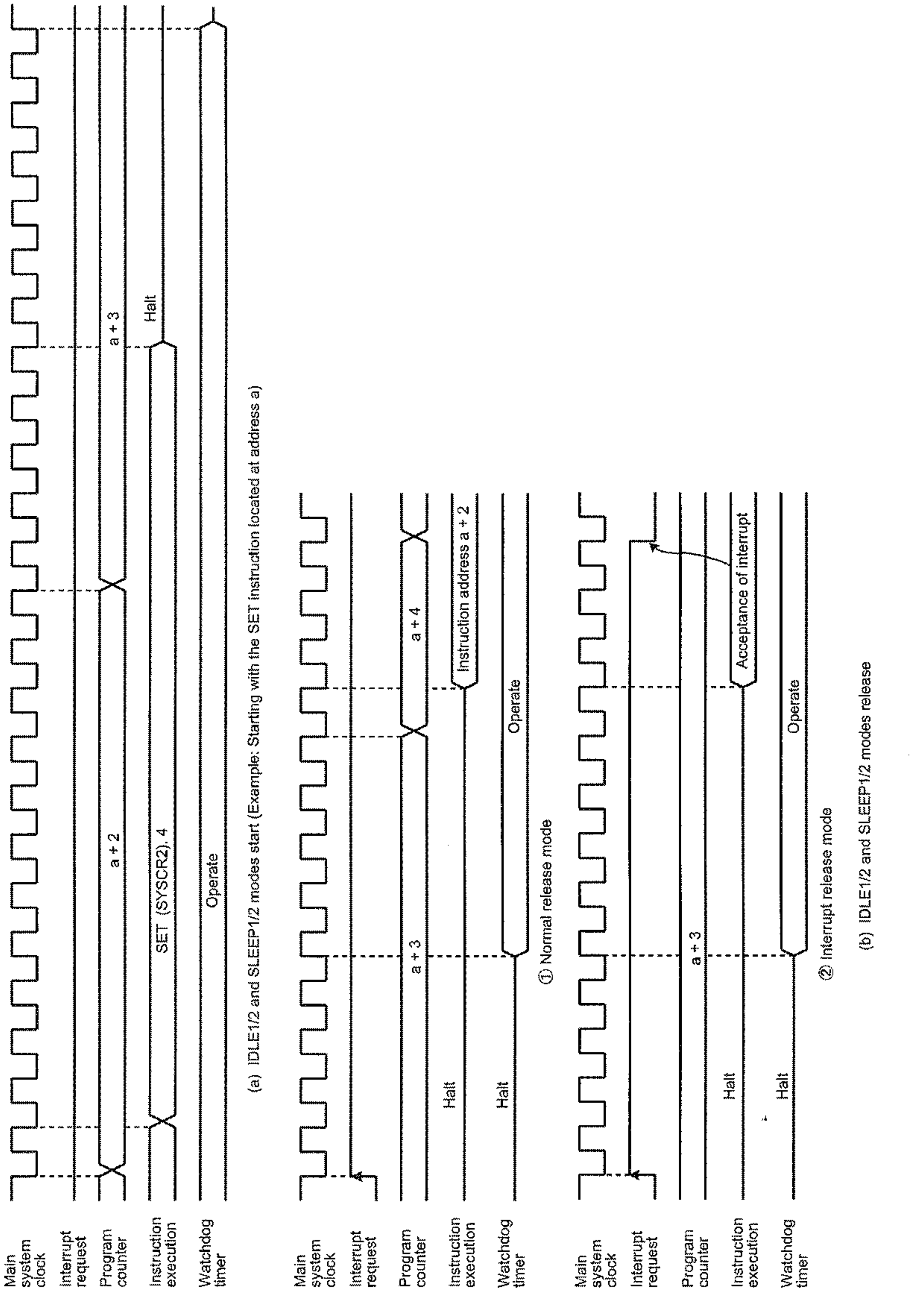
(3) Normal release mode (IMF = "0")

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.

(4) 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) and the interrupt processing is started. 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.



(a) IDLE1/2 and SLEEP1/2 modes start (Example: Starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE1/2 and SLEEP1/2 modes release

Figure 2-11 IDLE1/2 and SLEEP1/2 Modes Start/Release

2.2.4.3 IDLE0 and SLEEP0 modes (IDLE0, SLEEP0)

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

1. Timing generator stops feeding clock to peripherals except TBT.
2. 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.
3. 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) peripherals.

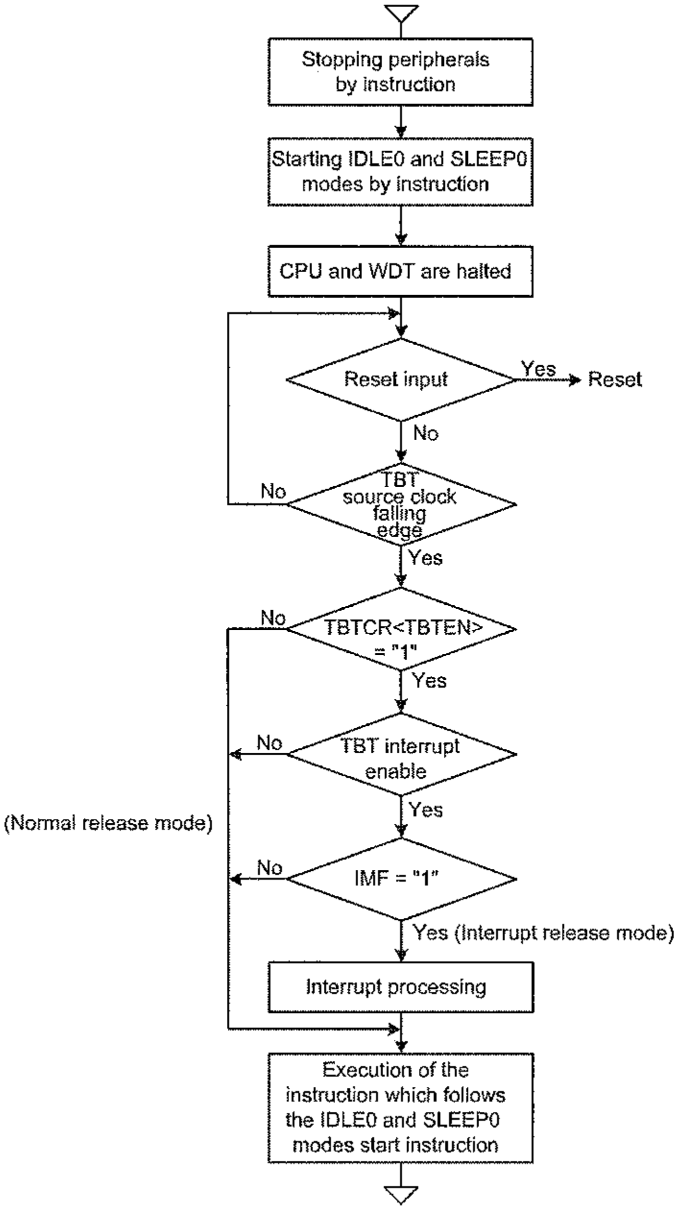

Stopping peripherals by instruction
Starting IDLE0 and SLEEP0 modes by instruction
CPU and WDT are halted
Reset input
Yes
Reset
No
TBT source clock falling edge
No
Yes
TBTCR<TBTEN> = "1"
No
Yes
TBT interrupt enable
No
Yes
(Normal release mode)
IMF = "1"
No
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE0 and SLEEP0 modes start instruction

Figure 2-12 IDLE0 and SLEEP0 Modes

(1) 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".

(2) Release the IDLE0 and SLEEP0 modes

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

These modes are selected by interrupt master flag (IMF), the individual interrupt enable flag of TBT 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".

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

(3) Normal release mode (IMF • EF7 • TBTCR<TBTEN> = "0")

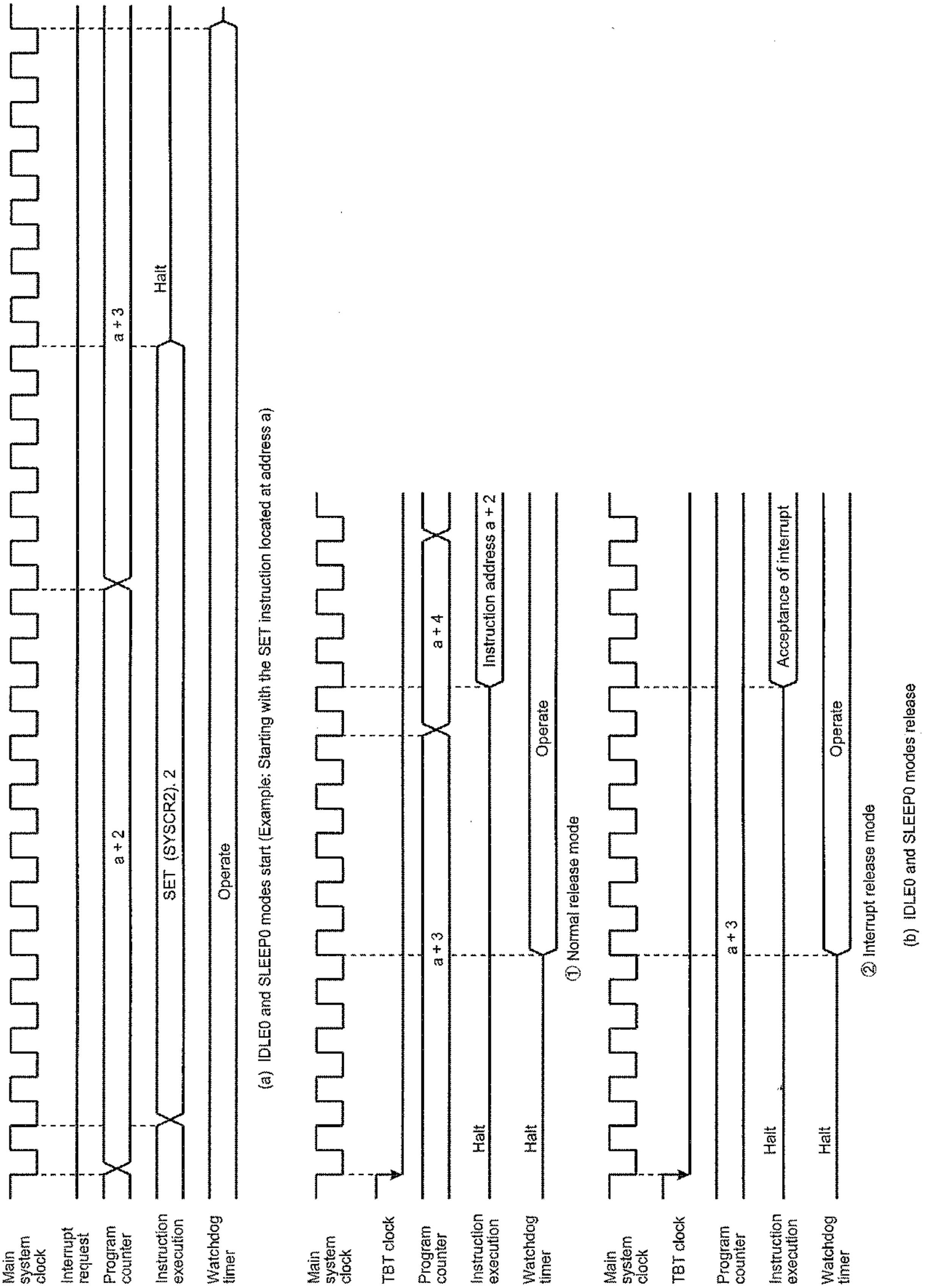
IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> without reference to individual interrupt enable flag (EF). After the falling edge is detected, the program operation is resumed from the instruction following the IDLE0 and SLEEP0 modes start instruction.

(4) Interrupt release mode (IMF • EF7 • TBTCR<TBTEN> = "1")

IDLE0 and SLEEP0 modes are released by the source clock falling edge, which is setting by the TBTCR<TBTCK> at INTTBT interrupt source enabled with the individual interrupt enable flag (EF) 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.



(a) IDLE0 and SLEEP0 modes start (Example: Starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE0 and SLEEP0 modes release

Figure 2-13 IDLE0 and SLEEP0 Modes Start/Release

2.2.4.4 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).

(1) Switching from NORMAL2 mode to SLOW1 mode

First, set SYSCK (Bit5 in SYSCR2) to switch the main system clock to the low-frequency clock for SLOW2 mode.

Next, clear XEN (Bit7 in SYSCR2) to turn off high-frequency oscillation.

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

When the low-frequency clock oscillation is unstable, wait until oscillation stabilizes before performing the above operations. The timer/counter 6, 5, 4, 3 (TC6, TC5, 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)
```

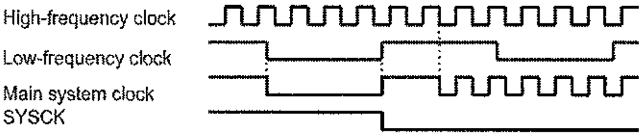
Example 2 :Switching to the SLOW1 mode after low-frequency clock has stabilized.

```
            SET      (SYSCR2). 6          ; SYSCR2<XTEN> ← 1
            LD       (TC3CR), 43H         ; Sets mode for TC4, TC3 (16-bit TC, fs for source)
            LD       (TC4CR), 05H
            LDW      (TTREG3), 8000H      ; Sets warm-up time (Depend on oscillator accompanied)
            DI                            ; IMF ← 0
            SET      (EIRH). 1            ; 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
```

(2) Switching from SLOW1 mode to NORMAL2 mode

First, set XEN (Bit7 in SYSCR2) to turn on the high-frequency oscillation. When time for stabilization (Warm up) has been taken by the timer/counter 4, 3 (TC4, TC3), clear SYSCK (Bit5 in SYSCR2) to switch the main system clock to the high-frequency clock.

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



Note: SLOW mode can also be released by inputting low level on the RESET pin, which immediately performs the reset operation. After reset, TMP86FS49UG are 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)
          LD     (TC4CR), 05H
          LD     (TTREG4), 0F8H     ; Sets warm-up time
          DI                        ; IMF ← 0
          SET    (EIRH). 1          ; 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
```

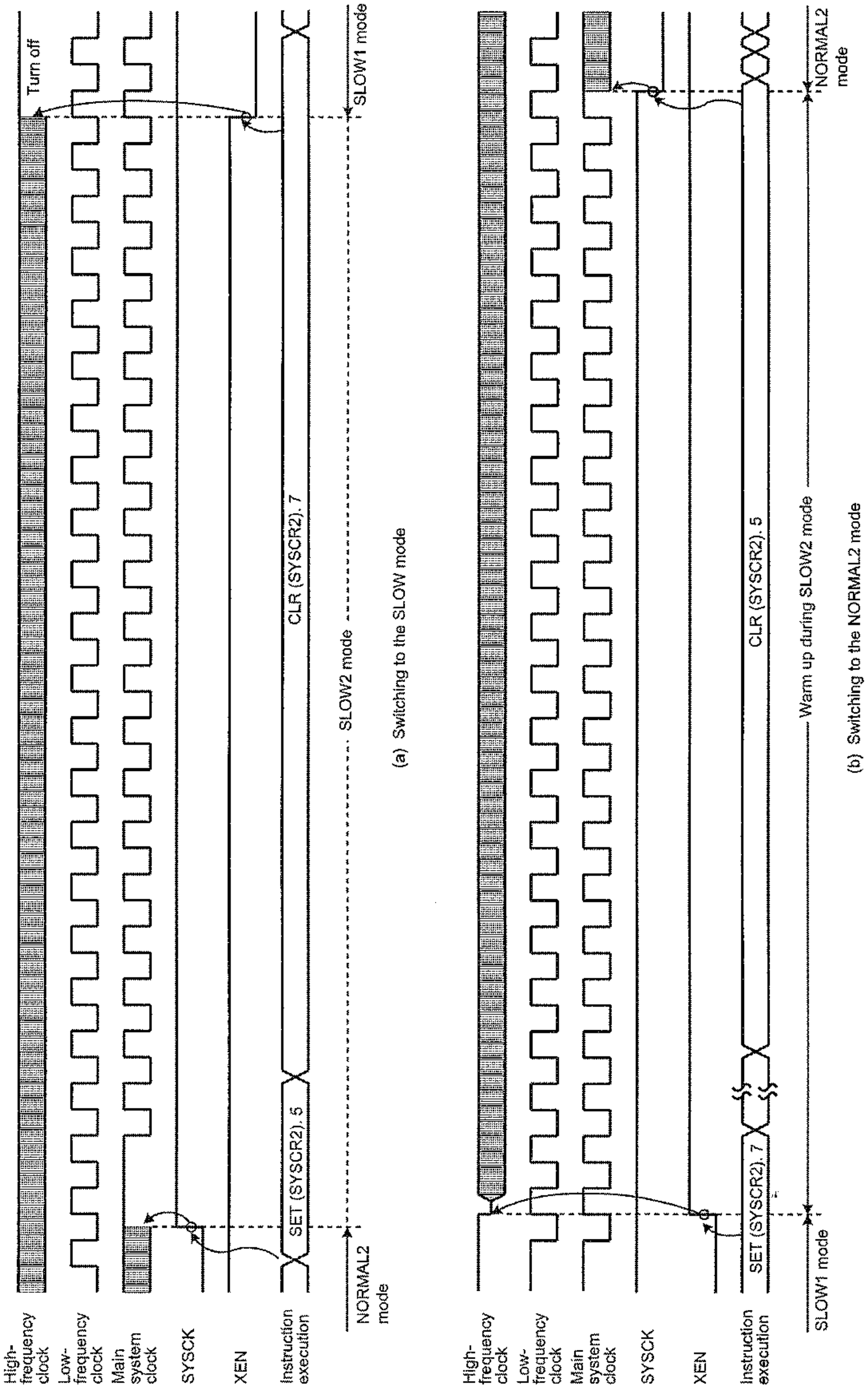


Figure 2-14 Switching between the NORMAL2 and SLOW Modes

2.3 Reset Circuit

The TMP86FS49UG have four types of reset generation procedures: An external reset input, an address trap reset, a watchdog timer reset and a system clock reset. Table 2-2 shows on-chip hardware initialization by reset action.

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 $\overline{\text{RESET}}$ pin can reset state at the maximum 24/fc [s] (1.5 µs at 16.0 MHz) when power is turned on.

Table 2-2 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

2.3.1 External Reset Input

The $\overline{\text{RESET}}$ pin contains a Schmitt trigger (Hysteresis) with an internal pull-up resistor.

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

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

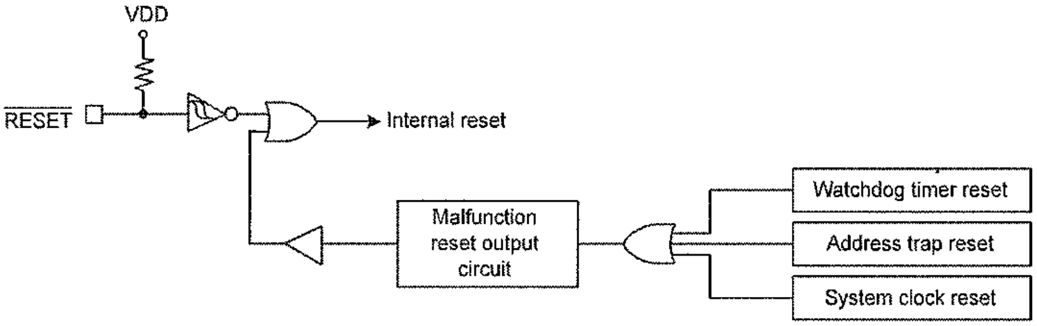
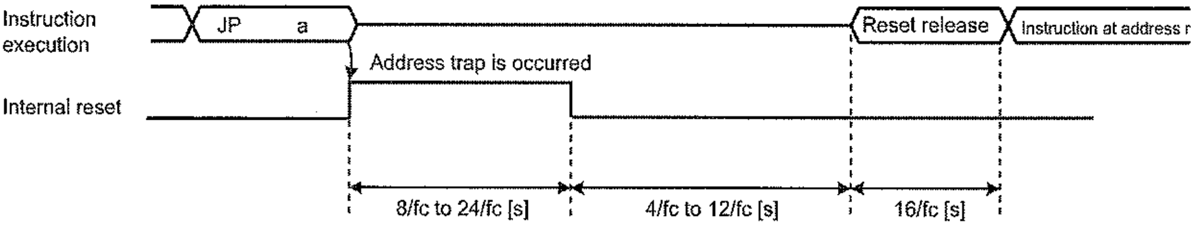


Figure 2-15 Reset Circuit

2.3.2 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 will be generated. The reset time is about 8/fc to 24/fc [s] (0.5 to 1.5 μs at 16.0 MHz).

Note: The operating mode under address trapped is alternative of reset or interrupt. The address trap area is alternative.



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.

Figure 2-16 Address Trap Reset

2.3.3 Watchdog timer reset

Refer to 2.4 "Watchdog Timer".

2.3.4 System clock reset

Clearing both XEN and XTEN (Bits 7 and 6 in SYSCR2) to "0", clearing XEN to "0" when SYSCK = "0", or clearing XTEN to "0" when SYSCK = "1" stops system clock, and causes the microcomputer to deadlock. This can be prevented by automatically generating a reset signal whenever XEN = XTEN = "0", XEN = SYSCK = "0", or XTEN = "0"/SYSCK = "1" is detected to continue the oscillation. The reset time is about 8/fc to 24/fc [s] (0.5 to 1.5 μs at 16.0 MHz).

3. Interrupt Control Circuit

The TMP86FS49UG has a total of 24 interrupt sources excluding reset, of which 0 source levels are multiplexed. Interrupts can be nested with priorities. Four of the internal interrupt sources are non-maskable while the rest are maskable.

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.

Interrupt Factors		Enable Condition	Interrupt Latch	Vector Address	Priority
Internal/External	(Reset)	Non-maskable	–	FFFEH	1
Internal	INTSWI (Software interrupt)	Non-maskable	–	FFFCH	2
Internal	INTUNDEF (Executed the undefined instruction interrupt)	Non-maskable	–	FFFCH	2
Internal	INTATRAP (Address trap interrupt)	Non-maskable	IL2	FFFAH	2
Internal	INTWDT (Watchdog timer interrupt)	Non-maskable	IL3	FFF8H	2
External	$\overline{\text{INT0}}$	IMF• EF4 = 1, INT0EN = 1	IL4	FFF6	5
Internal	INTTC1	IMF• EF5 = 1	IL5	FFF4	6
External	INT1	IMF• EF6 = 1	IL6	FFF2	7
Internal	INTTBT	IMF• EF7 = 1	IL7	FFF0	8
External	INT2	IMF• EF8 = 1	IL8	FFEE	9
Internal	INTTC4	IMF• EF9 = 1	IL9	FFEC	10
Internal	INTTC3	IMF• EF10 = 1	IL10	FFEA	11
Internal	INTSBI	IMF• EF11 = 1	IL11	FFE8	12
External	INT3	IMF• EF12 = 1	IL12	FFE6	13
Internal	INTSIO1	IMF• EF13 = 1	IL13	FFE4	14
Internal	INTSIO2	IMF• EF14 = 1	IL14	FFE2	15
Internal	INTADC	IMF• EF15 = 1	IL15	FFE0	16
Internal	INTRXD1	IMF• EF16 = 1	IL16	FFBE	17
Internal	INTTXD1	IMF• EF17 = 1	IL17	FFBC	18
Internal	INTTC6	IMF• EF18 = 1	IL18	FFBA	19
Internal	INTTC5	IMF• EF19 = 1	IL19	FFB8	20
Internal	INTRXD2	IMF• EF20 = 1	IL20	FFB6	21
Internal	INTTXD2	IMF• EF21 = 1	IL21	FFB4	22
Internal	INTTC2	IMF• EF22 = 1	IL22	FFB2	23
External	$\overline{\text{INT5}}$	IMF• EF23 = 1	IL23	FFB0	24
-	Reserved	IMF• EF24 = 1	IL24	FFAE	25
-	Reserved	IMF• EF25 = 1	IL25	FFAC	26
-	Reserved	IMF• EF26 = 1	IL26	FFAA	27
-	Reserved	IMF• EF27 = 1	IL27	FFA8	28
-	Reserved	IMF• EF28 = 1	IL28	FFA6	29
-	Reserved	IMF• EF29 = 1	IL29	FFA4	30
-	Reserved	IMF• EF30 = 1	IL30	FFA2	31
-	Reserved	IMF• EF31 = 1	IL31	FFA0	32

Note 1: To use the address trap interrupt (INTATRAP), clear WDTCR1<ATOUT> to "0" (It is set for the "reset request" after reset is cancelled). For details, see "Address Trap".

3.1 Interrupt latches (IL24 to IL2)

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

The interrupt latches are located on address 002EH, 002FH, 003CH and 003DH in SFR area. Except for IL3 and IL2, 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: Before manipulating each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts). The interrupt enable flags (EFs) and interrupt latches (ILs) must be manipulated under the following conditions; otherwise operation cannot be guaranteed.
1. After the DI instruction is executed.
2. The IMF is automatically cleared to "0" when an interrupt is accepted. However, when interrupt nesting is enabled, be sure to manipulate the EFs and ILs before setting the IMF to "1" (to enable interrupts).

Example 1 :Clears interrupt latches

```
DI                                      ; IMF ← 0
LDW      (ILL), 1110100000111111B       ; IL12, IL10 to IL6 ← 0
EI                                      ; IMF ← 1
```

Example 2 :Reads interrupt latchess

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

Example 3 :Tests interrupt latches

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

3.2 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 002CH, 002DH, 003AH and 003BH in SFR area, and they can be read and written by an instructions (Including read-modify-write instructions such as bit manipulation or operation instructions).

3.2.1 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".

3.2.2 Individual interrupt enable flags (EF24 to EF4)

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. During reset, all the individual interrupt enable flags (EF20 to EF4) are initialized to "0" and all maskable interrupts are not accepted until they are set to "1".

Note: Before manipulating each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts). The interrupt enable flags (EFs) and interrupt latches (ILs) must be manipulated under the following conditions; otherwise operation cannot be guaranteed.
1. After the DI instruction is executed.
2. The IMF is automatically cleared to "0" when an interrupt is accepted. However, when interrupt nesting is enabled, be sure to manipulate the EFs and ILs before setting the IMF to "1" (to enable interrupts).

Example 1 :Enables interrupts individually and sets IMF

```
DI                                          ; IMF ← 0

LDW       (EIRL), 1110100010100000B         ; EF15 to 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

(Initial value: 00000000000000**)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL (0030DH, 003CH)	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8	IL7	IL6	IL5	IL4	IL3	IL2		
	ILH (003DH)								ILL (003CH)							

(Initial value: 0000000000000000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIL (002FH, 002EH)	IL31	IL30	IL29	IL28	IL27	IL26	IL25	IL24	IL23	IL22	IL21	IL20	IL19	IL18	IL17	IL16
	ILD (002FH)								ILE (002EH)							

IL20~IL2	Interrupt latches	at RD 0: No interrupt request 1: Interrupt request	at WR 0: Clears the interrupt request 1: (Interrupt latch is not set.)	R/W

Note 1: To clear any one of bits IL7 to IL4, be sure to write "1" into IL2 and IL3.

Note 2: Before manipulating each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts). The interrupt enable flags (EFs) and interrupt latches (ILs) must be manipulated under the following conditions; otherwise operation cannot be guaranteed.

1. After the DI instruction is executed.
2. The IMF is automatically cleared to "0" when an interrupt is accepted. However, when interrupt nesting is enabled, be sure to manipulate the EFs and ILs before setting the IMF to "1" (to enable interrupts).

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

Interrupt Enable Registers

(Initial value: 000000000000***0)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIR (003BH, 003AH)	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8	EF7	EF6	EF5	EF4				IMF
	EIRH (003BH)								EIRL (003AH)							

(Initial value: 0000000000000000)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEIR (002DH, 002CH)	EF31	EF30	EF29	EF28	EF27	EF26	EF25	EF24	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
	EIRD (002DH)								EIRE (002CH)							

EF20~EF4	Individual-interrupt enable flag (Specified for each bit)	0: 1:	0: Disables the acceptance of each maskable interrupt. 1: Enables the acceptance of each maskable interrupt.	R/W
IMF	Interrupt master enable flag	0: 1:	0: Disables the acceptance of all maskable interrupts 1: Enables the acceptance of all maskable interrupts	

Note 1: *: Don't care

Note 2: Do not set IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 3: Before manipulating each interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (to disable interrupts). The interrupt enable flags (EFs) and interrupt latches (ILs) must be manipulated under the following conditions; otherwise operation cannot be guaranteed.

1. After the DI instruction is executed.
2. The IMF is automatically cleared to "0" when an interrupt is accepted. However, when interrupt nesting is enabled, be sure to manipulate the EFs and ILs before setting the IMF to "1" (to enable interrupts).

3.3 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 (2 ms @16 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 3-1 shows the timing chart of interrupt acceptance processing.

3.3.1 Interrupt acceptance processing is packaged as follows.

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

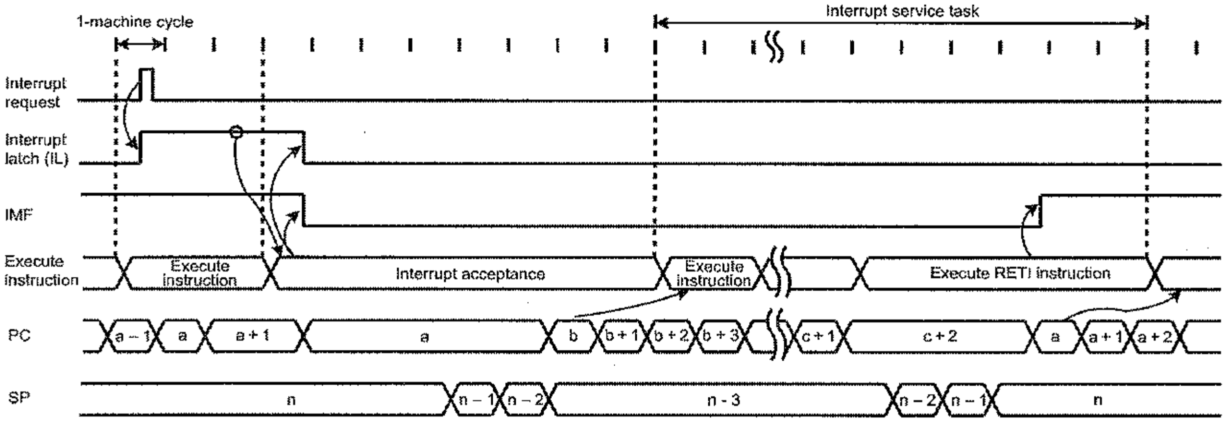
b. The interrupt latch (IL) for the interrupt source accepted is cleared to "0".

c. 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.

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

e. 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.

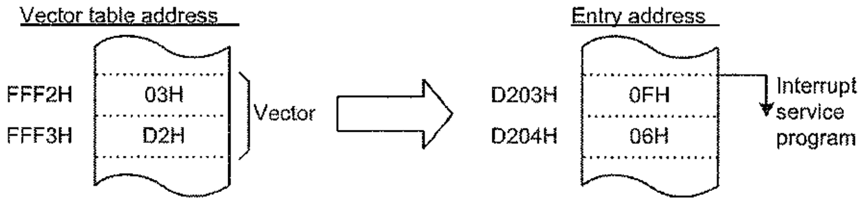


Note 1: a: Return address entry address, b: Entry address, c: Address which RETI instruction is stored

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 3-1 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 shorten compared with length between interrupt requests; otherwise the status cannot be recovered as non-maskable interrupt would simply nested.

3.3.2 Saving/restoring general-purpose registers

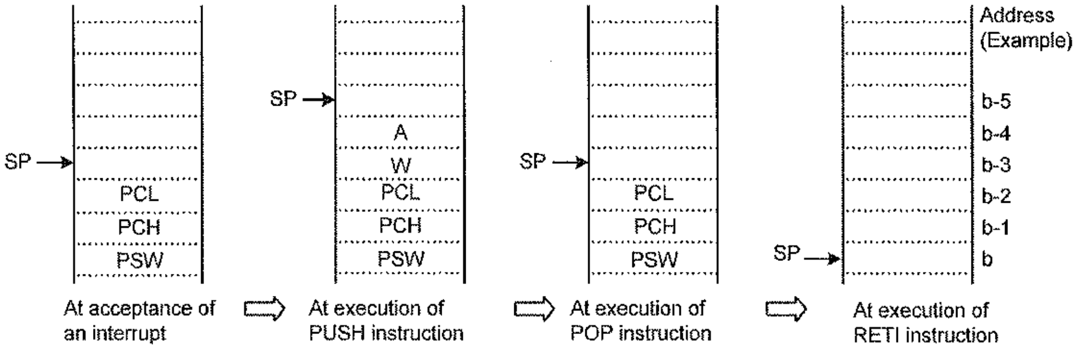
During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the accumulator and others are not. These registers are saved by software if necessary. When multiple interrupt services are nested, it is also necessary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

3.3.2.1 Using PUSH and POP instructions

If only a specific register is saved or interrupts of the same source are nested, general-purpose registers can be saved/restored using the PUSH/POP instructions.

Example :Save/store register using PUSH and POP instructions

```
PINTxx:     PUSH     WA               ; Save WA register
            (interrupt processing)
            POP      WA               ; Restore WA register
            RETI                      ; RETURN
```

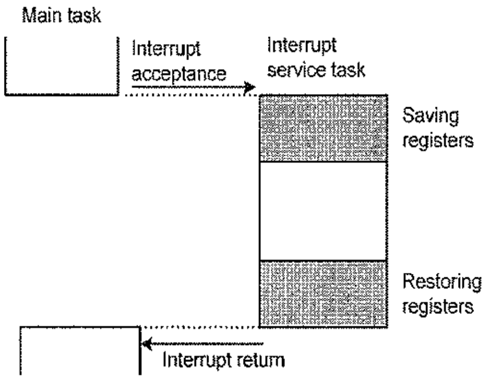

SP
PCL
PCH
PSW
At acceptance of an interrupt
SP
A
W
PCL
PCH
PSW
At execution of PUSH instruction
SP
PCL
PCH
PSW
At execution of POP instruction
SP
At execution of RETI instruction
Address (Example)
b-5
b-4
b-3
b-2
b-1
b

3.3.2.2 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 data transfer instruction

Figure 3-2 Saving/Restoring General-purpose Registers under Interrupt Processing

3.3.3 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 (INTATRAP), it is required to alter stacked data for program counter (PC) to restarting address, during interrupt service program.

Note: If [RETN] is executed with the above data unaltered, the program returns to the address trap area and INTATRAP occurs again.When interrupt acceptance processing has completed, stacked data for PCL and PCH are located on address (SP + 1) and (SP + 2) respectively.

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)
            RETN                          ; RETURN
```

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
```

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 1: 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).

Note 2: When the interrupt processing time is longer than the interrupt request generation time, the interrupt service task is performed but not the main task.

3.4 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.

3.4.1 Address error detection

FFH is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address during single chip mode. Code FFH is the SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing FFH to unused areas of the program memory. Address trap reset is generated in case that an instruction is fetched from RAM, SFR or DBR areas.

3.4.2 Debugging

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

3.5 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.

3.6 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).

3.7 External Interrupts

The TMP86FS49UG have 5 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 INT0 to INT5. The $\overline{\text{INT0}}$/P00 pin can be configured as either an external interrupt input pin or an input/output port, and is configured as an input port during reset.

Edge selection, noise reject control and $\overline{\text{INT0}}$/P00 pin function selection are performed by the external interrupt control register (EINTCR).

Source	Pin	seconday function	Enable Conditions	Edge	Digital Noise Reject
INT0	$\overline{\text{INT0}}$	P00	IMF = 1, EF4 = 1, INT0EN = 1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT1	INT1	P03	IMF•EF6=1	Falling edge or Rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 49/fc or 193/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.
INT2	INT2	P07	IMF•EF8=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	INT3	P15	IMF•EF12=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.
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF•EF23=1	Falling edge	Pulses of less than 2/fc [s] are eliminated as noise. Pulses of 7/fc [s] or more are considered to be signals. In the SLOW or the SLEEP mode, pulses of less than 1/fs [s] are eliminated as noise. Pulses of 3.5/fs [s] or more are considered to be signals.

Note 1: In NORMAL1/2 or IDLE1/2 mode, if a signal with no noise is input on an external interrupt pin, it takes a maximum of "signal establishment time + 6/fs[s]" from the input signal's edge to set the interrupt latch.

Note 2: When INT0EN = "0", IL4 is not set even if a falling edge is detected on the INT0 pin input.

Note 3: When a pin with more than one function is used as an output and a change occurs in data or input/output status, an interrupt request signal is generated in a pseudo manner. In this case, it is necessary to perform appropriate processing such as disabling the interrupt enable flag.

External Interrupt Control Register

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

INT1NC	Noise reject time select	0: Pulses of less than 63/fc [s] are eliminated as noise 1: Pulses of less than 15/fc [s] are eliminated as noise	R/W
INT0EN	P00/$\overline{\text{INT0}}$ pin configuration	0: P00 input/output port 1: $\overline{\text{INT0}}$ pin (Port P00 should be set to an input mode)	R/W
INT3 ES	INT3 edge select	0: Rising edge 1: Falling edge	R/W
INT2 ES	INT2 edge select	0: Rising edge 1: Falling edge	R/W
INT1 ES	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).

Note 3: The maximum time from modifying INT1NC until a noise reject time is changed is 2^6/fc.

4. Special Function Register (SFR)

The TMP86FS49UG 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 0000H to 003FH, DBR is mapped on address 0F80H to 0FFFH.

This chapter shows the arrangement of the special function register (SFR) and data buffer register (DBR) for TMP86FS49UG.

4.1 SFR

Address	Read	Write
0000H	P0DR	
0001H	P1DR	
0002H	P2DR	
0003H	P3DR	
0004H	P4DR	
0005H	P5DR	
0006H	P6DR	
0007H	P7DR	
0008H	P0OUTCR	
0009H	P1CR	
000AH	P4OUTCR	
000BH	P0PRD	-
000CH	P2PRD	-
000DH	P3PRD	-
000EH	P4PRD	-
000FH	P5PRD	-
0010H	TC1DRAL	
0011H	TC1DRAH	
0012H	TC1DRBL	
0013H	TC1DRBH	
0014H	TTREG3	
0015H	TTREG4	
0016H	TTREG5	
0017H	TTREG6	
0018H	PWREG3	
0019H	PWREG4	
001AH	PWREG5	
001BH	PWREG6	
001CH	ADCCR1	
001DH	ADCCR2	
001EH	ADCDR2	-
001FH	ADCDR1	-
0020H	SIO1CR	
0021H	SIO1SR	-
0022H	SIO1RDB	SIO1TDB
0023H	TC2CR	
0024H	TC2DRL	
0025H	TC2DRH	

Address	Read	Write
0026H	TC1CR	
0027H	TC3CR	
0028H	TC4CR	
0029H	TC5CR	
002AH	TC6CR	
002BH	SIO2RDB	SIO2TDB
002CH	EIRE	
002DH	EIRD	
002EH	ILE	
002FH	ILD	
0030H	Reserved	
0031H	SIO2CR	
0032H	SIO2SR	-
0033H	Reserved	
0034H	-	WDTCR1
0035H	-	WDTCR2
0036H	TBTCR	
0037H	EINTCR	
0038H	SYSCR1	
0039H	SYSCR2	
003AH	EIRL	
003BH	EIRH	
003CH	ILL	
003DH	ILH	
003EH	Reserved	
003FH	PSW	

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.).

4.2 DBR

Address	Read	Write
0F80H	Reserved	
0F81H	Reserved	
0F82H	Reserved	
0F83H	Reserved	
0F84H	Reserved	
0F85H	Reserved	
0F86H	Reserved	
0F87H	Reserved	
0F88H	Reserved	
0F89H	Reserved	
0F8AH	Reserved	
0F8BH	Reserved	
0F8CH	Reserved	
0F8DH	Reserved	
0F8EH	Reserved	
0F8FH	Reserved	
0F90H	SBISRA	SBICRA
0F91H	SBIDBR	
0F92H	-	I2CAR
0F93H	SBISRB	SBICRB
0F94H	-	
0F95H	UART1SR	UART1CR1
0F96H	-	UART1CR2
0F97H	RDBUF1	TDBUF1
0F98H	UART2SR	UART2CR1
0F99H	-	UART2CR2
0F9AH	RDBUF2	TDBUF2
0F9BH	P6CR1	
0F9CH	P6CR2	
0F9DH	P7CR1	
0F9EH	P7CR2	
0F9FH	-	STOPCR

Address	Read	Write
0FA0H	Reserved	
0FA1H	Reserved	
0FA2H	Reserved	
0FA3H	Reserved	
0FA4H	Reserved	
0FA5H	Reserved	
0FA6H	Reserved	
0FA7H	Reserved	
0FA8H	Reserved	
0FA9H	Reserved	
0FAAH	Reserved	
0FABH	Reserved	
0FACH	Reserved	
0FADH	Reserved	
0FAEH	Reserved	
0FAFH	Reserved	
0FB0H	Reserved	
0FB1H	Reserved	
0FB2H	Reserved	
0FB3H	Reserved	
0FB4H	Reserved	
0FB5H	Reserved	
0FB6H	Reserved	
0FB7H	Reserved	
0FB8H	Reserved	
0FB9H	Reserved	
0FBAH	Reserved	
0FBBH	Reserved	
0FBCH	Reserved	
0FBDH	Reserved	
0FBEH	Reserved	
0FBFH	Reserved	
0FC0H	Reserved	
0FC1H	Reserved	
0FC2H	Reserved	
0FC3H	Reserved	
0FC4H	Reserved	
0FC5H	Reserved	
0FC6H	Reserved	
0FC7H	Reserved	
0FC8H	Reserved	
0FC9H	Reserved	
0FCAH	Reserved	
0FCBH	Reserved	
0FCCH	Reserved	
0FCDH	Reserved	
0FCEH	Reserved	
0FCFH	Reserved	
0FD0H	Reserved	
0FD1H	Reserved	

Address	Read	Write
0FD2H	Reserved	
0FD3H	Reserved	
0FD4H	Reserved	
0FD5H	Reserved	
0FD6H	Reserved	
0FD7H	Reserved	
0FD8H	Reserved	
0FD9H	Reserved	
0FDAH	Reserved	
0FDBH	Reserved	
0FDCH	Reserved	
0FDDH	Reserved	
0FDEH	Reserved	
0FDFH	Reserved	
0FE0H	Reserved	
0FE1H	Reserved	
0FE2H	Reserved	
0FE3H	Reserved	
0FE4H	Reserved	
0FE5H	Reserved	
0FE6H	Reserved	
0FE7H	Reserved	
0FE8H	Reserved	
0FE9H	Reserved	
0FEAH	Reserved	
0FEBH	Reserved	
0FECH	Reserved	
0FEDH	Reserved	
0FEEH	Reserved	
0FEFH	Reserved	
0FF0H	Reserved	
0FF1H	Reserved	
0FF2H	Reserved	
0FF3H	Reserved	
0FF4H	Reserved	
0FF5H	Reserved	
0FF6H	Reserved	
0FF7H	Reserved	
0FF8H	Reserved	
0FF9H	Reserved	
0FFAH	Reserved	
0FFBH	Reserved	
0FFCH	Reserved	
0FFDH	Reserved	
0FFEH	Reserved	
0FFFH	FLSCR	

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.).

5. I/O Ports

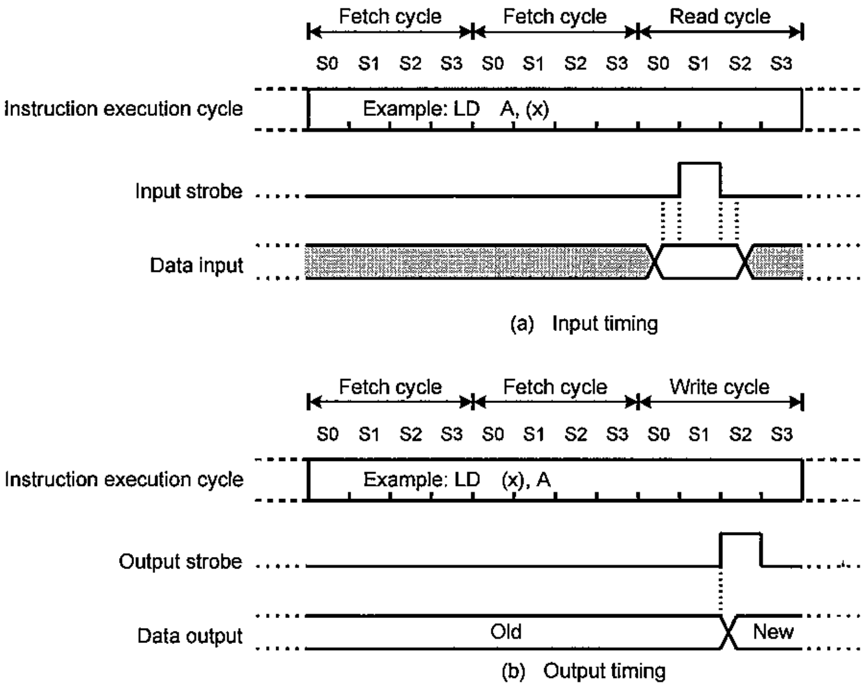
The TMP86FS49UG has 8 parallel input/output ports (56 pins) as follows.

	Primary Function	Secondary Functions
Port P0	8-bit I/O port	External interrupt, serial interface input/output and UART input/output.
Port P1	8-bit I/O port	External interrupt and timer counter input/output.
Port P2	3-bit I/O port	Low-frequency resonator connections, external interrupt input, STOP mode release signal input.
Port P3	8-bit I/O port	
Port P4	8-bit I/O port	Serial interface input/output and UART input/output.
Port P5	4-bit I/O port	Serial bus interface input/output.
Port P6	8-bit I/O port	Analog input and STOP mode release signal input.
Port P7	8-bit I/O port	Analog input.

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 timer before processing. Figure 5-1 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.



Note: The positions of the read and write cycles may vary, depending on the instruction.

Figure 5-1 Input/Output Timing (Example)

5.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input/output port.

Port P0 is also used as an external interrupt input, a serial interface input/output and an UART input/output.

When used as an input port, an external interrupt input, a serial interface input/output and an UART input/output, the corresponding output latch (P0DR) should be set to "1".

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

It can be selected whether output circuit of P0 port is a C-MOS output or a sink open drain individually, by setting P0OUTCR. When a corresponding bit of P0OUTCR is "0". the output circuit is selected to a sink open drain and when a corresponding bit of P0OUTCR is "1", the output circuit is selected to a C-MOS output.

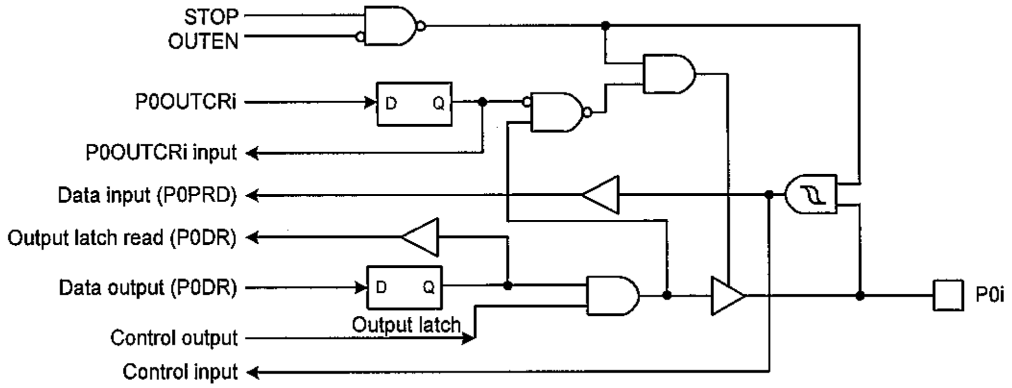
When used as an input port, an external interrupt input, a serial interface input and an UART input, the corresponding output control (P0OUTCR) should be set to "0" after P0DR is set to "1".

P0 port output latch (P0DR) and P0 port terminal input (P0PRD) are located on their respective address.

When read the output latch data, the P0DR should be read. When read the terminal input data, the P0PRD register should be read.

Table 5-1 Register Programming for Multi-function Ports (P07 to P00)

Function	Programmed Value	
	P0DR	P0OUTCR
Port input, external input, serial interface input or UART input	"1"	"0"
Port "0" output	"0"	Programming for each applica-tions
Port "1" output, serial interface output or UART output	"1"	



Note: i = 7 to 0

Figure 5-2 Port 0 and P0OUTCR

P0DR
(0000H)
R/W

7	6	5	4	3	2	1	0
P07 INT2	P06 $\overline{\text{SCK1}}$	P05 SO1	P04 SI1	P03 INT1	P02 TXD1	P01 RXD1	P00 $\overline{\text{INT0}}$

(Initial value: 1111 1111)

P0OUTCR
(0008H)

(Initial value: 0000 0000)

P0OUTCR	Port P0 output circuit control (Set for each bit individually)	0: Sink open-drain output 1: C-MOS output	R/W

P0PRD
(000BH)
Read only

P07	P06	P05	P04	P03	P02	P01	P00

5.2 Port P1 (P17 to P10)

Port P1 is an 8-bit input/output port which can be configured as an input or output in one-bit unit.

Port P1 is also used as a timer/counter input/output, an external interrupt input and a divider output.

Input/output mode is specified by the P1 control register (P1CR).

During reset, the P1CR is initialized to "0" and port P1 becomes an input mode. And the P1DR is initialized to "0".

When used as an input port, a timer/counter input and an external interrupt input, the corresponding bit of P1CR should be set to "0".

When used as an output port, the corresponding bit of P1CR should be set to "1".

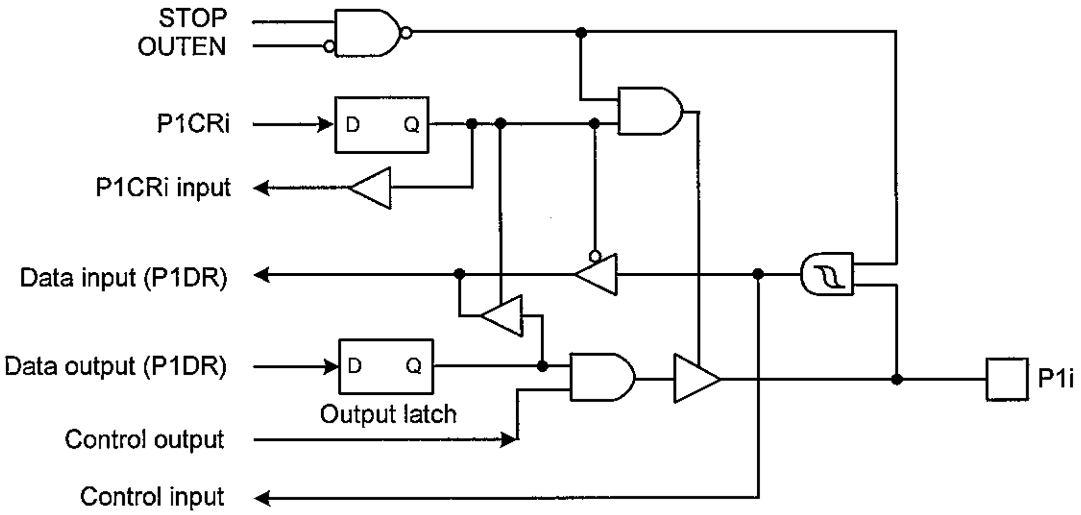
When used as a timer/counter output and a divider output, P1DR is set to "1" beforehand and the corresponding bit of P1CR should be set to "1".

When P1CR is "1", the content of the corresponding output latch is read by reading P1DR.

Table 5-2 Register Programming for Multi-function Ports

Function	Programmed Value	
	P1DR	P1CR
Port input, timer/counter input or external interrupt input	*	"0"
Port "0" output	"0"	"1"
Port "1" output, a timer output or a divider output	"1"	"1"

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.



Note: i = 7 to 0

Figure 5-3 Port 1 and P1CR

Note: The port set to an input mode reads the terminal input data. Therefore, when the input and output modes are used together, the content of the output latch which is specified as input mode might be changed by executing a bit Manipulation instruction.

	7	6	5	4	3	2	1	0	
P1DR (0001H) R/W	P17 TC6 $\overline{\text{PWM6}}$ $\overline{\text{PDO6}}$ $\overline{\text{PPG6}}$	P16 TC5 $\overline{\text{PWM5}}$ $\overline{\text{PDO5}}$	P15 TC2 INT3	P14 TC4 $\overline{\text{PWM4}}$ $\overline{\text{PDO4}}$ $\overline{\text{PPG4}}$	P13 TC3 $\overline{\text{PWM3}}$ $\overline{\text{PDO3}}$	P12 $\overline{\text{PPG}}$	P11 DVO	P10 TC1	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P1CR (0009H)									(Initial value: 0000 0000)

P1CR	I/O control for port P1 (Specified for each bit)	0: Input mode 1: Output mode	R/W

5.3 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.

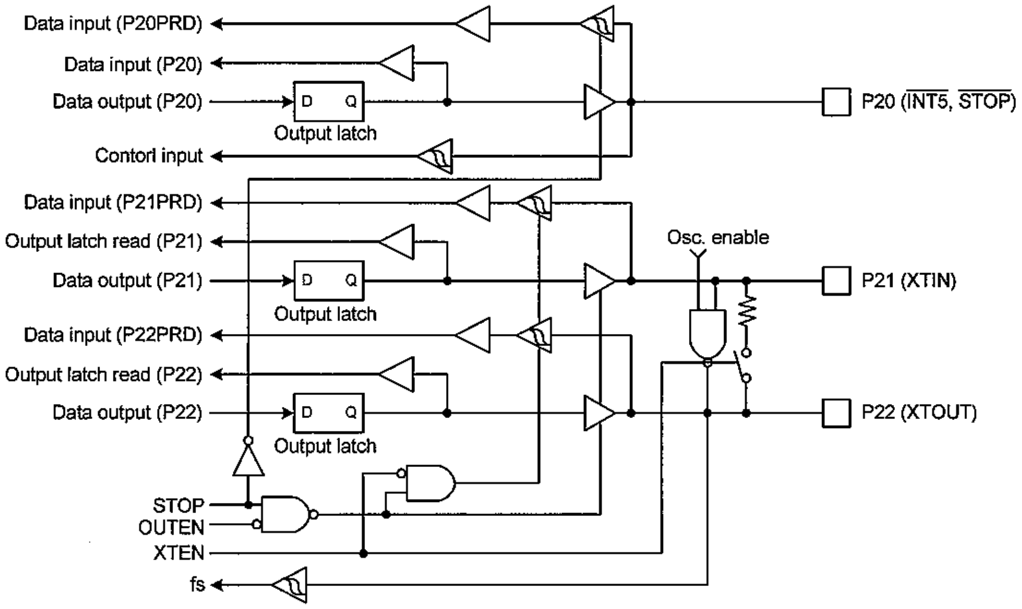


Figure 5-4 Port 2

	7	6	5	4	3	2	1	0	
P2DR (0002H) R/W						P22 XTOUT	P21 XTIN	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$	(Initial value: **** *111)

P2PRD (000CH) Read only						P22	P21	P20

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

5.4 Port P3 (P37 to P30) (Large Current Port)

Port P3 is an 8-bit input/output port.

When used as an input port, the corresponding output latch (P3DR) should be set to "1".

During reset, the P3DR is initialized to "1".

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. When read the terminal input data, the P3PRD register should be read.

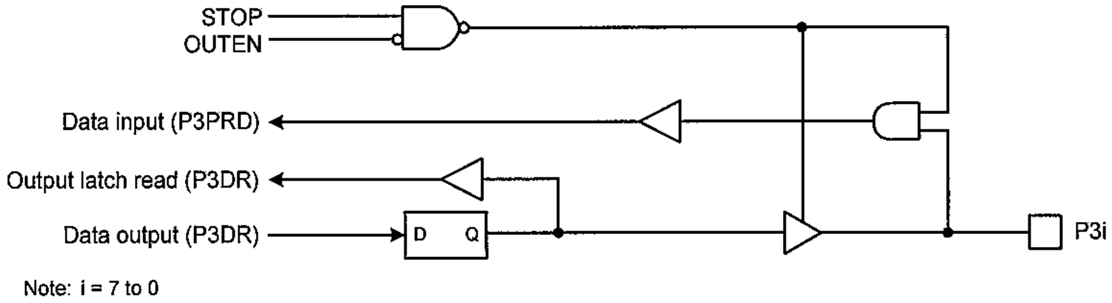


Figure 5-5 Port 3

P3DR (0003H) R/W	7	6	5	4	3	2	1	0	
	P37	P36	P35	P34	P33	P32	P31	P30	(Initial value: 1111 1111)

P3PRD (000DH) Read only								
	P37	P36	P35	P34	P33	P32	P31	P30

5.5 Port P4 (P47 to P40)

Port P4 is an 8-bit input/output port.

Port P4 is also used as a serial interface input/output and an UART input/output.

When used as an input port, a serial interface input/output and an UART input/output, the corresponding output latch (P4DR) should be set to "1".

During reset, the P4DR is initialized to "1", and the P4OUTCR is initialized to "0".

It can be selected whether output circuit of P4 port is a C-MOS output or a sink open drain individually, by setting P4OUTCR. When a corresponding bit of P4OUTCR is "0". the output circuit is selected to a sink open drain and when a corresponding bit of P4OUTCR is "1", the output circuit is selected to a C-MOS output.

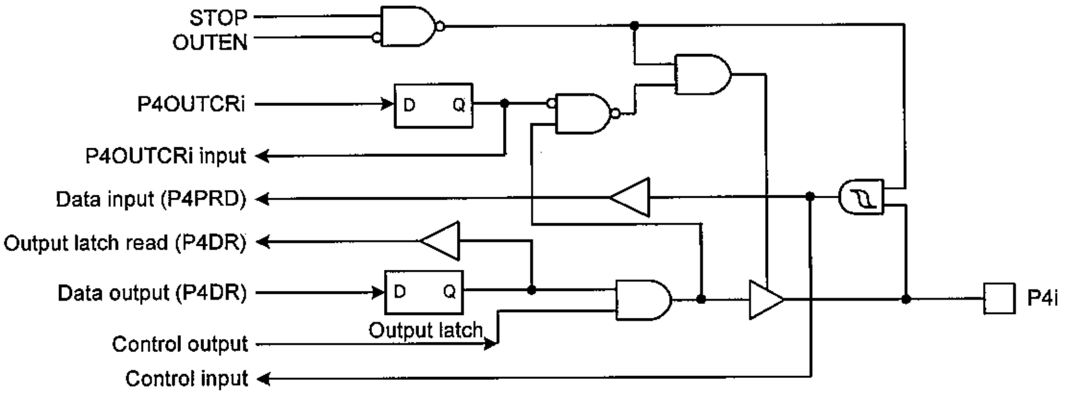
When used as an input port, a serial interface input and an UART input, the corresponding output control (P4OUTCR) should be set to "0" after P4DR is set to "1".

P4 port output latch (P4DR) and P4 port terminal input (P4PRD) are located on their respective address.

When read the output latch data, the P4DR should be read. When read the terminal input data, the P4PRD register should be read.

Table 5-3 Register Programming for Multi-function Ports (P47 to P40)

Function	Programmed Value	
	P4DR	P4OUTCR
Port input UART input or serial interface input	"1"	"0"
Port "0" output	"0"	Programming for each applications
Port "1" output UART output or serial interface output	"1"	



Note: i = 7 to 0

Figure 5-6 Port 4

	7	6	5	4	3	2	1	0	
P4DR (0004H) R/W	P47	P46 $\overline{\text{SCK2}}$	P45 SO2	P44 SI2	P43	P42 TXD2	P41 RXD2	P40	(Initial value: 1111 1111)

P4OUTCR (000AH)									(Initial value: 0000 0000)

P4OUTCR	Port P4 output circuit control (Set for each bit individually)	0: Sink open-drain output 1: C-MOS output	R/W

P4PRD (000EH) Read only	P47	P46	P45	P44	P43	P42	P41	P40

5.6 Port P5 (P574 to P50) (Large Current Port)

Port P5 is an 5-bit input/output port.

Port P5 is also used as an I^2C Bus input/output.

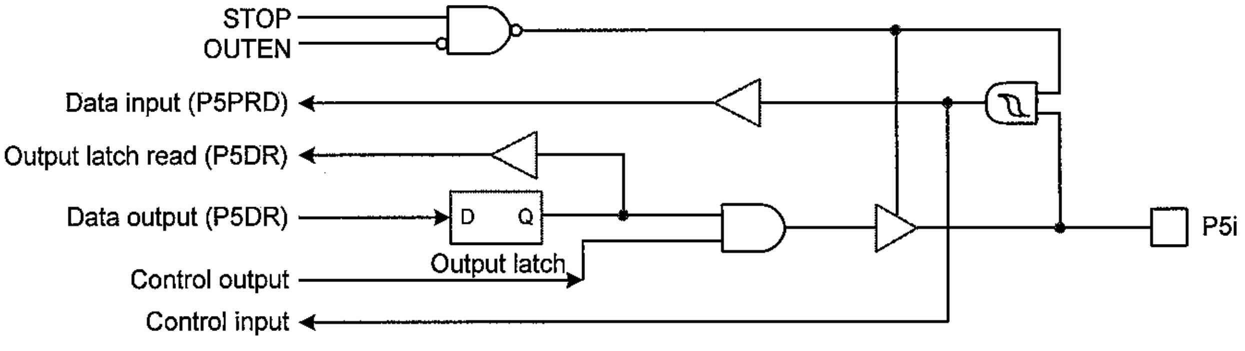
When used as an input port and I^2C Bus input/output, the corresponding output latch (P5DR) should be set to "1".

During reset, the P5DR is initialized to "1".

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. When read the terminal input data, the P5PRD register should be read.

If a read instruction is executed for port P5, read data of bit 7 to 5 are unstable.



Note: i = 4 to 0

Figure 5-7 Port 5

P5DR (0005H) R/W	7	6	5	4	3	2	1	0	
				P54	P53	P52	P51 SDA	P50 SCL	(Initial value: ***1 1111)

P5PRD (000FH) Read only								
				P54	P53	P52	P51	P50

5.7 Port P6 (P67 to P60)

Port P6 is an 8-bit input/output port which can be configured as an input or output in one-bit unit.

Port P6 is also used as an analog input and key-on wakeup input.

Input/output mode is specified by the P6 control register (P6CR1) and P6 input control register (P6CR2).

During reset, the P6CR1 is initialized to "0" the P6CR2 is initialized to "1" and port P6 becomes an input mode. And the P6DR is initialized to "0".

When used as an output port, the corresponding bit of P6CR1 should be set to "1".

When used as an input port and a key-on wakeup input , the corresponding bit of P6CR1 should be set to "0" and then, the corresponding bit of P6CR2 should be set to "1".

When used as an analog input, the corresponding bit of P6CR1 should be set to "0" and then, the corresponding bit of P6CR2 should be set to "0".

When P6CR1 is "1", the content of the corresponding output latch is read by reading P6DR.

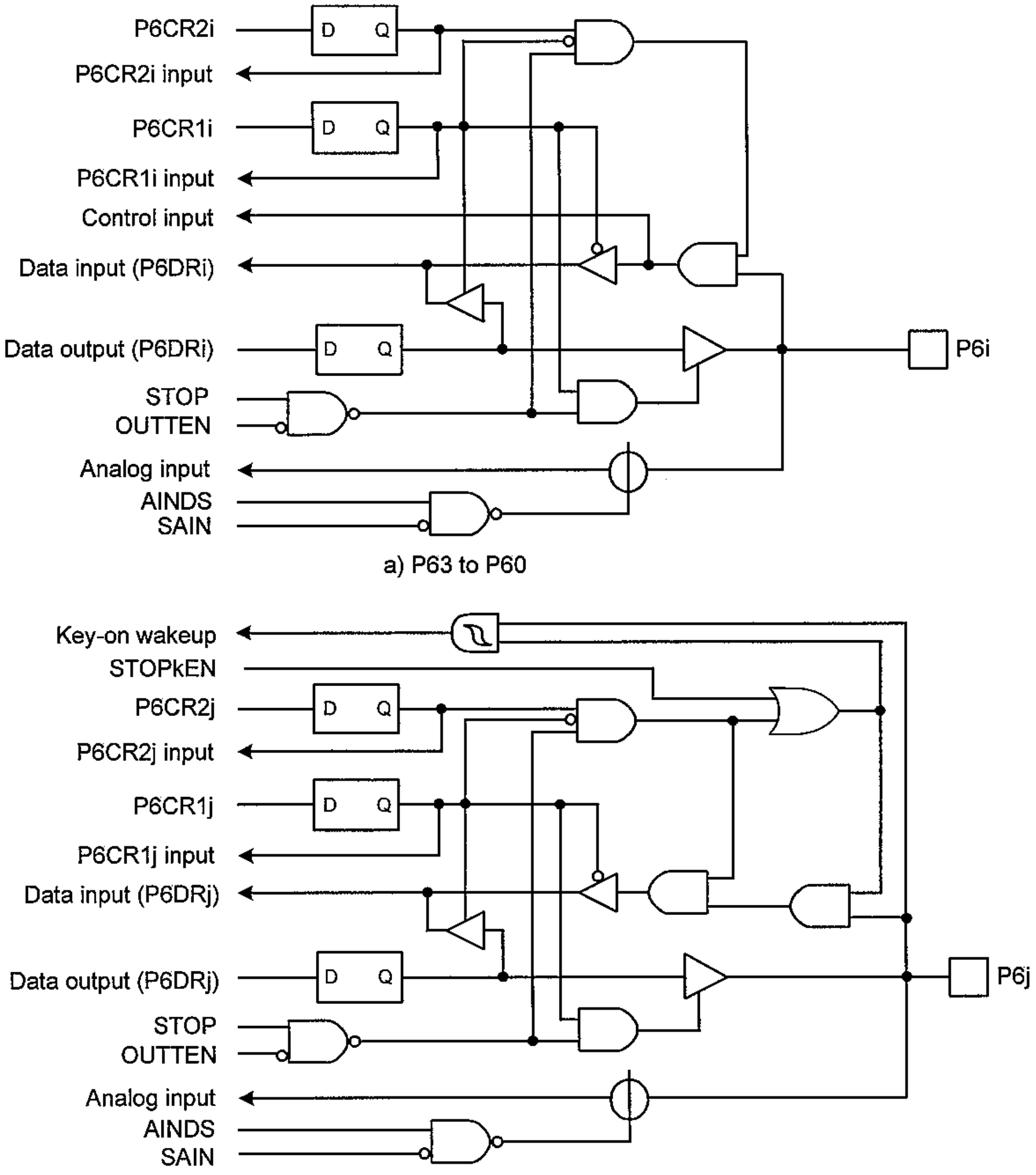
Table 5-4 Register Programming for Multi-function Ports

Function	Programmed Value		
	P6DR	P6CR1	P6CR2
Port input or key-on wakeup input	*	"0"	"1"
Analog input	*	"0"	"0"
Port "0" output	"0"	"1"	*
Port "1" output	"1"	"1"	*

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-5 Values Read from P6DR and Register Programming

Conditions		Values Read from P6DR
P6CR1	P6CR2	
"0"	"0"	"0"
"0"	"1"	Terminal input data
"1"	"0"	Output latch contents
	"1"	



a) P63 to P60

b) P67 to P64

Note 1: i = 3 to 0, j = 7 to 4, k = 3 to 0

Note 2: STOP is bit7 in SYSCR1.

Note 3: SAIN is AD input select signal.

Note 4: STOPkEN is input select signal in a key-on wakeup.

Figure 5-8 Port 6, P6CR1 and P6CR2

P6DR (0006H) R/W	7	6	5	4	3	2	1	0	
	P67 AIN07 STOP3	P66 AIN06 STOP2	P65 AIN05 STOP1	P64 AIN04 STOP0	P63 AIN03	P62 AIN02	P61 AIN01	P60 AIN00	(Initial value: 0000 0000)

P6CR1 (0F9BH)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000)

P6CR1	I/O control for port P6 (Specified for each bit)	0: Input mode 1: Output mode	R/W

P6CR2 (0F9CH)	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

P6CR2	P6 port input control (Specified for each bit)	0: Analog input 1: Port input, external interrupt input or key-on wakeup input	R/W

Note 1: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

Note 2: When used as an analog inport, be sure to clear the corresponding bit of P6CR2 to disable the port input.

Note 3: Do not set the output mode (P6CR1 = "1") for the pin used as an analog input pin.

Note 4: 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.

5.8 Port P7 (P77 to P70)

Port P7 is an 8-bit input/output port which can be configured as an input or output in one-bit unit.

Port P7 is also used as an analog input.

Input/output mode is specified by the P7 control register (P7CR1) and P7 input control register (P7CR2).

During reset, the P7CR1 is initialized to "0" the P7CR2 is initialized to "1" and port P7 becomes an input mode. And the P7DR is initialized to "0".

When used as an output port, the corresponding bit of P7CR1 should be set to "1".

When used as an input port, the corresponding bit of P7CR1 should be set to "0" and then, the corresponding bit of P7CR2 should be set to "1".

When used as an analog input, the corresponding bit of P7CR1 should be set to "0" and then, the corresponding bit of P7CR2 should be set to "0".

When P7CR1 is "1", the content of the corresponding output latch is read by reading P7DR.

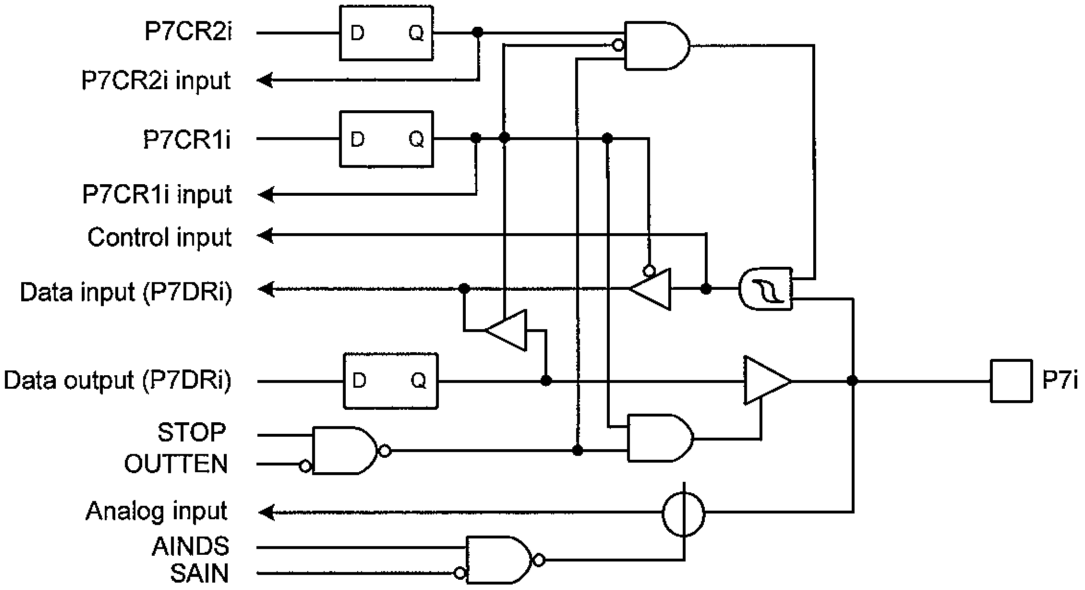
Table 5-6 Register Programming for Multi-function Ports

Function	Programmed Value		
	P7DR	P7CR1	P7CR2
Port input external interrupt input or key-on wakeup input	*	"0"	"1"
Analog input	*	"0"	"0"
Port "0" output	"0"	"1"	*
Port "1" output	"1"	"1"	*

Note: Asterisk (*) indicates "1" or "0" either of which can be selected.

Table 5-7 Values Read from P7DR and Register Programming

Conditions		Values Read from P7DR
P7CR1	P7CR2	
"0"	"0"	"0"
"0"	"1"	Terminal input data
"1"	"0"	Output latch contents
	"1"	



Note 1: i = 7 to 0

Note 2: STOP is bit7 in SYSCR1.

Note 3: SAIN is AD input select signal.

Figure 5-9 Port 7, P7CR1 and P7CR2

P7DR (0007H) R/W

	7	6	5	4	3	2	1	0	
P7DR (0007H) R/W	P77 AIN17	P76 AIN16	P75 AIN15	P74 AIN14	P73 AIN13	P72 AIN12	P71 AIN11	P70 AIN10	(Initial value: 0000 0000)

	7	6	5	4	3	2	1	0	
P7CR1 (0F9DH)									(Initial value: 0000 0000)

P7CR1	I/O control for port P7 (Specified for each bit)	0: Input mode 1: Output mode	R/W

	7	6	5	4	3	2	1	0	
P7CR2 (0F9EH)									(Initial value: 1111 1111)

P7CR2	P7 port input control (Specified for each bit)	0: Analog input 1: Port input, external interrupt input or key-on wakeup input	R/W

Note 1: The port placed in input mode reads the pin input state. Therefore, when the input and output modes are used together, the output latch contents for the port in input mode might be changed by executing a bit manipulation instruction.

Note 2: When used as an analog inport, be sure to clear the corresponding bit of P7CR2 to disable the port input.

Note 3: Do not set the output mode (P7CR1 = "1") for the pin used as an analog input pin.

Note 4: 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.

6. Time Base Timer (TBT)

6.1 Time Base Timer

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 6-2).

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.

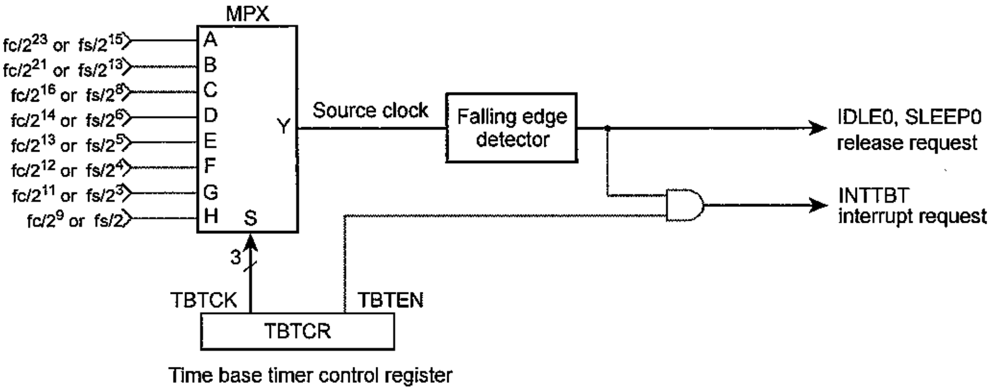


Figure 6-1 Time Base Timer Configuration

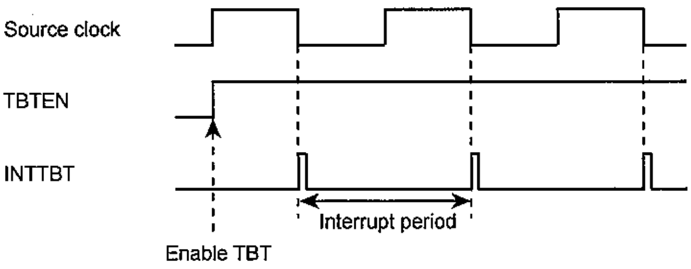


Figure 6-2 Time Base Timer Interrupt

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) . 7
EI
```

Time Base Timer Control Register

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

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

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

Table 6-1 Time Base Timer Interrupt Frequency (Example : fc = 16.0 MHz, fs = 32.768 kHz)

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

6.2 Divider Output (DVO)

Approximately 50% duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from pin P30 ($\overline{\text{DVO}}$). Divider output is from pin P30 ($\overline{\text{DVO}}$). The P30 output latch (P3DR) 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.

Divider Output Control Register

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

DVOEN	Divider output enable/disable	0: Disable 1: Enable				R/W
DVOCK	Divider output ($\overline{\text{DVO}}$) frequency selection [Hz]		NORMAL1/2, IDLE1/2 Mode		SLOW, SLEEP Mode	R/W
			DV7CK = 0	DV7CK = 1		
		00	$fc/2^{13}$	$fs/2^{5}$	$fs/2^{5}$	
		01	$fc/2^{12}$	$fs/2^{4}$	$fs/2^{4}$	
		10	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	
		11	$fc/2^{10}$	$fs/2^{2}$	$fs/2^{2}$	

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

```
LD      (TBTCR) , 00000000B        ; DVOCK ← "00"
LD      (TBTCR) , 10000000B        ; DVOEN ← "1"
```

Table 6-2 Divider Output Frequency (Example: at fc = 16.0 MHz, fs = 32.768 kHz)

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

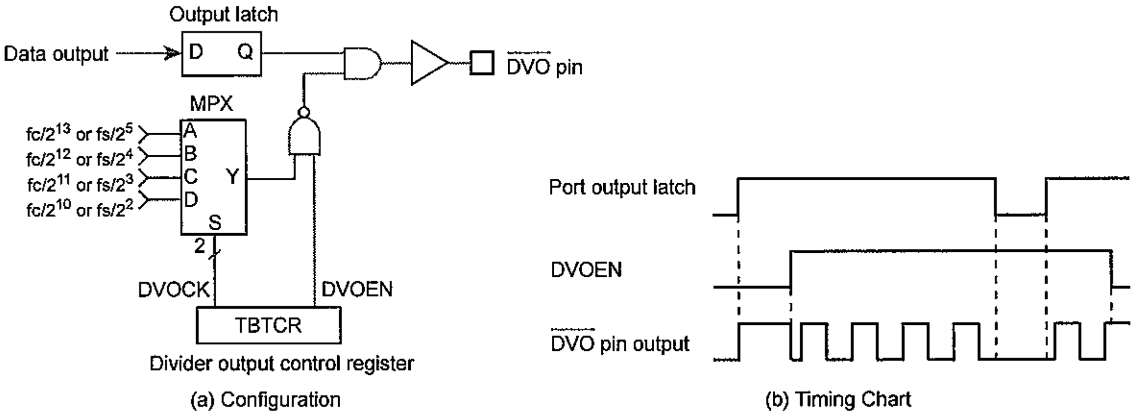


Figure 6-3 Divider Output

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

7.1 Watchdog Timer Configuration

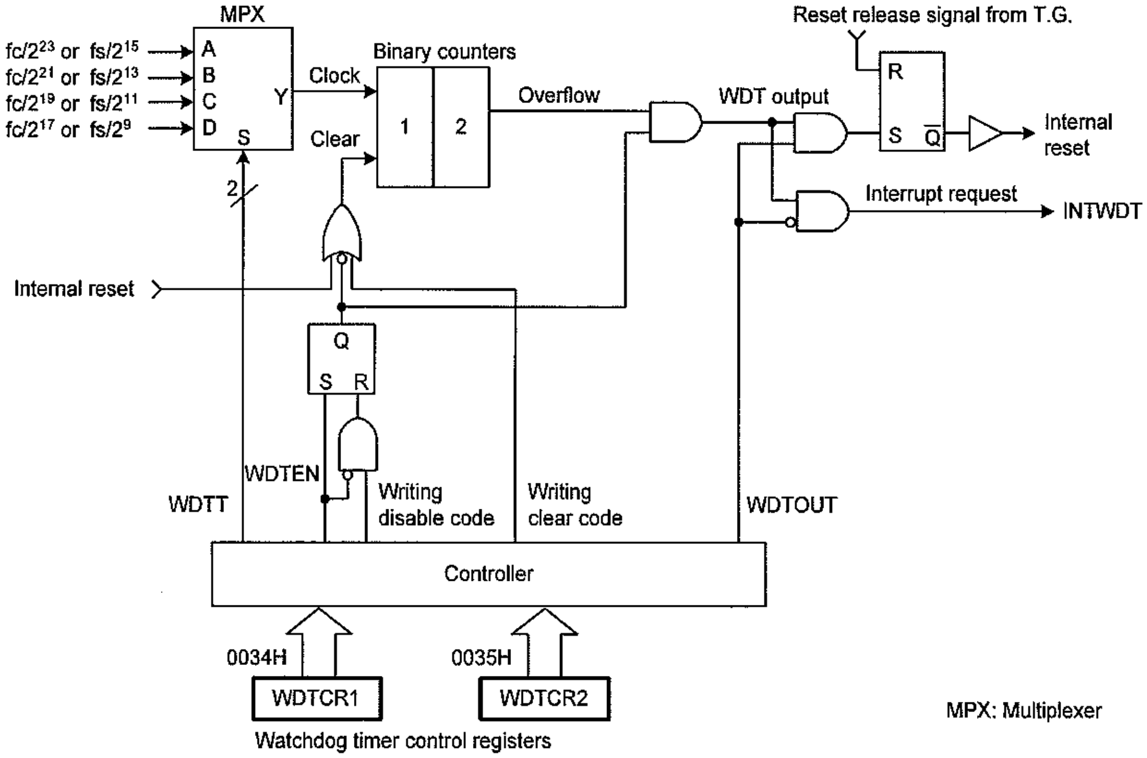


Figure 7-1 Watchdog Timer Configuration

7.2 Watchdog Timer Control

Figure 7-1 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

7.2.1 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 WDTOUT = 1 a reset is generated, reset the internal hardware. When WDTOUT = 0, a watchdog timer interrupt (INTWDT) is generated.

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

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

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

```
                      LD      (WDTCR2), 4EH         ; Clears the binary counters
                      LD      (WDTCR1), 00001101B   ; WDTT ← 10, WDTOUT ← 1
                      LD      (WDTCR2), 4EH         ;Clears the binary counters (always clear immediately before and after
                      :                             changing WDTT)
Within 3/4 of WDT     :
detection time        :
                      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 enabled/disable	0: Disable (It is necessary to write the disable code to WDTCR2) 1: Enable				Write only
WDTT	Watchdog timer detection time [s]		NORMAL1/2 mode		SLOW mode	
			DV7CK = 0	DV7CK = 1		
		00	2^{25}/fc	2^{17}/fs	2^{17}/fs	
		01	2^{23}/fc	2^{15}/fs	2^{15}/fs	
		10	2^{21}/fc	2^{13}/fs	2^{13}/fs	
		11	2^{19}/fc	2^{11}/fs	2^{11}/fs	
WDTOUT	Watchdog timer output select	0: Interrupt request 1: Interrupt reset generate				

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

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

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

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

Note 5: To disable the watchdog timer, always write "4EH" (Clear code) to WDTCR2 for clearing the binary counter before writing "0" to WDTEN, and then write "B1H" (Disable code) to WDTCR2.
Also, immediately before these procedure, disable the interrupt mater flag (IMF) by DI instruction.

Watchdog Timer Register 2

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

WDTCR2	Watchdog timer control code write register	4EH: Watchdog timer binary counter clear (Clear code) B1H: Watchdog timer disable (Disable code) D2H: Enable assigning address trap area Others: Invalid	Write only

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

Note 2: *: Don't care

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

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

7.2.2 Watchdog timer enable

The watchdog timer is enabled by setting WDTEN (Bit3 in WDTCR1) to "1". WDTEN is initialized to "1" during reset, so the watchdog timer operates immediately after reset is released.

7.2.3 Watchdog timer disable

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

Example :Disables watchdog timer

```
        DI                                  ; IMF ← 0
        LD      (WDTCR2), 04EH              ; Clear the binary coutner
        LD      (WDTCR1), 0B101H            ; WDTEN ← 0, WDTCR2 ← Disable code
        EI                                  ; IMF ← 1
```

Table 7-1 Watchdog Timer Detection Time (Example: fc = 16.0 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

7.3 Watchdog Timer Interrupt (INTWDT)

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

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

Example :Watchdog timer interrupt setting up

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

7.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, to reset the internal hardware. The reset time is about 8/fc to 24/fc [s] (0.5 to 1.5 µs at fc = 16.0 MHz).

Note: The high-frequency clock oscillator also turns on when a watchdog timer reset is generated in SLOW mode. The reset output time is 8/fc to 24/fc [s]. Therefore, 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. Thus, the reset time must be considered an approximated value.

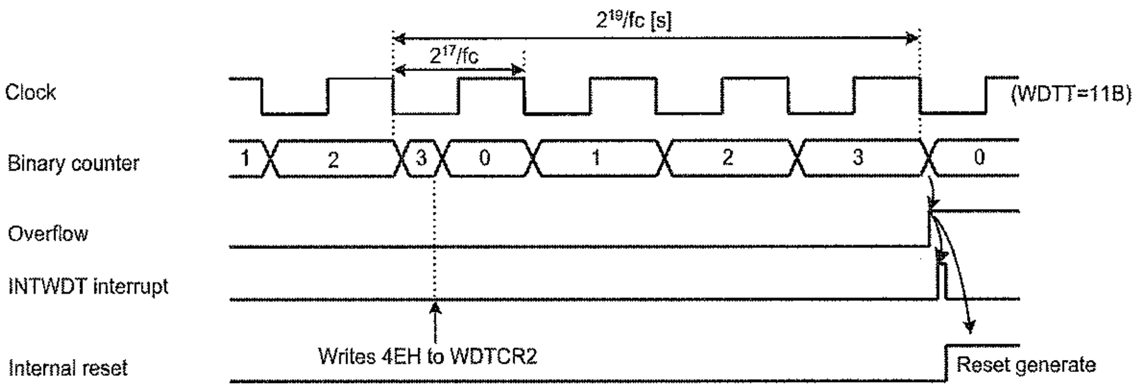


Figure 7-2 Watchdog Timer Interrupt/Reset

7.5 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 as follows.

Watchdog Timer Control Register 1

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

ATAS	Selection of address trap in internal RAM	0: No address trap 1:Address trap (after setting ATAS to "1", it is neceaary to write the control code D2H to WDTCR2)	Write only
ATOUT	Selection of opertion at address trap	0: Interrupt 1: Internal reset generate	

Watchdog Timer Control Register 2

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

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

7.5.1 Selection of address trap in internal RAM (ATAS)

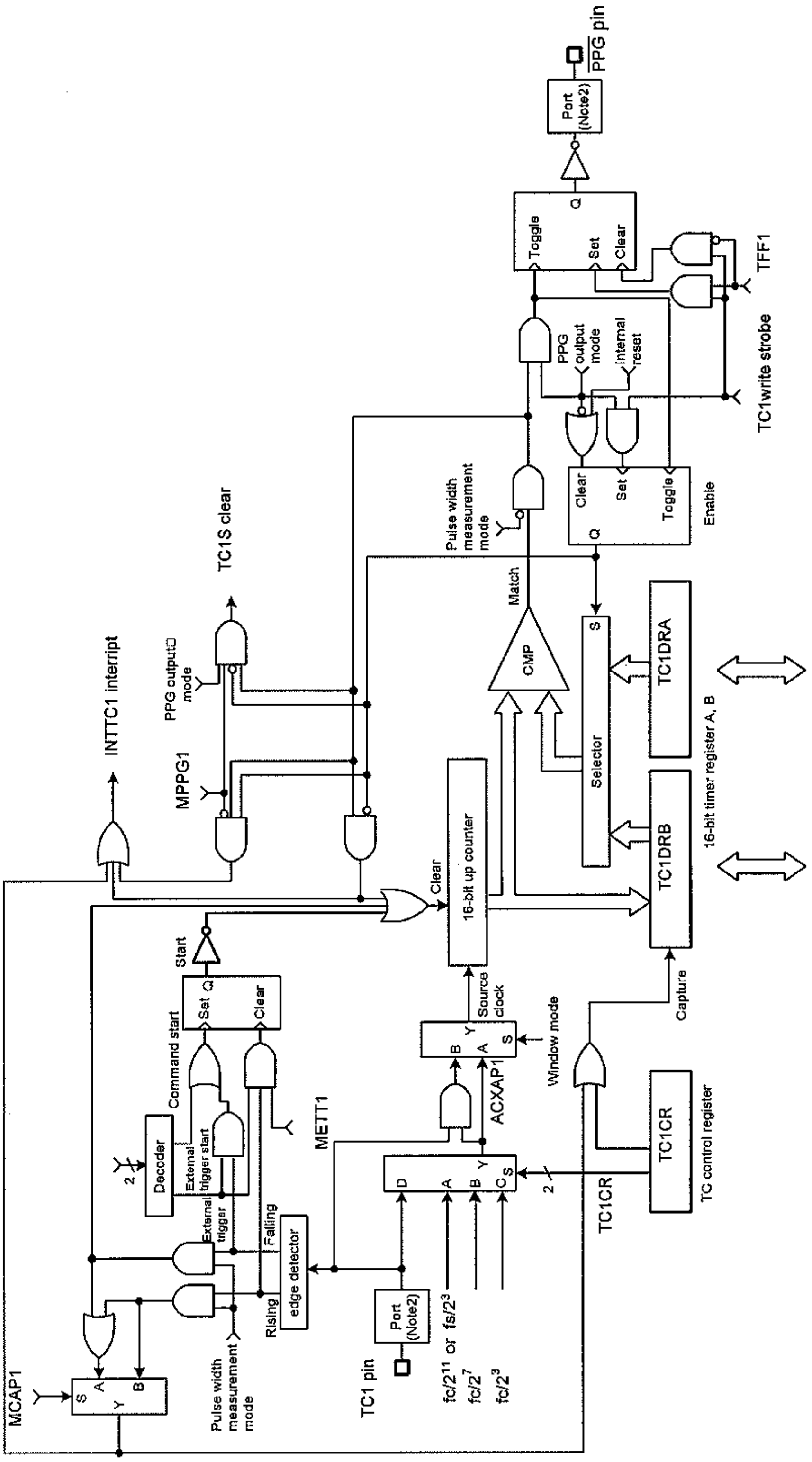
Using WDTCR1<ATAS>, address trap or no address trap can be selected for the internal RAM area. To execute an instruction in the internal RAM area, set "0" in WDTCR1<ATAS>. Setting in WDTCR1<ATAS> becomes valid after control code D2H is written in WDTCR2. Executing an instruction in the SFR/DBR area generates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

7.5.2 Selection of operation at address trap (ATOUT)

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

8. 16-Bit Timer/Counter 1

8.1 Configuration



Note 1: MPX: Multiplexer
CMP: Comparator
Note 2: When control input/output is used, I/O port setting should de set correctly.
For details, reter to "2.2 I/O poets".

Figure 8-1 Timer/Counter 1 (TC1)

8.2 Control

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

Timer Registers

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TC1DRA (0011H, 0010H) R/W	TC1DRAH (0011H)								TC1DRAL (0010H)							
									(Initial Value : 1111 1111 1111 1111)							
TC1DRB (0013H, 0012H) Read (Write: PPG output mode only)	TC1DRBH (0013H)								TC1DRBL (0012H)							
									(Initial Value : 1111 1111 1111 1111)							

Note: TC1DRB should not be written except PPG mode.

TC1 Control Register

	7	6	5	4	3	2	1	0	
TC1CR (0026H) R/W	TFF1	ACAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		(Initial Value : 0000 0000)

<table>
<tr><td>TC1M</td><td>TC1 operating mode select</td><td colspan="8">00: Timer/external trigger timer/event counter mode
01: Window mode
10: Pulse width measurement mode
11: PPG (Programmable pulse generate) output mode</td><td rowspan="17">R/W</td></tr>
<tr><td rowspan="6">TC1CK</td><td rowspan="6">TC1 source clock select [Hz]</td><td rowspan="2"></td><td colspan="5">NORMAL1/2, IDLE1/2 mode</td><td colspan="2" rowspan="2">SLOW1/2, SLEEP1/2 mode</td></tr>
<tr><td>DV7CK = 0</td><td colspan="4">DV7CK = 1</td></tr>
<tr><td>00</td><td>fc/2¹¹</td><td colspan="4">fs/2³</td><td colspan="2">fs/2³</td></tr>
<tr><td>01</td><td>fc/2⁷</td><td colspan="4">fc/2⁷</td><td colspan="2">–</td></tr>
<tr><td>10</td><td>fc/2³</td><td colspan="4">fc/2³</td><td colspan="2">–</td></tr>
<tr><td>11</td><td colspan="7">External clock (TC1 pin input)</td></tr>
<tr><td rowspan="5">TC1S</td><td rowspan="5">TC1 start control</td><td colspan="2"></td><td>Timer</td><td>Ex-trigger</td><td>Event</td><td>Win-dow</td><td>Pulse</td><td>PPG</td></tr>
<tr><td colspan="2">00:Stop and counter clear</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td colspan="2">01:Command start</td><td>O</td><td>×</td><td>×</td><td>×</td><td>×</td><td>O</td></tr>
<tr><td colspan="2">10:External trigger start at the rising edge</td><td>×</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td colspan="2">11:External trigger start at the falling edge</td><td>×</td><td>O</td><td>O</td><td>O</td><td>O</td><td>O</td></tr>
<tr><td>ACAP1</td><td>Auto capture control</td><td colspan="4">0: Auto-capture disable</td><td colspan="4">1:Auto-capture enable</td></tr>
<tr><td>MCAP1</td><td>Pulse width measurement mode control</td><td colspan="4">0:Double edge capture</td><td colspan="4">1:Single edge capture</td></tr>
<tr><td>METT1</td><td>External trigger timer mode control</td><td colspan="4">0:Trigger start</td><td colspan="4">1:Trigger start and stop</td></tr>
<tr><td>MPPG1</td><td>PPG output control</td><td colspan="4">0:Continuous pulse generation</td><td colspan="4">1:One-shot</td></tr>
<tr><td>TFF1</td><td>Time F/F1 control</td><td colspan="4">0:Clear</td><td colspan="4">1:Set</td></tr>
</table>

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

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

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

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

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

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

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

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

8.3 Function

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

8.3.1 Timer mode

In this mode, counting up is performed using the internal clock. The contents of TC1DRA are compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to “0”. Counting up resumes after the counter is cleared. The current contents of up counter can be transferred to TC1DRB by setting TC1CR<ACAP1> to “1” (Auto capture function).

Table 8-1 Source Clock (internal clock) for Timer/Counter 1
(Example: at fc = 16 MHz, fs = 32.768kHz)

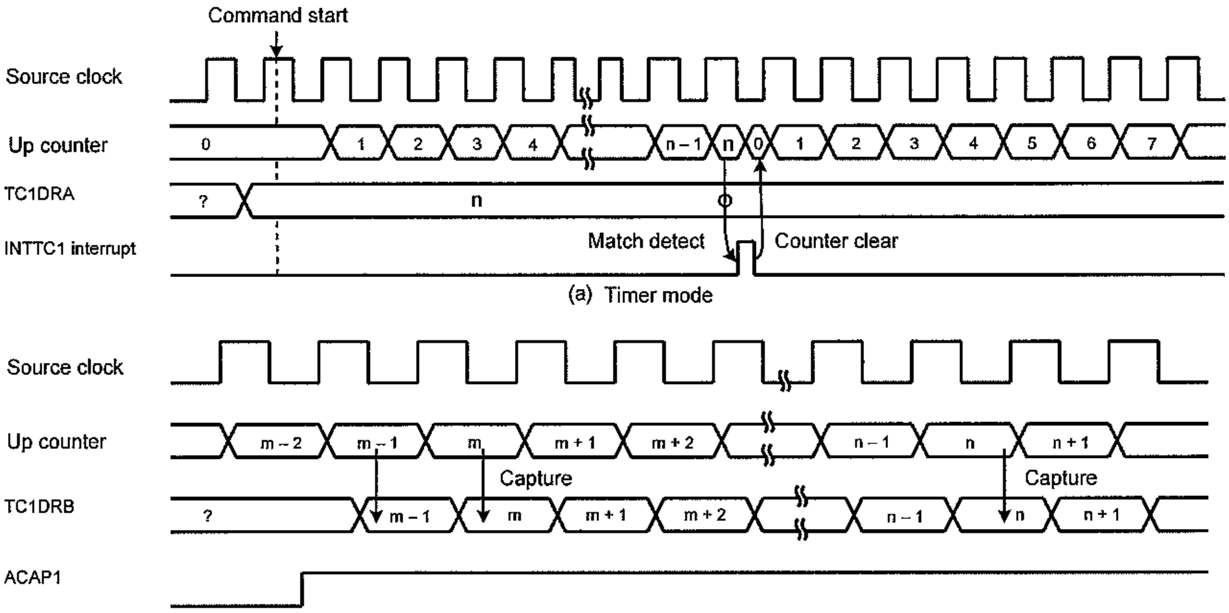
TC1CK	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode	
	DV7CK = 0		DV7CK = 1			
	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]	Resolution [μs]	Maximum Time Setting [s]
00	128	8.39	244.14	16.0	244.14	16.0
01	8.0	0.524	8.0	0.524	–	–
10	0.5	32.77 m	0.5	32.77 m	–	–

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

```
LDW     (TC1DRA), 1E84H     ; Sets the timer register (1 s ÷ 2^11/fc = 1E84H)
DI                          ;IMF = “0”
SET     (EIRL). 5           ; Enable INTTC1
EI                          ;IMF = “1”
LD      (TC1CR), 00000000B  ; TFF1 ¨ “0”, TC1CK ¨ “00”, TC1M ¨ “00”
LD      (TC1CR), 00010000B  ; Starts TC1
```

Example 2 :Auto-capture

```
LD      (TC1CR), 01010000B  ; ACAP1 ¨ “1” (Capture)
LD      WA, (TC1DRB)        ; Reads the capture value
```



(a) Timer mode

(b) Auto capture

Figure 8-2 Timer Mode Timing Chart

8.3.2 External trigger timer mode

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

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

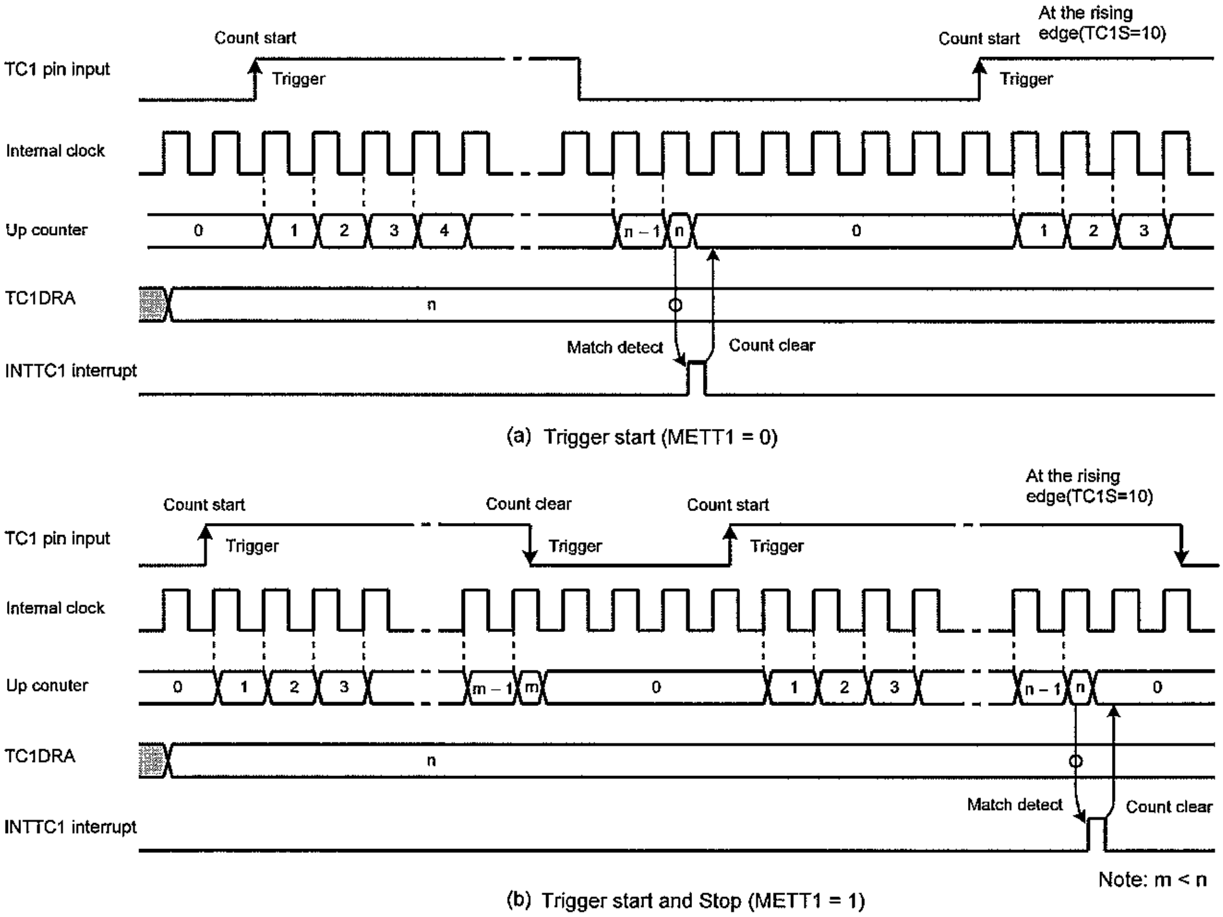
The TC1 pin input has the noise rejection; therefore, pulses of 4/fc [s] or less are rejected as noise. A pulse width of 12/fc [s] or more is required for edge detection in NORMAL1/2 or IDLE1/2 mode. The noise rejection circuit is turned off in SLOW1/2 and SLEEP1/2 modes. But, a pulse width of one machine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

Example 1 :Detects rising edge in TC1 pin input and generates an interrupt 100 ms later.
(at fc = 16 MHz, DV7CK = 0)

```
LDW     (TC1DRA), 00C8H         ; 100 ms ÷ 2^3/fc = C8H
DI                              ;IMF = "0"
SET     (EIRL). 5               ; INTTC1 interrupt enable
EI                              ; IMF = "1"
LD      (TC1CR), 00001000B      ; TFF1 = "0", TC1CK = "10", TC1M = "00"
LD      (TC1CR), 00101000B      ; TC1 external trigger start, METT1 ="0"
```

Example 2 :Generates an interrupt, inputting "L" level pulse (pulse width: 4 ms or more) to the TC1 pin.
(at fc = 16 MHz))

```
LDW     (TC1DRA), 01F4H         ; 4 ms ÷ 2^7/fc = 01F4H
DI                              ; IMF = "0"
SET     (EIRL). 5               ; INTTC1 interrupt enable
EI                              ; IMF = "1"
LD      (TC1CR), 00000100B      ; TFF1 = "0", TC1CK = "10", TC1M = "00"
LD      (TC1CR), 01110100B      ; TC1 external trigger start, METT1 = 1
```



(a) Trigger start (METT1 = 0)

(b) Trigger start and Stop (METT1 = 1)

Figure 8-3 External Trigger Timer Mode Timing Chart

8.3.3 Event counter mode

In this mode, events are counted at the edge of the TC1 pin input (either the rising or falling edge can be selected with the external trigger TC1CR<TC1S>). The contents of TC1DRA are compared with the contents of up counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. After the counter is cleared, the up counter starts counting by TC1 input edge. Match detect is executed on other edge of count-up. A match can not be detected and INTTC1 is not generated when the pulse is still in same state. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

Setting TC1CR<ACAP1> to "1" transfers the current contents.of up counter to TC1DRB (Auto-capture function).

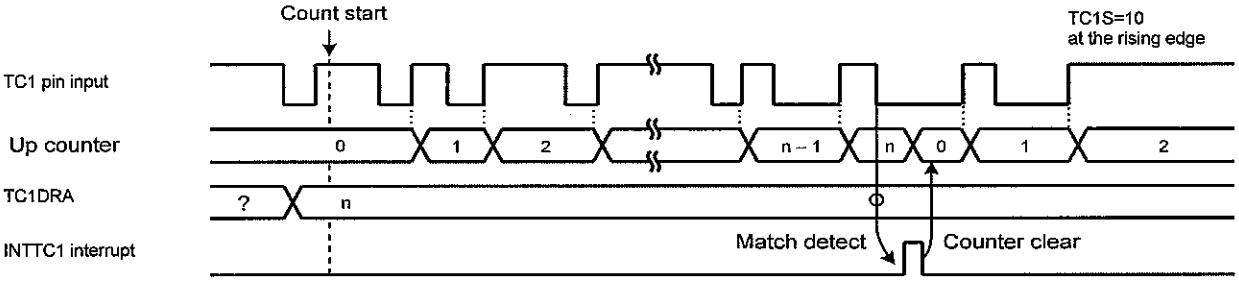


Figure 8-4 Event Counter Mode Timing Chart

Table 8-2 Timer/Counter 1 External Clock Source

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

8.3.4 Window mode

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

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

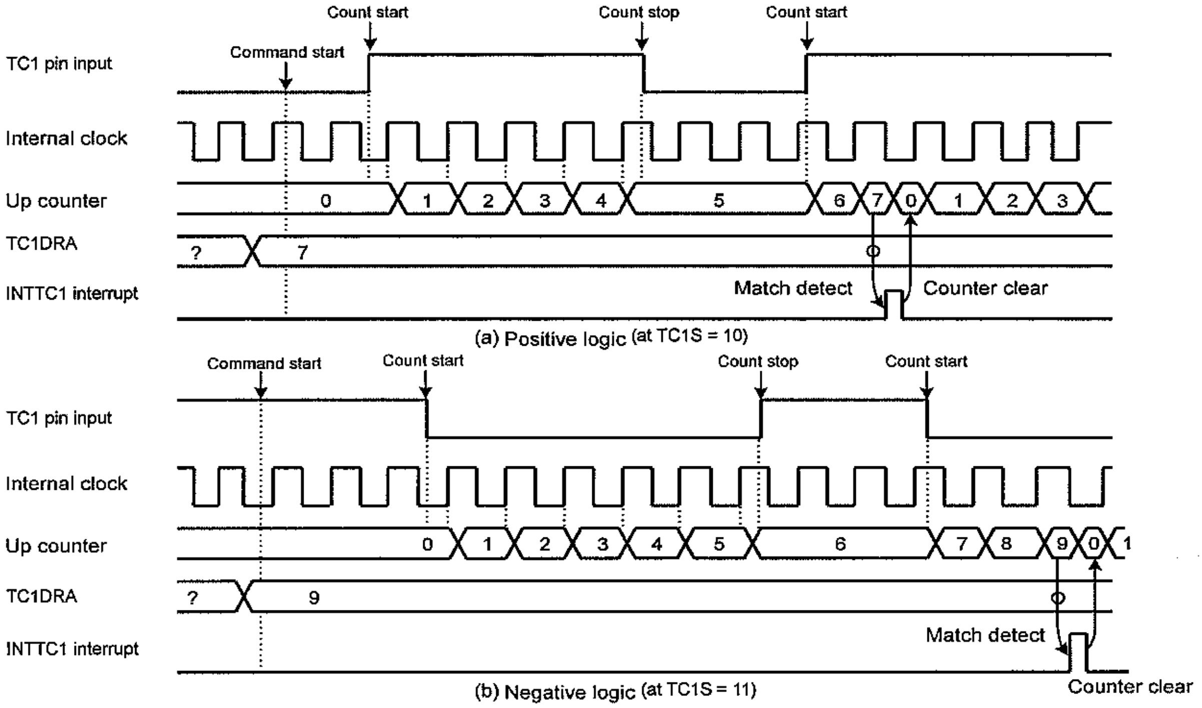


Figure 8-5 Window Mode Timing Chart

8.3.5 Pulse width measurement mode

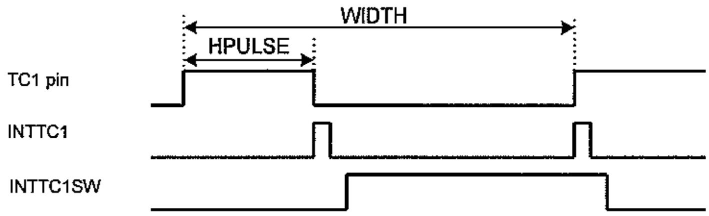
In this mode, counting is started by the external trigger (Set to external trigger start by TC1CR<TC1S>). The trigger can be selected either the rising or falling edge of the TC1 pin input. The source clock is used an internal clock. On the next falling (Rising) edge, the counter contents are transferred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

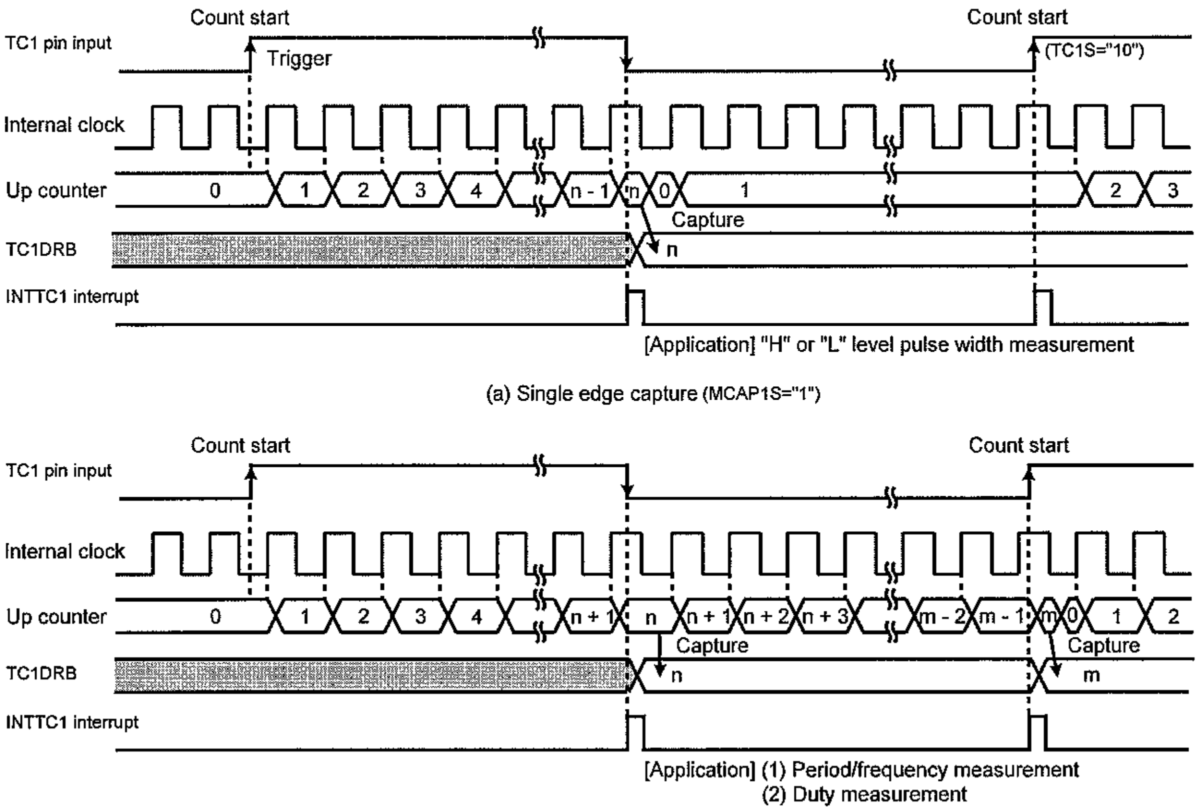
Note: Be sure to read the captured value from TC1DRB before the next trigger edge is detected. If fail to read it, it becomes undefined. It is recommended that a 16-bit access instruction be used to read from TC1DRB.

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

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

```
              CLR     (INTTC1SW). 0       ; INTTC1 service switch initial setting
              LD      (TC1CR), 00000110B  ; Sets the TC1 mode and source clock
              DI                          ; IMF = "0"
              SET     (EIRL). 5           ; Enables INTTC1
              EI                          ; IMF = "1"
              LD      (TC1CR), 00100110B  ; Starts TC1 with an external trigger at MCAP1 = 0
               :
PINTTC1:      CPL     (INTTC1SW). 0       ; Inverts INTTC1 service switch
              JRS     F, SINTTC1
              LD      WA, (TC1DRBL)       ; Reads TC1DRB ("H" level pulse width)
              LD      (HPULSE), WA
              RETI
SINTTC1:      LD      WA, (TC1DRBL)       ; Reads TC1DRB (Period)
              LD      (WIDTH), WA
               :
              RETI                        ; Duty calculation
               :
VINTTC1:      DW      PINTTC1             ; INTTC1 Interrupt vector
```





(a) Single edge capture (MCAP1S="1")

(b) Double edge capture (MCAP1S="0")

Figure 8-6 Pulse Measurement Mode Timing Chart

8.3.6 Programmable pulse generate (PPG) output mode

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

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

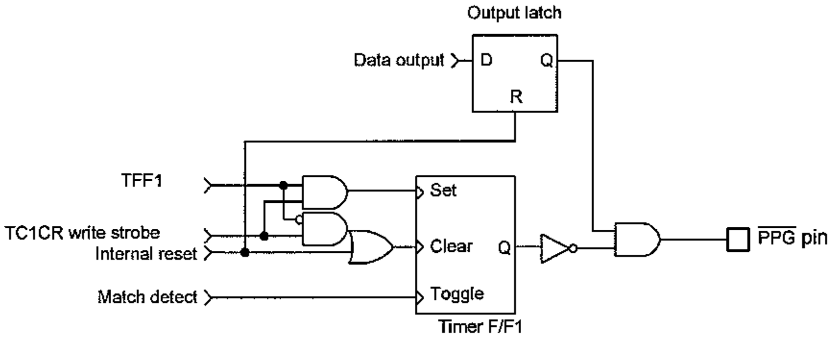
Note 1: To change the content of the timer register when the timer is running, change it to a sufficiently large value, compared with the current count. If the timer register content is changed to a value smaller than the current count when the timer is running, it is likely that unintended pulses may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

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

Note 3: To restart the PPG output mode, it is necessary to initialize timer F/F1. To initialize timer F/F1, Change to the timer mode once (the timer mode need not be restarted), Then set to the PPG output mode and also set TFF1 again.

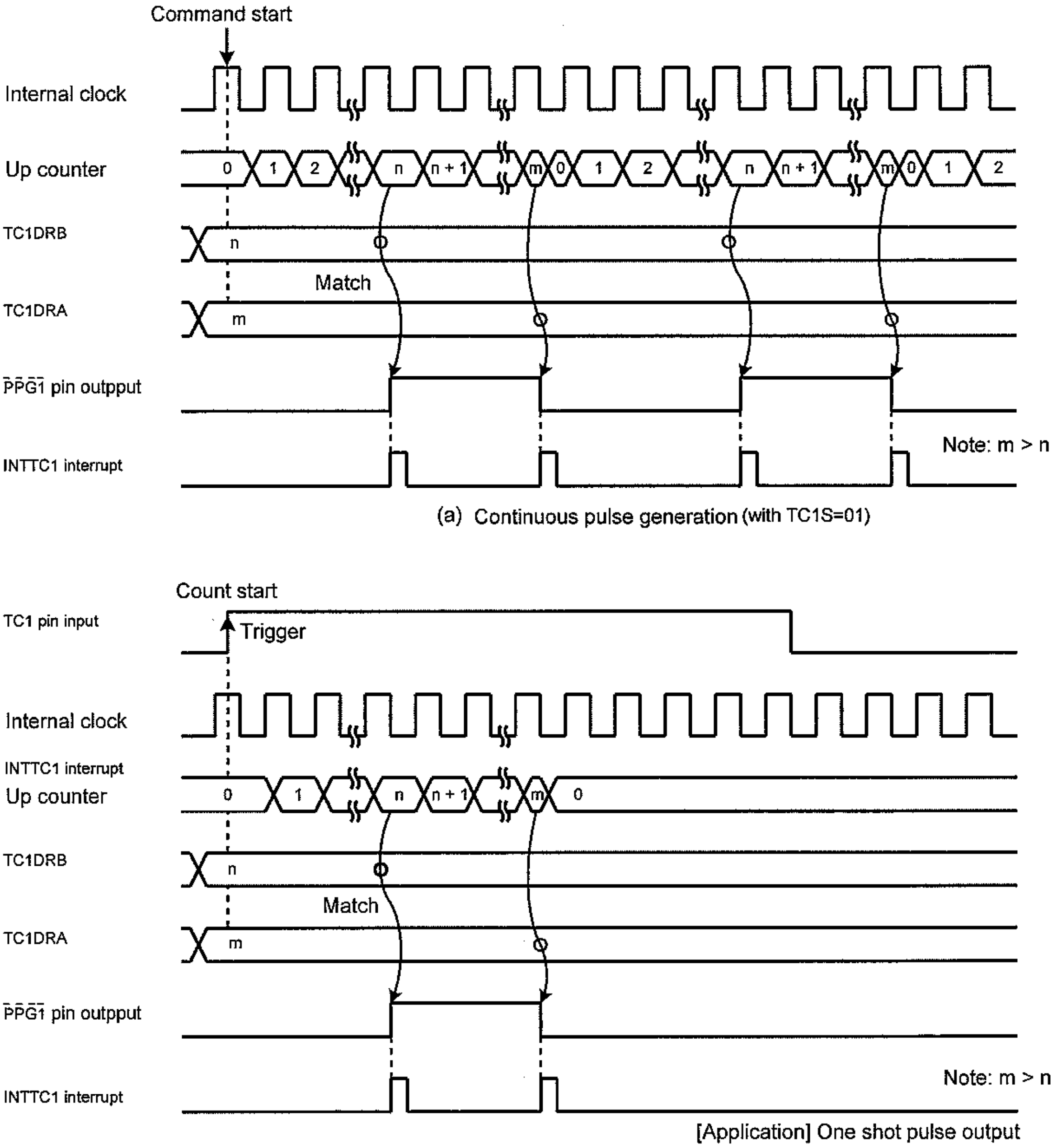
Example :Pulse output “H” level 800 ms, “L” level 200 ms (at fc = 16 MHz)

```
             Port setting
LD           (TC1CR), 10001011B        ; Sets the PPG output mode
LDW          (TC1DRA), 07D0H           ; Sets the period (1 ms ÷ 2^3/fc = 07D0H)
LDW          (TC1DRB), 0190H           ; Sets "L" level pulse width (200 µs ÷ 2^3/fc = 0190H)
LD           (TC1CR), 10011011B        ; Starts
```



MPX: Multiplexer

Figure 8-7 $\overline{\text{PPG}}$ Output



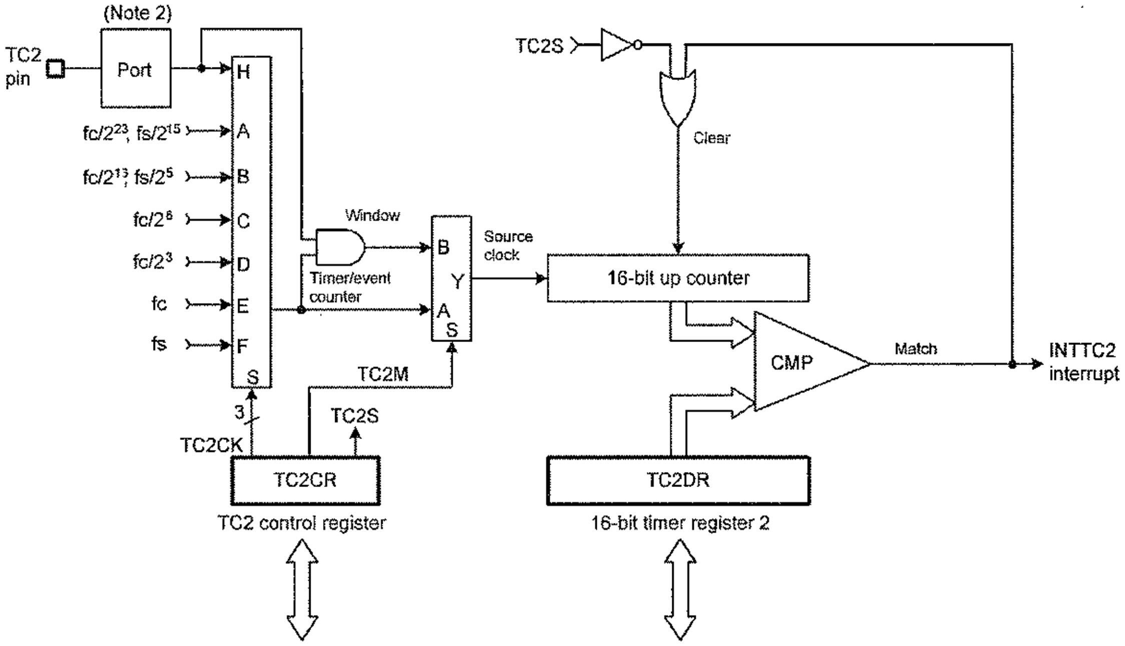
(a) Continuous pulse generation (with TC1S=01)

(b) One-shot (with TC1S=10)

Figure 8-8 PPG Output Mode Timing Chart

9. 16-Bit Timer/Counter 2

9.1 Configuration



Note 1: MPX: Multiplexer
CMP: Comparator

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

Figure 9-1 Timer/Counter 2 (TC2A)

9.2 Control

The timer/counter 2 is controlled by a timer/counter 2 control register (TC2CR) and a 16-bit timer register 2 (TC2DR).

TC2DR (0025H, 0024H)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TC2DRH (0025H)								TC2DRL (0024H)							
	(Initial value: 1111 1111 1111 1111)								R/W							

TC2CR (0023H)	7	6	5	4	3	2	1	0	
			TC2S	TC2CK				TC2M	(Initial value: **00 00*0)

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

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

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

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

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

Note 5: Values to be loaded to the timer register must satisfy the following condition.
TC2DR > 1 (TC2DR15 to TC2DR11 > 1 at warm up)

Note 6: If a read instruction is executed for TC2CR, read data of bit 7, 6 and 1 are unstable.

Note 7: On entering STOP mode, the TC2 start control (TC2S) is cleared to "0" automatically. So, the timer stops. Once the STOP mode has been released, to start using the timer counter, set TC2S again.

Note 8: The high-frequency clock (fc) canbe selected only when the time mode at SLOW2 mode is selected.

9.3 Function

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

And when warm-up for switching from SLOW to NORMAL2, the timer/counter2 is used as a time mode.

9.3.1 Timer mode

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

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

Table 9-1 Source Clock (Internal clock) for Timer/Counter 2 (at fc = 16 MHz, DV7CK=0)

TC2CK	NORMAL1/2, IDLE1/2 mode				SLOW1/2 Mode		SLEEP1/2 Mode	
	DV7CK = 0		DV7CK = 1		Resolution	Maximum Time Setting	Resolution	Maximum Time Setting
	Resolution	Maximum Time Setting	Resolution	Maximum Time Setting				
000	524.3 [ms]	9.54 [h]	1.0 [s]	18.2 [h]	1 [s]	18.20 [h]	1.0 [s]	18.2 [h]
001	512.0 [μs]	33.6 [s]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]	0.98 [ms]	1.07 [min]
010	16.0 [μs]	1.05 [s]	16.0 [μs]	1.05 [s]	–	–	–	–
011	0.5 [μs]	32.8 [ms]	0.5 [μs]	32.8 [ms]	–	–	–	–
100	–	–	–	–	62.5 [ns] (Note)	–	–	–
101	30.5 [μs]	2.0 [s]	30.5 [ms]	2.0 [s]	–	–	–	–

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

Example :Sets the timer mode with source clock fc/2^3 [Hz] and generates an interrupt every 25 ms (at fc = 16 MHz, DV7CK=0).

```
LDW     (TC2DR), C350H       ; Sets TC2DR (25 ms ÷ 2^3/fc = C350H)
DI                           ; IMF = "0"
SET     (EIRE). 6            ; Enables INTTC2 interrupt
EI                           ; IMF = "1"
LD      (TC2CR), 00001100B   ; TC2CK ← "011", TC2M ← "0"
LD      (TC2CR), 00101100B   ; Starts TC2
```

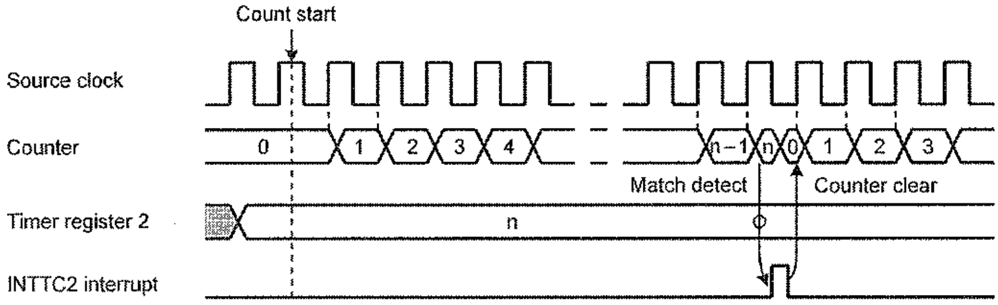


Figure 9-2 Timer Mode Timing Chart

9.3.2 Event counter mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TC2DR are compared with the contents of the up counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The minimum input pulse width of TC2 pin is shown in Table 9-2. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width. Match detect is executed on the falling edge of the TC2 pin. A match can not be detected and INTTC2 is not generated when the pulse is still in a falling state.

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

```
LDW     (TC2DR), 640          ; Sets TC2DR
DI                            ; IMF = "0"
SET     (EIRE). 6             ; Enables INTTC2 interrupt
EI                            ; IMF = "1"
LD      (TC2CR), 00011100B    ; TC2CK ← "111", TC2M ← "0"
LD      (TC2CR), 00111100B    ; Starts TC2
```

Table 9-2 Timer/Counter 2 External Clock Source

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

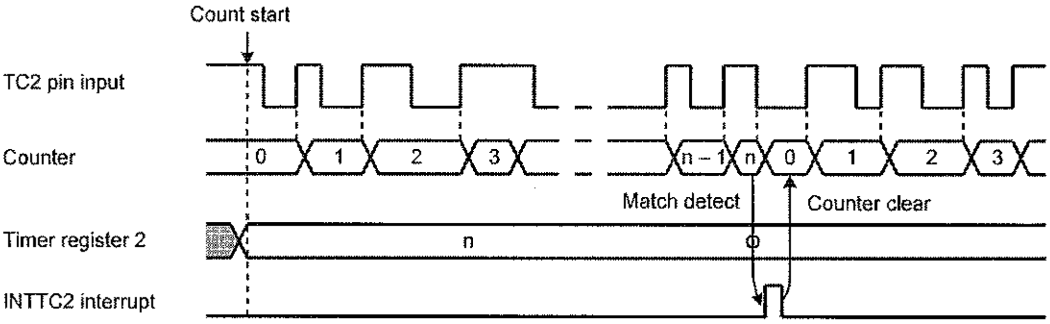


Figure 9-3 Event Counter Mode Timing Chart

9.3.3 Window mode

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

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

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

Example :Generates an interrupt, inputting "H" level pulse width of 120 ms or more. (at fc = 16 MHz, DV7CK = 0)

```
LDW     (TC2DR), 00EAH        ; Sets TC2DR (120 ms ÷ 2^13/fc = 00EAH)
DI                            ; IMF = "0"
SET     (EIRE). 6             ; Enables INTTC2 interrupt
EI                            ; IMF = "1"
LD      (TC2CR), 00000101B    ; TC2CK ← "001", TC1M ← "1"
LD      (TC2CR), 00100101B    ; Starts TC2
```

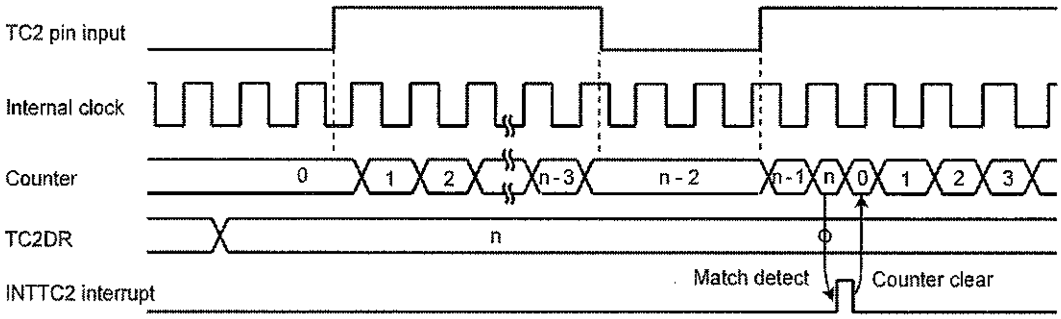


Figure 9-4 Window Mode Timing Chart

10. 8-Bit Timer/Counter (TC3, TC4)

10.1 Configuration

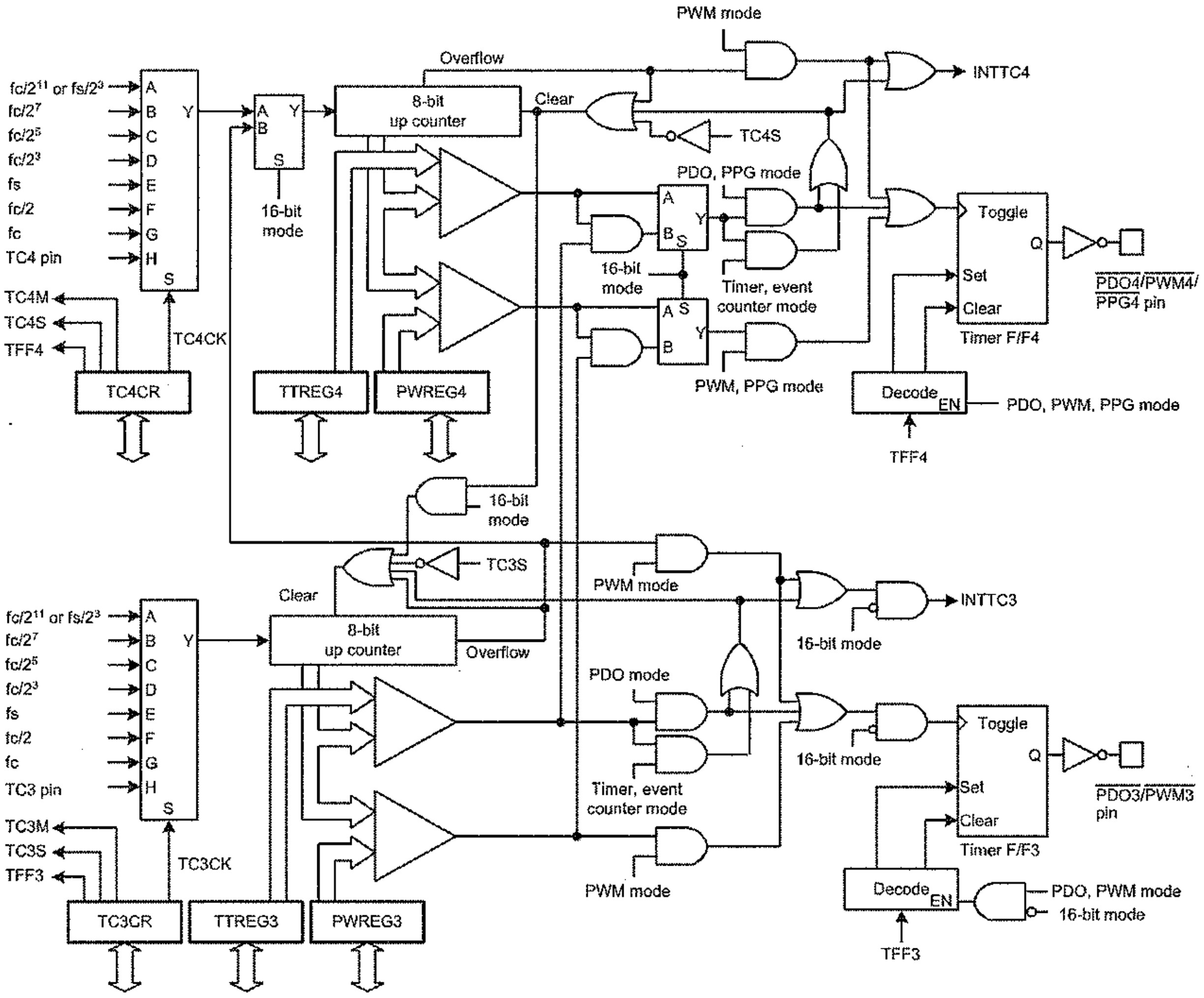


Figure 10-1 8-Bit Timer 3, 4

10.2 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

	7	6	5	4	3	2	1	0	
TTREG3 (0014H) R/W									(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
PWREG3 (0018H) R/W									(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 modes.

Timer/Counter 3 Control Register

TC3CR	7	6	5	4	3	2	1	0	
(0027H)	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 and IDLE1/2 modes		SLOW1/2 and SLEEP1/2 modes	
			DV7CK = 0	DV7CK = 1		
		000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		001	$fc/2^7$	$fc/2^7$	–	
		010	$fc/2^5$	$fc/2^5$	–	
		011	$fc/2^3$	$fc/2^3$	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	fc	
		111	TC3 pin input			
TC3S	TC3 start control	0: Stop and counter clear 1: Command start				
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				

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" Æ "0"), do not change TC3M, TC3CK and TFF3. But it is possible to change TC3M, TC3CK and TFF3 at the start timing (TC3S = "0" Æ "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 10-1 and Table 10-2 for details.

Note 7: Value of timer register depends on the operating mode, refer to Table 10-3 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.

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

TTREG4
(0015H)
R/W

7	6	5	4	3	2	1	0

(Initial value: 1111 1111)

PWREG4
(0019H)
R/W

7	6	5	4	3	2	1	0

(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 modes.

Timer/Counter 4 Control Register

TC4CR
(0028H)

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

(Initial value: 0000 0000)

TFF4	Timer F/F4 control	0: Clear 1: Set				R/W
TC4CK	TC4 source clock select [Hz]		NORMAL1/2 and IDLE1/2 modes		SLOW1/2 and SLEEP1/2 modes	
			DV7CK = 0	DV7CK = 1		
		000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		001	$fc/2^7$	$fc/2^7$	–	
		010	$fc/2^5$	$fc/2^5$	–	
		011	$fc/2^3$	$fc/2^3$	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	–	
		111	TC4 pin input			
TC4S	TC4 start control	0: Stop and counter clear 1: Command start				
TC4M	TC4 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				

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” Æ “0”), do not change TC4M, TC4CK and TFF4. But it is possible to change TC4M, TC4CK and TFF4 at the start timing (TC4S = “0” Æ “1”).

Note 4: When TC4M is selected to “1**” (16-bit mode), the source clock is automatically selected to the overflowing 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 10-1 and Table 10-2 for details.

Note 8: Value of timer register depends on the operating mode, refer to Table 10-3 for details.

Table 10-1 Operating Mode and Available Source Clock (under NORMAL1/2 mode, IDLE1/2 mode)

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

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).

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

Table 10-2 Operating Mode and Available Source Clock (under SLOW1/2 mode, SLEEP1/2 mode)

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

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).

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

Table 10-3 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, 3) £ 65535
fc warm-up counter	256 £ (TTREG4, 3) £ 65535
16-bit PWM	2 £ (PWREG4, 3) £ 65534
16-bit PPG	1 £ (PWREG4, 3) < (TTREG4, 3) £ 65535 and (PWREG4, 3) + 1 < (TTREG4, 3)

Note: n = 3, 4

10.3 Function

Timer/counter 3, 4 have eight operating modes: 8-bit timer, 8-bit event cointer, 8-bit programmable divider output mode, 8-bit pulse width modulation output mode, 16-bit timer, 16-bit event cointer, warm-up counter mode,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 by cascade connection.

10.3.1 8-bit timer mode (Timer/counter 3, 4)

In this mode, counting up is performed using the internal clock. 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 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 $\overline{\text{PDOj/PWMj/PPGj}}$ pin.

Note 2: In the timer 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

Table 10-4 Timer/Counter 3, 4 Source Clock (Internal clock)

Source Clock			Resolution		Maximum Time Setting	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc=16 MHz	At fs = 32.768 kHz	At fc=16 MHz	At 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]	32.6 [ms]	62.3 [ms]
$fc/2^{7}$	$fc/2^{7}$	–	8 [µs]	–	2.0 [ms]	–
$fc/2^{5}$	$fc/2^{5}$	–	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).

```
LDW     (TTREG4), 0AH       ; Sets the timer register (80 µs ÷ 2^7/fc = 0AH)
DI
SET     (EIRH). 1           ; Enables INTTC4 interrupt
EI
LD      (TC4CR), 00010000B  ; Sets the 8-bit timer mode and source clock (fc/2^7)
LD      (TC4CR), 00011000B  ; Starts TC4
```

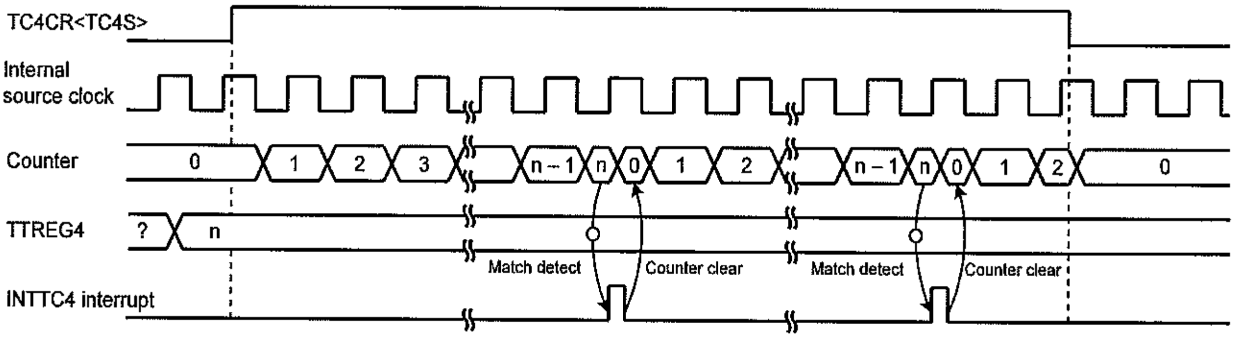


Figure 10-2 8-Bit Timer Mode Timing Chart (in case of timer/counter 4)

10.3.2 8-bit event counter mode (Timer/counter 3, 4)

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

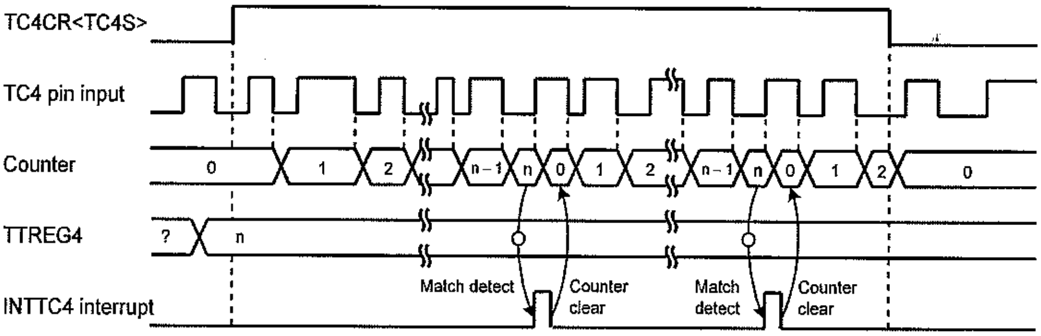


Figure 10-3 Event Counter Mode Timing Chart (in case of timer/counter 4)

10.3.3 8-bit programmable divider output (PDO) mode (Timer/counter 3, 4)

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 $\overline{\text{PDOj}}$ output is toggled.

Example :Output a 1024 Hz pulse (at fc = 16 MHz = "0", in case of TC4)

```
LD        (TTREG4), 3DH              ; (1/1024 ÷ 2^7/fc ) ÷ 2 = 3DH

LD        (TC4CR), 00010001B         ; Sets 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: j = 3, 4

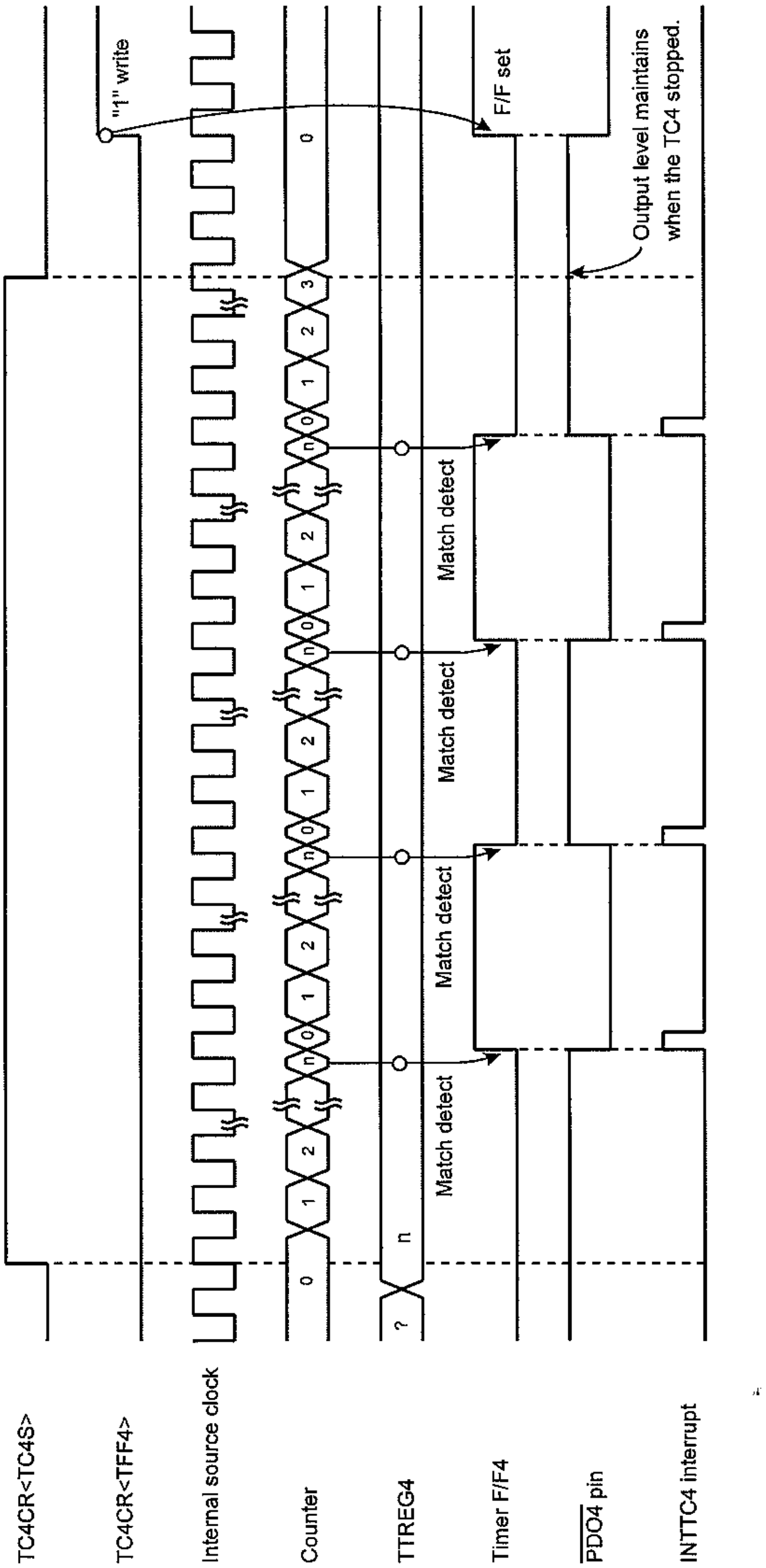


Figure 10-4 8-Bit PDO Mode Timing Chart (in case of timer/counter 4)

10.3.4 8-bit pulse width modulation (PWM) output mode (Timer/counter 3, 4)

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   (TCjCR). 3   ; Stops timer/counter
  CLR   (TCjCR). 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

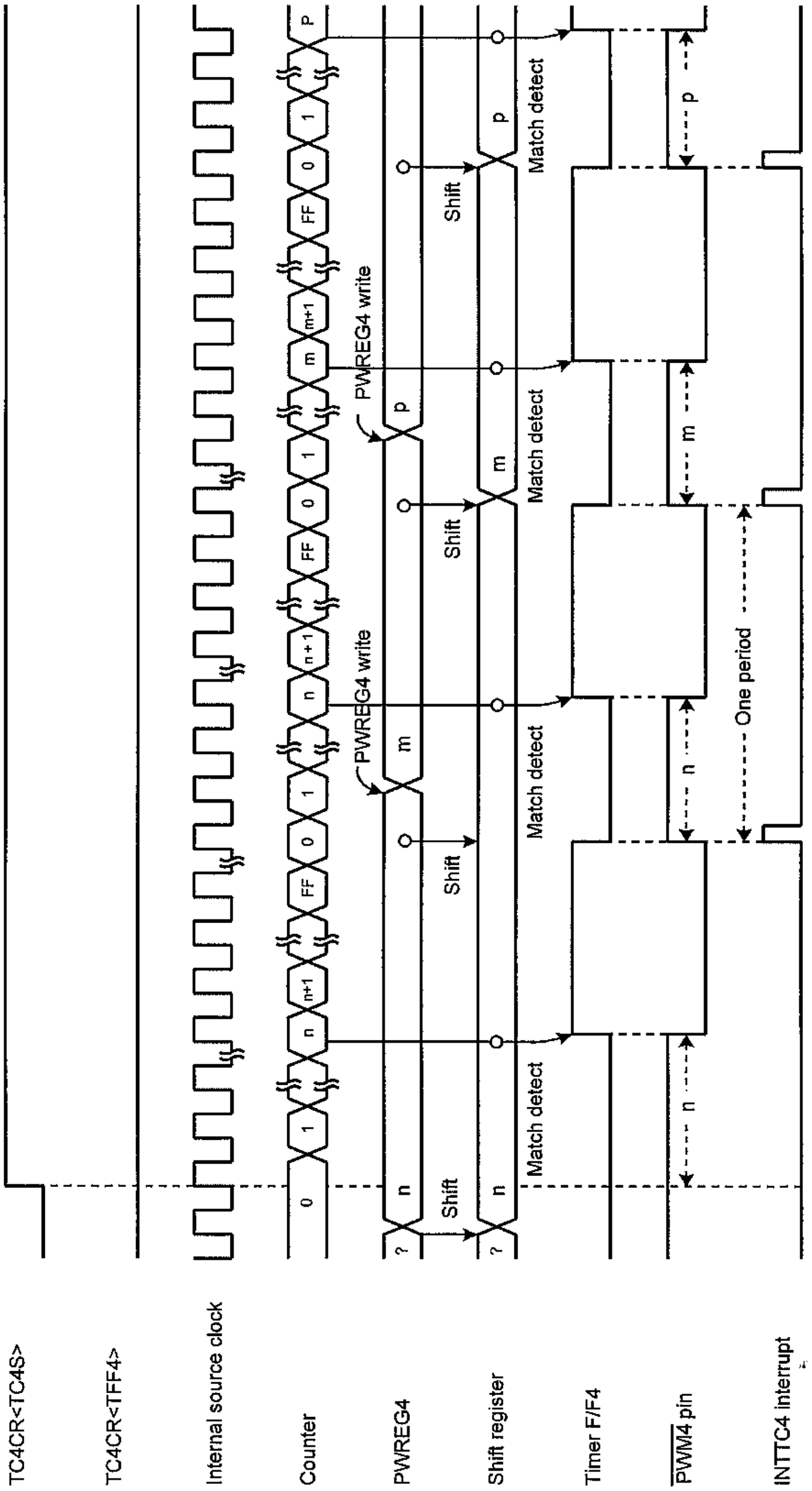


Figure 10-5 8-Bit PWM Mode Timing Chart (in case of timer/counter 4)

Table 10-5 PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At 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]
$fc/2^{7}$	$fc/2^{7}$	–	8 [μs]	–	2.05 [ms]	–
$fc/2^{5}$	$fc/2^{5}$	–	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]	7.81 [ms]
fc/2	fc/2	–	125 [ns]	–	32 [μs]	–
fc	fc	–	62.5 [ns]	–	16 [μs]	–

10.3.5 16-bit timer mode (Timer/counter 3 and 4)

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

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

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.

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

Note 2: In the timer 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

Table 10-6 Source Clock of 16-Bit Timer Mode

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

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). 1            ; 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
```

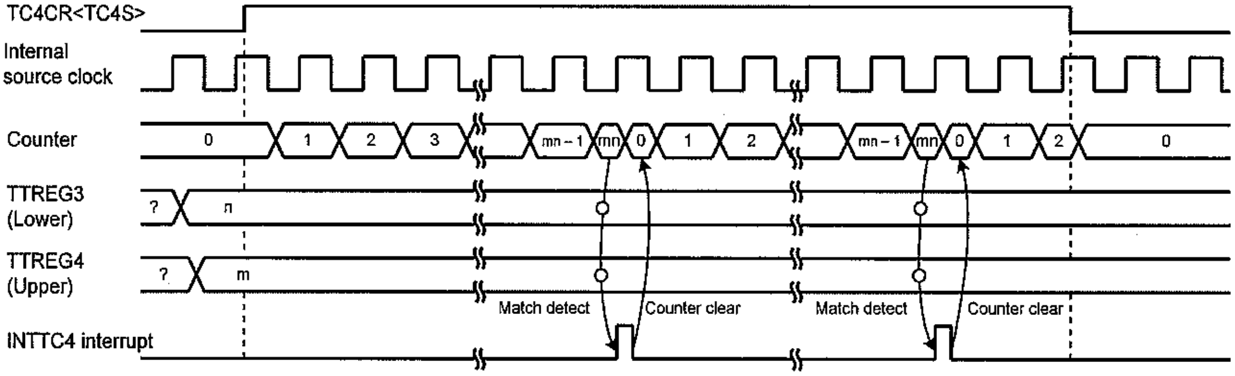


Figure 10-6 16-Bit Timer Mode Timing Chart (in case of timer/counter 3 and 4)

10.3.6 16-bit event counter mode (Timer/counter 3 and 4)

In this mode, event are counted on the falling edge of the TC3 pin input. Timer/counter 3 and 4 are also available as a 16-bit event counter mode by cascade connection.

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

10.3.7 16-bit pulse width modulation (PWM) output mode (Timer/counter 3 and 4)

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

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.

Note 1: In PWM mode, write to the timer registers PWREG4, PWREG3 immediately after an INTTC4 interrupt is generated (Normally during the INTTC4 interrupt service routine). If writing to PWREG4, PWREG3 and INTTC4 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 INTTC4 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 $\overline{\text{PWM4}}$, modify TC4CR<TTF4> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFF4 simultaneously.

Example: Fixes $\overline{\text{PWM4}}$ output at high level after timer/counter is stopped

```
  CLR   (TC4CR). 3   ; Stops timer/counter
  CLR   (TC4CR). 7   ; Sets PWM4 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.

Table 10-7 16-Bit PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At 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	fc/2^7	–	8 [μs]	–	524.3 [ms]	–
fc/2^5	fc/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 $\overline{\text{PWM4}}$ pin with 16-bit PWM mode (at fc = 16 MHz)

```
LDW     (PWREG3), 07D0H     ; Sets pulse width

LD      (TC3CR), 33H        ; Sets the 16-bit PWM mode (Lower) and source clock (fc/2^3)

LD      (TC4CR), 056H       ; Sets the TFF4 to "1" and sets the 16-bit PWM mode (Upper)

LD      (TC4CR), 05EH       ; Starts timer/counter
```

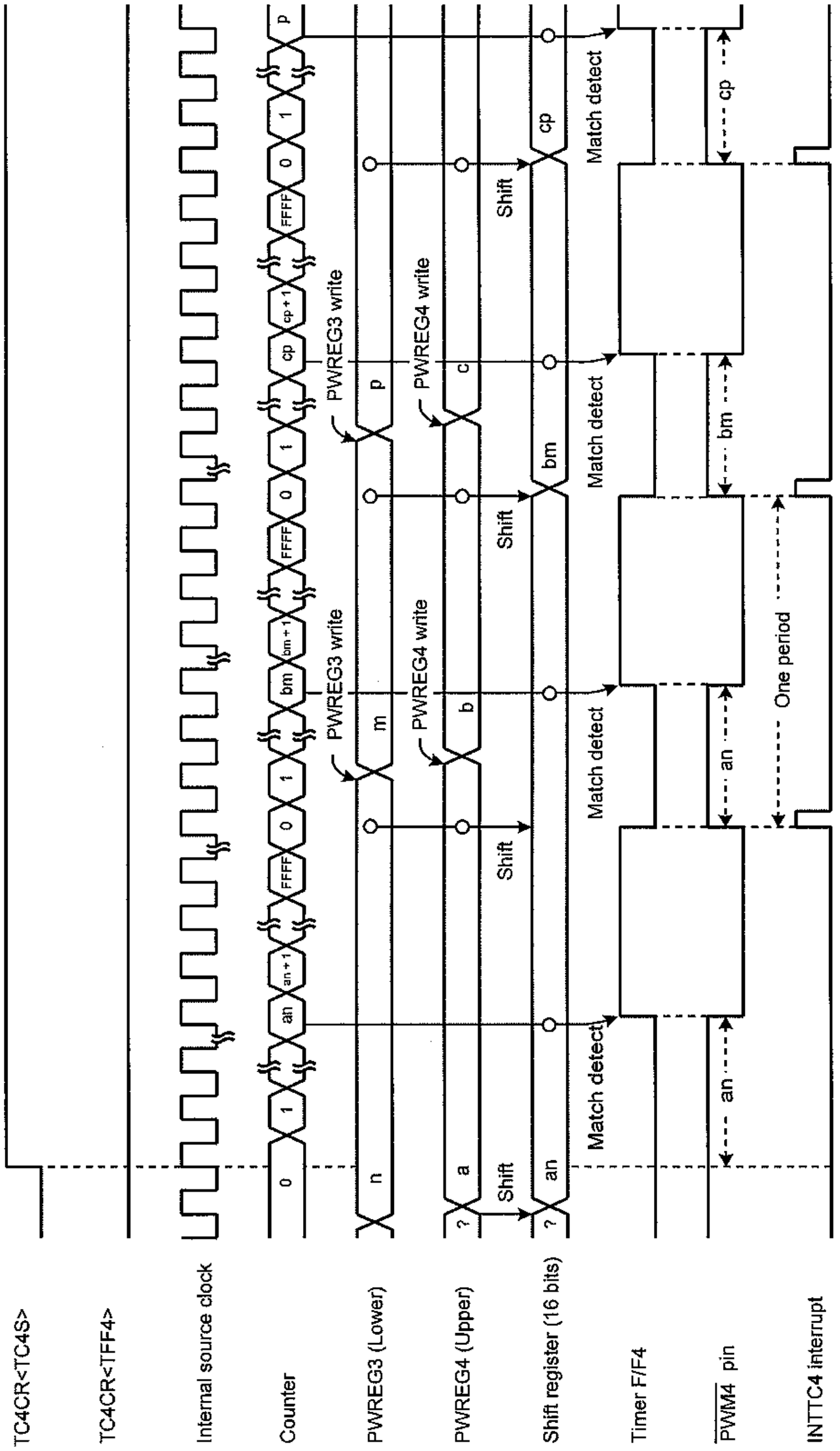


Figure 10-7 16-Bit PWM Mode Timing Chart (in case of timer/counter 3 and 4)

10.3.8 16-bit programmable pulse generate (PPG) output mode (Timer/counter 3 and 4)

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

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.

Example :Extract the pulse, whose term and "high" width is 16.385 ms and 1 ms respectively, from $\overline{\text{PPG4}}$ pin with 16-bit PPG mode (at fc = 16 MHz)

```
LDW     (PWREG3), 07D0H     ; Sets pulse width

LDW     (TTREG3), 8002H     ; Sets pulse term

LD      (TC3CR), 33H        ; Sets the 16-bit PPG mode (Lower) and source clock (fc/2^3)

LD      (TC4CR), 057H       ; Sets the TFF4 to "1" and sets the 16-bit PPG mode (Upper)

LD      (TC4CR), 05FH       ; 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 $\overline{\text{PPG4}}$, modify TC4CR<TTF4> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

Example: Fixes $\overline{\text{PPG4}}$ output at high level after timer/counter is stopped

```
CLR  (TC4CR). 3  ; Stops timer/counter
CLR  (TC4CR). 7  ; Sets PPG4 output to high level output
```

Note 3: i = 3, 4

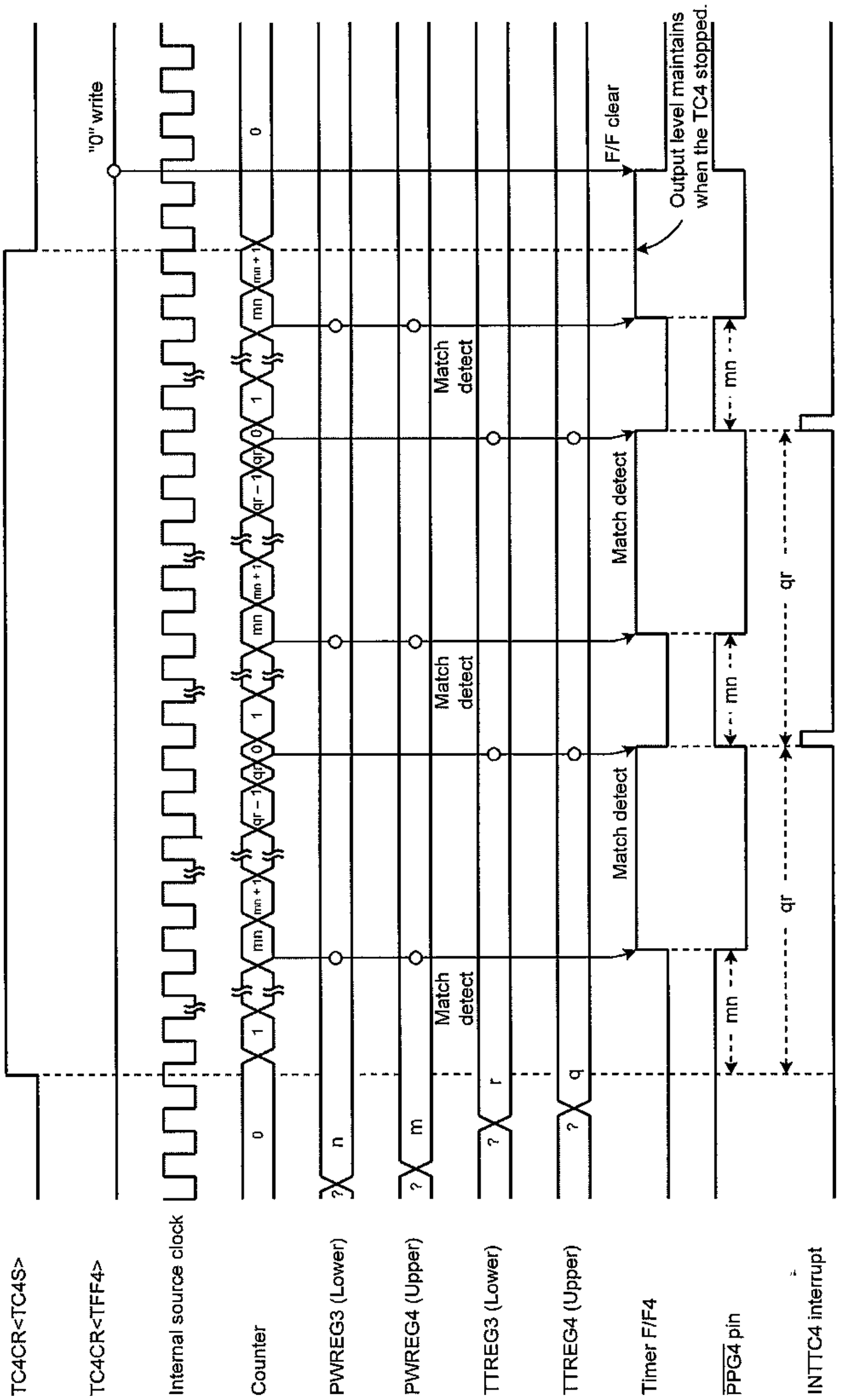


Figure 10-8 16-Bit PPG Mode Timing Chart (in case of timer/counter 3 and 4)

10.3.9 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 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 $\overline{\text{PDOi/PWMi/PPGi}}$ pin.

Note 2: In the warm-up mode, the lower 8 bits of TTREGm, TTREGn are ignored and an interrupt is generated by matching the upper 8 bits.

Note 3: i = 3, 4

10.3.9.1 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, 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 low-frequency clock by setting SYSCR2<SYSCK> to "1".
After that, halt the high-frequency oscillation by clearing SYSCR2<XEN> to "0".

Table 10-8 Warm-up Period for Low-frequency Oscillation (at fs = 32.768 kHz)

Min (at TTREGm, TTREGn = 0100H)	Max (at TTREGm, TTREGn = FF00H)
7.81 ms	1.99 s

Note: m = 4, n = 3

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). 1          ; 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)

                                       ; SYSCR2<XEN> ← "0" (Turns off low-frequency oscilla-
            CLR     (SYSCR2). 7        tion)

            RETI

             :       :

VINTTC4:    DW      PINTTC4            ; INTTC4 vector table
```

10.3.9.2 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 10-9 Warm-up Period for High-frequency Oscillation (at fc = 16 MHz)

Min (at TTREGm, TTREGn = 0100H)	Max (at TTREGm, TTREGn = FF00H)
16 µs	4.08 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). 1           ; 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)

                                        ; SYSCR2<XTEN> ←— "0" (Turns off high-frequency oscilla-
            CLR     (SYSCR2). 6         tion)

            RETI

             :       :

VINTTC4:    DW      PINTTC4             ; INTTC4 vector table
```

11. 8-Bit Timer/Counter (TC5, TC6)

11.1 Configuration

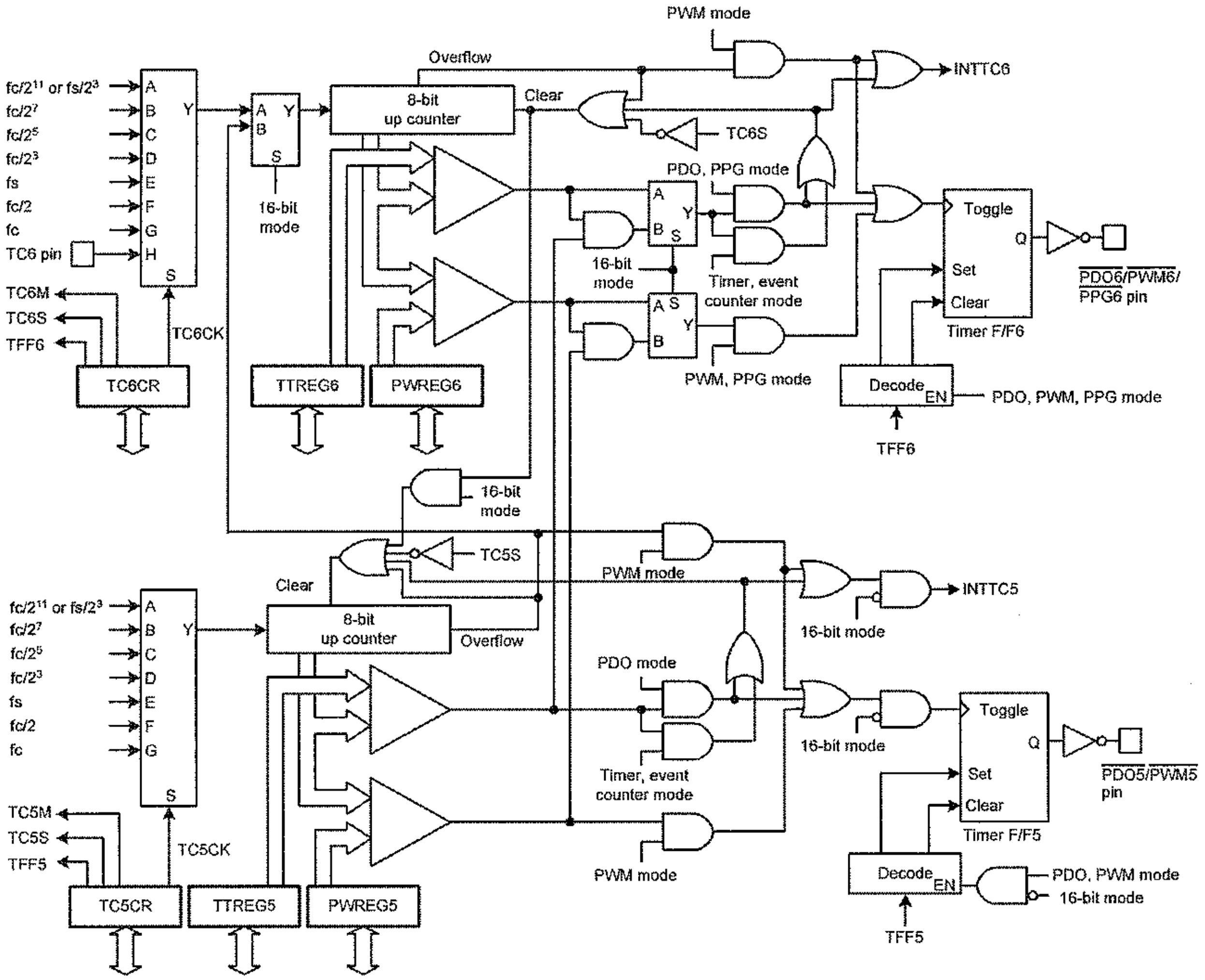


Figure 11-1 8-Bit Timer 5, 6

11.2 Control

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 (0016H) R/W	7	6	5	4	3	2	1	0	
									(Initial value: 1111 1111)

PWREG5 (001AH) 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 8-bit PWM and 16-bit PWM modes.

Timer/Counter 5 Control Register

TC5CR (0029H)	7	6	5	4	3	2	1	0	
	TFF5	TC5CK			TC5S	TC5M			(Initial value: 0000 0000)

TFF5	Timer F/F5 control	0: Clear 1: Set				R/W
TC5CK	TC5 source clock select [Hz]		NORMAL1/2 and IDLE1/2 modes		SLOW1/2 and SLEEP1/2 modes	
			DV7CK = 0	DV7CK = 1		
		000	$fc/2^{11}$	$fs/2^3$	$fs/2^3$	
		001	$fc/2^7$	$fc/2^7$	–	
		010	$fc/2^5$	$fc/2^5$	–	
		011	$fc/2^3$	$fc/2^3$	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	fc	
		111	TC5 pin input			
TC5S	TC5 start control	0: Stop and counter clear 1: Command start				
TC5M	TC5 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 TC6M) 1**: Reserved				

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

Note 2: During TC5 operation, do not change TC5M, TC5CK and TFF5.

Note 3: When TC5 operation is stopped (TC5S = “1” Æ “0”), do not change TC5M, TC5CK and TFF5. But it is possible to change TC5M, TC5CK and TFF5 at the start timing (TC5S = “0” Æ “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 11-1 and Table 11-2 for details.

Note 7: Value of timer register depends on the operating mode, refer to Table 11-3 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.

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 (0017H) R/W									(Initial value: 1111 1111)

	7	6	5	4	3	2	1	0	
PWREG6 (001BH) 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 modes.

Timer/Counter 6 Control Register

	7	6	5	4	3	2	1	0	
TC6CR (002AH)	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 and IDLE1/2 modes		SLOW1/2 and SLEEP1/2 modes	
			DV7CK = 0	DV7CK = 1		
		000	$fc/2^{11}$	$fs/2^{3}$	$fs/2^{3}$	
		001	$fc/2^{7}$	$fc/2^{7}$	–	
		010	$fc/2^{5}$	$fc/2^{5}$	–	
		011	$fc/2^{3}$	$fc/2^{3}$	–	
		100	fs	fs	fs	
		101	fc/2	fc/2	–	
		110	fc	fc	–	
		111	TC6 pin input			
TC6S	TC6 start control	0: Stop and counter clear 1: Command start				
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				

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” Æ “0”), do not change TC6M, TC6CK and TFF6. But it is possible to change TC6M, TC6CK and TFF6 at the start timing (TC6S = “0” Æ “1”).

Note 4: When TC6M is selected to “1**” (16-bit mode), the source clock is automatically selected to the overflowing 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 11-1 and Table 11-2 for details.

Note 8: Value of timer register depends on the operating mode, refer to Table 11-3 for details.

Table 11-1 Operating Mode and Available Source Clock (under NORMAL1/2 mode, IDLE1/2 mode)

Operating Mode	$fc/2^{11}$ or $fs/2^{3}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{3}$	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 (TC5CK).

Note 2: i = 5, 6 (8-bit mode)
i = 5 (16-bit mode)

Table 11-2 Operating Mode and Available Source Clock (under SLOW1/2 mode, SLEEP1/2 mode)

Operating Mode	$fc/2^{11}$ or $fs/2^{3}$	$fc/2^{7}$	$fc/2^{5}$	$fc/2^{3}$	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 (TC5CK).

Note 2: i = 5, 6 (8-bit mode)
i = 5 (16-bit mode)

Table 11-3 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 £ (TTREG6, 5) £ 65535
fc warm-up counter	256 £ (TTREG6, 5) £ 65535
16-bit PWM	2 £ (PWREG6, 5) £ 65534
16-bit PPG	1 £ (PWREG6, 5) < (TTREG6, 5) £ 65535 and (PWREG6, 5) + 1 < (TTREG6, 5)

Note: n = 5, 6

11.3 Function

Timer/counter 5, 6 have eight operating modes: 8-bit timer, 8-bit event cointer, 8-bit programmable divider output mode, 8-bit pulse width modulation output mode, 16-bit timer, 16-bit event cointer, warm-up counter mode,16-bit pulse width modulation output mode, 16-bit programmable pulse generator output mode.

16-bit timer mode can use timer counter 5 and 6 by cascade connection.

11.3.1 8-bit timer mode (Timer/counter 5, 6)

In this mode, counting up is performed using the internal clock. 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 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 $\overline{\text{PDOj/PWMj/PPGj}}$ pin.

Note 2: In the timer 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 = 5, 6

Table 11-4 Timer/Counter 5, 6 Source Clock (Internal clock)

Source Clock			Resolution		Maximum Time Setting	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc=16 MHz	At fs = 32.768 kHz	At fc=16 MHz	At 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]	32.6 [ms]	62.3 [ms]
fc/2^7	fc/2^7	–	8 [μs]	–	2.0 [ms]	–
fc/2^5	fc/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).

```
LDW     (TTREG6), 0AH         ; Sets the timer register (80 µs ÷ 2^7/fc = 0AH)
DI
SET     (EIRE). 2             ; Enables INTTC6 interrupt
EI
LD      (TC6CR), 00010000B    ; Sets the 8-bit timer mode and source clock (fc/2^7)
LD      (TC6CR), 00011000B    ; Starts TC6
```

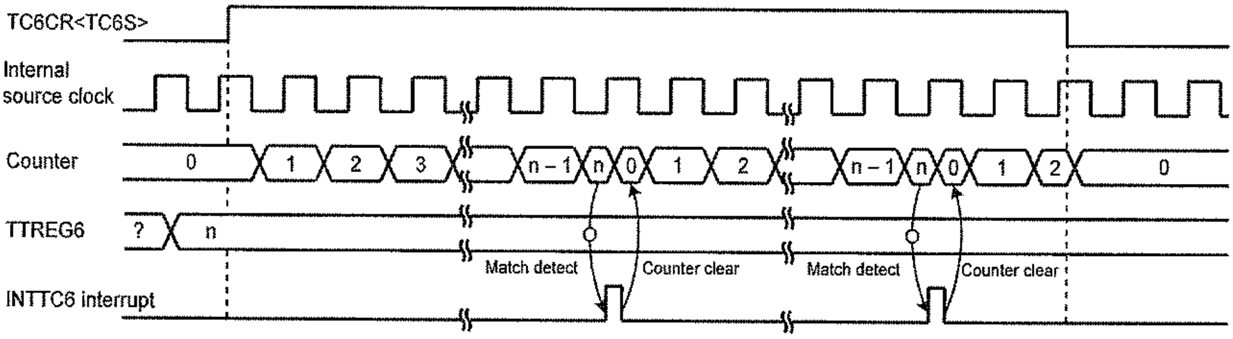


Figure 11-2 8-Bit Timer Mode Timing Chart (in case of timer/counter 6)

11.3.2 8-bit event counter mode (Timer/counter 5, 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 = 5, 6

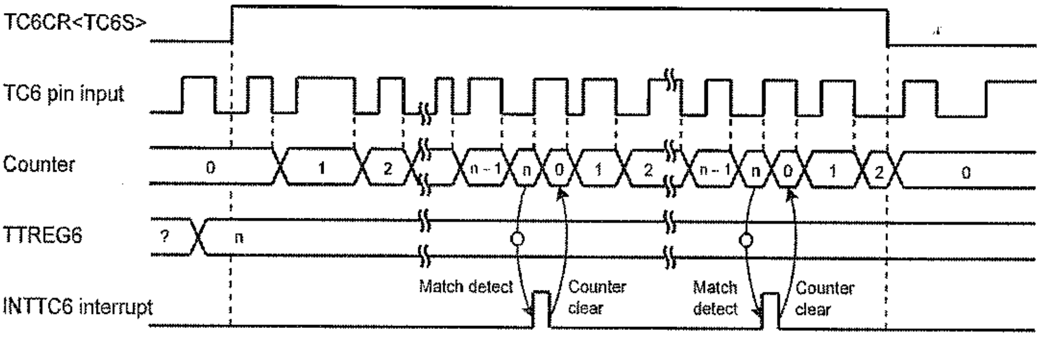


Figure 11-3 Event Counter Mode Timing Chart (in case of timer/counter 6)

11.3.3 8-bit programmable divider output (PDO) mode (Timer/counter 5, 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 $\overline{\text{PDOj}}$ output is toggled.

Example :Output a 1024 Hz pulse (at fc = 16 MHz = "0", in case of TC6)

```
LD      (TTREG6), 3DH             ; (1/1024 ÷ 2^7/fc ) ÷ 2 = 3DH

LD      (TC6CR), 00010001B        ; Sets the 8-bit PDO mode and source clock (fc/2^7)

LD      (TC6CR), 00011001B        ; Starts TC6
```

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: j = 5, 6

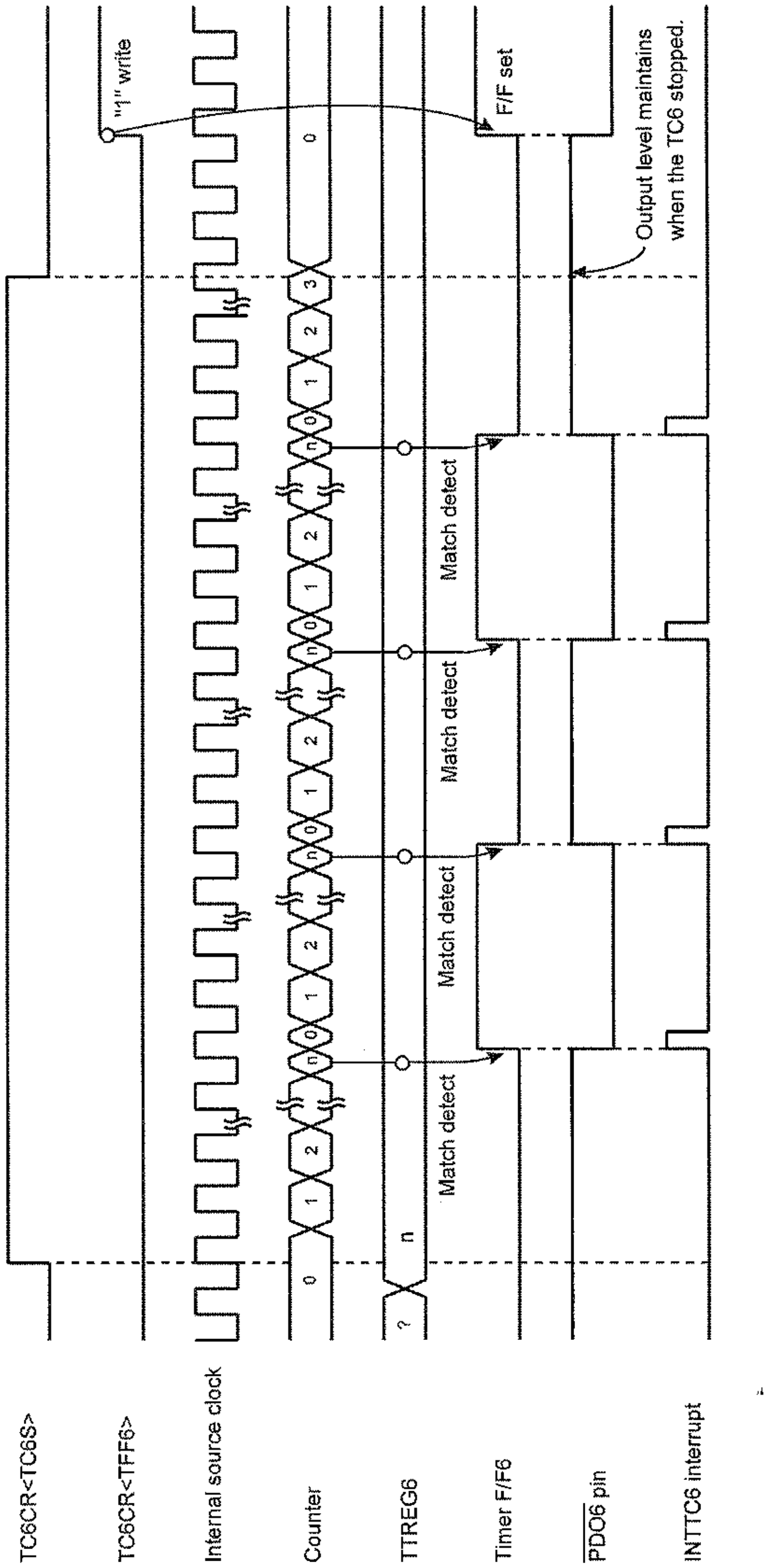


Figure 11-4 8-Bit PDO Mode Timing Chart (in case of timer/counter 6)

11.3.4 8-bit pulse width modulation (PWM) output mode (Timer/counter 5, 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   (TCjCR). 3   ; Stops timer/counter
  CLR   (TCjCR). 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 = 5, 6

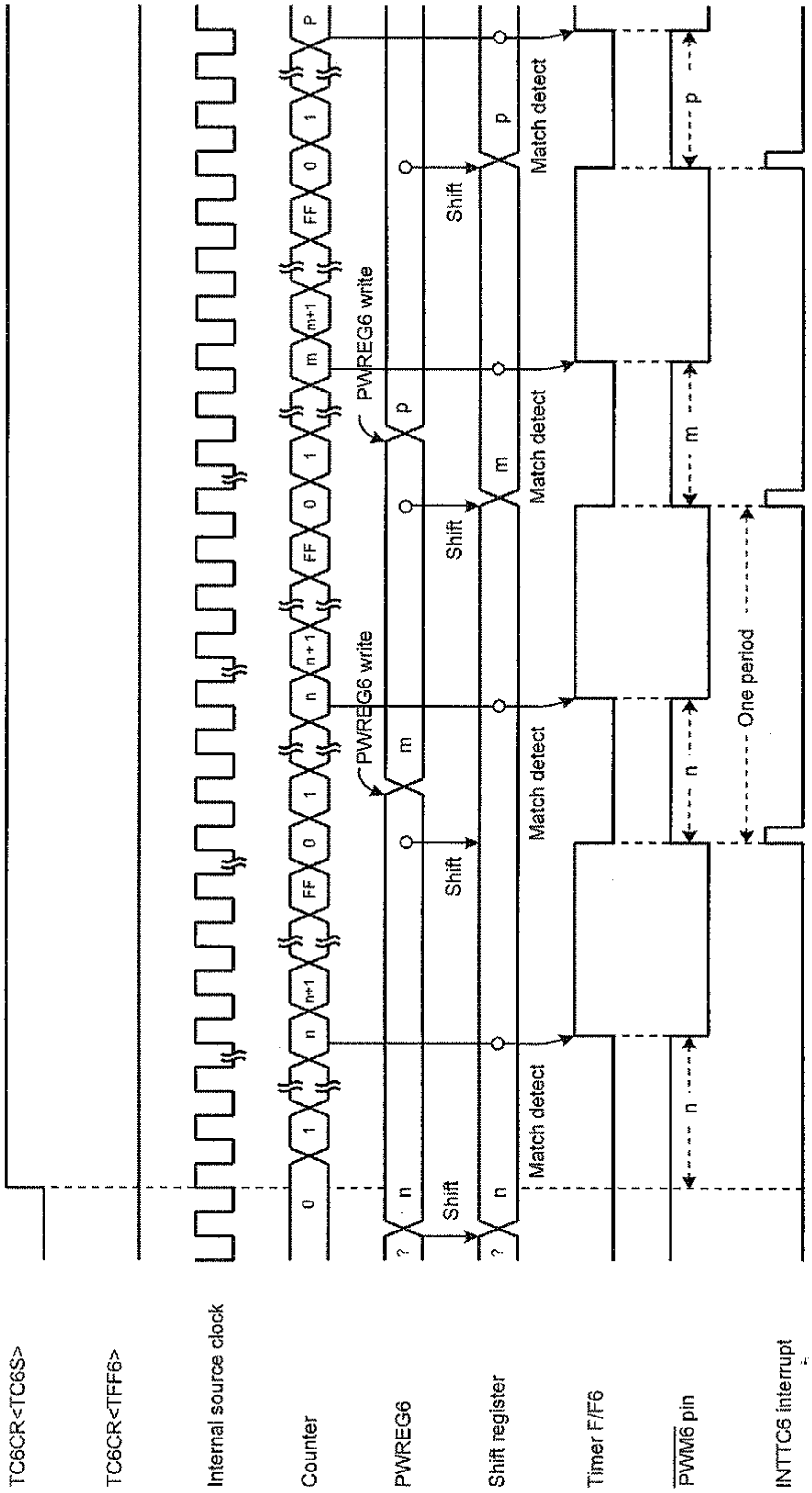


Figure 11-5 8-Bit PWM Mode Timing Chart (in case of timer/counter 6)

Table 11-5 PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At 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}	fc/2^{7}	–	8 [μs]	–	2.05 [ms]	–
fc/2^{5}	fc/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]	7.81 [ms]
fc/2	fc/2	–	125 [ns]	–	32 [μs]	–
fc	fc	–	62.5 [ns]	–	16 [μs]	–

11.3.5 16-bit timer mode (Timer/counter 5 and 6)

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

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

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 $\overline{\text{PDOj/PWMj/PPGj}}$ pin.

Note 2: In the timer 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 = 5, 6

Table 11-6 Source Clock of 16-Bit Timer Mode

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

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     (TTREG5), 927CH     ; Sets the timer register (300 ms ÷ 2^7/fc = 927CH)
DI
SET     (EIRE). 2           ; Enable INTTC6 interrupt
EI
LD      (TC5CR), 13H        ; Sets the 16-bit timer mode (Lower) and source clock
LD      (TC6CR), 04H        ; Sets the 16-bit timer mode (Upper)
LD      (TC6CR), 0CH        ; Starts timer/counter
```

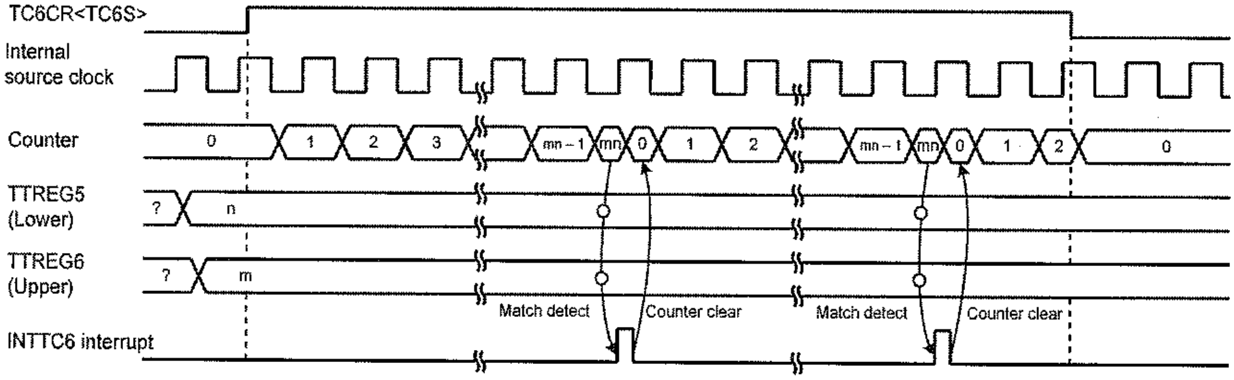


Figure 11-6 16-Bit Timer Mode Timing Chart (in case of timer/counter 5 and 6)

11.3.6 16-bit event counter mode (Timer/counter 5 and 6)

In this mode, event are counted on the falling edge of the TC5 pin input. Timer/counter 5 and 6 are also available as a 16-bit event counter mode by cascade connection.

16-bit event counter mode of timer/counter 5 and 6
If a match is found, the INTTC6 interrupt is generated and the counter is cleared to “0”. After the counter is cleared, counting up resumes every falling edge of TC5 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 TTREG5 more first than TTREG6. The timer register must not write only either TTREG5 or TTREG6.

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 = 5, 6

11.3.7 16-bit pulse width modulation (PWM) output mode (Timer/counter 5 and 6)

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

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 PWM6 pin. An INTTC6 interrupt is generated when an overflow occurs. When used as 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 registers PWREG6, PWREG5 immediately after an INTTC6 interrupt is generated (Normally during the INTTC6 interrupt service routine). If writing to PWREG6, PWREG5 and INTTC6 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 INTTC6 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 PWM6, modify TC6CR<TTF6> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFF6 simultaneously.

Example: Fixes PWM6 output at high level after timer/counter is stopped

```
CLR  (TC6CR). 3  ; Stops timer/counter
CLR  (TC6CR). 7  ; Sets PWM6 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.

Table 11-7 16-Bit PWM Output Mode

Source Clock			Resolution		Maximum Setting Time	
NORMAL1/2 and IDLE1/2 Modes		SLOW1/2 and SLEEP1/2 Modes	At fc = 16 MHz	At fs = 32.768 kHz	At fc = 16 MHz	At 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]
$fc/2^7$	$fc/2^7$	–	8 [μs]	–	524.3 [ms]	–
$fc/2^5$	$fc/2^5$	–	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 PWM6 pin with 16-bit PWM mode (at fc = 16 MHz)

```
LDW   (PWREG5), 07D0H   ; Sets pulse width
LD    (TC5CR), 33H      ; Sets the 16-bit PWM mode (Lower) and source clock (fc/2^3)
LD    (TC6CR), 056H     ; Sets the TFF6 to "1" and sets the 16-bit PWM mode (Upper)
LD    (TC6CR), 05EH     ; Starts timer/counter
```

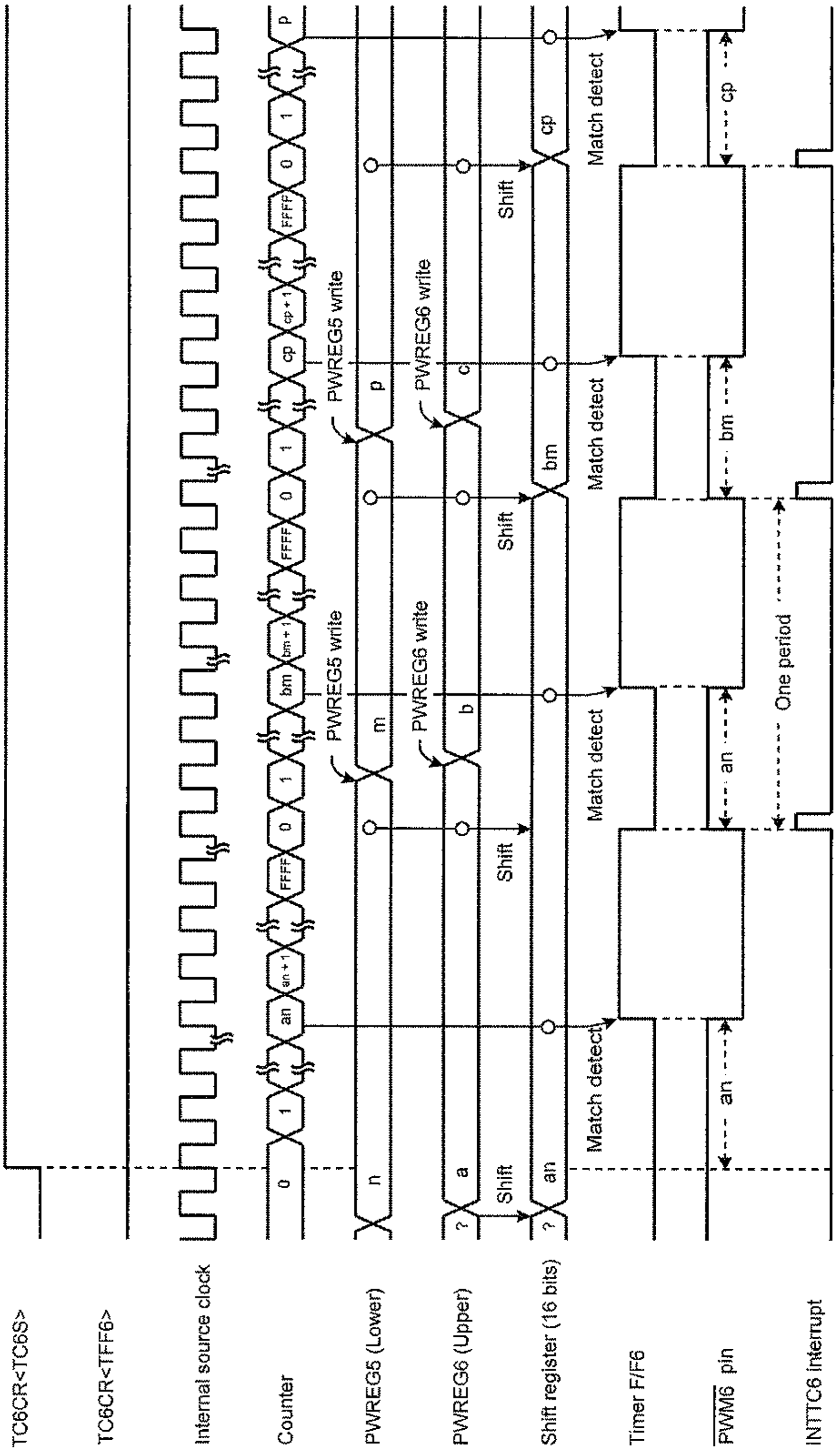


Figure 11-7 16-Bit PWM Mode Timing Chart (in case of timer/counter 5 and 6)

11.3.8 16-bit programmable pulse generate (PPG) output mode (Timer/counter 5 and 6)

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

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 $\overline{\text{PPG6}}$ pin with 16-bit PPG mode (at fc = 16 MHz)

```
LDW     (PWREG5), 07D0H     ; Sets pulse width

LDW     (TTREG5), 8002H     ; Sets pulse term

LD      (TC5CR), 33H        ; Sets the 16-bit PPG mode (Lower) and source clock (fc/2^3)

LD      (TC6CR), 057H       ; Sets the TFF6 to "1" and sets the 16-bit PPG mode (Upper)

LD      (TC6CR), 05FH       ; 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 $\overline{\text{PPG6}}$, modify TC6CR<TTF6> after timer/counter has been stopped. Do not execute halt of timer/counter and modification of TFFj simultaneously.

Example: Fixes $\overline{\text{PPG6}}$ output at high level after timer/counter is stopped

```
CLR   (TC6CR). 3   ; Stops timer/counter
CLR   (TC6CR). 7   ; Sets PPG6 output to high level output
```

Note 3: i = 5, 6

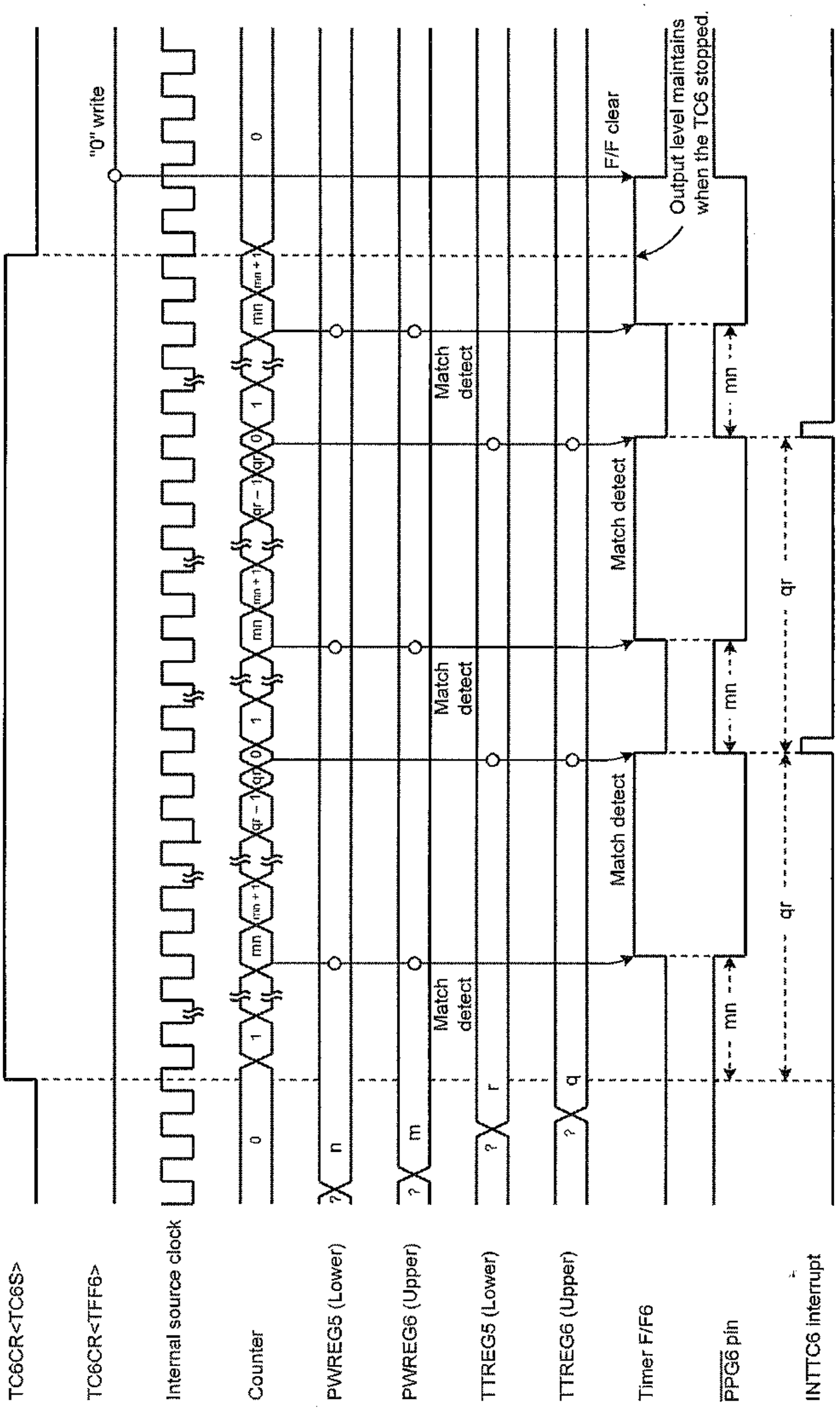


Figure 11-8 16-Bit PPG Mode Timing Chart (in case of timer/counter 5 and 6)

11.3.9 Warm-up counter mode

In this mode, the warm-up period for switching the main system clock can be generated. Timer/counter 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 $\overline{\text{PDOi/PWMi/PPGi}}$ pin.

Note 2: In the warm-up mode, the lower 8 bits of TTREGm, TTREGn are ignored and an interrupt is generated by matching the upper 8 bits.

Note 3: i = 5, 6

11.3.9.1 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, 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 low-frequency clock by setting SYSCR2<SYSCK> to "1".
After that, halt the high-frequency oscillation by clearing SYSCR2<XEN> to "0".

Table 11-8 Warm-up Period for Low-frequency Oscillation (at fs = 32.768 kHz)

Min (at TTREGm, TTREGn = 0100H)	Max (at TTREGm, TTREGn = FF00H)
7.81 ms	1.99 s

Note: m = 6, n = 5

Example :Switching to the SLOW1 mode after low-frequency clock has stabilized by using TC6, 5.

```
            SET     (SYSCR2). 6        ; SYSCR2<XTEN> ← "1"

            LD      (TC5CR), 43H       ; TFF5 = "0",  fs for source clock, sets 16-bit mode

            LD      (TC6CR),  05H      ; TFF6 = "0",  sets warm-up counter mode

            LD      (TTREG5), 8000H    ; Sets warm-up time (Depend on oscillator characteristics)

            DI                         ; IMF ← "0"

            SET     (EIRE). 2          ; Enables INTTC6

            EI                         ; IMF ← "1"

            SET     (TC6CR). 5         ; Starts TC6, 5

             :        :

PINTTC6:    CLR     (TC6CR). 5         ; Stops TC6, 5

            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 oscilla-
                                       tion)

            RETI

             :        :

VINTTC6:    DW      PINTTC6            ; INTTC6 vector table
```

11.3.9.2 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 11-9 Warm-up Period for High-frequency Oscillation (at fc = 16 MHz)

Min (at TTREGm, TTREGn = 0100H)	Max (at TTREGm, TTREGn = FF00H)
16 μs	4.08 ms

Example :Switching to the NORMAL1 mode after high-frequency clock has stabilized by using TC6, 5.

```
            SET     (SYSCR2). 7         ; SYSCR2<XEN> ← "1"

            LD      (TC5CR), 63H        ; TFF5 = "0",  fc for source clock, sets 16-bit mode

            LD      (TC6CR), 05H        ; TFF6 = "0",  sets warm-up counter mode

            LD      (TTREG5), 0F800H    ; Sets warm-up time (Depend on oscillator characteristics)

            DI                          ; IMF ← "0"

            SET     (EIRE). 2           ; Enables INTTC6

            EI                          ; IMF ← "1"

            SET     (TC6CR). 5          ; Starts TC6, 5

             :       :

PINTTC6:    CLR     (TC6CR). 5          ; Stops TC6, 5

            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 oscilla-
                                        tion)

            RETI

             :       :

VINTTC6:    DW      PINTTC6             ; INTTC6 vector table
```

12. UART 1 (Asynchronous serial interface 1)

12.1 Configuration

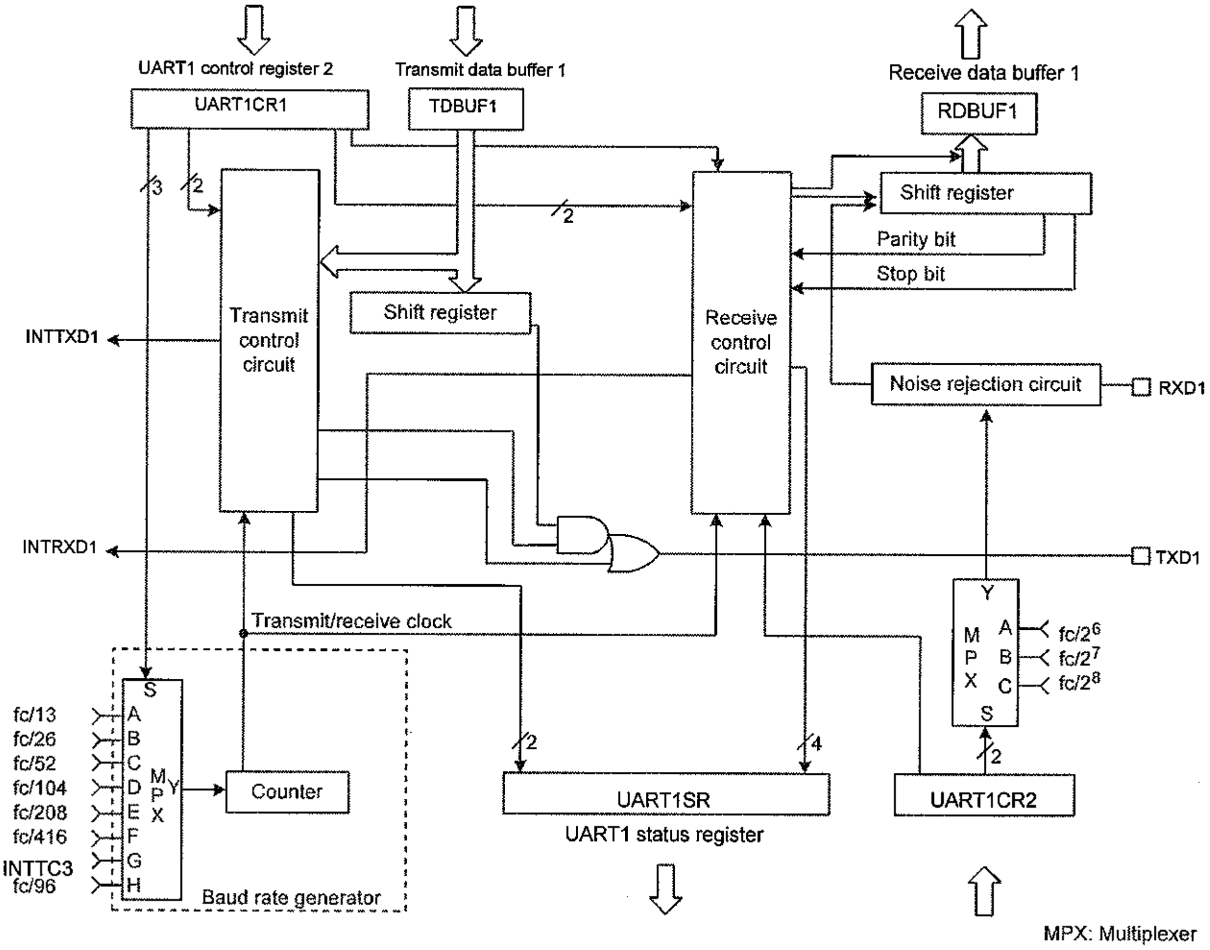


Figure 12-1 UART1

12.2 Control

UART is controlled by the UART control registers (UART1CR1, UART1CR2). The operating status can be monitored using the UART status register (UART1SR).

UART1 Control Registers

UART1CR1	7	6	5	4	3	2	1	0	
(0F95H)	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

TXE	Transfer operation	0: Disable 1: Enable	Write only
RXE	Receive operation	0: Disable 1: Enable	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
PE	Parity addition	0: No parity 1: Parity	
BRG	Transmit clock select	000: fc/13 [Hz] 001: fc/26 010: fc/52 011: fc/104 100: fc/208 101: fc/416 110: TC3 (INTTC3) 111: fc/96	

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: UART1CR1<RXE> and UART1CR1<TXE> should be set to "0" before UART1CR1<BRG> is changed.

UART1CR2	7	6	5	4	3	2	1	0	
(0F96H)						RXDNC		STOPBR	(Initial value: **** *000)

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	Write only
STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits	

Note: When UART1CR2<RXDNC> = "01", pulses longer than 96/fc [s] are always regarded as signals; when UART1CR2<RXDNC> = "10", longer than 192/fc [s]; and when UART1CR2<RXDNC> = "11", longer than 384/fc [s].

UART1 Status Register and Data Buffer Registers

UART1SR (0F95H)	7	6	5	4	3	2	1	0	
	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

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

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

UART1 Receive Data Buffer

RDBUF1 (0F97H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000) Read only

UART1 Transmit Data Buffer

TDBUF1 (0F97H)	7	6	5	4	3	2	1	0	
									(Initial value: 0000 0000) Write only

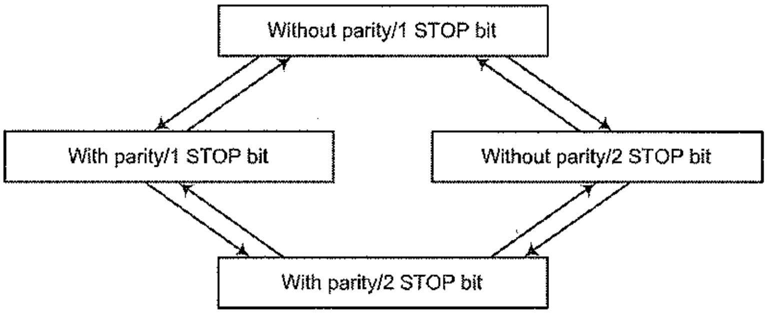
12.3 Transfer Data Format

In UART1, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART1CR1<STBT>), and parity (Select parity in UART1CR1<PE>; even- or odd-numbered parity by UART1CR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.

Table 12-1 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.



12.4 Transfer Rate

The baud rate of UART1 is set of UART1CR1<BRG>. The example of the baud rate are shown as follows.

Table 12-2 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

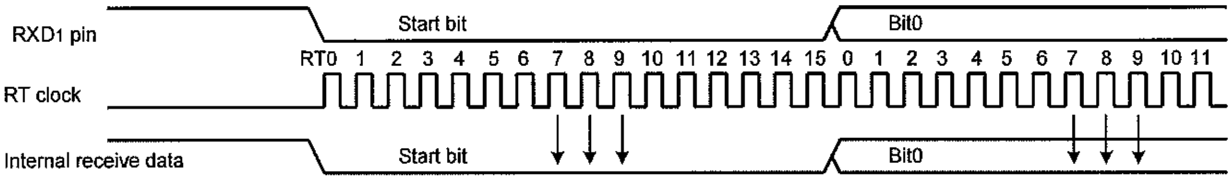
When TC3 is used as the UART1 transfer rate (when UART1CR1<BRG> = "110"), the transfer clock and transfer rate are determined as follows:

$$\text{Transfer clock} = \frac{\text{TC3 source clock}}{\text{TTREG3 set value}}$$

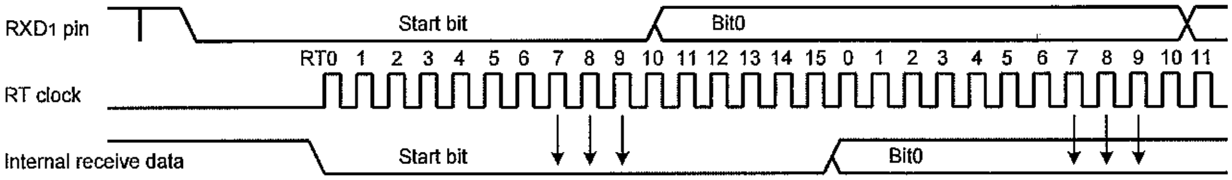
$$\text{Transfer rate} = \frac{\text{Transrer clock}}{16}$$

12.5 Data Sampling

The UART receiver keeps sampling input using the clock selected by UART1CR1<BRG> until a start bit is detected in RXD1 pin input. RT clock starts detecting "L" level of the RXD1 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).


RXD1 pin
Start bit
Bit0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit0

a) Without noise rejection circuit


RXD1 pin
Start bit
Bit0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit0

b) With noise rejection circuit

Figure 12-2 Data Sampling

12.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART1CR1<STBT>.

12.7 Parity

Set parity/no parity by UART1CR1<PE>; set parity type (Odd- or even-numbered) by UART1CR1<EVEN>.

12.8 Transmit/Receive

12.8.1 Data transmit

Set UART1CR1<TXE> to “1”. Read UART1SR to check UART1SR<TBEP> = “1”, then write data in TDBUF1 (Transmit data buffer). Writing data in TDBUF1 zero-clears UART1SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD1 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART1CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UART1CR1. When data transmit starts, transmit buffer empty flag UART1SR<TBEP> is set to “1” and an INTTXD1 interrupt is generated.

While UART1CR1<TXE> = “0” and from when “1” is written to UART1CR1<TXE> to when send data are written to TDBUF1, the TXD1 pin is fixed at high level. When transmitting data, first read UART1SR, then write data in TDBUF1. Otherwise, UART1SR<TBEP> is not zero-cleared and transmit does not start.

12.8.2 Data receive

Set UART1CR1<RXE> to “1”. When data are received via the RXD1 pin, the receive data are transferred to RDBUF1 (Receive data buffer). At this time, the data transmitted includes 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 RDBUF1 (Receive data buffer). Then the receive buffer full flag UART1SR<RBFL> is set and an INTRXD1 interrupt is generated. Select the data transfer baud rate using bits 0 to 2 in UART1CR1.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF1 (Receive data buffer) but discarded; data in the RDBUF1 are not affected.

Note: When a receive operation is disabled by setting UART1CR1<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.

12.9 Status Flag/Interrupt Signal

12.9.1 Parity error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UART1SR<PERR> is set to “1”. The UART1SR<PERR> is cleared to “0” when the RDBUF1 is read after reading the UART1SR.

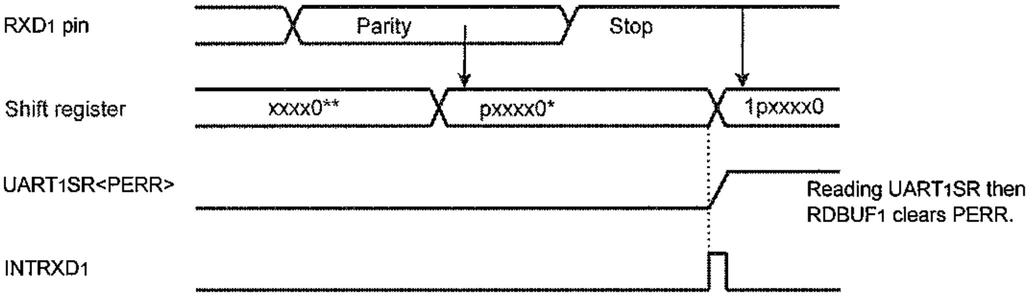


Figure 12-3 Generation of Parity Error

12.9.2 Framing error

When "0" is sampled as the stop bit in the receive data, framing error flag UART1SR<FERR> is set to "1". The UART1SR<FERR> is cleared to "0" when the RDBUF1 is read after reading the UART1SR.

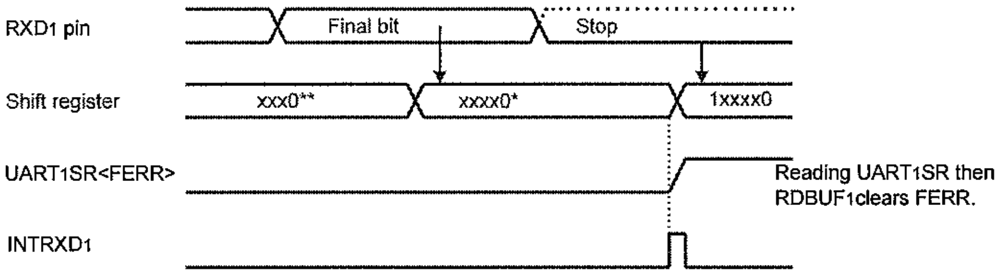


Figure 12-4 Generation of Framing Error

12.9.3 Overrun error

When all bits in the next data are received while unread data are still in RDBUF1, overrun error flag UART1SR<OERR> is set to "1". In this case, the receive data is discarded; data in RDBUF1 are not affected. The UART1SR<OERR> is cleared to "0" when the RDBUF1 is read after reading the UART1SR.

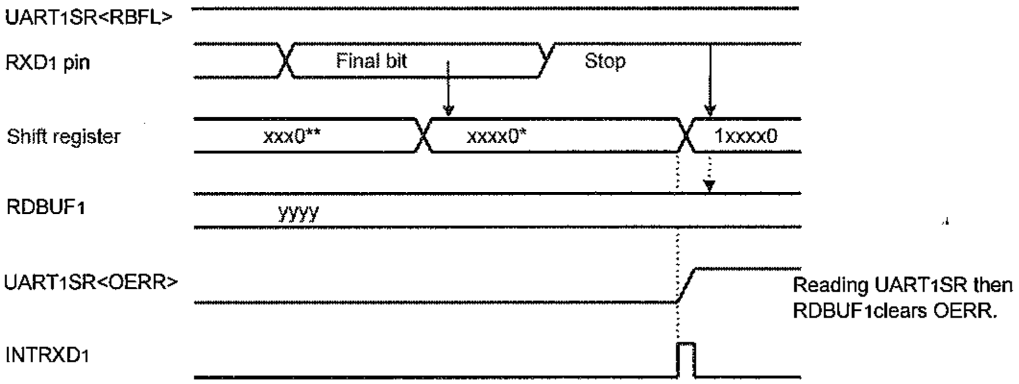


Figure 12-5 Generation of Overrun Error

12.9.4 Receive data buffer full

Loading the received data in RDBUF1 sets receive data buffer full flag UART1SR<RBFL>. The UART1SR<RBFL> is cleared to "0" when the RDBUF1 is read after reading the UART1SR.

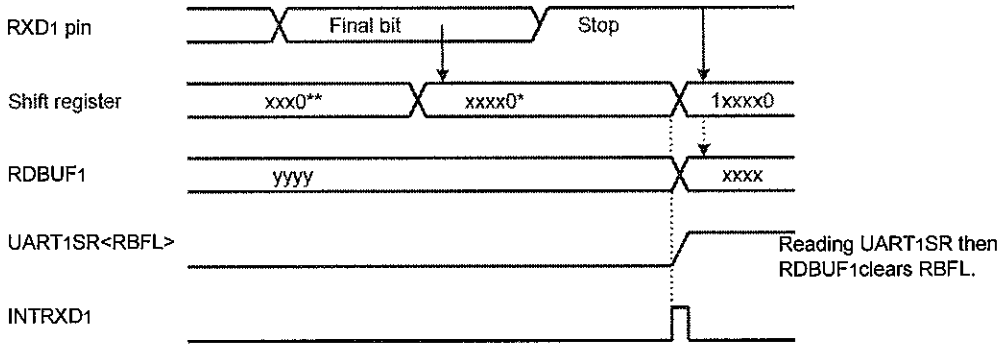


Figure 12-6 Generation of Receive Buffer Full

12.9.5 Transmit data buffer empty

When no data is in the transmit buffer TDBUF1, UART1SR<TBEP> is set to "1", that is, when data in TDBUF1 are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART1SR<TBEP> is set to "1". The UART1SR<TBEP> is cleared to "0" when the TDBUF1 is written after reading the UART1SR.

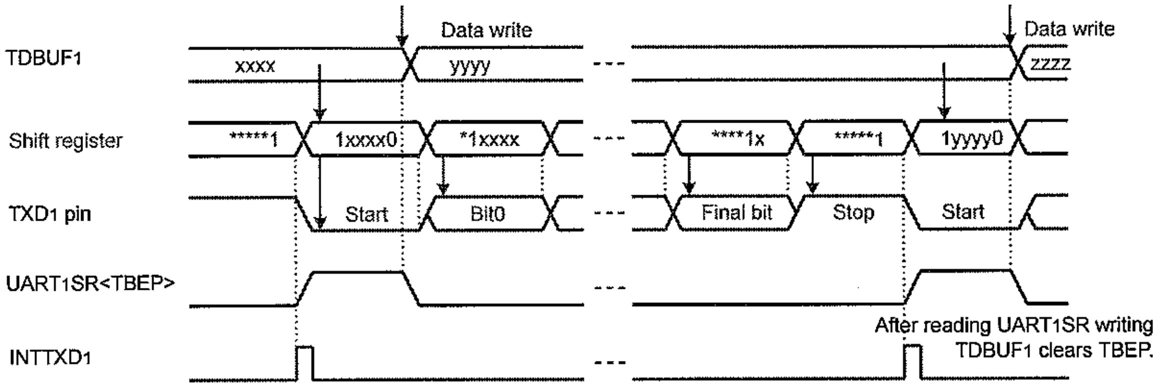


Figure 12-7 Generation of Transmit Buffer Empty

12.9.6 Transmit end flag

When data are transmitted and no data is in TDBUF1 (UART1SR<TBEP> = "1"), transmit end flag UART1SR<TEND> is set to "1". The UART1SR<TEND> is cleared to "0" when the data transmit is stated after writing the TDBUF1.

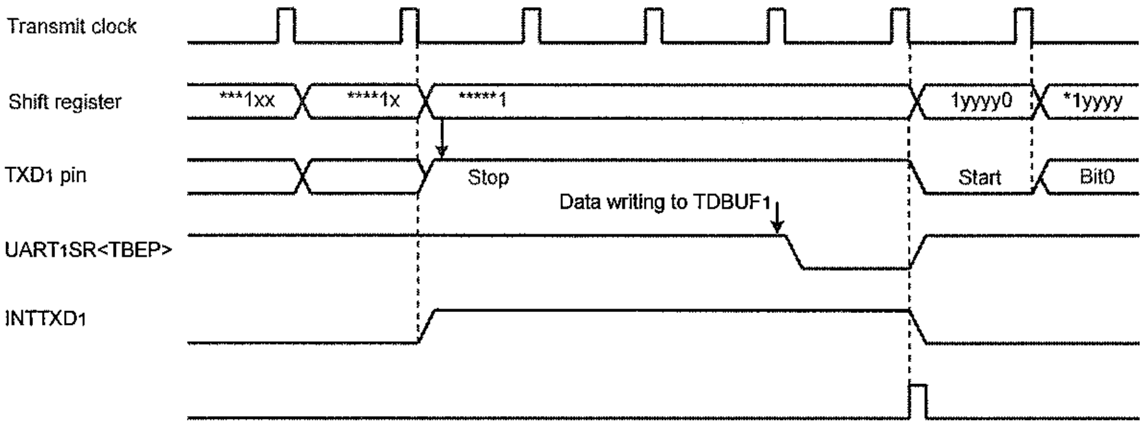


Figure 12-8 Generation of Transmit Buffer Empty

13. UART 2 (Asynchronous serial interface 2)

13.1 Configuration

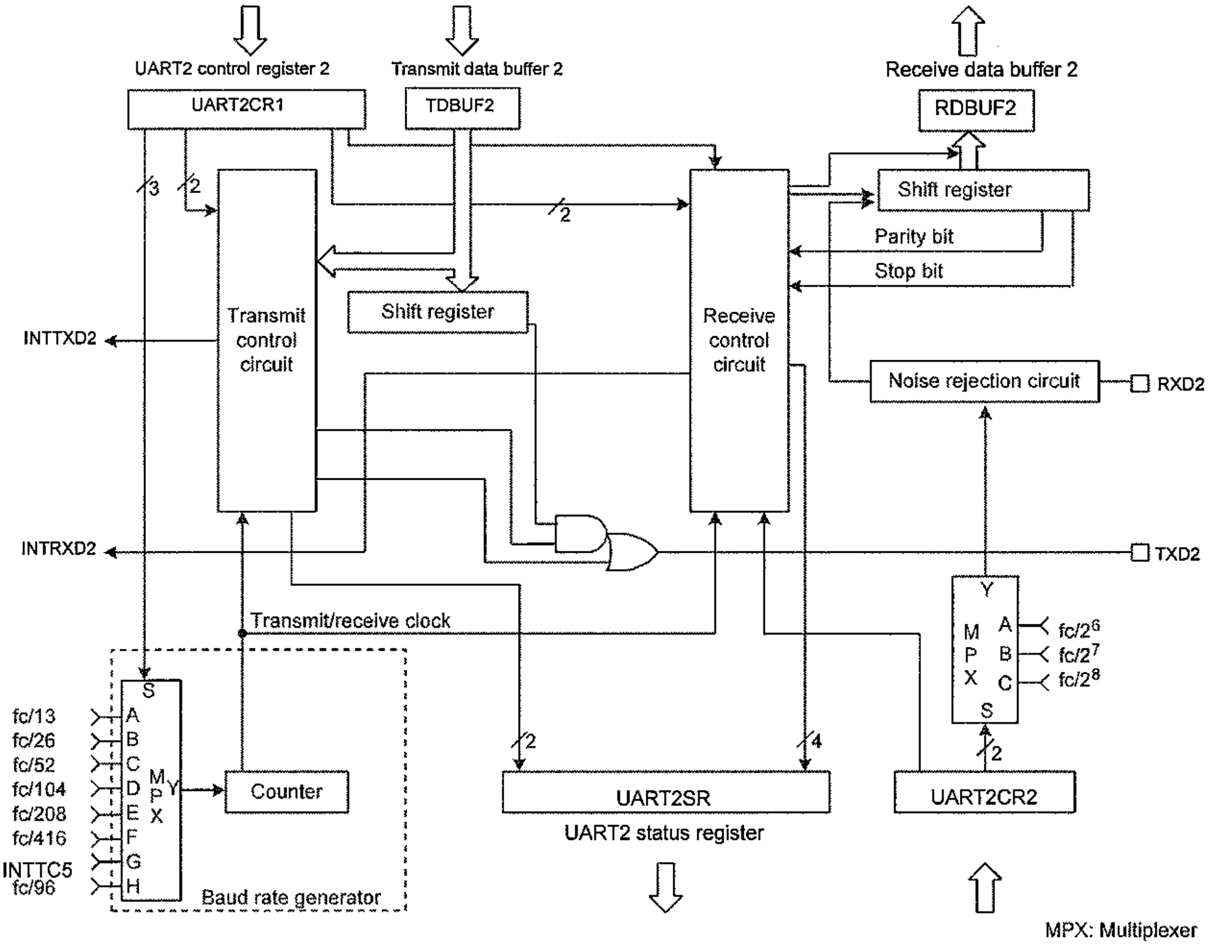


Figure 13-1 UART2

13.2 Control

UART is controlled by the UART control registers (UART2CR1, UART2CR2). The operating status can be monitored using the UART status register (UART2SR).

UART2 Control Registers

UART2CR1 (0F98H)	7	6	5	4	3	2	1	0	
	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value: 0000 0000)

TXE	Transfer operation	0: Disable 1: Enable	Write only
RXE	Receive operation	0: Disable 1: Enable	
STBT	Transmit stop bit length	0: 1 bit 1: 2 bits	
EVEN	Even-numbered parity	0: Odd-numbered parity 1: Even-numbered parity	
PE	Parity addition	0: No parity 1: Parity	
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	

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: UART2CR1<RXE> and UART2CR1<TXE> should be set to "0" before UART2CR1<BRG> is changed.

UART2CR2 (0F99H)	7	6	5	4	3	2	1	0	
						RXDNC		STOPBR	(Initial value: **** *000)

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	Write only
STOPBR	Receive stop bit length	0: 1 bit 1: 2 bits	

Note: When UART2CR2<RXDNC> = "01", pulses longer than 96/fc [s] are always regarded as signals; when UART2CR2<RXDNC> = "10", longer than 192/fc [s]; and when UART2CR2<RXDNC> = "11", longer than 384/fc [s].

UART2 Status Register and Data Buffer Registers

	7	6	5	4	3	2	1	0	
UART2SR (0F98H)	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value: 0000 11**)

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

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

UART2 Receive Data Buffer

	7	6	5	4	3	2	1	0	
RDBUF2 (0F9AH)									(Initial value: 0000 0000) Read only

UART2 Transmit Data Buffer

	7	6	5	4	3	2	1	0	
TDBUF2 (0F9AH)									(Initial value: 0000 0000) Write only

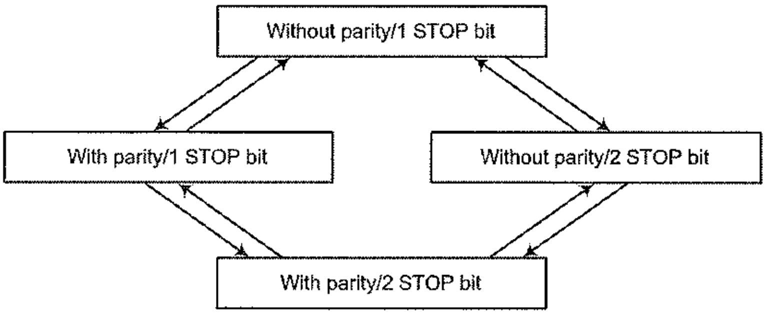
13.3 Transfer Data Format

In UART2, an one-bit start bit (Low level), stop bit (Bit length selectable at high level, by UART2CR1<STBT>), and parity (Select parity in UART2CR1<PE>; even- or odd-numbered parity by UART2CR1<EVEN>) are added to the transfer data. The transfer data formats are shown as follows.

Table 13-1 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.



13.4 Transfer Rate

The baud rate of UART2 is set of UART2CR1<BRG>. The example of the baud rate are shown as follows.

Table 13-2 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

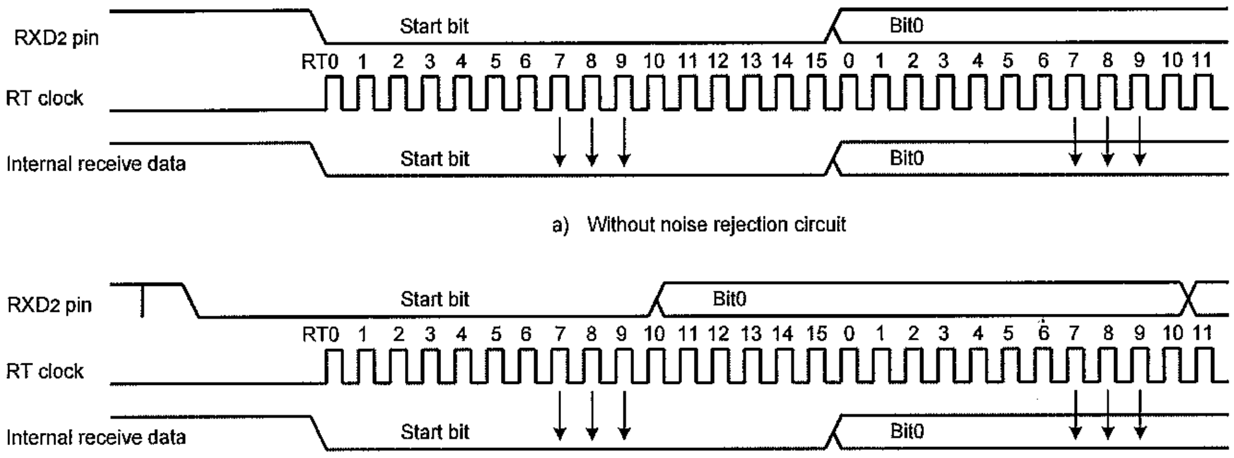
When TC5 is used as the UART2 transfer rate (when UART2CR1<BRG> = "110"), the transfer clock and transfer rate are determined as follows:

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

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

13.5 Data Sampling

The UART receiver keeps sampling input using the clock selected by UART2CR1<BRG> until a start bit is detected in RXD2 pin input. RT clock starts detecting "L" level of the RXD2 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).


RXD2 pin
Start bit
Bit0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit0
RXD2 pin
Start bit
Bit0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
Bit0

a) Without noise rejection circuit

b) With noise rejection circuit

Figure 13-2 Data Sampling

13.6 STOP Bit Length

Select a transmit stop bit length (1 bit or 2 bits) by UART2CR1<STBT>.

13.7 Parity

Set parity/no parity by UART2CR1<PE>; set parity type (Odd- or even-numbered) by UART2CR1<EVEN>.

13.8 Transmit/Receive

13.8.1 Data transmit

Set UART2CR1<TXE> to "1". Read UART2SR to check UART2SR<TBEP> = "1", then write data in TDBUF2 (Transmit data buffer). Writing data in TDBUF2 zero-clears UART2SR<TBEP>, transfers the data to the transmit shift register and the data are sequentially output from the TXD2 pin. The data output include a one-bit start bit, stop bits whose number is specified in UART2CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using bits 0 to 2 in UART2CR1. When data transmit starts, transmit buffer empty flag UART2SR<TBEP> is set to "1" and an INTTXD2 interrupt is generated.

While UART2CR1<TXE> = "0" and from when "1" is written to UART2CR1<TXE> to when send data are written to TDBUF2, the TXD2 pin is fixed at high level. When transmitting data, first read UART2SR, then write data in TDBUF2. Otherwise, UART2SR<TBEP> is not zero-cleared and transmit does not start.

13.8.2 Data receive

Set UART2CR1<RXE> to "1". When data are received via the RXD2 pin, the receive data are transferred to RDBUF2 (Receive data buffer). At this time, the data transmitted includes 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 RDBUF2 (Receive data buffer). Then the receive buffer full flag UART2SR<RBFL> is set and an INTRXD2 interrupt is generated. Select the data transfer baud rate using bits 0 to 2 in UART2CR1.

If an overrun error (OERR) occurs when data are received, the data are not transferred to RDBUF2 (Receive data buffer) but discarded; data in the RDBUF2 are not affected.

Note:When a receive operation is disabled by setting UART2CR1<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.

13.9 Status Flag/Interrupt Signal

13.9.1 Parity error

When parity determined using the receive data bits differs from the received parity bit, the parity error flag UART2SR<PERR> is set to "1". The UART2SR<PERR> is cleared to "0" when the RDBUF2 is read after reading the UART2SR.

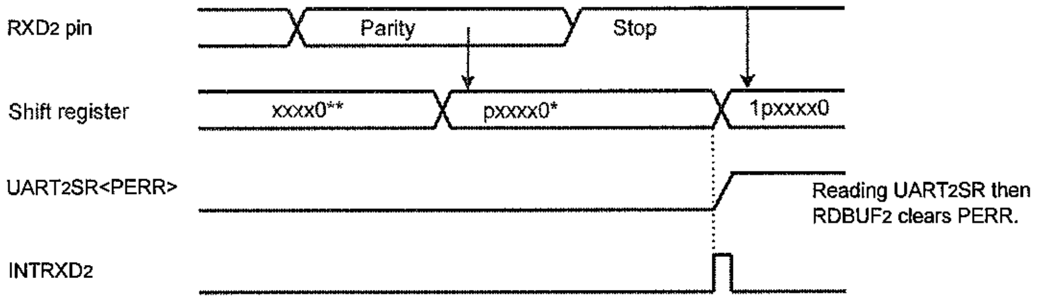


Figure 13-3 Generation of Parity Error

13.9.2 Framing error

When "0" is sampled as the stop bit in the receive data, framing error flag UART2SR<FERR> is set to "1". The UART2SR<FERR> is cleared to "0" when the RDBUF2 is read after reading the UART2SR.

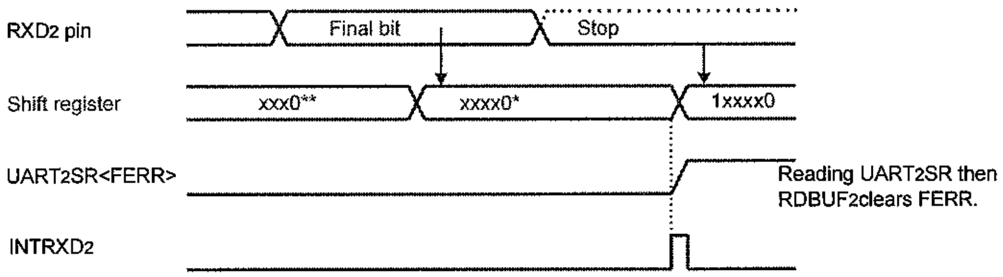


Figure 13-4 Generation of Framing Error

13.9.3 Overrun error

When all bits in the next data are received while unread data are still in RDBUF2, overrun error flag UART2SR<OERR> is set to "1". In this case, the receive data is discarded; data in RDBUF2 are not affected. The UART2SR<OERR> is cleared to "0" when the RDBUF2 is read after reading the UART2SR.

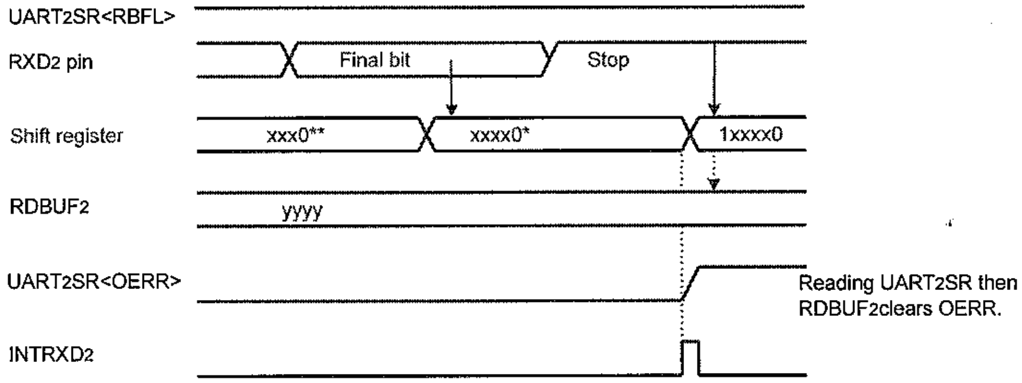


Figure 13-5 Generation of Overrun Error

13.9.4 Receive data buffer full

Loading the received data in RDBUF2 sets receive data buffer full flag UART2SR<RBFL>. The UART2SR<RBFL> is cleared to "0" when the RDBUF2 is read after reading the UART2SR.

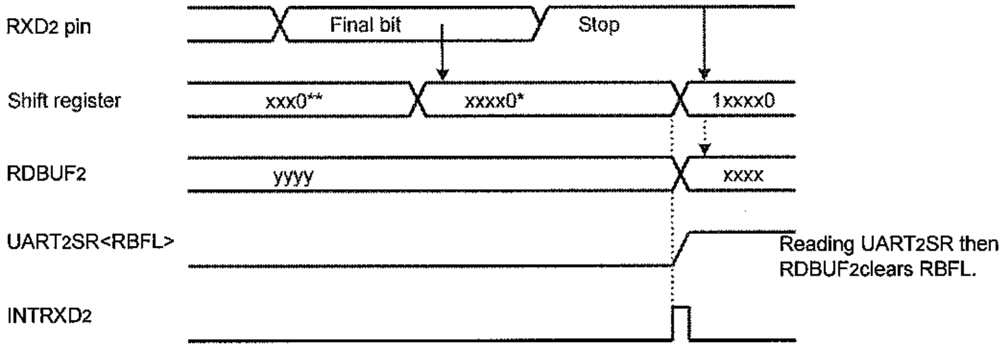


Figure 13-6 Generation of Receive Buffer Full

13.9.5 Transmit data buffer empty

When no data is in the transmit buffer TDBUF2, UART2SR<TBEP> is set to "1", that is, when data in TDBUF2 are transferred to the transmit shift register and data transmit starts, transmit data buffer empty flag UART2SR<TBEP> is set to "1". The UART2SR<TBEP> is cleared to "0" when the TDBUF2 is written after reading the UART2SR.

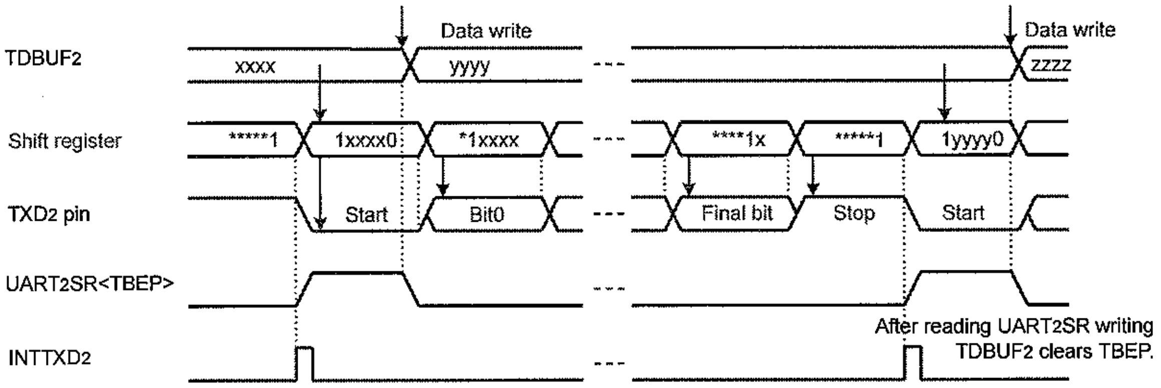


Figure 13-7 Generation of Transmit Buffer Empty

13.9.6 Transmit end flag

When data are transmitted and no data is in TDBUF2 (UART2SR<TBEP> = "1"), transmit end flag UART2SR<TEND> is set to "1". The UART2SR<TEND> is cleared to "0" when the data transmit is stated after writing the TDBUF2.

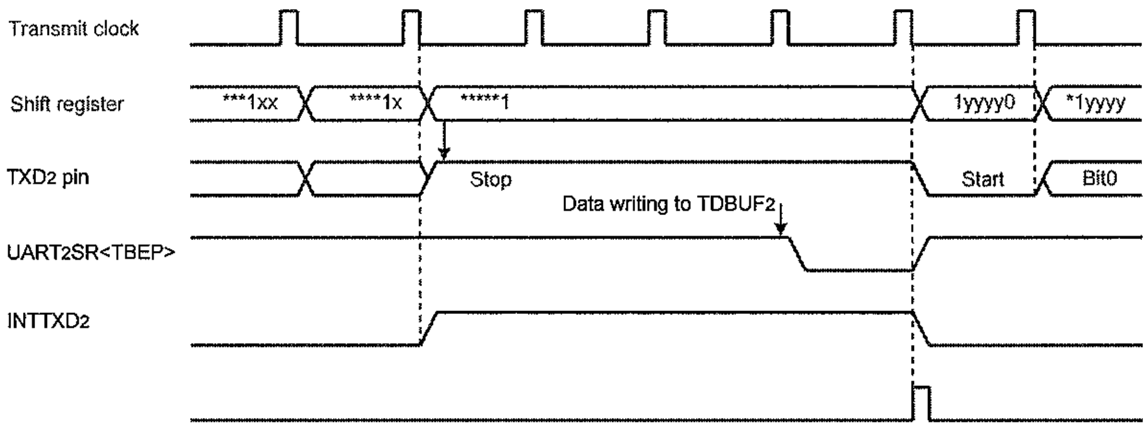


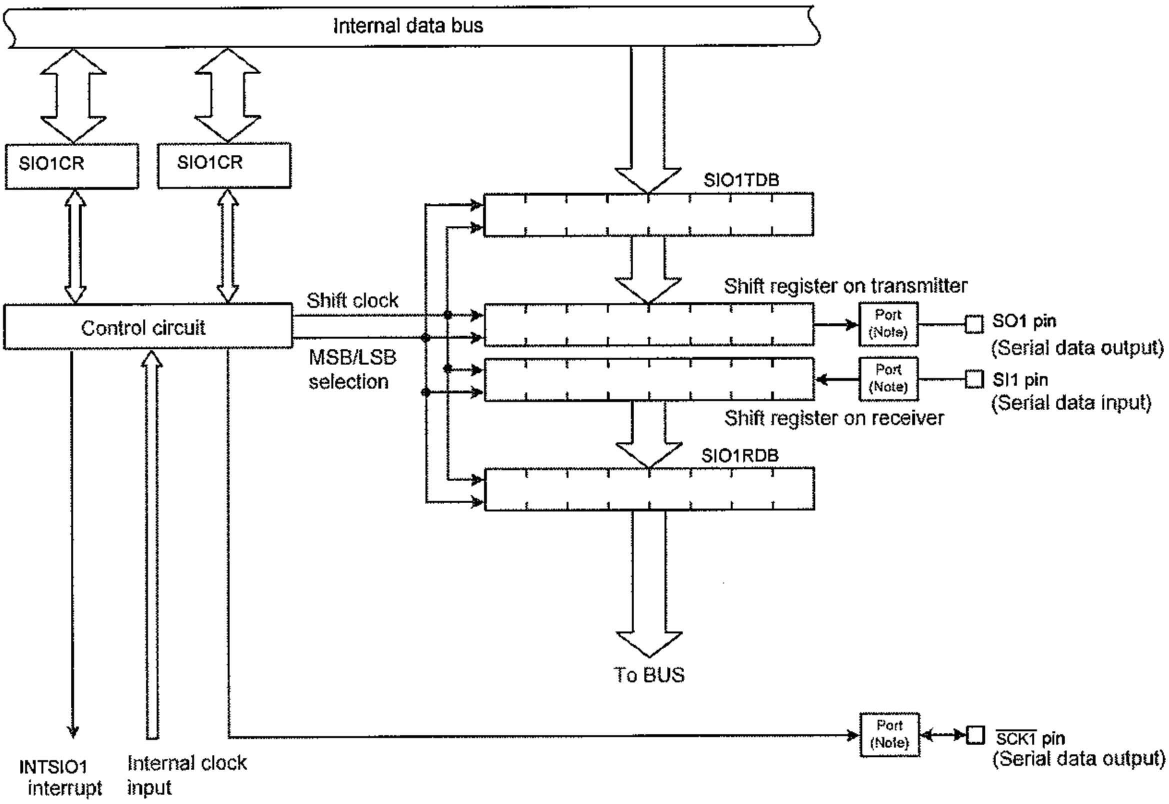
Figure 13-8 Generation of Transmit Buffer Empty

14. HSIO1 (High-Speed Synchronous Serial Interface)

The serial interfaces connect to an external device via SI1, SO1, and $\overline{\text{SCK1}}$ pins.

When these pins are used as serial interface, the output latches for each port.

14.1 Configuration



Note: Set the register of port correctly for the port assigned as serial interface pins.
For details, see the description of the input/output port control register.

Figure 14-1 Serial Interfaces

14.2 Control

The SIO is controlled using the serial interface control register (SIO1CR). The operating status of the serial interface can be inspected by reading the status register (SIO1CR).

Serial Interface Control Register

SIO1CR (0020H)	7	6	5	4	3	2	1	0	
	SIOS	SIOINH	SIOM		SIODIR	SCK			(Initial value: 0000 0000)

SIOS	Specify start/stop of transfer	0: Stop 1: Start				R/W
SIOINH	Forcibly stops transfer (Note 1)	0: – 1: Forcibly stop (Automatically cleared to "0" after stopping)				
SIOM	Selects transfer mode	00: Transmit mode 01: Receive mode 10: Transmit/receive mode 11: Reserved				
SIODIR	Selects direction of transfer	0: MSB (Transfer beginning with bit7) 1: LSB (Transfer beginning with bit0)				
SCK	Selects serial clock		NORMAL1/2 or IDLE1/2 modes		SLOW/SLEEP mode	
			TBTCR <DV7CK> = "0"	TBTCR <DV7CK> = "1"		
		000	fc/2^{12}	fs/2^4	fs/2^4	
		001	fc/2^8	fc/2^8	Reserved	
		010	fc/2^7	fc/2^7	Reserved	
		011	fc/2^6	fc/2^6	Reserved	
		100	fc/2^5	fc/2^5	Reserved	
		101	fc/2^4	fc/2^4	Reserved	
		110	fc/2^3	fc/2^3	Reserved	
		111	External clock (Input from $\overline{\text{SCK1}}$ pin)			

Note 1: When SIO1CR<SIOINH> is set to "1", SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

Note 2: Transfer mode, direction of transfer and serial clock must be select during the transfer is stopping (when SIO1SR<SIOF> "0").

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

Serial Interface Status Register

	7	6	5	4	3	2	1	0	
SIO1SR (0021H)	SIOF	SEF	TXF	RXF	TXERR	RXERR			(Initial value: 0010 00**)

SIOF	Serial transfer operation status monitor	0: Transfer finished 1: Transfer in progress	Read only
SEF	Number of clocks monitor	0: 8 clocks 1: 1 to 7 clocks	
TXF	Transmit buffer empty flag	0: Data exists in transmit buffer 1: No data exists in transmit buffer	
RXF	Receive buffer full flag	0: No data exists in receive buffer 1: Data exists in receive buffer	
TXERR	Transfer operation error flag	Read 0: – (No error exist) 1: Transmit buffer under run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored)	R/W
RXERR	Receive operation error flag	Read 0: – (No error exist) 1: Receive buffer over run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored)	

Note 1: The operation error flag (TXERR and RXERR) are not automatically cleared by stopping transfer with SIO1CR<SIOS> "0". Therefore, set these bits to "0" for clearing these error flag. Or set SIO1CR<SIOINH> to "1".

Note 2: *: Don't care

Receive buffer register

	7	6	5	4	3	2	1	0	Read only
SIO1RDB (0022H)									(Initial value: 0000 0000)

Transmit buffer register

	7	6	5	4	3	2	1	0	Write only
SIO1TDB (0022H)									(Initial value: **** ****)

Note 1: SIO1TDB is write only register. A bit manipulation should not be performed on the transmit buffer register using a read-modify-write instruction.

Note 2: The SIO1TDB should be written after checking SIO1SR<TXF> "1". When SIO1SR<TXF> is "0", the writing data can't be transferred to SIO1TDB even if write instruction is executed to SIO1TDB

Note 3: *: Don't care

14.2.1 Serial clock

14.2.1.1 Clock source

The serial clock can be selected by using SIO1CR<SCK>. When the serial clock is changed, the writing instruction to SIO1CR<SCK> should be executed while the transfer is stopped (when SIO1SR<SIOF> “0”)

(1) Internal clock

Setting the SIO1CR<SCK> to other than “111” outputs the clock (Shown in Table 2.8.2) as serial clock outputs from $\overline{\text{SCK1}}$ pin. At the before beginning or finishing of a transfer, $\overline{\text{SCK1}}$ pin is kept in high level.

When writing (in the transmit mode) or reading (in the receive mode) data can not follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is completed. The maximum time from releasing the automatic-wait function by reading or writing a data is 1 cycle of the selected serial clock until the serial clock comes out from $\overline{\text{SCK1}}$ pin.

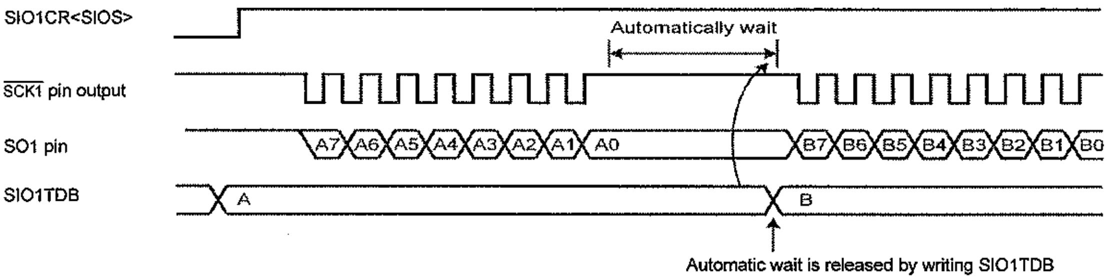


Figure 14-2 Automatic-wait Function (Example of transmit mode)

Table 14-1 Serial Clock Rate (fc = 16 MHz, fs = 32.768kHz)

	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode	
	TBTCR<DV7CK> = "0"		TBTCR<DV7CK> = "1"		Serial Clock	Baud Rate
SCK	Serial Clock	Baud Rate	Serial Clock	Baud Rate		
000	$fc/2^{12}$	3.906 kbps	$fs/2^{4}$	2048 bps	$fs/2^{4}$	2048 bps
001	$fc/2^{8}$	62.5 kbps	$fc/2^{8}$	62.5 kbps	Reserved	-
010	$fc/2^{7}$	125 kbps	$fc/2^{7}$	125 kbps	Reserved	-
011	$fc/2^{6}$	250 kbps	$fc/2^{6}$	250 kbps	Reserved	-
100	$fc/2^{5}$	500 kbps	$fc/2^{5}$	500 kbps	Reserved	-
101	$fc/2^{4}$	1.00 Mbps	$fc/2^{4}$	1.00 Mbps	Reserved	-
110	$fc/2^{3}$	2.00 Mbps	$fc/2^{3}$	2.00 Mbps	Reserved	

(2) External clock

When an external clock is selected by setting SIO1CR<SCK> to "111", the clock via the $\overline{\text{SCK1}}$ pin from an external source is used as the serial clock.

To ensure shift operation, the serial clock pulse width must be 4/fc or more for both "H" and "L" levels.

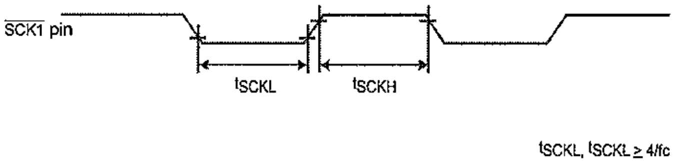


Figure 14-3 External Clock

14.2.1.2 Shift edge

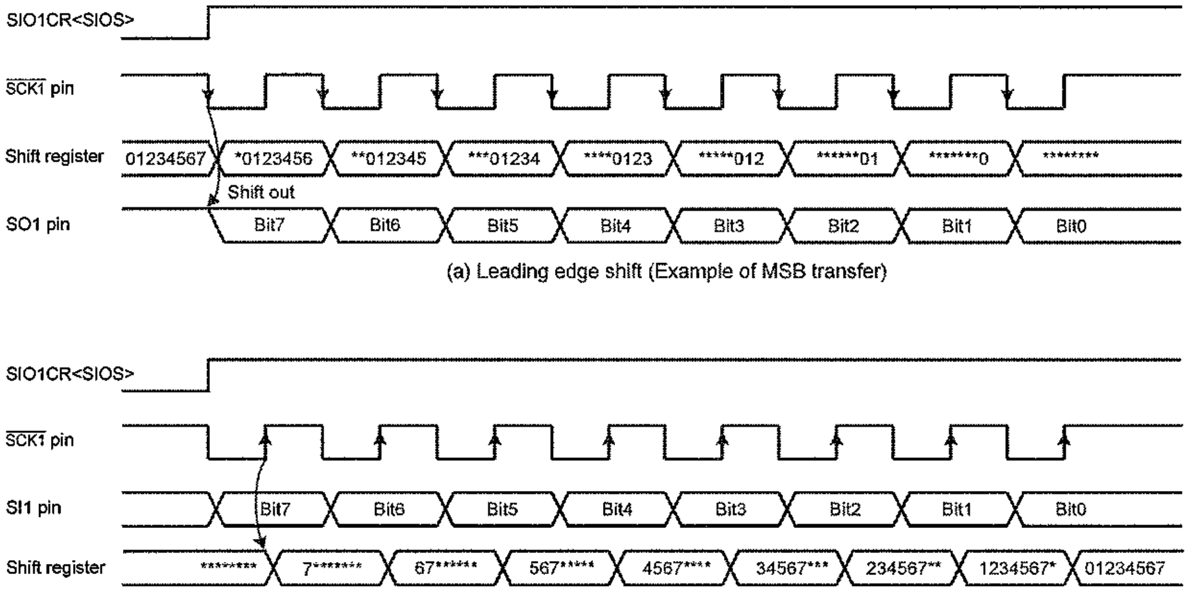
The leading edge is used to transmit data, and the trailing edge is used to receive data.

(1) Leading edge shift

Data is shifted on the leading edge of the serial clock (falling edge of the $\overline{\text{SCK1}}$ pin input/output).

(2) Trailing edge shift

Data is shifted on the trailing edge of the serial clock (rising edge of the $\overline{\text{SCK1}}$ pin input/output).



(a) Leading edge shift (Example of MSB transfer)

(b) Trailing edge shift (Example of MSB transfer)

Figure 14-4 Shift Edge

14.2.2 Transfer bit direction

Transfer data direction can be selected by using SIO1CR<SIODIR>. The transfer data direction can't be set individually for transmit and receive operations.

When the data direction is changed, the writing instruction to SIO1CR<SIODIR> should be executed while the transfer is stopped (when SIO1CR<SIOF> “0”)

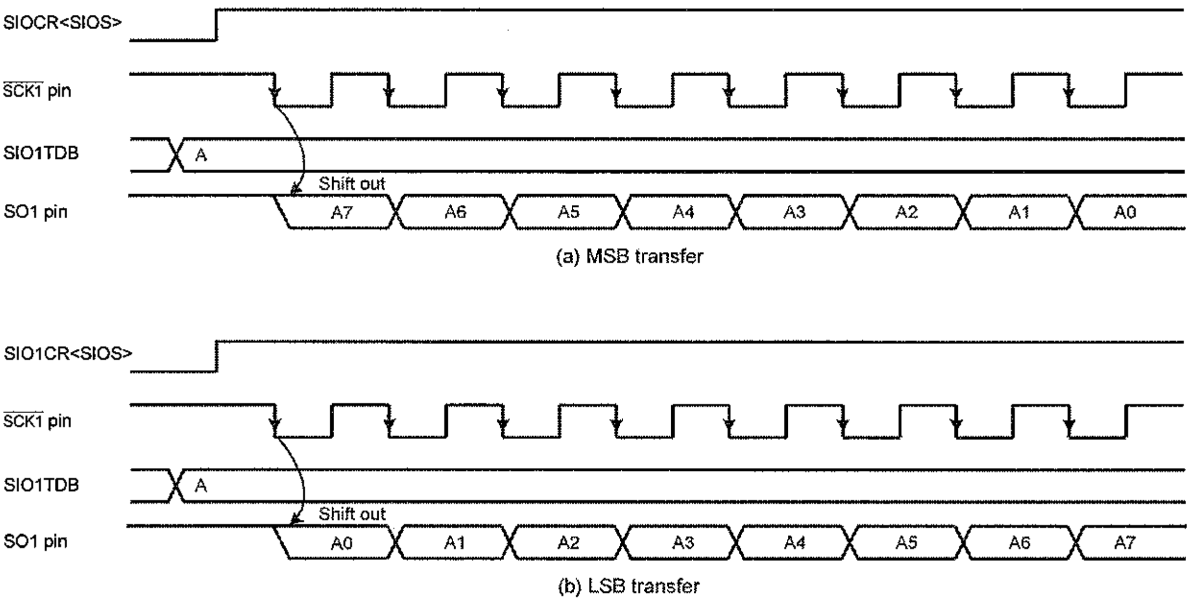


Figure 14-5 Transfer Bit Direction (Example of transmit mode)

14.2.2.1 Transmit mode

(1) MSB transmit mode

MSB transmit mode is selected by setting SIO1CR<SIODIR> to "0", in which case the data is transferred sequentially beginning with the most significant bit (Bit7).

(2) LSB transmit mode

LSB transmit mode is selected by setting SIO1CR<SIODIR> to "1", in which case the data is transferred sequentially beginning with the least significant bit (Bit0).

14.2.2.2 Receive mode

(1) MSB receive mode

MSB receive mode is selected by setting SIO1CR<SIODIR> to "0", in which case the data is received sequentially beginning with the most significant bit (Bit7).

(2) LSB receive mode

LSB receive mode is selected by setting SIO1CR<SIODIR> to "1", in which case the data is received sequentially beginning with the least significant bit (Bit0).

14.2.2.3 Transmit/receive mode

(1) MSB transmit/receive mode

MSB transmit/receive mode are selected by setting SIO1CR<SIODIR> to “0” in which case the data is transferred sequentially beginning with the most significant bit (Bit7) and the data is received sequentially beginning with the most significant (Bit7).

(2) LSB transmit/receive mode

LSB transmit/receive mode are selected by setting SIO1CR<SIODIR> to “1”, in which case the data is transferred sequentially beginning with the least significant bit (Bit0) and the data is received sequentially beginning with the least significant (Bit0).

14.2.3 Transfer modes

Transmit, receive and transmit/receive mode are selected by using SIO1CR<SIOM>.

14.2.3.1 Transmit mode

Transmit mode is selected by writing “00” to SIO1CR<SIOM>.

(1) Starting the transmit operation

Transmit mode is selected by setting “00” to SIO1CR<SIOM>. Serial clock is selected by using SIO1CR<SCK>. Transfer direction is selected by using SIO1CR<SIODIR>.

When a transmit data is written to the transmit buffer register (SIO1TDB), SIO1SR<TXF> is cleared to “0”.

After SIO1CR<SIOS> is set to “1”, SIO1SR<SIOF> is set synchronously to “1” the falling edge of $\overline{\text{SCK1}}$ pin.

The data is transferred sequentially starting from SO1 pin with the direction of the bit specified by SIO1CR<SIODIR>, synchronizing with the $\overline{\text{SCK1}}$ pin's falling edge.

SIO1SR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK1}}$ pin and eighth clock falling edge.

SIO1SR<TXF> is set to “1” at the rising edge of pin after the data written to the SIO1TDB is transferred to shift register, then the INTSIO1 interrupt request is generated, synchronizing with the next falling edge on $\overline{\text{SCK1}}$ pin.

Note 1: In internal clock operation, when SIO1CR<SIOS> is set to "1", transfer mode does not start without writing a transmit data to the transmit buffer register (SIO1TDB).

Note 2: In internal clock operation, when the SIO1CR<SIOS> is set to "1", SIO1TDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK1}}$ pin.

Note 3: In external clock operation, when the falling edge is input from $\overline{\text{SCK1}}$ pin after SIO1CR<SIOS> is set to "1", SIO1TDB is transferred to shift register immediately.

(2) During the transmit operation

When data is written to SIO1TDB, SIO1SR<TXF> is cleared to “0”.

In internal clock operation, in case a next transmit data is not written to SIO1TDB, the serial clock stops to “H” level by an automatic-wait function when all of the bit set in the SIO1TDB has been transmitted. Automatic-wait function is released by writing a transmit data to SIO1TDB. Then, transmit operation is restarted after maximum 1-cycle of serial clock.

When the next data is written to the SIO1TDB before termination of previous 8-bit data with SIO1SR<TXF> "1", the next data is continuously transferred after transmission of previous data.

In external clock operation, after SIO1SR<TXF> is set to "1", the transmit data must be written to SIO1TDB before the shift operation of the next data begins.

If the transmit data is not written to SIO1TDB, transmit error occurs immediately after shift operation is started. Then, INTSIO1 interrupt request is generated after SIO1SR<TXERR> is set to "1".

(3) Stopping the transmit operation

There are two ways for stopping transmits operation.

- The way of clearing SIO1CR<SIOS>.
  When SIO1CR<SIOS> is cleared to "0", transmit operation is stopped after all transfer of the data is finished. When transmit operation is finished, SIO1SR<SIOF> is cleared to "0" and SO1 pin is kept in high level.
  In external clock operation, SIO1CR<SIOS> must be cleared to "0" before SIO1SR<SEF> is set to "1" by beginning next transfer.
- The way of setting SIO1CR<SIOINH>.
  Transmit operation is stopped immediately after SIO1CR<SIOINH> is set to "1". In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

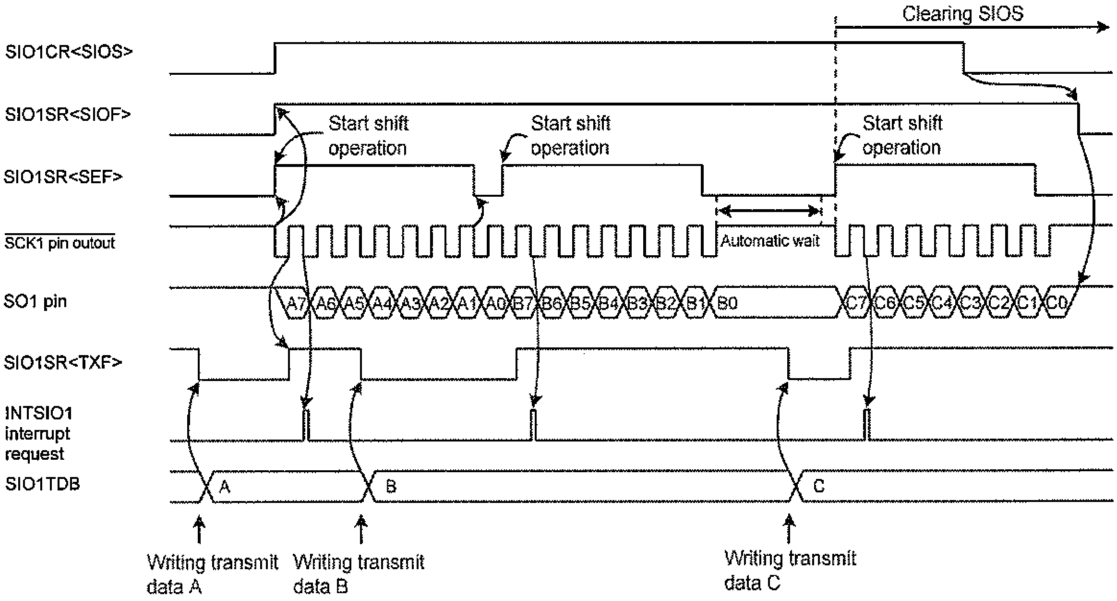


Figure 14-6 Example of Internal Clock and MSB Transmit Mode

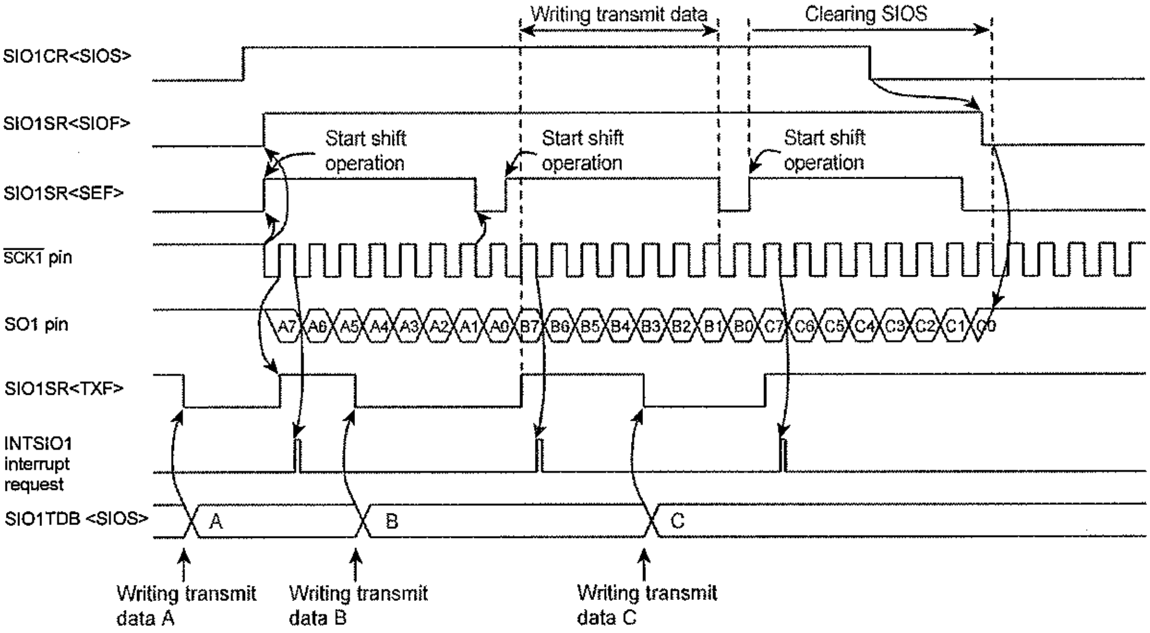
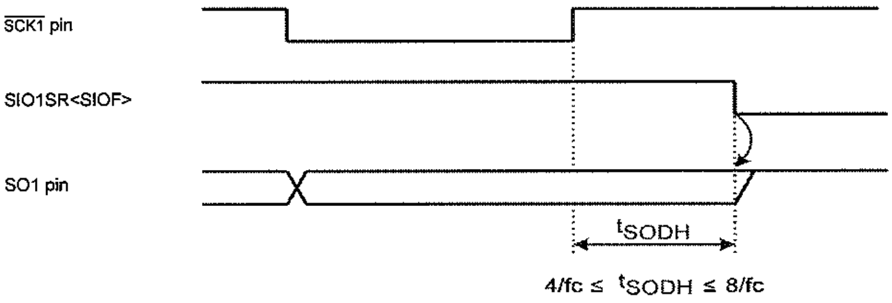


Figure 14-7 Exaple of External Clock and MSB Transmit Mode



(4) Transmit error processing

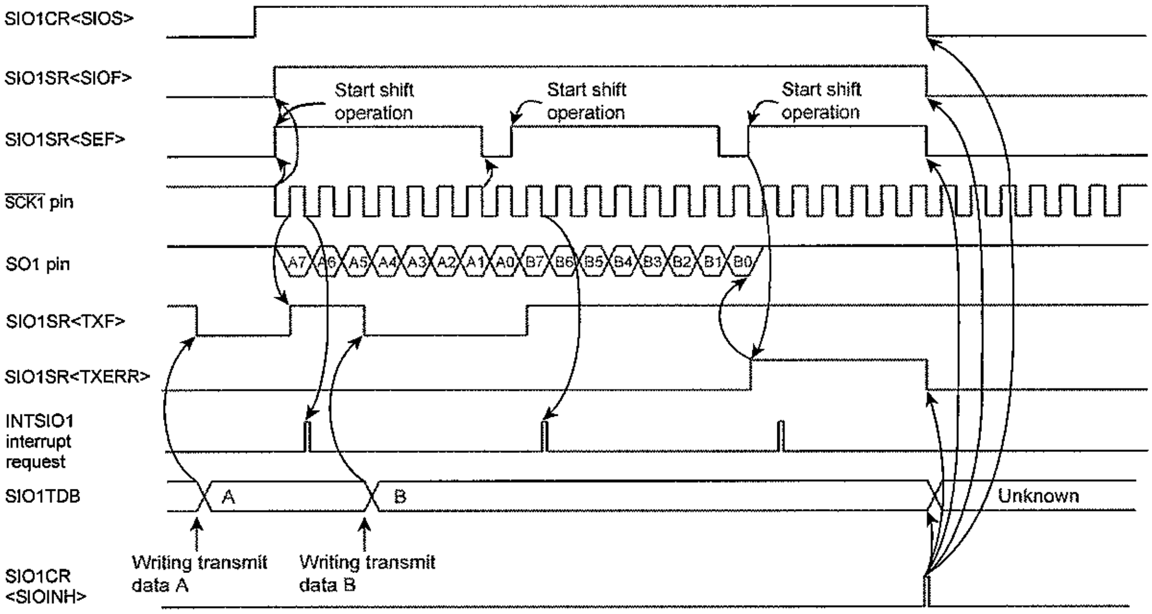
Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIO1TDB in external clock operation.

  If transmit errors occur during transmit operation, SIO1SR<TXERR> is set to "1" immediately after starting shift operation. Synchronizing with the next serial clock falling edge, INTSIO1 interrupt request is generated.

  If shift operation starts before writing data to SIO1TDB after SIO1CR<SIOS> is set to "1", SIO1SR<TXERR> is set to "1" immediately after shift operation is started and then INTSIO1 interrupt request is generated.

  SIO1 pin is kept in high level when SIO1SR<TXERR> is set to "1". When transmit error occurs, transmit operation must be forcibly stop by writing SIO1CR<SIOINH> to "1". In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.



14.2.3.2 Receive mode

The receive mode is selected by writing “01” to SIO1CR<SIOM>.

(1) Starting the receive operation

Receive mode is selected by setting “01” to SIO1CR<SIOM>. Serial clock is selected by using SIO1CR<SCK>. Transfer direction is selected by using SIO1CR<SIODIR>.

After SIO1CR<SIOS> is set to “1”, SIO1SR<SIOF> is set synchronously to “1” the falling edge of SCK1 pin.

Synchronizing with the SCK1 pin's rising edge, the data is received sequentially from SI1 pin with the direction of the bit specified by SBI1DIR<SIODIR>.

SIO1SR<SEF> is kept in high level, between the first clock falling edge of SCK1 pin and eighth clock falling edge.

When 8-bit data is received, the data is transferred to SIO1RDB from shift register. INTSIO1 interrupt request is generated and SIO1SR<RXF> is set to “1”

Note: In internal clock operation, when the SIO1CR<SIOS> is set to "1", the serial clock is generated from SCK1 pin after maximum 1-cycle of serial clock frequency.

(2) During the receive operation

The SIO1SR<RXF> is cleared to “0” by reading a data from SIO1RDB.

In the internal clock operation, the serial clock stops to “H” level by an automatic-wait function when the all of the 8-bit data has been received. Automatic-wait function is released by reading a received data from SIO1RDB. Then, receive operation is restarted after maximum 1-cycle of serial clock.

In external clock operation, after SIO1SR<RXF> is set to “1”, the received data must be read from SIO1RDB, before the next data shift-in operation is finished.

If received data is not read out from SIO1RDB receive error occurs immediately after shift operation is finished. Then INTSIO1 interrupt request is generated after SIO1SR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIO1CR<SIOS>.
  When SIO1CR<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIO1SR<SIOF> is cleared to "0". In external clock operation, SIO1CR<SIOS> must be cleared to "0" before SIO1SR<SEF> is set to "1" by starting the next shift operation.
- The way of setting SIO1CR<SIOINH>.
  Receive operation is stopped immediately after SIO1CR<SIOINH> is set to "1". In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

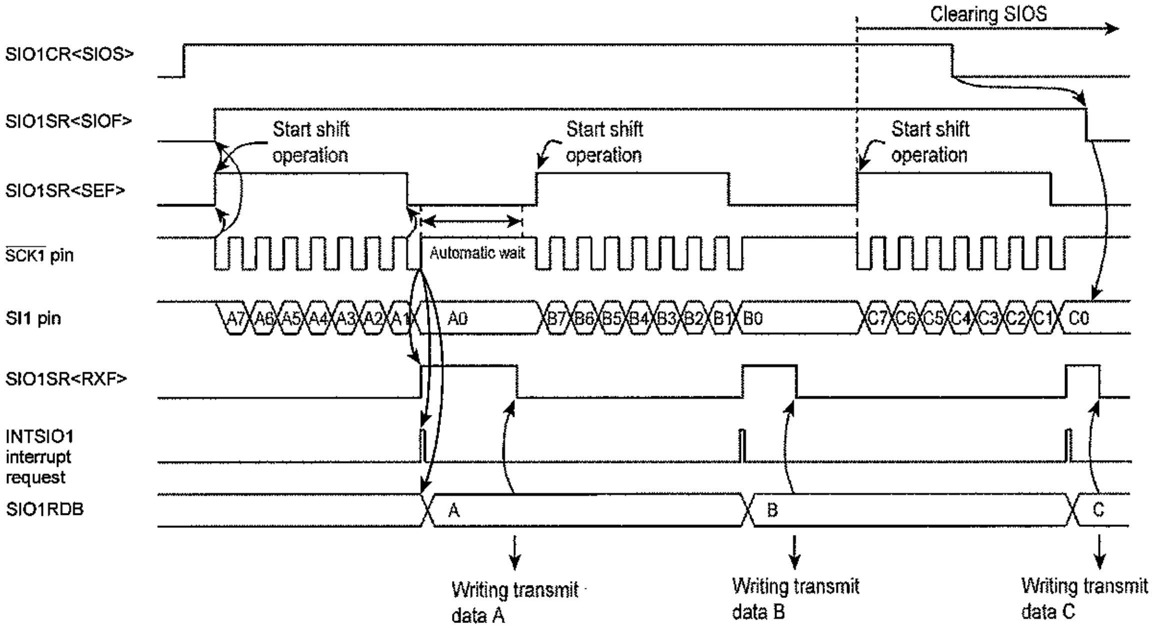


Figure 14-8 Example of Internal Clock and MSB Receive Mode

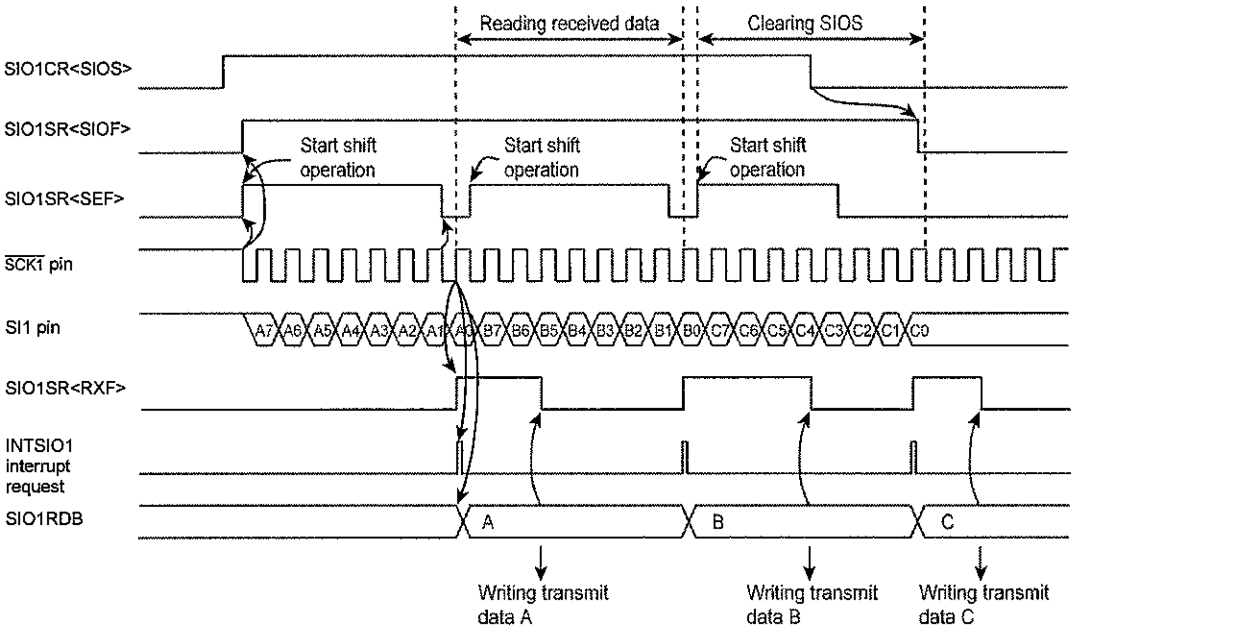


Figure 14-9 Example of External Clock and MSB Receive Mode

(4) Receive error processing

Receive errors occur on the following situation. To protect SIO1RDB and the shift register contents, the received data is ignored while the SIO1SR<RXERR> is “1”.

- Shift operation is finished before reading out received data from SIO1RDB at SIO1SR<RXF> is “1” in an external clock operation.
  If receive error occurs, set the SIO1CR<SIOS> to “0” for reading the data that received immediately before error occurence. And read the data from SIO1RDB. Data in shift register (at errors occur) can be read by reading the SIO1RDB again.
  When SIO1SR<RXERR> is cleared to “0” after reading the received data, SIO1SR<RXF> is cleared to “0”.
  After clearing SIO1CR<SIOS> to “0”, when 8-bit serial clock is input to SCK1 pin, receive operation is stopped. To restart the receive operation, confirm that SIO1SR<SIOF> is cleared to “0”.
  If the receive error occurs, set the SIO1CR<SIOINH> to “1” for stopping the receive operation immediately. In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

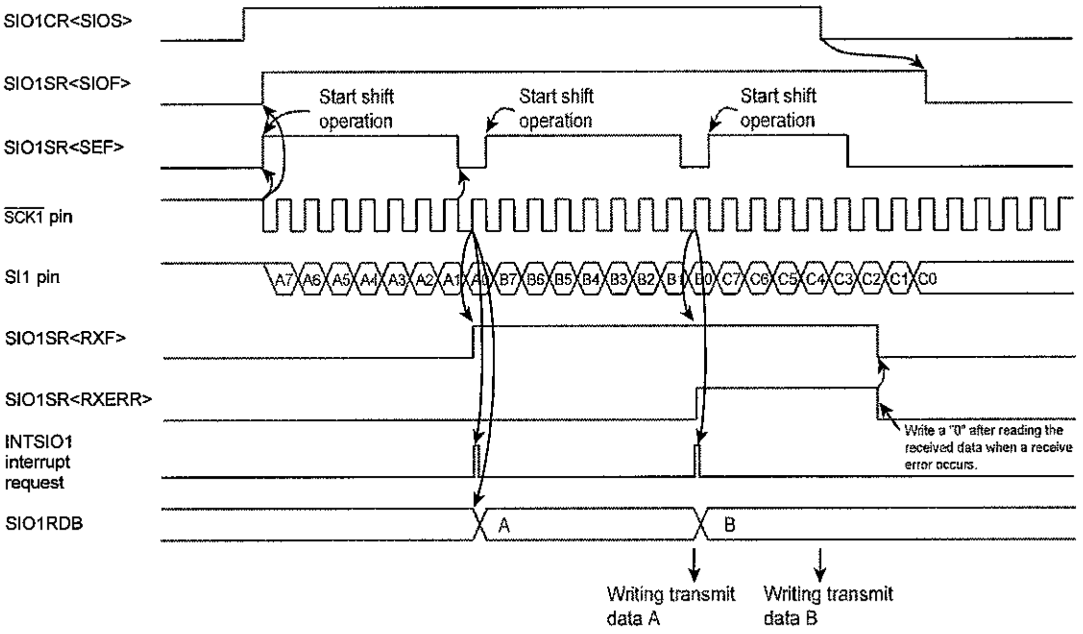

SIO1CR<SIOS>
SIO1SR<SIOF>
SIO1SR<SEF>
$\overline{\text{SCK1}}$ pin
SI1 pin
SIO1SR<RXF>
SIO1SR<RXERR>
INTSIO1 interrupt request
SIO1RDB
Start shift operation
Start shift operation
Start shift operation
A7 A6 A5 A4 A3 A2 A1 A0 B7 B6 B5 B4 B3 B2 B1 B0 C7 C6 C5 C4 C3 C2 C1 C0
Write a "0" after reading the received data when a receive error occurs.
A
B
Writing transmit data A
Writing transmit data B

Figure 14-10 Example of Receive Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

14.2.3.3 Transmit/receive mode

The transmit/receive mode are selected by writing "10" to SIO1CR<SIOM>.

(1) Starting the transmit/receive operation

Transmit/receive mode is selected by writing "10" to SIO1CR<SIOM>. Serial clock is selected by using SIO1CR<SCK>. Transfer direction is selected by using SIO1CR<SIODIR>.

When a transmit data is written to the transmit buffer register (SIO1TDB), SIO1SR<TXF> is cleared to "0".

After SIO1CR<SIOS> is set to "1", SIO1SR<SIOF> is set synchronously to the falling edge of $\overline{\text{SCK1}}$ pin.

The data is transferred sequentially starting from SO1 pin with the direction of the bit specified by SIO1CR<SIODIR>, synchronizing with the $\overline{\text{SCK1}}$ pin's falling edge. And receiving operation also starts with the direction of the bit specified by SIO1CR<SIODIR>, synchronizing with the $\overline{\text{SCK1}}$ pin's rising edge.

SIO1SR<SEF> is kept in high level between the first clock falling edge of $\overline{\text{SCK1}}$ pin and eighth clock falling edge.

SIO1SR<TXF> is set to "1" at the rising edge of $\overline{\text{SCK1}}$ pin after the data written to the SIO1TDB is transferred to shift register. When 8-bit data has been received, the received data is transferred to SIO1RDB from shift register, then the INTSIO1 interrupt request occurs, synchronizing with setting SIO1SR<RXF> to "1".

Note 1: In internal clock operation, when the SIO1CR<SIOS> is set to "1", SIO1TDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK1}}$ pin.

Note 2: In external clock operation, when the falling edge is input from $\overline{\text{SCK1}}$ pin after SIO1CR<SIOS> is set to "1", SIO1TDB is transferred to shift register immediately. When the rising edge is input from $\overline{\text{SCK1}}$ pin, receive operation also starts.

(2) During the transmit/receive operation

When data is written to SIO1TDB, SIO1SR<TXF> is cleared to "0" and when a data is read from SIO1RDB, SIO1SR<RXF> is cleared to "0".

In internal clock operation, in case of the condition described below, the serial clock stops to "H" level by an automatic-wait function when all of the bit set in the data has been transmitted.

- Next transmit data is not written to SIO1TDB after reading a received data from SIO1RDB.
- Received data is not read from SIO1RDB after writing a next transmit data to SIO1TDB.
- Neither SIO1TDB nor SIO1RDB is accessed after transmission.

The automatic wait function is released by writing the next transmit data to SIO1TDB after reading the received data from SIO1RDB, or reading the received data from SIO1RDB after writing the next data to SIO1TDB.
Then, transmit/receive operation is restarted after maximum 1 cycle of serial clock.
In external clock operation, reading the received data from SIO1RDB and writing the next data to SIO1TDB must be finished before the shift operation of the next data begins.
If the transmit data is not written to SIO1TDB after SIO1SR<TXF> is set to "1", transmit error occurs immediately after shift operation is started. When the transmit error occurred, SIO1SR<TXERR> is set to "1".
If received data is not read out from SIO1RDB before next shift operation starts after setting SIO1SR<RXF> to "1", receive error occurs immediately after shift operation is finished. When the receive error has occurred, SIO1SR<RXERR> is set to "1".

(3) Stopping the transmit/receive operation

There are two ways for stopping the transmit/receive operation.

- The way of clearing SIO1CR<SIOS>.
  When SIO1CR<SIOS> is cleared to "0", transmit/receive operation is stopped after all transfer of the data is finished. When transmit/receive operation is finished, SIO1SR<SIOF> is cleared to "0" and SO1 pin is kept in high level.
  In external clock operation, SIO1CR<SIOS> must be cleared to "0" before SIO1SR<SEF> is set to "1" by beginning next transfer.
- The way of setting SIO1CR<SIOINH>.
  Transmit/receive operation is stopped immediately after SIO1CR<SIOINH> is set to "1". In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

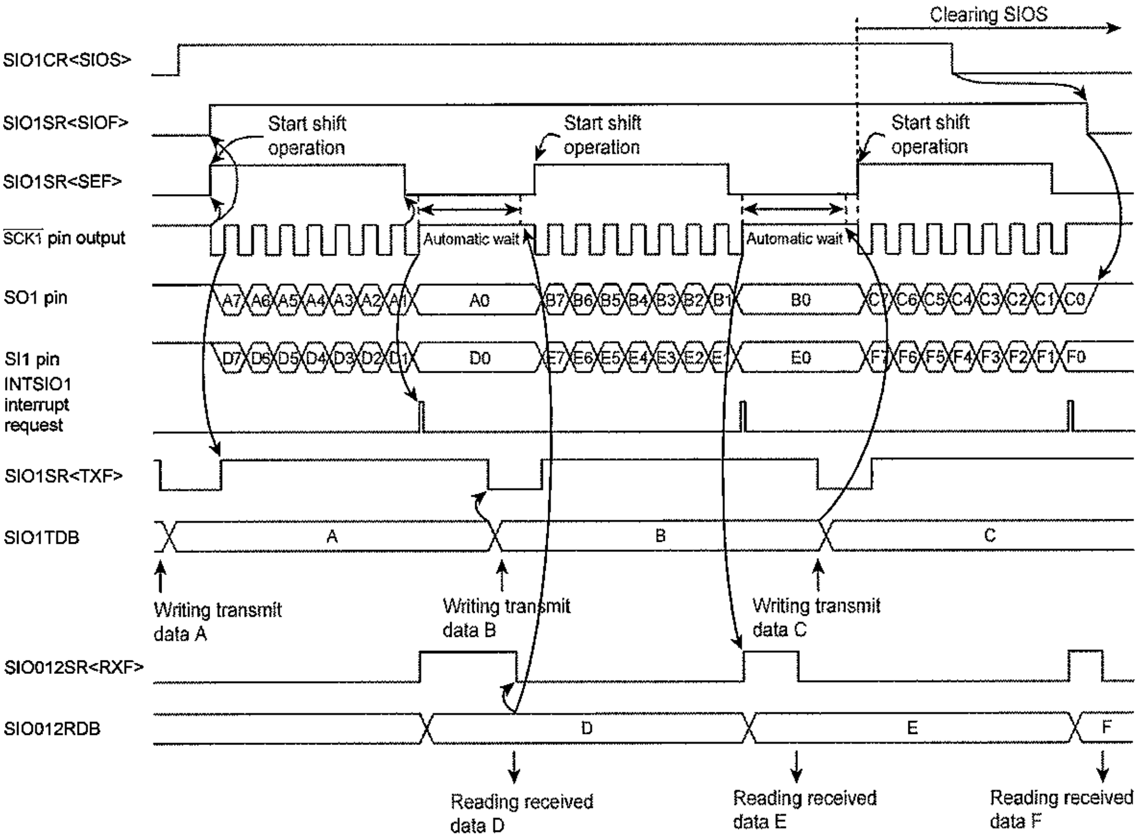


Figure 14-11 Example of Internal Clock and MSB Transmit/Receive Mode

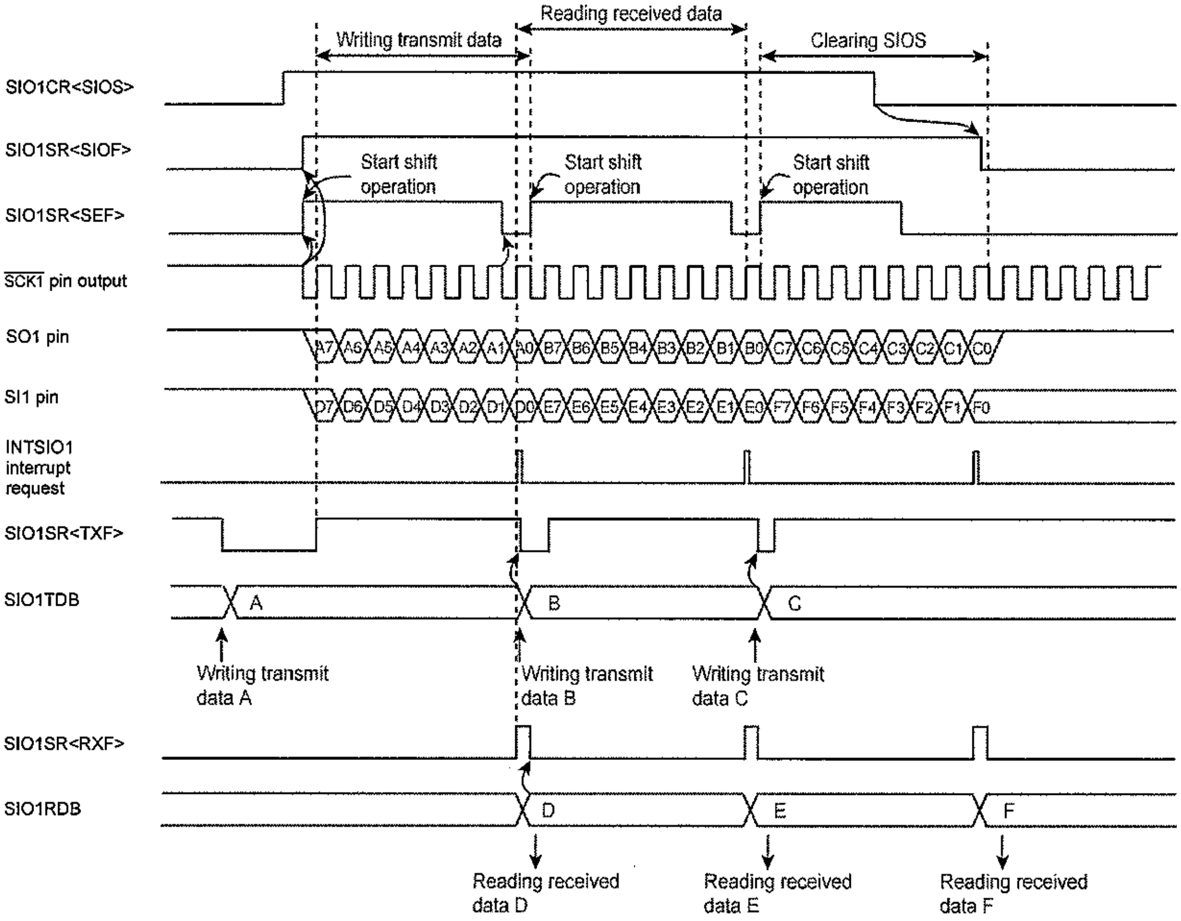


Figure 14-12 Example of External Clock and MSB Transmit/Receive Mode

(4) Transmit/receive error processing

Transmit/receive errors occur on the following situation. Corrective action is different, which errors occur transmits or receives.

(a) Transmit errors

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIO1TDB in external clock operation.
  If transmit errors occur during transmit operation, SIO1SR<TXERR> is set to "1" immediately after starting shift operation. And INTSIO1 interrupt request is generated after all of the 8-bit data has been received.
  If shift operation starts before writing data to SIO1TDB after SIO1CR<SIOS> is set to "1", SIO1SR<TXERR> is set immediately after starting shift operation. And INTSIO1 interrupt request is generated after all of the 8-bit data has been received.
  SO1 pin is kept in high level when SIO1SR<TXERR> is set to "1". When transmit error occurs, transmit operation must be forcibly stop by writing SIO1CR<SIOINH> to "1" after the received data is read from SIO1RDB. In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

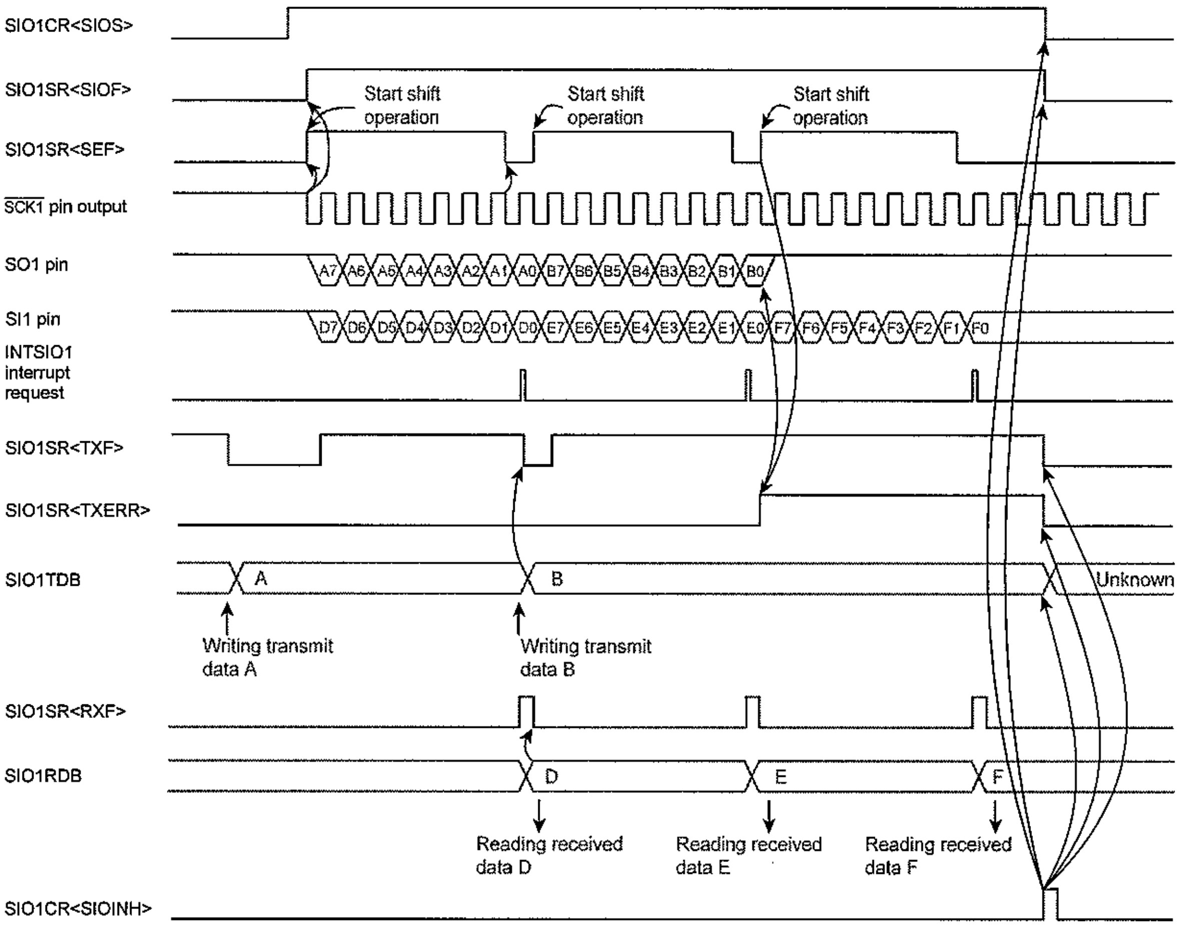


Figure 14-13 Example of Transmit/Receive (Transmit) Error Processing

(b) Receive errors

Receive errors occur on the following situation. To protect SIO1RDB and the shift register contents, the received data is ignored while the SIO1SR<RXERR> is "1".

- Shift operation is finished before reading out received data from SIO1RDB at SIO1SR<RXF> is "1" in an external clock operation.
  If receive error occurs, set the SIO1CR<SIOS> to "0" for reading the data that received immediately before error occurence. And read the data from SIO1RDB. Data in shift register (at errors occur) can be read by reading the SIO1RDB again.
  When SIO1SR<RXERR> is cleared to "0" after reading the received data, SIO1SR<RXF> is cleared to "0".
  After clearing SIO1CR<SIOS> to "0", when 8-bit serial clock is input to SCK1 pin, receive operation is stopped. To restart the receive operation, confirm that SIO1SR<SIOF> is cleared to "0".
  If the received error occurs, set the SIO1CR<SIOINH> to "1" for stopping the receive operation immediately. In this case, SIO1CR<SIOS>, SIO1SR register, SIO1RDB register and SIO1TDB register are initialized.

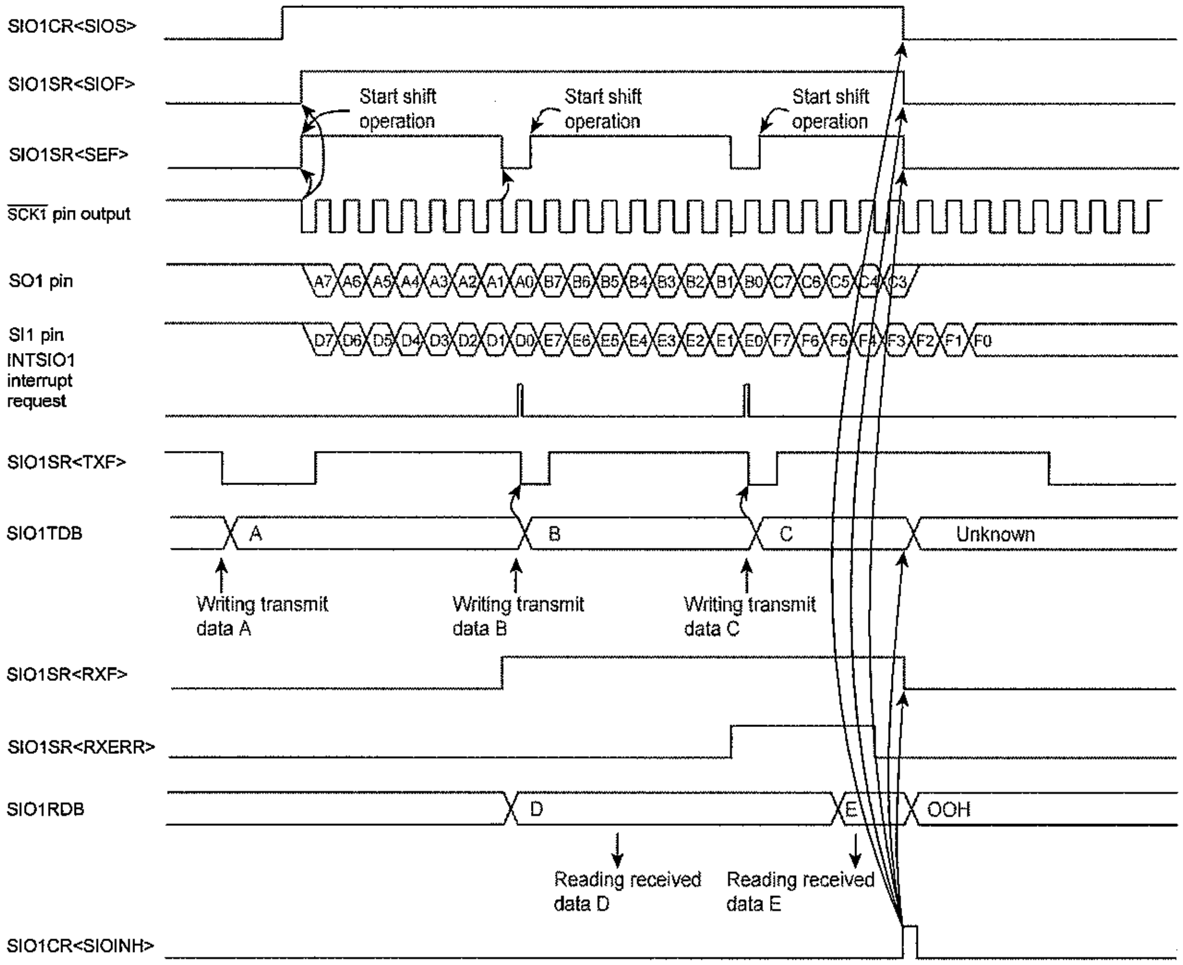


Figure 14-14 Example of Transmit/Receive (Receive) Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

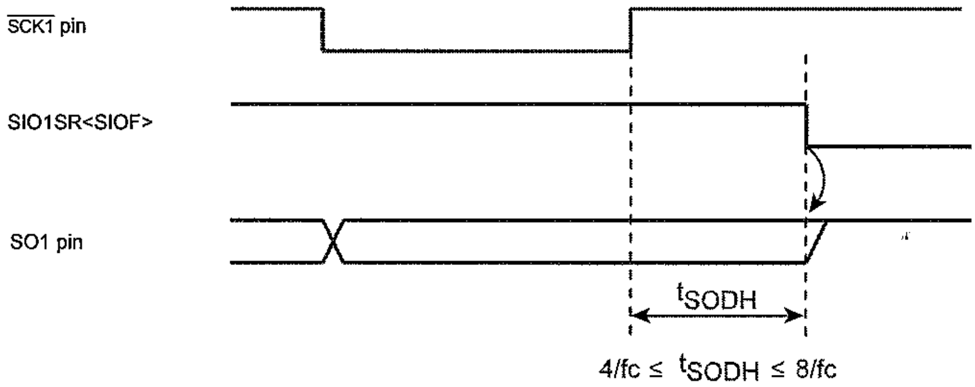


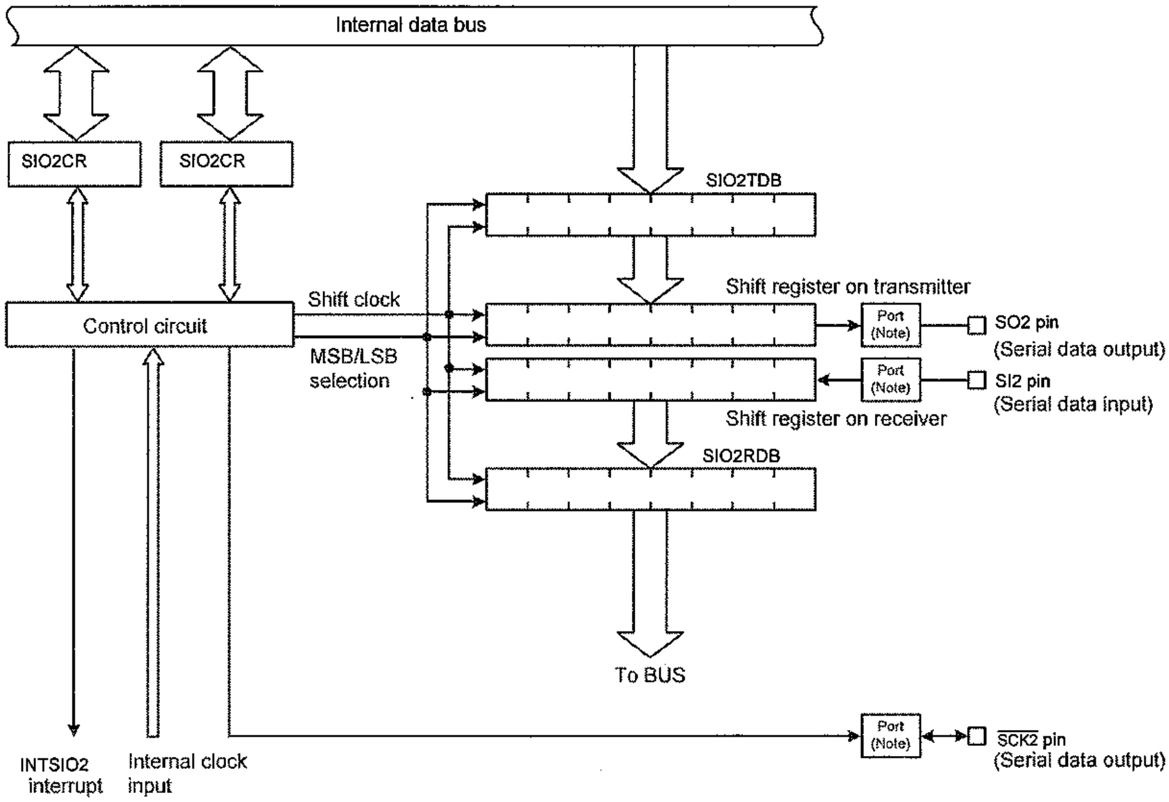
Figure 14-15 Hold Time of the End of Transmit/Receive Mode

15. HSIO2 (High-Speed Synchronous Serial Interface)

The serial interfaces connect to an external device via SI2, SO2, and $\overline{\text{SCK2}}$ pins.

When these pins are used as serial interface, the output latches for each port.

15.1 Configuration



Note: Set the register of port correctly for the port assigned as serial interface pins.
For details, see the description of the input/output port control register.

Figure 15-1 Serial Interfaces

15.2 Control

The SIO is controlled using the serial interface control register (SIO2CR). The operating status of the serial interface can be inspected by reading the status register (SIO2CR).

Serial Interface Control Register

<table>
<tr><td>SIO2CR</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td>(0031H)</td><td>SIOS</td><td>SIOINH</td><td colspan="2">SIOM</td><td>SIODIR</td><td colspan="3">SCK</td><td>(Initial value: 0000 0000)</td></tr>
</table>

<table>
<tr><td>SIOS</td><td>Specify start/stop of transfer</td><td colspan="4">0: Stop
1: Start</td><td rowspan="5">R/W</td></tr>
<tr><td>SIOINH</td><td>Forcibly stops transfer (Note 1)</td><td colspan="4">0: –
1: Forcibly stop (Automatically cleared to "0" after stopping)</td></tr>
<tr><td>SIOM</td><td>Selects transfer mode</td><td colspan="4">00: Transmit mode
01: Receive mode
10: Transmit/receive mode
11: Reserved</td></tr>
<tr><td>SIODIR</td><td>Selects direction of transfer</td><td colspan="4">0: MSB (Transfer beginning with bit7)
1: LSB (Transfer beginning with bit0)</td></tr>
<tr><td rowspan="10">SCK</td><td rowspan="10">Selects serial clock</td><td rowspan="2"></td><td colspan="2">NORMAL1/2 or IDLE1/2 modes</td><td rowspan="2">SLOW/SLEEP mode</td></tr>
<tr><td>TBTCR &lt;DV7CK&gt; = "0"</td><td>TBTCR &lt;DV7CK&gt; = "1"</td></tr>
<tr><td>000</td><td>$fc/2^{12}$</td><td>$fs/2^4$</td><td>$fs/2^4$</td></tr>
<tr><td>001</td><td>$fc/2^8$</td><td>$fc/2^8$</td><td>Reserved</td></tr>
<tr><td>010</td><td>$fc/2^7$</td><td>$fc/2^7$</td><td>Reserved</td></tr>
<tr><td>011</td><td>$fc/2^6$</td><td>$fc/2^6$</td><td>Reserved</td></tr>
<tr><td>100</td><td>$fc/2^5$</td><td>$fc/2^5$</td><td>Reserved</td></tr>
<tr><td>101</td><td>$fc/2^4$</td><td>$fc/2^4$</td><td>Reserved</td></tr>
<tr><td>110</td><td>$fc/2^3$</td><td>$fc/2^3$</td><td>Reserved</td></tr>
<tr><td>111</td><td colspan="3">External clock (Input from $\overline{\text{SCK1}}$ pin)</td></tr>
</table>

Note 1: When SIO2CR<SIOINH> is set to "1", SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

Note 2: Transfer mode, direction of transfer and serial clock must be select during the transfer is stopping (when SIO2SR<SIOF> "0").

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

Serial Interface Status Register

	7	6	5	4	3	2	1	0	
SIO2SR (0032H)	SIOF	SEF	TXF	RXF	TXERR	RXERR			(Initial value: 0010 00**)

SIOF	Serial transfer operation status monitor	0: Transfer finished 1: Transfer in progress	Read only
SEF	Number of clocks monitor	0: 8 clocks 1: 1 to 7 clocks	
TXF	Transmit buffer empty flag	0: Data exists in transmit buffer 1: No data exists in transmit buffer	
RXF	Receive buffer full flag	0: No data exists in receive buffer 1: Data exists in receive buffer	
TXERR	Transfer operation error flag	Read 0: – (No error exist) 1: Transmit buffer under run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored)	R/W
RXERR	Receive operation error flag	Read 0: – (No error exist) 1: Receive buffer over run occurs in an external clock mode Write 0: Clear the flag 1: – (A write of "1" to this bit is ignored)	

Note 1: The operation error flag (TXERR and RXERR) are not automatically cleared by stopping transfer with SIO2CR<SIOS> "0". Therefore, set these bits to "0" for clearing these error flag. Or set SIO2CR<SIOINH> to "1".

Note 2: *: Don't care

Receive buffer register

	7	6	5	4	3	2	1	0	Read only
SIO2RDB (002BH)									(Initial value: 0000 0000)

Transmit buffer register

	7	6	5	4	3	2	1	0	Write only
SIO2TDB (002BH)									(Initial value: **** ****)

Note 1: SIO2TDB is write only register. A bit manipulation should not be performed on the transmit buffer register using a read-modify-write instruction.

Note 2: The SIO2TDB should be written after checking SIO2SR<TXF> "1". When SIO2SR<TXF> is "0", the writing data can't be transferred to SIO2TDB even if write instruction is executed to SIO2TDB .

Note 3: *: Don't care

15.2.1 Serial clock

15.2.1.1 Clock source

The serial clock can be selected by using SIO2CR<SCK>. When the serial clock is changed, the writing instruction to SIO2CR<SCK> should be executed while the transfer is stopped (when SIO2SR<SIOF> "0")

(1) Internal clock

Setting the SIO2CR<SCK> to other than "111" outputs the clock (Shown in Table 2.8.2) as serial clock outputs from $\overline{\text{SCK2}}$ pin. At the before beginning or finishing of a transfer, $\overline{\text{SCK2}}$ pin is kept in high level.

When writing (in the transmit mode) or reading (in the receive mode) data can not follow the serial clock rate, an automatic-wait function is executed to stop the serial clock automatically and hold the next shift operation until reading or writing is completed. The maximum time from releasing the automatic-wait function by reading or writing a data is 1 cycle of the selected serial clock until the serial clock comes out from $\overline{\text{SCK2}}$ pin.

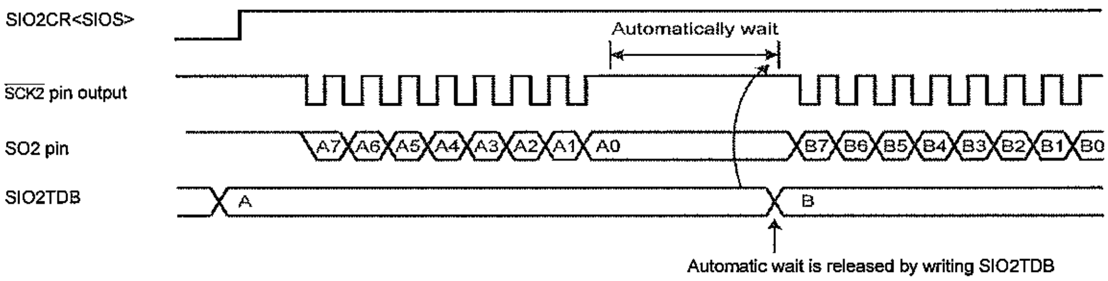


Figure 15-2 Automatic-wait Function (Example of transmit mode)

Table 15-1 Serial Clock Rate (fc = 16 MHz, fs = 32.768kHz)

	NORMAL1/2, IDLE1/2 Mode				SLOW1/2, SLEEP1/2 Mode	
	TBTCR<DV7CK> = "0"		TBTCR<DV7CK> = "1"		Serial Clock	Baud Rate
SCK	Serial Clock	Baud Rate	Serial Clock	Baud Rate		
000	fc/2^{12}	3.906 kbps	fs/2^4	2048 bps	fs/2^4	2048 bps
001	fc/2^8	62.5 kbps	fc/2^8	62.5 kbps	Reserved	–
010	fc/2^7	125 kbps	fc/2^7	125 kbps	Reserved	–
011	fc/2^6	250 kbps	fc/2^6	250 kbps	Reserved	–
100	fc/2^5	500 kbps	fc/2^5	500 kbps	Reserved	–
101	fc/2^4	1.00 Mbps	fc/2^4	1.00 Mbps	Reserved	–
110	fc/2^3	2.00 Mbps	fc/2^3	2.00 Mbps	Reserved	

(2) External clock

When an external clock is selected by setting SIO2CR<SCK> to "111", the clock via the $\overline{\text{SCK2}}$ pin from an external source is used as the serial clock.

To ensure shift operation, the serial clock pulse width must be 4/fc or more for both "H" and "L" levels.

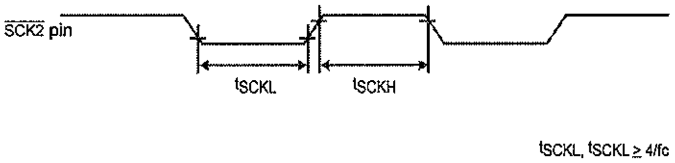

$\overline{\text{SCK2}}$ pin
t_{SCKL}
t_{SCKH}
t_{SCKL}, $t_{SCKL} \geq 4/fc$

Figure 15-3 External Clock

15.2.1.2 Shift edge

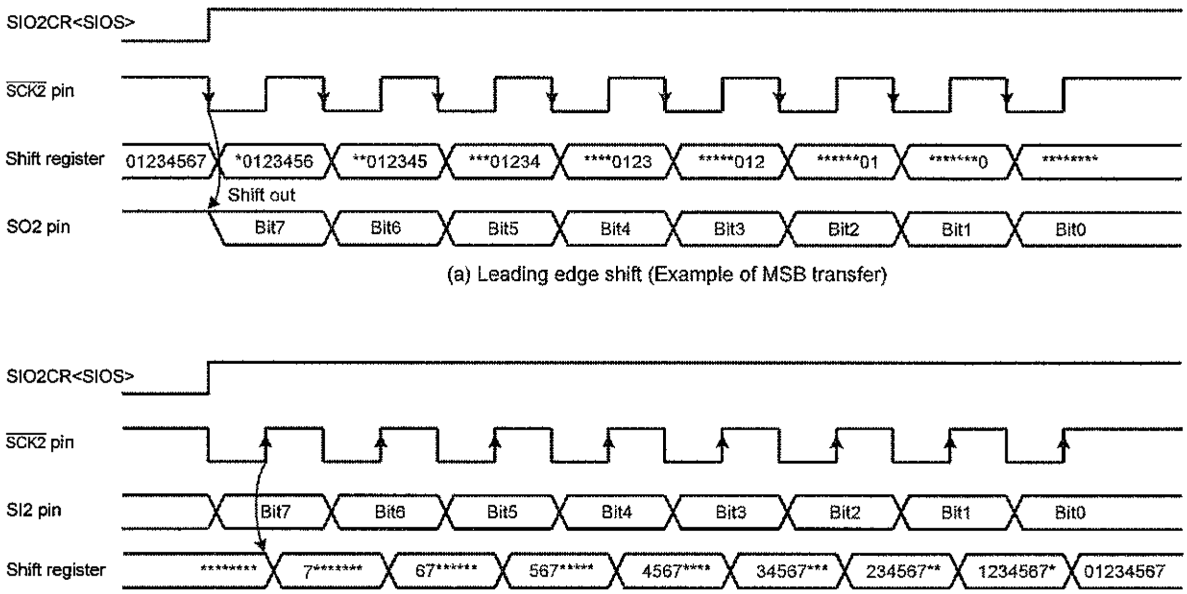
The leading edge is used to transmit data, and the trailing edge is used to receive data.

(1) Leading edge shift

Data is shifted on the leading edge of the serial clock (falling edge of the $\overline{\text{SCK2}}$ pin input/output).

(2) Trailing edge shift

Data is shifted on the trailing edge of the serial clock (rising edge of the $\overline{\text{SCK2}}$ pin input/output).



(a) Leading edge shift (Example of MSB transfer)

(b) Trailing edge shift (Example of MSB transfer)

Figure 15-4 Shift Edge

15.2.2 Transfer bit direction

Transfer data direction can be selected by using SIO2CR<SIODIR>. The transfer data direction can't be set individually for transmit and receive operations.

When the data direction is changed, the writing instruction to SIO2CR<SIODIR> should be executed while the transfer is stopped (when SIO2CR<SIOF> “0”)

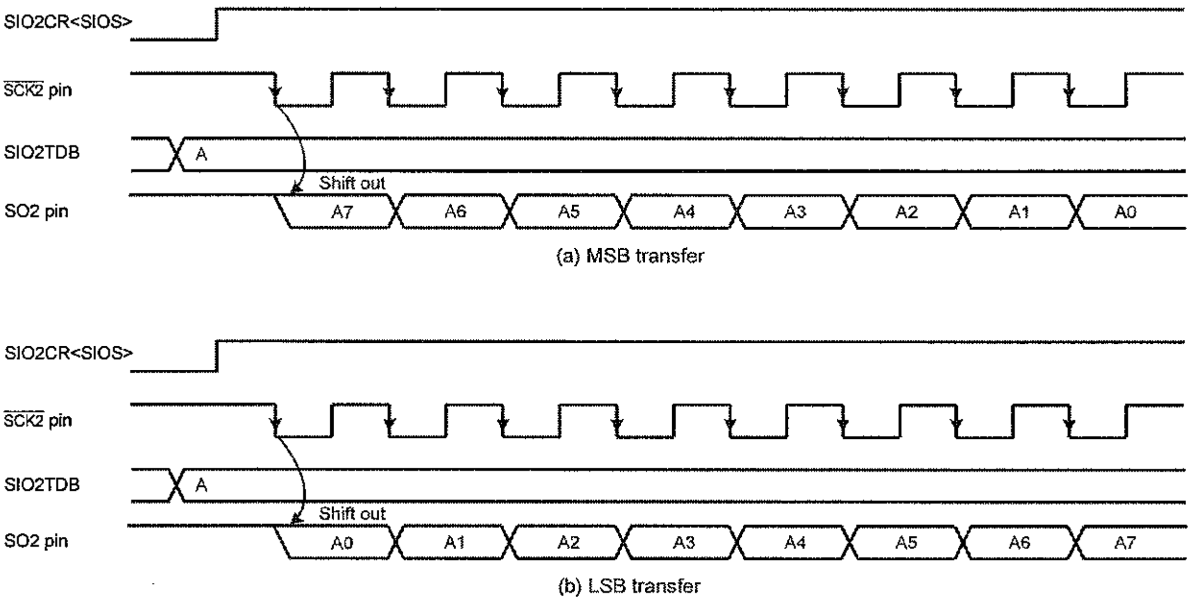


Figure 15-5 Transfer Bit Direction (Example of transmit mode)

15.2.2.1 Transmit mode

(1) MSB transmit mode

MSB transmit mode is selected by setting SIO2CR<SIODIR> to "0", in which case the data is transferred sequentially beginning with the most significant bit (Bit7).

(2) LSB transmit mode

LSB transmit mode is selected by setting SIO2CR<SIODIR> to "1", in which case the data is transferred sequentially beginning with the least significant bit (Bit0).

15.2.2.2 Receive mode

(1) MSB receive mode

MSB receive mode is selected by setting SIO2CR<SIODIR> to "0", in which case the data is received sequentially beginning with the most significant bit (Bit7).

(2) LSB receive mode

LSB receive mode is selected by setting SIO2CR<SIODIR> to "1", in which case the data is received sequentially beginning with the least significant bit (Bit0).

15.2.2.3 Transmit/receive mode

(1) MSB transmit/receive mode

MSB transmit/receive mode are selected by setting SIO2CR<SIODIR> to "0" in which case the data is transferred sequentially beginning with the most significant bit (Bit7) and the data is received sequentially beginning with the most significant (Bit7).

(2) LSB transmit/receive mode

LSB transmit/receive mode are selected by setting SIO2CR<SIODIR> to "1", in which case the data is transferred sequentially beginning with the least significant bit (Bit0) and the data is received sequentially beginning with the least significant (Bit0).

15.2.3 Transfer modes

Transmit, receive and transmit/receive mode are selected by using SIO2CR<SIOM>.

15.2.3.1 Transmit mode

Transmit mode is selected by writing "00" to SIO2CR<SIOM>.

(1) Starting the transmit operation

Transmit mode is selected by setting "00" to SIO2CR<SIOM>. Serial clock is selected by using SIO2CR<SCK>. Transfer direction is selected by using SIO2CR<SIODIR>.

When a transmit data is written to the transmit buffer register (SIO2TDB), SIO2SR<TXF> is cleared to "0".

After SIO2CR<SIOS> is set to "1", SIO2SR<SIOF> is set synchronously to "1" the falling edge of $\overline{\text{SCK2}}$ pin.

The data is transferred sequentially starting from SO2 pin with the direction of the bit specified by SIO2CR<SIODIR>, synchronizing with the $\overline{\text{SCK2}}$ pin's falling edge.

SIO2SR<SEF> is kept in high level, between the first clock falling edge of $\overline{\text{SCK2}}$ pin and eighth clock falling edge.

SIO2SR<TXF> is set to "1" at the rising edge of pin after the data written to the SIO2TDB is transferred to shift register, then the INTSIO2 interrupt request is generated, synchronizing with the next falling edge on $\overline{\text{SCK2}}$ pin.

Note 1: In internal clock operation, when SIO2CR<SIOS> is set to "1", transfer mode does not start without writing a transmit data to the transmit buffer register (SIO2TDB).

Note 2: In internal clock operation, when the SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from $\overline{\text{SCK2}}$ pin.

Note 3: In external clock operation, when the falling edge is input from $\overline{\text{SCK2}}$ pin after SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register immediately.

(2) During the transmit operation

When data is written to SIO2TDB, SIO2SR<TXF> is cleared to "0".

In internal clock operation, in case a next transmit data is not written to SIO2TDB, the serial clock stops to "H" level by an automatic-wait function when all of the bit set in the SIO2TDB has been transmitted. Automatic-wait function is released by writing a transmit data to SIO2TDB. Then, transmit operation is restarted after maximum 1-cycle of serial clock.

When the next data is written to the SIO2TDB before termination of previous 8-bit data with SIO2SR<TXF> "1", the next data is continuously transferred after transmission of previous data.

In external clock operation, after SIO2SR<TXF> is set to "1", the transmit data must be written to SIO2TDB before the shift operation of the next data begins.

If the transmit data is not written to SIO2TDB, transmit error occurs immediately after shift operation is started. Then, INTSIO2 interrupt request is generated after SIO2SR<TXERR> is set to "1".

(3) Stopping the transmit operation

There are two ways for stopping transmits operation.

- The way of clearing SIO2CR<SIOS>.
  When SIO2CR<SIOS> is cleared to "0", transmit operation is stopped after all transfer of the data is finished. When transmit operation is finished, SIO2SR<SIOF> is cleared to "0" and SO2 pin is kept in high level.
  In external clock operation, SIO2CR<SIOS> must be cleared to "0" before SIO2SR<SEF> is set to "1" by beginning next transfer.
- The way of setting SIO2CR<SIOINH>.
  Transmit operation is stopped immediately after SIO2CR<SIOINH> is set to "1". In this case, SIO1CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

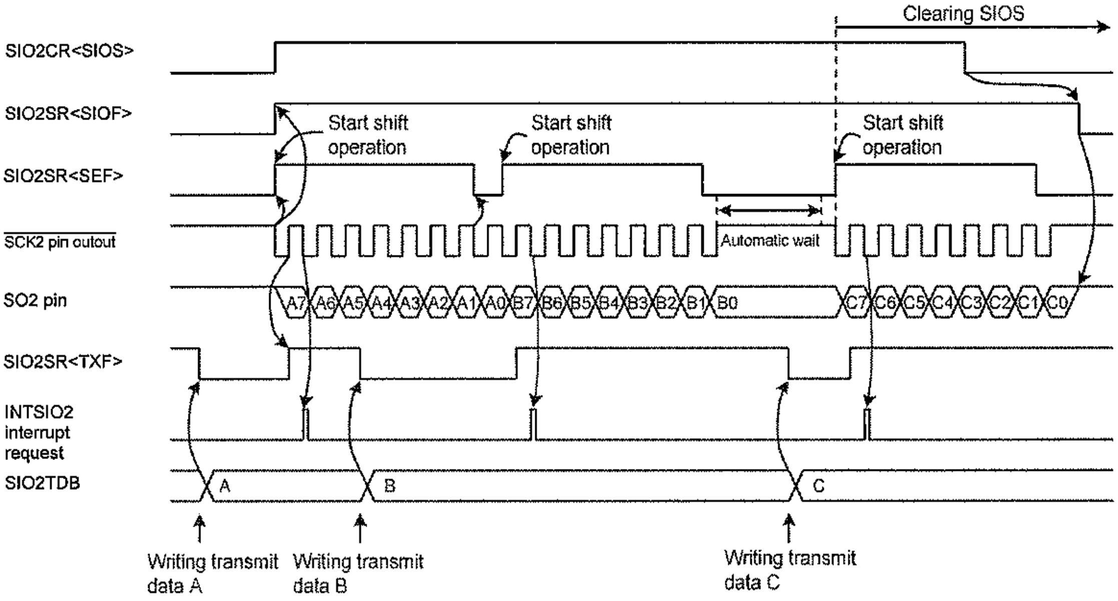


Figure 15-6 Example of Internal Clock and MSB Transmit Mode

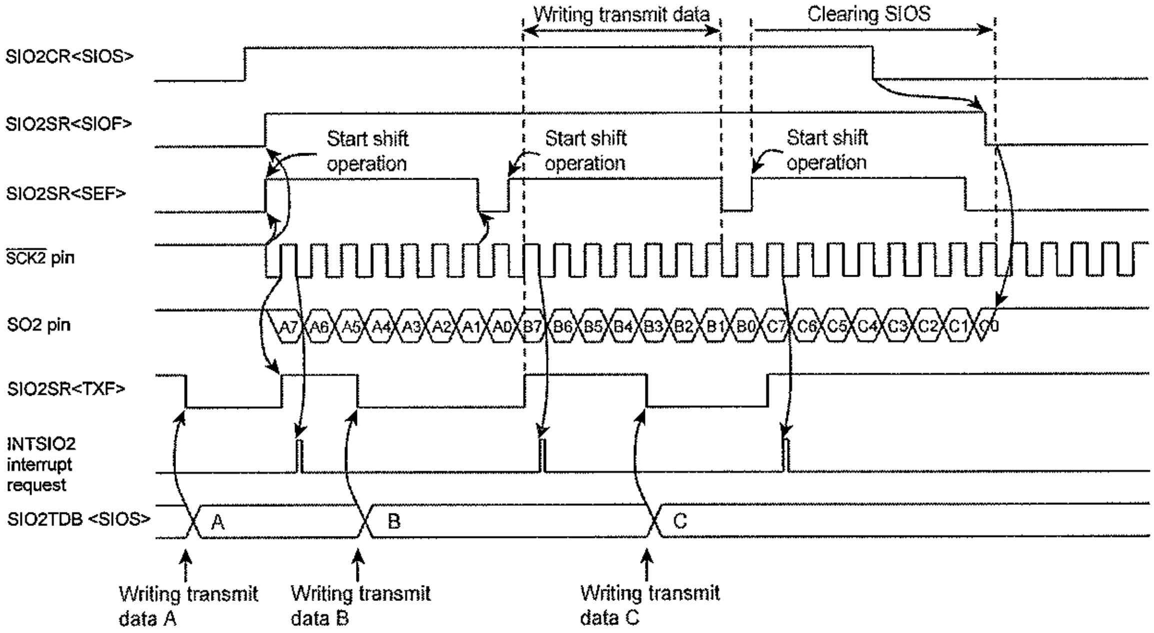
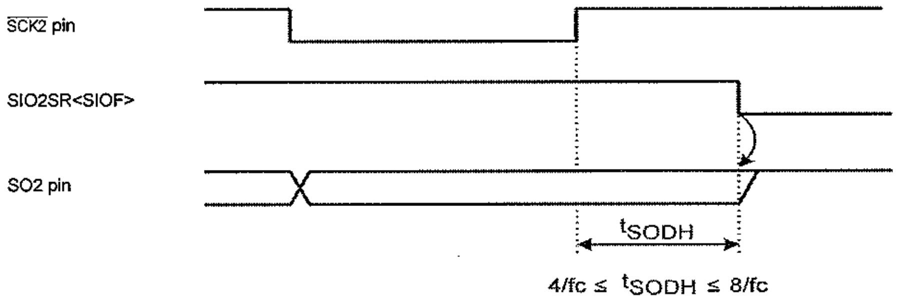


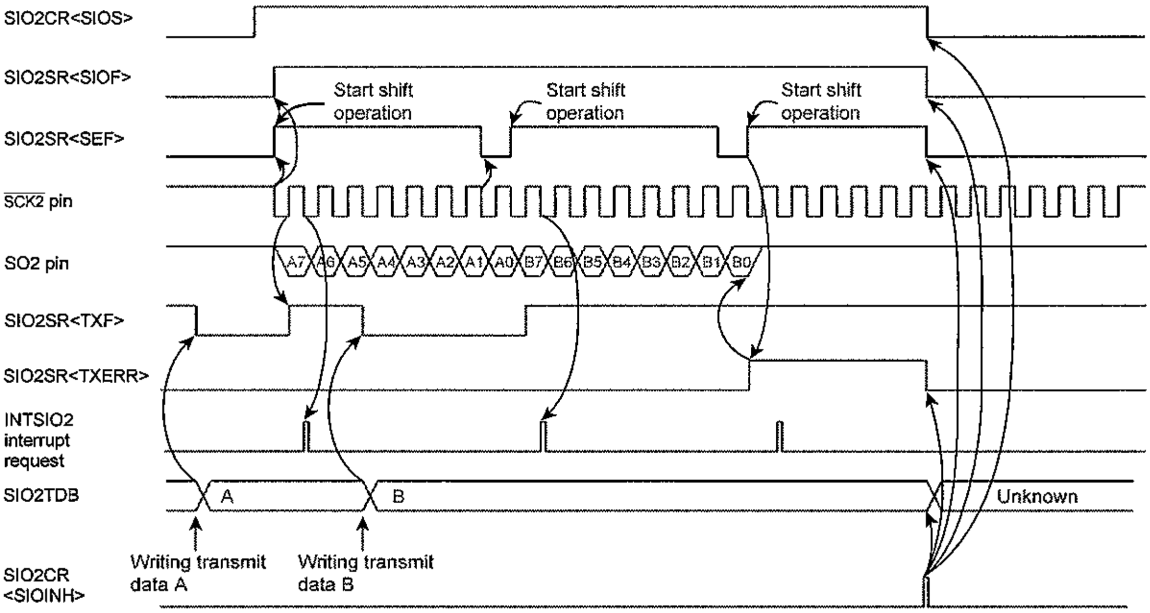
Figure 15-7 Exaple of External Clock and MSB Transmit Mode



(4) Transmit error processing

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIO2TDB in external clock operation.
  If transmit errors occur during transmit operation, SIO2SR<TXERR> is set to “1” immediately after starting shift operation. Synchronizing with the next serial clock falling edge, INTSIO2 interrupt request is generated.
  If shift operation starts before writing data to SIO2TDB after SIO2CR<SIOS> is set to “1”, SIO2SR<TXERR> is set to “1” immediately after shift operation is started and then INTSIO1 interrupt request is generated.
  SIO2 pin is kept in high level when SIO2SR<TXERR> is set to “1”. When transmit error occurs, transmit operation must be forcibly stop by writing SIO2CR<SIOINH> to “1”. In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.



15.2.3.2 Receive mode

The receive mode is selected by writing “01” to SIO2CR<SIOM>.

(1) Starting the receive operation

Receive mode is selected by setting “01” to SIO2CR<SIOM>. Serial clock is selected by using SIO2CR<SCK>. Transfer direction is selected by using SIO2CR<SIODIR>.

After SIO2CR<SIOS> is set to “1”, SIO2SR<SIOF> is set synchronously to “1” the falling edge of SCK2 pin.

Synchronizing with the SCK2 pin's rising edge, the data is received sequentially from SI2 pin with the direction of the bit specified by SBI2DIR<SIODIR>.

SIO1SR<SEF> is kept in high level, between the first clock falling edge of SCK2 pin and eighth clock falling edge.

When 8-bit data is received, the data is transferred to SIO2RDB from shift register. INTSIO2 interrupt request is generated and SIO2SR<RXF> is set to “1”

Note: In internal clock operation, when the SIO1CR<SIOS> is set to "1", the serial clock is generated from SCK2 pin after maximum 1-cycle of serial clock frequency.

(2) During the receive operation

The SIO2SR<RXF> is cleared to “0” by reading a data from SIO2RDB.

In the internal clock operation, the serial clock stops to “H” level by an automatic-wait function when the all of the 8-bit data has been received. Automatic-wait function is released by reading a received data from SIO2RDB. Then, receive operation is restarted after maximum 1-cycle of serial clock.

In external clock operation, after SIO2SR<RXF> is set to “1”, the received data must be read from SIO2RDB, before the next data shift-in operation is finished.

If received data is not read out from SIO2RDB receive error occurs immediately after shift operation is finished. Then INTSIO2 interrupt request is generated after SIO2SR<RXERR> is set to "1".

(3) Stopping the receive operation

There are two ways for stopping the receive operation.

- The way of clearing SIO2CR<SIOS>.
  When SIO2CR<SIOS> is cleared to "0", receive operation is stopped after all of the data is finished to receive. When receive operation is finished, SIO2SR<SIOF> is cleared to "0". In external clock operation, SIO2CR<SIOS> must be cleared to "0" before SIO2SR<SEF> is set to "1" by starting the next shift operation.
- The way of setting SIO2CR<SIOINH>.
  Receive operation is stopped immediately after SIO2CR<SIOINH> is set to "1". In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

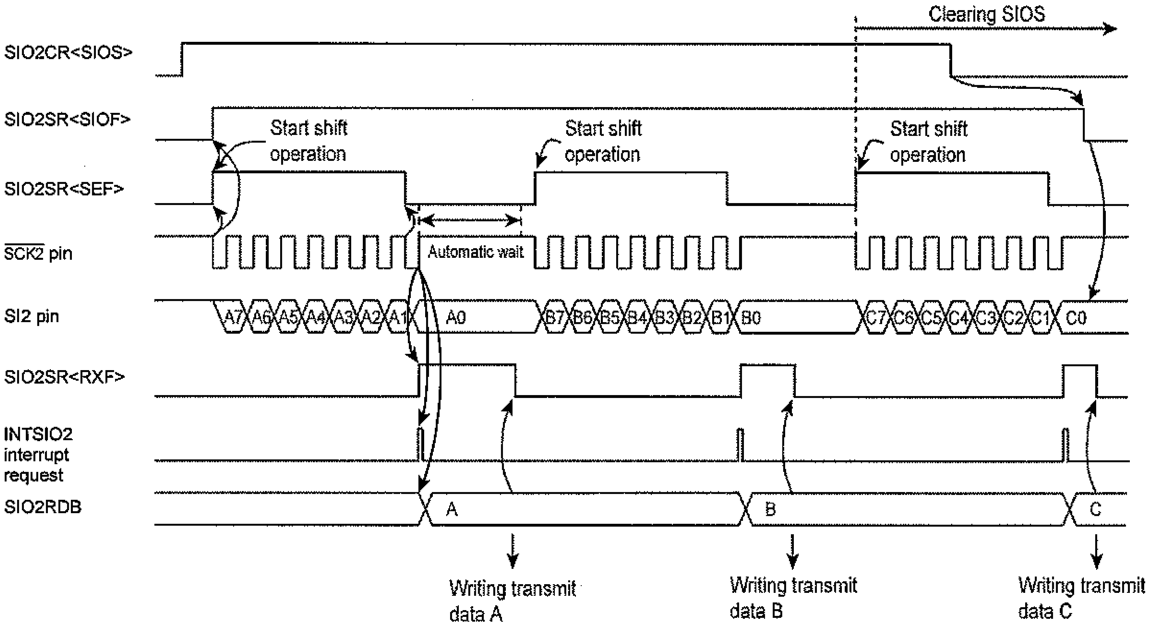


Figure 15-8 Example of Internal Clock and MSB Receive Mode

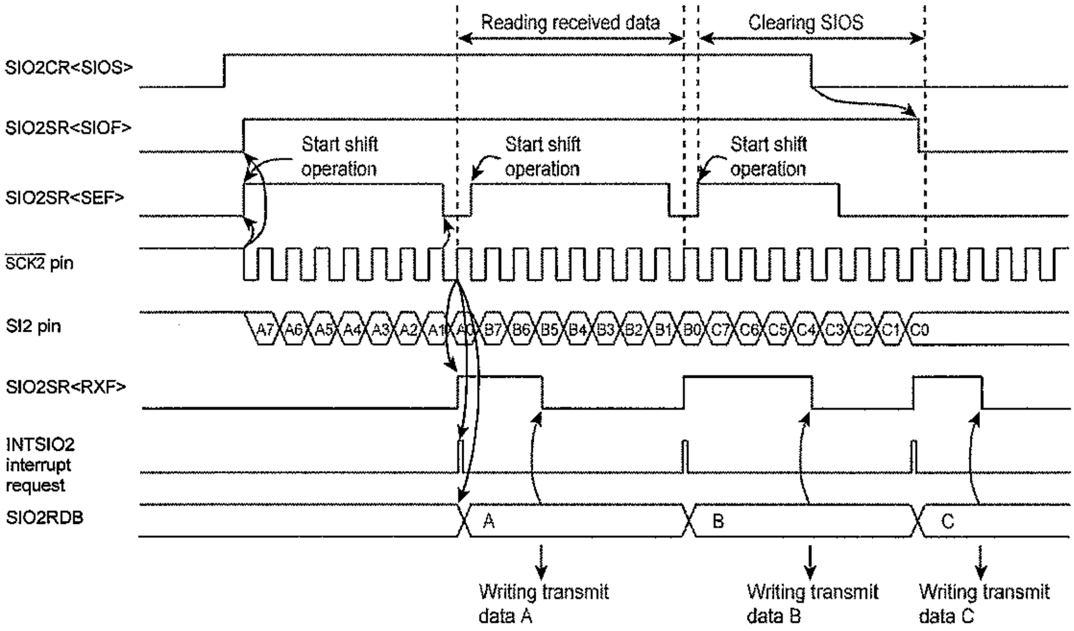


Figure 15-9 Example of External Clock and MSB Receive Mode

(4) Receive error processing

Receive errors occur on the following situation. To protect SIO2RDB and the shift register contents, the received data is ignored while the SIO2SR<RXERR> is "1".

- Shift operation is finished before reading out received data from SIO2RDB at SIO2SR<RXF> is "1" in an external clock operation.
  If receive error occurs, set the SIO2CR<SIOS> to "0" for reading the data that received immediately before error occurence. And read the data from SIO2RDB. Data in shift register (at errors occur) can be read by reading the SIO2RDB again.
  When SIO2SR<RXERR> is cleared to "0" after reading the received data, SIO2SR<RXF> is cleared to "0".
  After clearing SIO2CR<SIOS> to "0", when 8-bit serial clock is input to SCK2 pin, receive operation is stopped. To restart the receive operation, confirm that SIO2SR<SIOF> is cleared to "0".
  If the receive error occurs, set the SIO2CR<SIOINH> to "1" for stopping the receive operation immediately. In this case, SIO2CR<SIOS>, SIO1SR register, SIO2RDB register and SIO2TDB register are initialized.

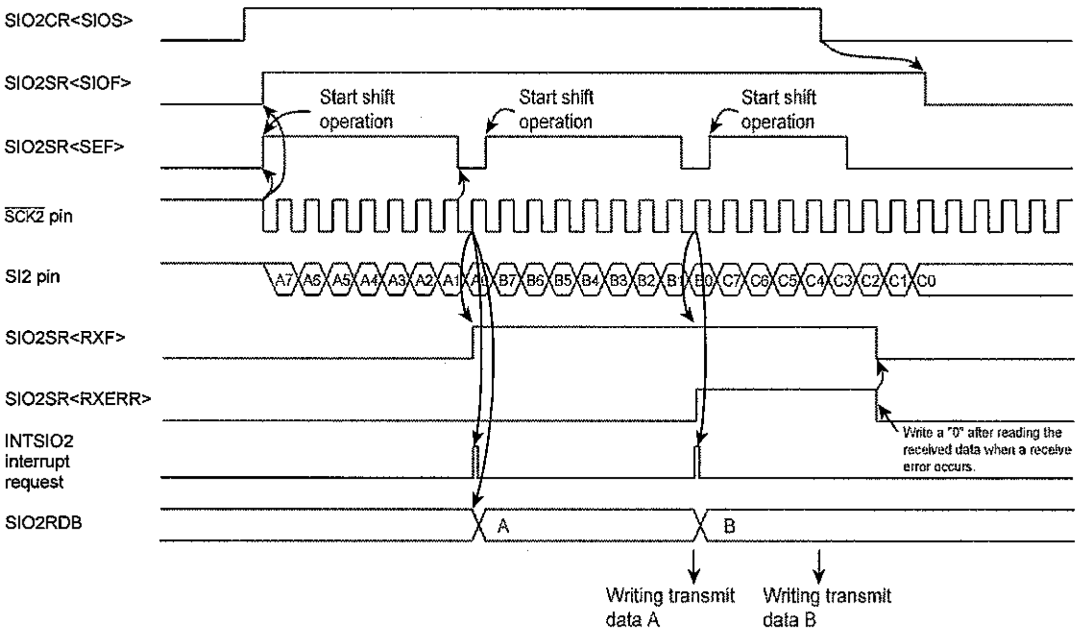


Figure 15-10 Example of Receive Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

15.2.3.3 Transmit/receive mode

The transmit/receive mode are selected by writing “10” to SIO2CR<SIOM>.

(1) Starting the transmit/receive operation

Transmit/receive mode is selected by writing “10” to SIO2CR<SIOM>. Serial clock is selected by using SIO2CR<SCK>. Transfer direction is selected by using SIO2CR<SIODIR>.

When a transmit data is written to the transmit buffer register (SIO1TDB), SIO2SR<TXF> is cleared to “0”.

After SIO2CR<SIOS> is set to “1”, SIO2SR<SIOF> is set synchronously to the falling edge of SCK2 pin.

The data is transferred sequentially starting from SO2 pin with the direction of the bit specified by SIO2CR<SIODIR>, synchronizing with the SCK2 pin's falling edge. And receiving operation also starts with the direction of the bit specified by SIO2CR<SIODIR>, synchronizing with the SCK2 pin's rising edge.

SIO2SR<SEF> is kept in high level between the first clock falling edge of SCK2 pin and eighth clock falling edge.

SIO2SR<TXF> is set to “1” at the rising edge of SCK2 pin after the data written to the SIO2TDB is transferred to shift register. When 8-bit data has been received, the received data is transferred to SIO2RDB from shift register, then the INTSIO2 interrupt request occurs, synchronizing with setting SIO2SR<RXF> to “1”.

Note 1: In internal clock operation, when the SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register after maximum 1-cycle of serial clock frequency, then a serial clock is output from SCK2 pin.

Note 2: In external clock operation, when the falling edge is input from SCK2 pin after SIO2CR<SIOS> is set to "1", SIO2TDB is transferred to shift register immediately. When the rising edge is input from SCK2 pin, receive operation also starts.

(2) During the transmit/receive operation

When data is written to SIO2TDB, SIO2SR<TXF> is cleared to "0" and when a data is read from SIO2RDB, SIO2SR<RXF> is cleared to "0".

In internal clock operation, in case of the condition described below, the serial clock stops to "H" level by an automatic-wait function when all of the bit set in the data has been transmitted.

- Next transmit data is not written to SIO2TDB after reading a received data from SIO2RDB.
- Received data is not read from SIO2RDB after writing a next transmit data to SIO2TDB.
- Neither SIO2TDB nor SIO2RDB is accessed after transmission.

The automatic wait function is released by writing the next transmit data to SIO2TDB after reading the received data from SIO2RDB, or reading the received data from SIO2RDB after writing the next data to SIO2TDB.
Then, transmit/receive operation is restarted after maximum 1 cycle of serial clock.
In external clock operation, reading the received data from SIO2RDB and writing the next data to SIO2TDB must be finished before the shift operation of the next data begins.
If the transmit data is not written to SIO2TDB after SIO2SR<TXF> is set to "1", transmit error occurs immediately after shift operation is started. When the transmit error occurred, SIO2SR<TXERR> is set to "1".
If received data is not read out from SIO2RDB before next shift operation starts after setting SIO2SR<RXF> to "1", receive error occurs immediately after shift operation is finished. When the receive error has occurred, SIO2SR<RXERR> is set to "1".

(3) Stopping the transmit/receive operation

There are two ways for stopping the transmit/receive operation.

- The way of clearing SIO2CR<SIOS>.
  When SIO2CR<SIOS> is cleared to "0", transmit/receive operation is stopped after all transfer of the data is finished. When transmit/receive operation is finished, SIO2SR<SIOF> is cleared to "0" and SO2 pin is kept in high level.
  In external clock operation, SIO2CR<SIOS> must be cleared to "0" before SIO2SR<SEF> is set to "1" by beginning next transfer.
- The way of setting SIO2CR<SIOINH>.
  Transmit/receive operation is stopped immediately after SIO2CR<SIOINH> is set to "1". In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

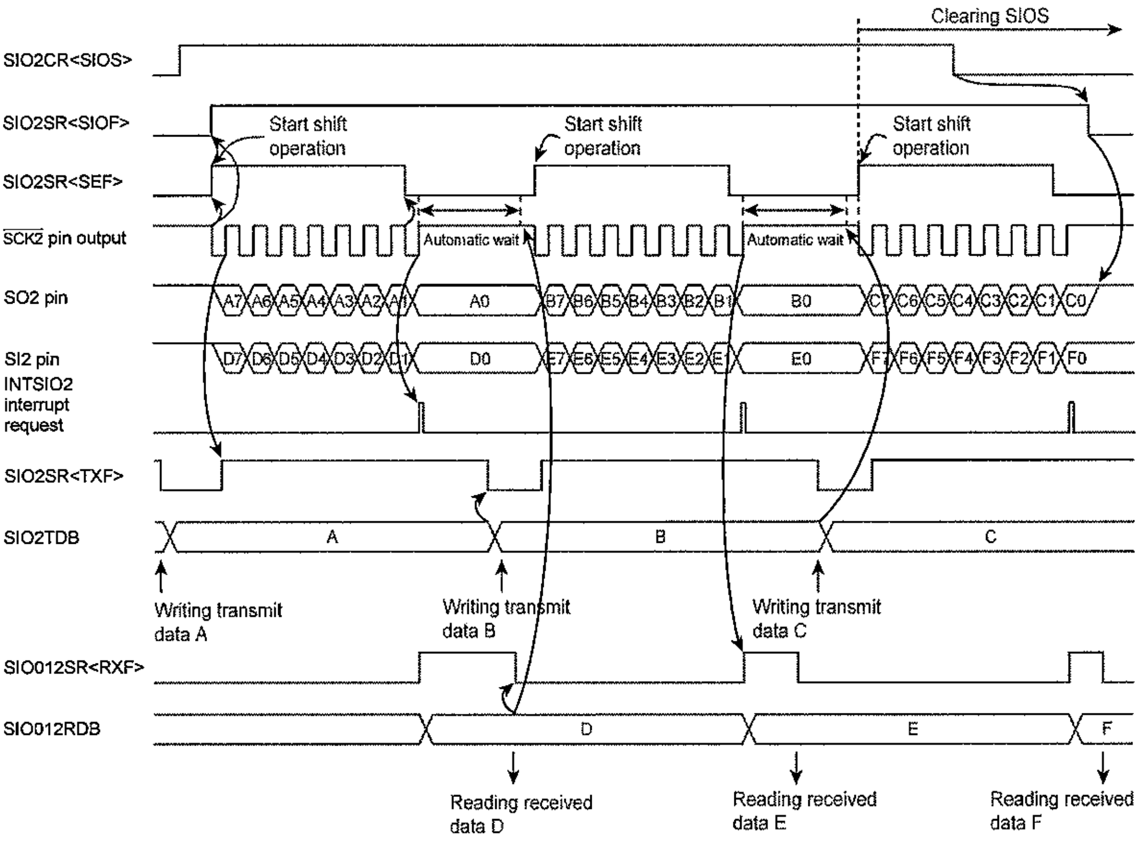


Figure 15-11 Example of Internal Clock and MSB Transmit/Receive Mode

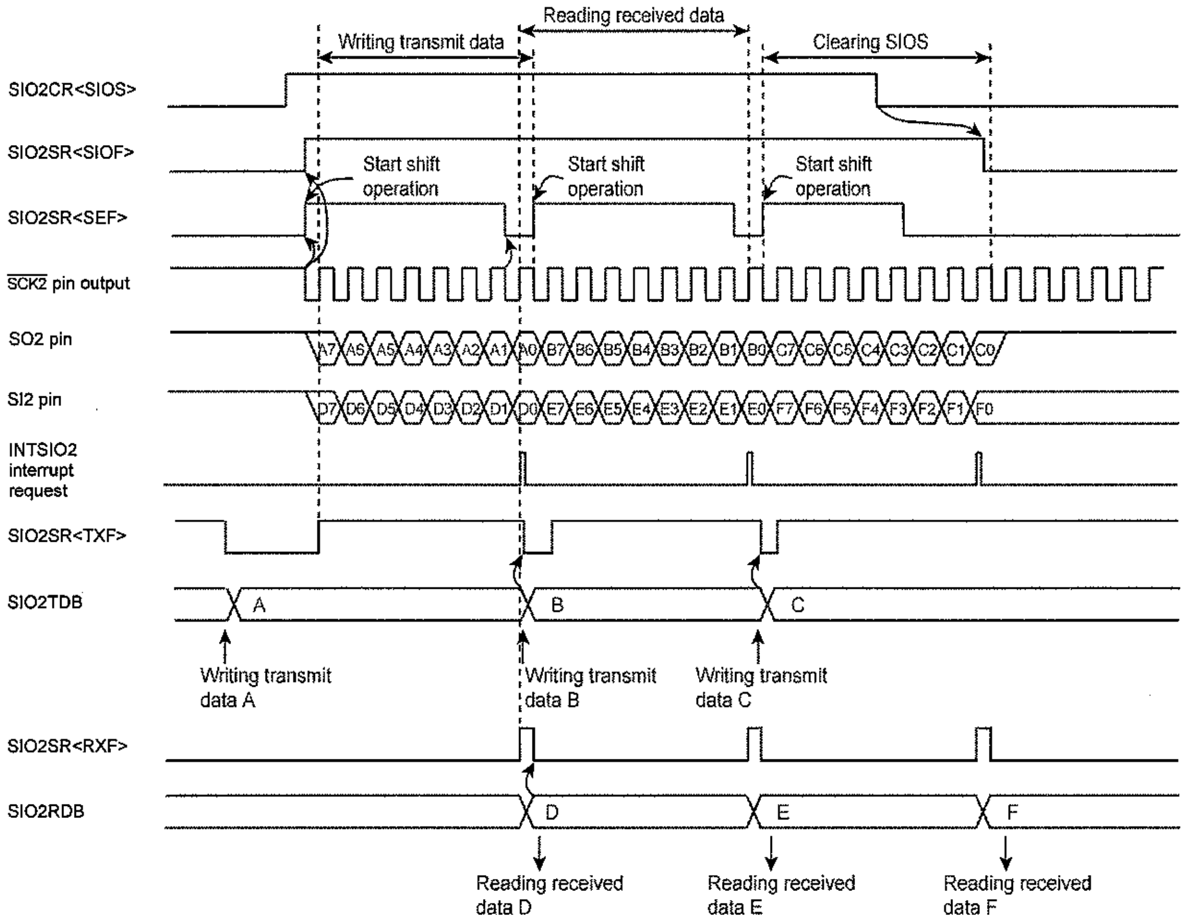


Figure 15-12 Example of External Clock and MSB Transmit/Receive Mode

(4) Transmit/receive error processing

Transmit/receive errors occur on the following situation. Corrective action is different, which errors occur transmits or receives.

(a) Transmit errors

Transmit errors occur on the following situation.

- Shift operation starts before writing next transmit data to SIO2TDB in external clock operation.
  If transmit errors occur during transmit operation, SIO2SR<TXERR> is set to "1" immediately after starting shift operation. And INTSIO2 interrupt request is generated after all of the 8-bit data has been received.
  If shift operation starts before writing data to SIO2TDB after SIO2CR<SIOS> is set to "1", SIO2SR<TXERR> is set immediately after starting shift operation. And INTSIO2 interrupt request is generated after all of the 8-bit data has been received.
  SO2 pin is kept in high level when SIO2SR<TXERR> is set to "1". When transmit error occurs, transmit operation must be forcibly stop by writing SIO2CR<SIOINH> to "1" after the received data is read from SIO2RDB. In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

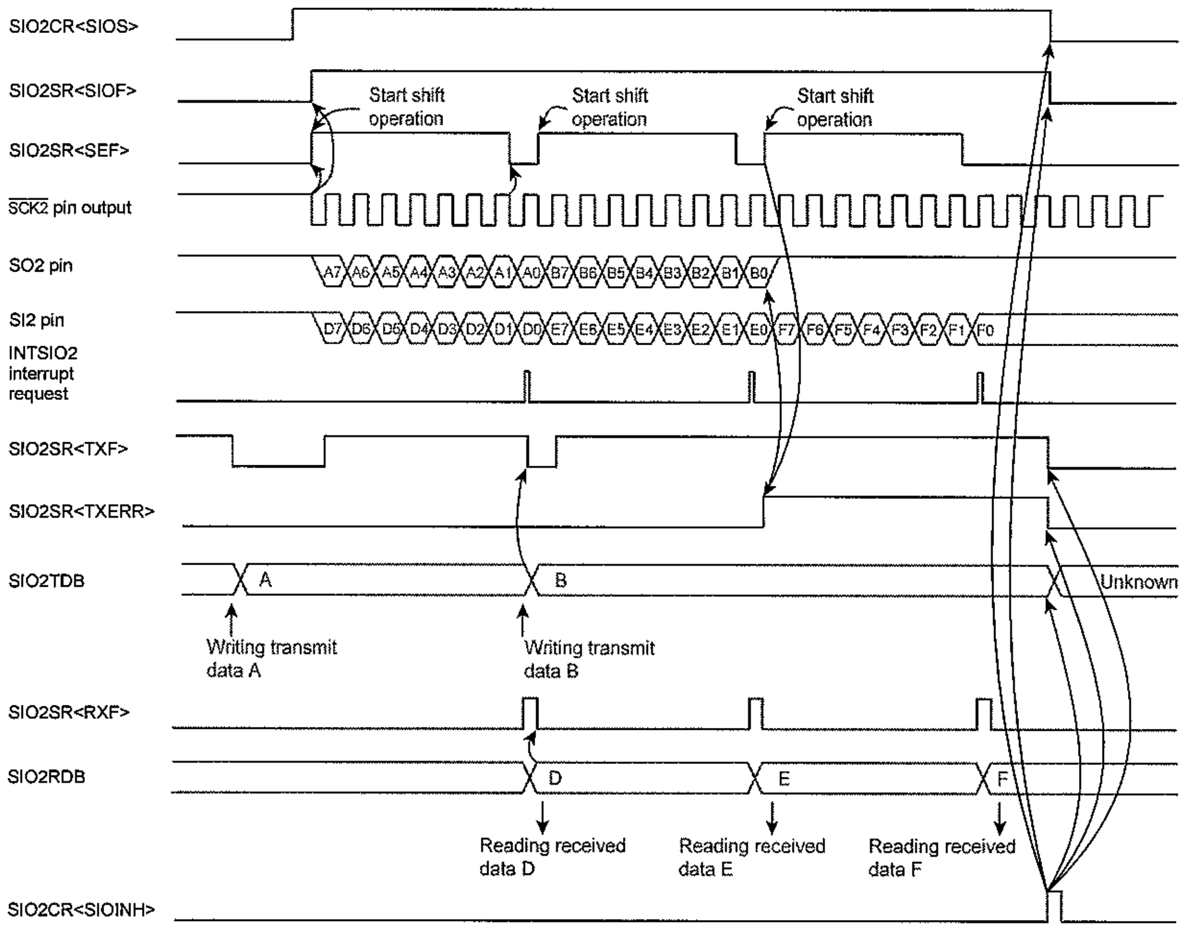


Figure 15-13 Example of Transmit/Receive (Transmit) Error Processing

(b) Receive errors

Receive errors occur on the following situation. To protect SIO2RDB and the shift register contents, the received data is ignored while the SIO2SR<RXERR> is “1”.

- Shift operation is finished before reading out received data from SIO2RDB at SIO2SR<RXF> is “1” in an external clock operation.
  If receive error occurs, set the SIO2CR<SIOS> to “0” for reading the data that received immediately before error occurence. And read the data from SIO2RDB. Data in shift register (at errors occur) can be read by reading the SIO1RDB again.
  When SIO2SR<RXERR> is cleared to “0” after reading the received data, SIO2SR<RXF> is cleared to “0”.
  After clearing SIO2CR<SIOS> to “0”, when 8-bit serial clock is input to $\overline{\text{SCK2}}$ pin, receive operation is stopped. To restart the receive operation, confirm that SIO2SR<SIOF> is cleared to “0”.
  If the received error occurs, set the SIO2CR<SIOINH> to “1” for stopping the receive operation immediately. In this case, SIO2CR<SIOS>, SIO2SR register, SIO2RDB register and SIO2TDB register are initialized.

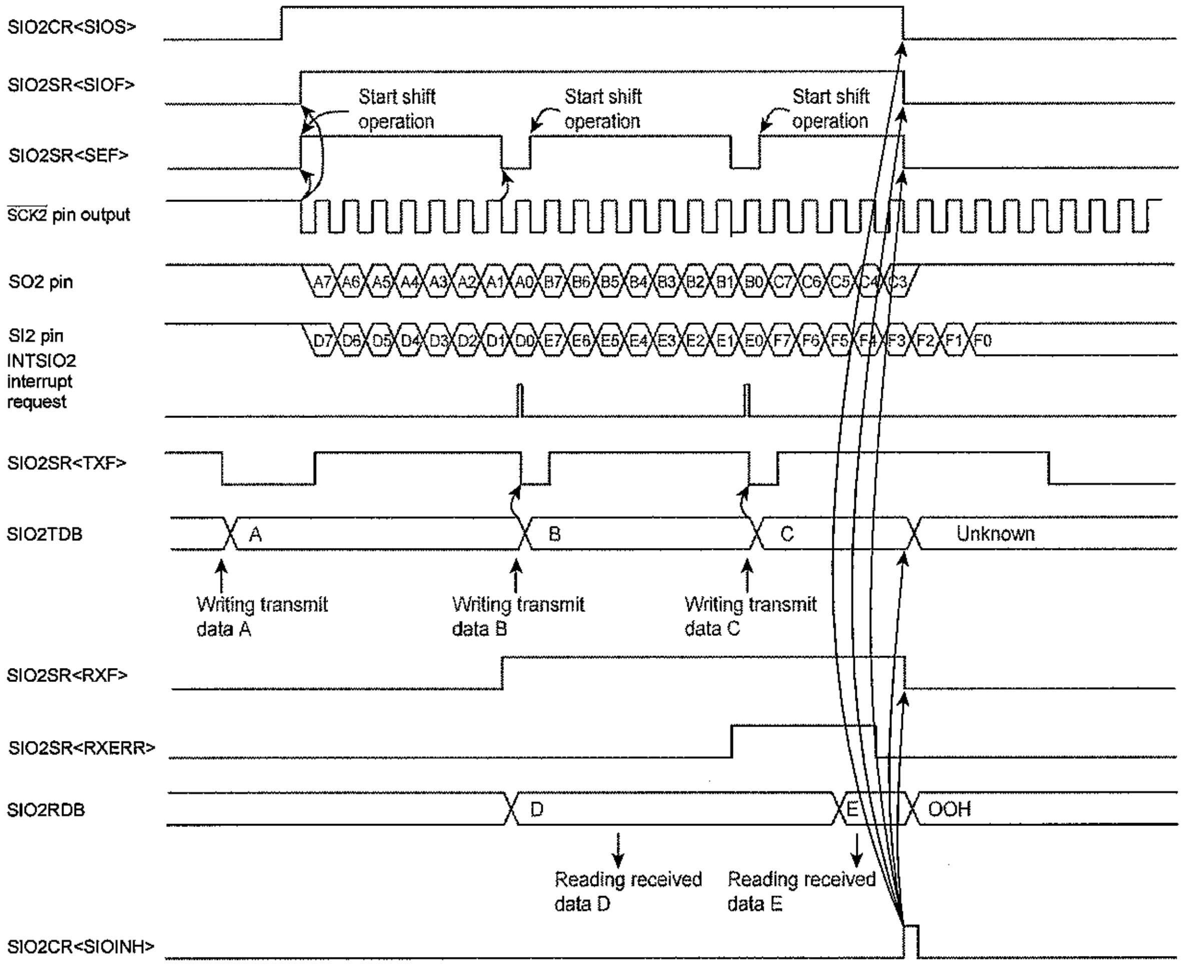


Figure 15-14 Example of Transmit/Receive (Receive) Error Processing

Note: If receive error is not corrected, an interrupt request does not generate after the error occurs.

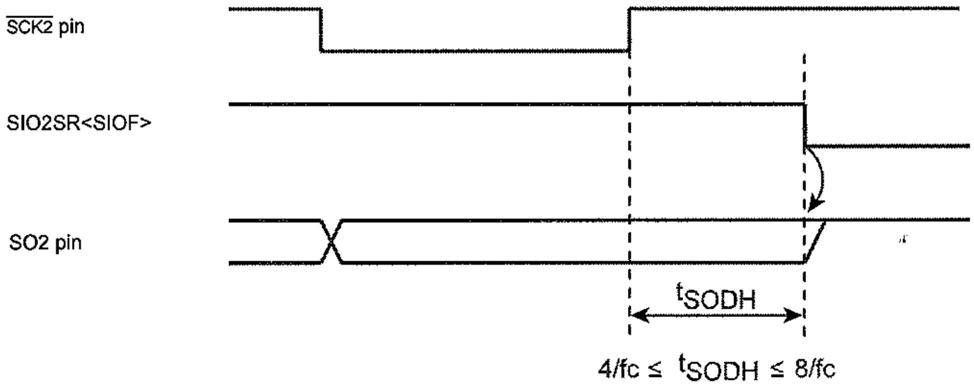


Figure 15-15 Hold Time of the End of Transmit/Receive Mode

16. Serial Bus Interface (SBI)

The TMP86FS49UG has an 1-channel serial bus interface which employs an I^2C bus (A bus system by Philips).

The serial interface is connected to an external devices through SDA and SCL.

Note 1: The serial bus interface can be used only in NORMAL1/2 and IDLE1/2 mode. It can not be used in IDLE0, SLOW1/2 and SLEEP0/1/2 mode.

Note 2: The serial bus interface can be used only in the Standard mode of I^2C. The fast mode and the high-speed mode can not be used.

16.1 Configuration

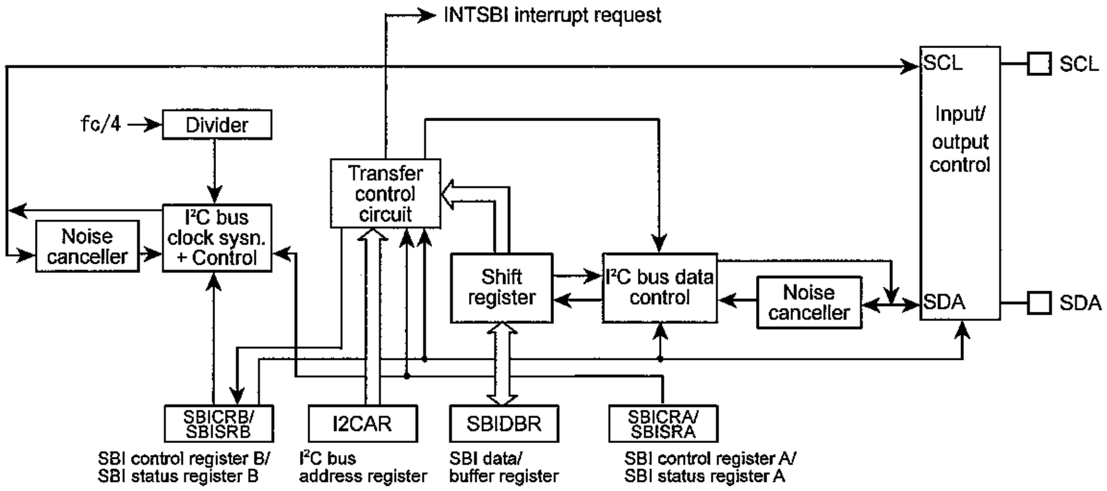


Figure 16-1 Serial Bus Interface (SBI)

16.2 Control

The following registers are used for control the serial bus interface and monitor the operation status.

- Serial bus interface control register A (SBICRA)
- Serial bus interface control register B (SBICRB)
- Serial bus interface data buffer register (SBIDBR)
- I^2C bus address register (I2CAR)
- Serial bus interface status register A (SBISRA)
- Serial bus interface status register B (SBISRB)

16.3 Software Reset

A serial bus interface circuit has a software reset function, when a serial bus interface circuit is locked by an external noise, etc.

To reset the serial bus interface circuit, write “10”, “01” into the SWRST (Bit1, 0 in SBICRB).

And a status of software reset canbe read from SWRMON (Bit0 in SBISRB).

16.4 The Data Format in the I^{2}C Bus Mode

The data format of the I^{2}C bus is shown below.

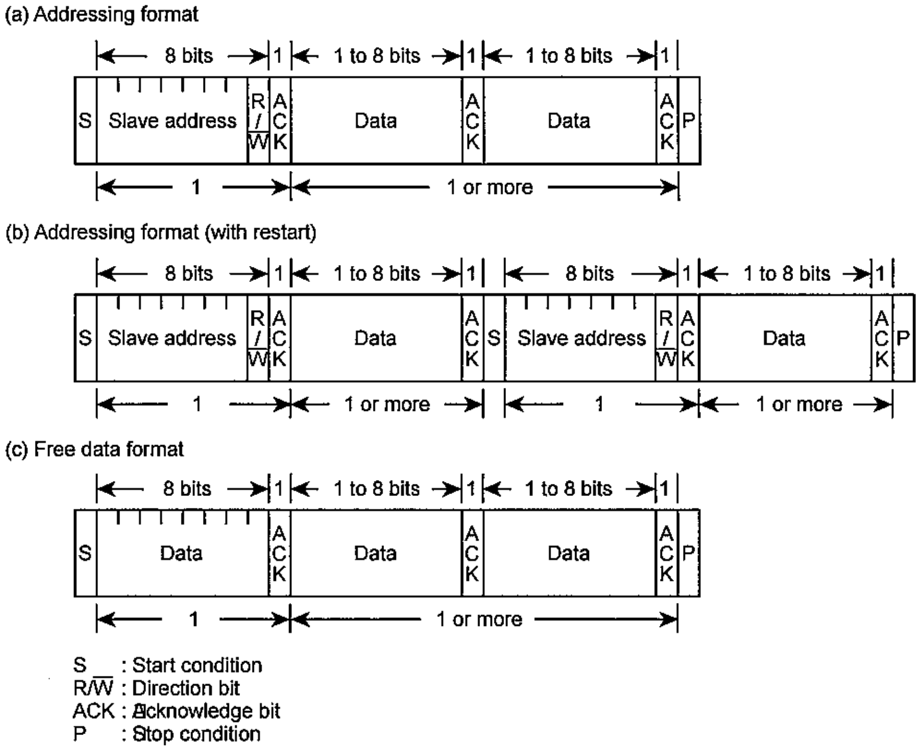


Figure 16-2 Data Format in of I^{2}C Bus

16.5 I^{2}C Bus Control

The following registers are used to control the serial bus interface (SBI) and monitor the operation status of the I^{2}C bus.

Serial Bus Interface Control Register A

	7	6	5	4	3	2	1	0	
SBICRA (0F90H)	BC			ACK		SCK			(Initial value: 0000 *000)

BC	Number of transferred bits	BC	ACK = 0		ACK = 1		Write only
			Number of Clock	Bits	Number of Clock	Bits	
		000:	8	8	9	8	
		001:	1	1	2	1	
		010:	2	2	3	2	
		011:	3	3	4	3	
		100:	4	4	5	4	
		101:	5	5	6	5	
		110:	6	6	7	6	
		111:	7	7	8	7	
ACK	Acknowledgement mode specification	ACK	Master mode		Slave mode		R/W
		0:	Not generate a clock pulse for an acknowledgement.		Not count a clock pulse for an acknowledgement.		
		1:	Generate a clock pulse for an acknowledgement.		Count a clock pulse for an acknowledgement.		
SCK	Serial clock (fscl) selection (Output on SCL pin) [fscl = $1/(2^{n+1}/fc + 8/fc)$]	SCK	n	At fc = 16 MHz	At fc = 8 MHz	At fc = 4 MHz	Write only
		000:	4	Reserved	Reserved	100.0 kHz	
		001:	5	Reserved	Reserved	55.6 kHz	
		010:	6	Reserved	58.8 kHz	29.4 kHz	
		011:	7	60.6 kHz	30.3 kHz	15.2 kHz	
		100:	8	30.8 kHz	15.4 kHz	7.7 kHz	
		101:	9	15.5 kHz	7.8 kHz	3.9 kHz	
		110:	10	7.8 kHz	3.9 kHz	1.9 kHz	
		111:	Reserved				

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

Note 2: Set the BC to "000" before switching to 8-bit SIO bus mode.

Note 3: SBICRA cannot be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 4: Do not set SCK as the frequency that is over 100 kHz.

Serial Bus Interface Data Buffer Register

	7	6	5	4	3	2	1	0	
SBIDBR (0F91H)									(Initial value: **** ****) R/W

Note 1: For writing transmitted data, start from the MSB (Bit7).

Note 2: The data which was written into SBIDBR can not be read, since a write data buffer and a read buffer are independent in SBIDBR. Therefore, SBIDBR cannot be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 3: *: Don't care

I^{2}C bus Address Register

I2CAR (0F92H)	7	6	5	4	3	2	1	0	
	Slave address							ALS	(Initial value: 0000 0000)
	SA6	SA5	SA4	SA3	SA2	SA1	SA0		

SA	Slave address selection		Write only
ALS	Address recognition mode specification	0: Slave address recognition 1: Non slave address recognition	

Note 1: I2CAR is write-only register, which cannot be used with any of read-modify-write instruction such as bit manipulation, etc.

Note 2: Do not set I2CAR to "00H" to avoid the incorrect response of acknowledgment in slave mode. If "00H" is set to I2CAR as the Slave Address and received "01H" in slave mode, the device might transmit the acknowledgment incorrectly.

Serial Bus Interface Control Register B

SBICRB (0F93H)	7	6	5	4	3	2	1	0	
	MST	TRX	BB	PIN	SBIM		SWRST1	SWRST0	(Initial value: 0001 0000)

MST	Master/slave selection	0: Slave 1: Master	Write only
TRX	Transmitter/receiver selection	0: Receiver 1: Transmitter	
BB	Start/stop generation	0: Generate a stop condition when MST, TRX and PIN are "1" 1: Generate a start condition when MST, TRX and PIN are "1"	
PIN	Cancel interrupt service request	0: – 1: Cancel interrupt service request	
SBIM	Serial bus interface operating mode selection	00: Port mode (Serial bus interface output disable) 01: Reserved 10: I^{2}C bus mode 11: Reserved	
SWRST1 SWRST0	Software reset start bit	Software reset starts by first writing "10" and next writing "01"	

Note 1: Switch a mode to port after confirming that the bus is free.

Note 2: Switch a mode to I^{2}C bus mode after confiming that the port is high level.

Note 3: SBICRB has write-only register and must not be used with any of read-modify-write instructions such as bit manipulation, etc.

Note 4: When the SWRST (Bit1, 0 in SBICRB) is written to "10", "01" in I^{2}C bus mode, software reset is occurred. In this case, the SBICRA, I2CAR, SBISRA and SBISRB registers are initialized and the bits of SBICRB except the SBIM (Bit3, 2 in SBI-CRB) are also initialized.

Serial Bus Interface Status Register A

SBISRA (0F90H)	7	6	5	4	3	2	1	0	
								SWRMON	(Initial value: **** ***1)

SWRMON	Software reset monitor	0: During software reset 1: –	Read only

Serial Bus Interface Status Register B

SBISRB (0F93H)	7	6	5	4	3	2	1	0	
	MST	TRX	BB	PIN	AL	AAS	AD0	LRB	(Initial value: 0001 0000)

MST	Master/slave selection status monitor	0: Slave 1: Master	Read only
TRX	Transmitter/receiver selection status monitor	0: Receiver 1: Transmitter	
BB	Bus status monitor	0: Bus free 1: Bus busy	
PIN	Interrupt service requests status monitor	0: Requesting interrupt service 1: Releasing interrupt service request	
AL	Arbitration lost detection monitor	0: - 1: Arbitration lost detected	
AAS	Slave address match detection monitor	0: Not detect slave address match or "GENERAL CALL" 1: Detect slave address match or "GENERAL CALL"	
AD0	"GENERAL CALL" detection monitor	0: Not detect "GENERAL CALL" 1: Detect "GENERAL CALL"	
LRB	Last received bit monitor	0: Last receive bit is "0" 1: Last receiv bit is "1"	

16.5.1 Acknowledgement mode specification

16.5.1.1 Acknowledgment mode (ACK = "1")

To set the device as an acknowledgment mode, the ACK (Bit4 in SBICRA) should be set to "1". When a serial bus interface circuit is a master mode, an additional clock pulse is generated for an acknowledge signal. In a slave mode, a clock is counted for the acknowledge signal.

In the master transmitter mode, the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In the master receiver mode, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle.

In a slave mode, when a received slave address matches to a slave address which is set to the I2CAR or when a "GENERAL CALL" is received, the SDA pin is set to low level generating an acknowledge signal. After the matching of slave address or the detection of "GENERAL CALL", in the transmitter, the SDA pin is released in order to receive an acknowledge signal from the receiver during additional clock pulse cycle. In a receiver, the SDA pin is set to low level generation an acknowledge signal during additional clock pulse cycle after the matching of slave address or the detection of "GENERAL CALL"

The Table 16-1 shows the SCL and SDA pins status in acknowledgment mode.

Table 16-1 SCL and SDA Pins Status in Acknowledgement Mode

Mode	Pin		Transmitter	Receiver
Master	SCL		An additional clock pulse is generated.	
	SDA		Released in order to receive an acknowledge signal.	Set to low level generating an acknowledge signal
Slave	SCL		A clock is counted for the acknowledge signal.	
	SDA	When slave address matches or a general call is detected	–	Set to low level generating an acknowledge signal.
		After matching of slave address or general call	Released in order to receive an acknowledge signal.	Set to low level generating an acknowledge signal.

16.5.1.2 Non-acknowledgment mode (ACK = “0”)

To set the device as a non-acknowledgement mode, the ACK should be cleared to “0”.

In the master mode, a clock pulse for an acknowledge signal is not generated.

In the slave mode, a clock for a acknowledge signal is not counted.

16.5.2 Number of transfer bits

The BC (Bits7 to 5 in SBICRA) is used to select a number of bits for next transmitting and receiving data.

Since the BC is cleared to “000” by a start condition, a slave address and direction bit transmissions are always executed in 8 bits. Other than these, the BC retains a specified value.

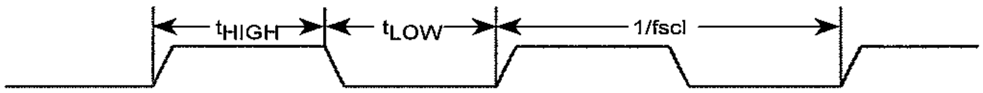
16.5.3 Serial clock

16.5.3.1 Clock source

The SCK (Bits2 to 0 in SBICRA) is used to select a maximum transfer frequency output from the SCL pin in the master mode.

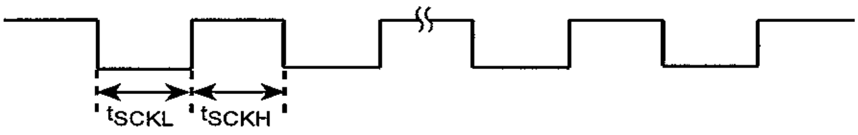
Four or more machine cycles are required for both high and low levels of pulse width in the external clock which is input from SCL pin.

Note: Since the serial bus interface can not be used as the fast mode and the high-speed mode, do not set SCK as the frequency that is over 100 kHz.



$t_{LOW} = 2^n/fc$
$t_{HIGH} = 2^n/fc + 8/fc$
$fscl = 1/(t_{LOW} + t_{HIGH})$

SCK (Bits2 to 0 in the SBICRA)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10



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

Note 1: fc = High-frequency clock
Note 2: tcyc = 4/fc (in NORMAL mode, IDLE mode)

Figure 16-3 Clock Source

16.5.3.2 Clock synchronization

In the I^2C bus, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate a clock pulse of another master device which generates a high-level clock pulse.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

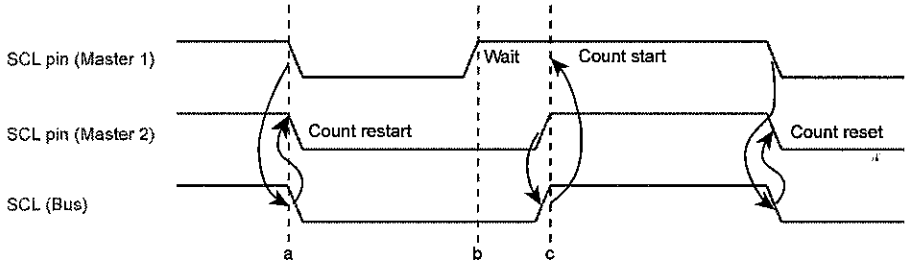


Figure 16-4 Clock Synchronization

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is determined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

16.5.4 Slave address and address recognition mode specification

When the serial bus interface circuit is used with an addressing format to recognize the slave address, clear the ALS (Bit0 in I2CAR) to "0", and set the SA (Bits7 to 1 in I2CAR) to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set the ALS to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after start condition.

16.5.5 Master/slave selection

To set a master device, the MST (Bit7 in SBICRB) should be set to "1". To set a slave device, the MST should be cleared to "0".

When a stop condition on the bus or an arbitration lost is detected, the MST is cleared to "0" by the hardware.

16.5.6 Transmitter/receiver selection

To set the device as a transmitter, the TRX (Bit6 in SBICRB) should be set to "1". To set the device as a receiver, the TRX should be cleared to "0". When data with an addressing format is transferred in the slave mode, the TRX is set to "1" by a hardware if the direction bit ($R/\overline{W}$) sent from the master device is "1", and is cleared to "0" by a hardware if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device, the TRX is cleared to "0" by a hardware if a transmitted direction bit is "1", and is set to "1" by a hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, the TRX is cleared to "0" by the hardware. Table 16-2 shows TRX changing conditions in each mode and TRX value after changing

Table 16-2 TRX changing conditions in each mode

Mode	Direction Bit	Conditions	TRX after Changing
Slave Mode	"0"	A received slave address is the same value set to I2CAR	"0"
	"1"		"1"
Master Mode	"0"	ACK signal is returned	"1"
	"1"		"0"

When a serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating a start condition. The TRX is not changed by a hardware.

16.5.7 Start/stop condition generation

When the BB (Bit5 in SBISRB) is "0", a slave address and a direction bit which are set to the SBIDBR are output on a bus after generating a start condition by writing "1" to the MST, TRX, BB and PIN. It is necessary to set ACK to "1" beforehand.

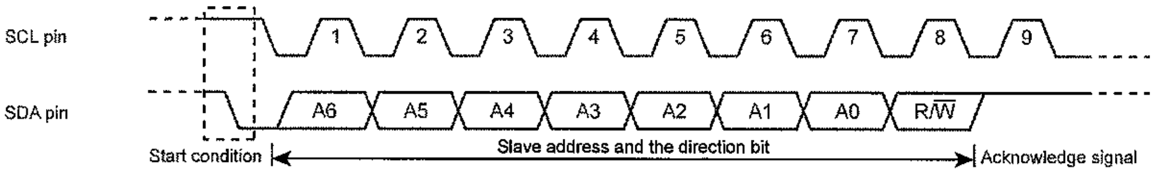


Figure 16-5 Start Condition Generation and Slave Address Generation

When the BB is "1", sequence of generating a stop condition is started by writing "1" to the MST, TRX and PIN, and "0" to the BB. Do not modify the contents of MST, TRX, BB and PIN until a stop condition is generated on a bus.

When a stop condition is generated and the SCL line on a bus is pulled-down to low level by another device, a stop condition is generated after releasing the SCL line.

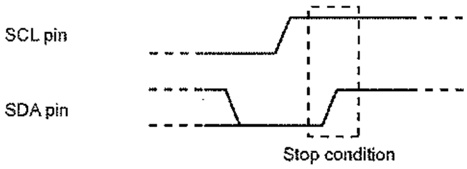


Figure 16-6 Stop Condition Generation

The bus condition can be indicated by reading the contents of the BB (Bit5 in SBISRB). The BB is set to "1" when a start condition on a bus is detected and is cleared to "0" when a stop condition is detected.

16.5.8 Interrupt service request and cancel

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by the BC and the ACK is complete, a serial bus interface interrupt request (INTSBI) is generated.

In the slave mode, the conditions of generating INTSBI are follows:

- At the end of acknowledge signal when the received slave address matches to the value set by the I2CAR
- At the end of acknowledge signal when a "GENERAL CALL" is received
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

When a serial bus interface interrupt request occurs, the PIN (Bit4 in SBISRB) is cleared to "0". During the time that the PIN is "0", the SCL pin is pulled-down to low level.

Either writing data to SBIDBR or reading data from the SBIDBR sets the PIN to "1".

The time from the PIN being set to "1" until the SCL pin is released takes tLOW.

Although the PIN (Bit4 in SBICRB) can be set to "1" by the program, the PIN can not be cleared to "0" by the program.

Note: Regarding the status of PIN when the arbitration lost occurs, refer to Table 16-3.

16.5.9 Setting of I^{2}C bus mode

The SBIM (Bit3 and 2 in SBICRB) is used to set I^{2}C bus mode.

Set the SBIM to “10” in order to set I^{2}C bus mode. Before setting of I^{2}C bus mode, confirm serial bus interface pins in a high level, and then, write “10” to SBIM. And switch a port mode after confirming that a bus is free.

16.5.10Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^{2}C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point “a”. After that, when Master 1 outputs “1” and Master 2 outputs “0”, since the SDA line of a bus is wired AND, the SDA line is pulled-down to the low level by Master 2. When the SCL line of a bus is pulled-up at point “b”, the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master 1 is called “arbitration lost”. A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

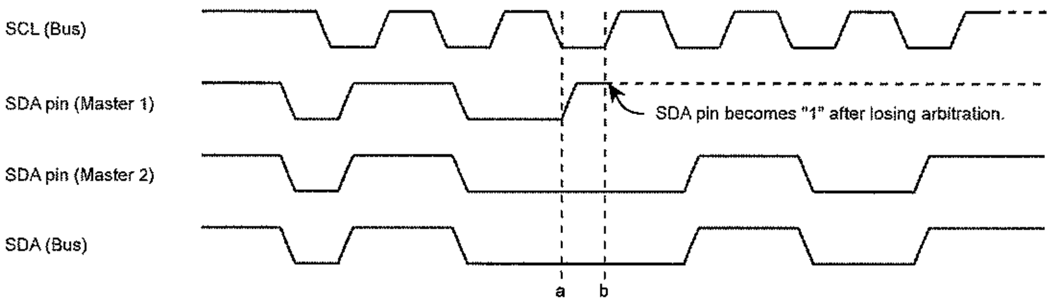


Figure 16-7 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (Bit3 in SBISRB) is set to “1”.

When the AL is set to “1”, the MST and TRX are cleared to “0” and the mode is switched to a slave receiver mode. Thus, the serial bus interface circuit stops output of clock pulses during data transfer after the AL is set to “1”.

The AL is cleared to “0” by writing data to the SBIDBR, reading data from the SBIDBR or writing data to the SBICRB.

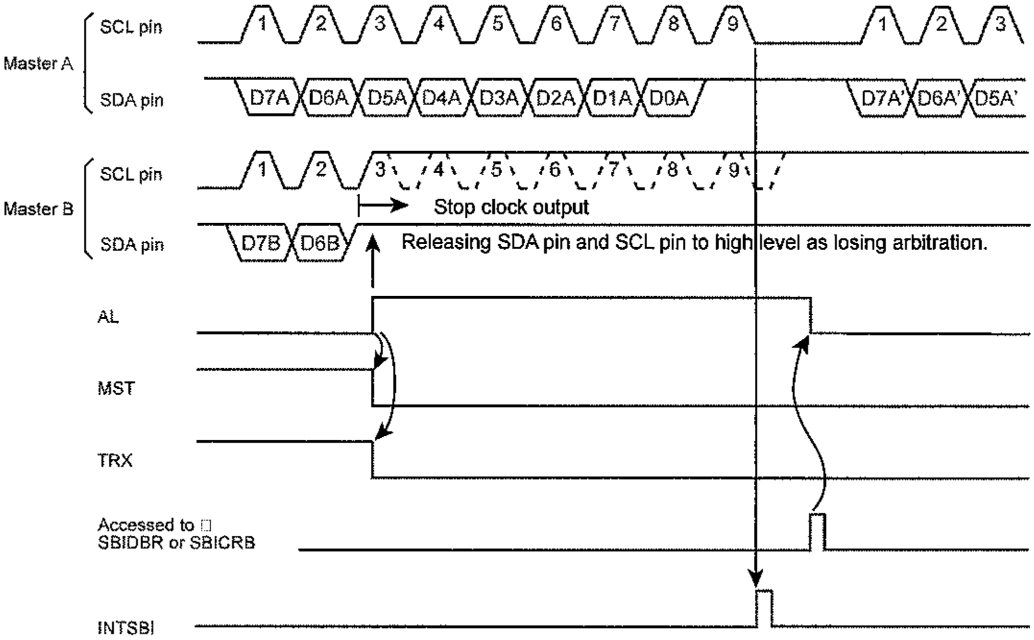

Master A
SCL pin
SDA pin
1 2 3 4 5 6 7 8 9 1 2 3
D7A D6A D5A D4A D3A D2A D1A D0A D7A' D6A' D5A'
Master B
SCL pin
SDA pin
D7B D6B
Stop clock output
Releasing SDA pin and SCL pin to high level as losing arbitration.
AL
MST
TRX
Accessed to SBIDBR or SBICRB
INTSBI

Figure 16-8 Example of when a Serial Bus Interface Circuit is a Master B

16.5.11Slave address match detection monitor

In the slave mode, the AAS (Bit2 in SBISRB) is set to "1" when the received data is "GENERAL CALL" or the received data matches the slave address setting by I2CAR with an address recognition mode (ALS = 0).

When a serial bus interface circuit operates in the free data format (ALS = 1), the AAS is set to "1" after receiving the first 1-word of data.

The AAS is cleared to "0" by writing data to the SBIDBR or reading data from the SBIDBR.

16.5.12GENERAL CALL detection monitor

The AD0 (Bit1 in SBISRB) is set to "1" when all 8-bit received data is "0" immediately after a start condition in a slave mode. The AD0 is cleared to "0" when a start or stop condition is detected on a bus.

16.5.13Last received bit monitor

The SDA value stored at the rising edge of the SCL is set to the LRB (Bit0 in SBISRB). In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the LRB.

16.6 Data Transfer of I^{2}C Bus

16.6.1 Device initialization

For initialization of device, set the ACK in SBICRA to "1" and the BC to "000". Specify the data length to 8 bits to count clocks for an acknowledge signal. Set a transfer frequency to the SCK in SBICRA.

Next, set the slave address to the SA in I2CAR and clear the ALS to "0" to set an addressing format.

After confirming that the serial bus interface pin is high level, for specifying the default setting to a slave receiver mode, clear "0" to the MST, TRX and BB in SBICRB, set "1" to the PIN, "10" to the SBIM, and "00" to bits SWRST1 and SWRST0.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

16.6.2 Start condition and slave address generation

Confirm a bus free status (BB = 0).

Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.

By writing "1" to the MST, TRX, BB and PIN, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBIDBR are output. An INTSBI interrupt request occurs at the 9th falling edge of a SCL clock cycle, and the PIN is cleared to "0". The SCL pin is pulled-down to the low level while the PIN is "0". When an interrupt request occurs, the TRX changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: Do not write a slave address to be output to the SBIDBR while data is transferred. If data is written to the SBIDBR, data to been outputting may be destroyed.

Note 2: The bus free must be confirmed by software within 98.0 μs (The shortest transmitting time according to the I^2C bus standard) after setting of the slave address to be output. Only when the bus free is confirmed, set "1" to the MST, TRX, BB, and PIN to generate the start conditions. If the writing of slave address and setting of MST, TRX, BB and PIN doesn't finish within 98.0 μs, the other masters may start the transferring and the slave address data written in SBIDBR may be broken.

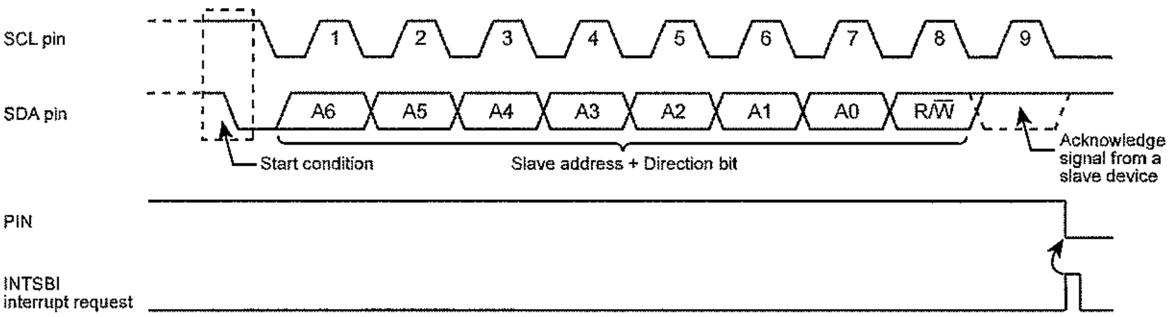


Figure 16-9 Start Condition Generation and Slave Address Transfer

16.6.3 1-word data transfer

Check the MST by the INTSBI interrupt process after an 1-word data transfer is completed, and determine whether the mode is a master or slave.

16.6.3.1 When the MST is "1" (Master mode)

Check the TRX and determine whether the mode is a transmitter or receiver.

(1) When the TRX is "1" (Transmitter mode)

Test the LRB. When the LRB is "1", a receiver does not request data. Implement the process to generate a stop condition (Described later) and terminate data transfer.

When the LRB is "0", the receiver requests next data. When the next transmitted data is other than 8 bits, set the BC, set the ACK to "1", and write the transmitted data to the SBIDBR. After writing the data, the PIN becomes "1", a serial clock pulse is generated for transferring a next 1 word of data from the SCL pin, and then the 1 word of data is transmitted. After the data is transmitted, and an INTSBI interrupt request occurs. The PIN become "0" and the SCL pin is set to low level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB test above.

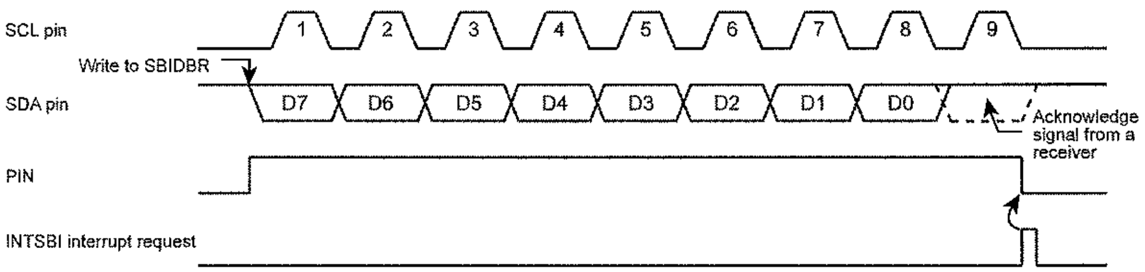


Figure 16-10 Example of when BC = "000", ACK = "1"

(2) When the TRX is "0" (Receiver mode)

When the next transmitted data is other than of 8 bits, set the BC again. Set the ACK to "1" and read the received data from the SBIDBR (Reading data is undefined immediately after a slave address is sent). After the data is read, the PIN becomes "1". A serial bus interface circuit outputs a serial clock pulse to the SCL to transfer next 1-word of data and sets the SDA pin to "0" at the acknowledge signal timing.

An INTSBI interrupt request occurs and the PIN becomes "0". Then a serial bus interface circuit outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

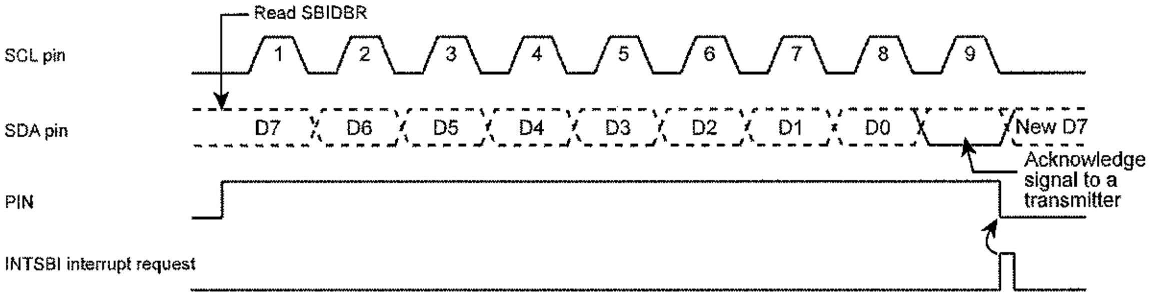


Figure 16-11 Example of when BC = "000", ACK = "1"

To make the transmitter terminate transmit, clear the ACK to "0" before reading data which is 1-word before the last data to be received. A serial bus interface circuit does not generate a clock pulse for the acknowledge signal by clearing ACK. In the interrupt routine of end of transmission, when the BC is set to "001" and read the data, PIN is set to "1" and generates a clock pulse for a 1-bit data transfer. In this case, since the master device is a receiver, the SDA line on a bus keeps the high-level. The transmitter receives the high-level signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

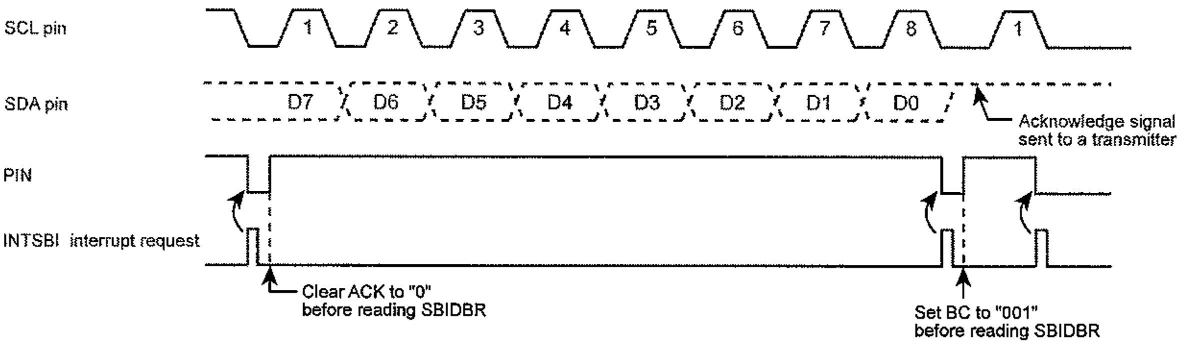

SCL pin
1
2
3
4
5
6
7
8
1
SDA pin
D7
D6
D5
D4
D3
D2
D1
D0
Acknowledge signal sent to a transmitter
PIN
INTSBI interrupt request
Clear ACK to "0" before reading SBIDBR
Set BC to "001" before reading SBIDBR

Figure 16-12 Termination of Data Transfer in Master Receiver Mode

16.6.3.2 When the MST is “0” (Slave mode)

In the slave mode, a serial bus interface circuit operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, the conditions of generating INTSBI are follows:

- When the received slave address matches to the value set by the I2CAR
- When a “GENERAL CALL” is received
- At the end of transferring or receiving after matching of slave address or receiving of “GENERAL CALL”

A serial bus interface circuit changes to a slave mode if arbitration is lost in the master mode. And an INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. The behavior of INTSBI and PIN after losing arbitration are shown in Table 16-3.

Table 16-3 The Behavior of INTSBI and PIN after Losing Arbitration

	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as a Master Transmit Mode
INTSBI	INTSBI is generated at the termination of word data.	
PIN	When the slave address matches the value set by I2CAR, the PIN is cleared to "0" by generating of INTSBI. When the slave address doesn't match the value set by I2CAR, the PIN keeps "1".	PIN keeps "1".

When an INTSBI occurs, the PIN (bit 4 in the SBICRB) is reset, and the SCL pin is set to low level. Either reading or writing from or to the SBIDBR or setting the PIN to “1” releases the SCL pin after taking t_{LOW} time.

Check the AL (Bit3 in the SBISR), the TRX (Bit6 in the SBISR), the AAS (Bit2 in the SBISR), and the AD0 (Bit1 in the SBISR) and implements processes according to conditions listed in Table 16-4.

Table 16-4 Operation in the Slave Mode

TRX				Conditions	Process
1	1	1	0	A serial bus interface circuit loses arbitration when transmitting a slave address. And receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR.
	0	1	0	In the slave receiver mode, a serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Test the LRB. If the LRB is set to "1", set the PIN to "1" since the receiver does not request next data. Then, clear the TRX to "0" to release the bus. If the LRB is set to "0", set the number of bits in 1 word to the BC and write transmitted data to the SBIDBR since the receiver requests next data.
0	1	1	1/0	A serial bus interface circuit loses arbitration when transmitting a slave address. And receives a slave address of which the value of the direction bit sent from another master is "0" or receives a "GENERAL CALL".	Read the SBIDBR for setting the PIN to "1" (Reading dummy data) or write "1" to the PIN.
		0	0	A serial bus interface circuit loses arbitration when transmitting a slave address or data. And terminates transferring word data.	A serial bus interface circuit is changed to slave mode. To clear AL to "0", read the SBIDBR or write the data to SBIDBR.
	0	1	1/0	In the slave receiver mode, a serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "0" or receives "GENERAL CALL".	Read the SBIDBR for setting the PIN to "1" (Reading dummy data) or write "1" to the PIN.
		0	1/0	In the slave receiver mode, a serial bus interface circuit terminates receiving of 1-word data.	Set the number of bits in 1-word to the BC and read received data from the SBIDBR.

Note: In the slave mode, if the slave address set in I2CAR is "00000000B", the TRX changes to "1" by receiving the start byte data "00000001B"

16.6.4 Stop condition generation

When the BB is "1", a sequence of generating a stop condition is started by setting "1" to the MST, TRX and PIN, and clear "0" to the BB. Do not modify the contents of the MST, TRX, BB, PIN until a stop condition is generated on a bus.

When a SCL line on a bus is pulled-down by other devices, a serial bus interface circuit generates a stop condition after they release a SCL line.

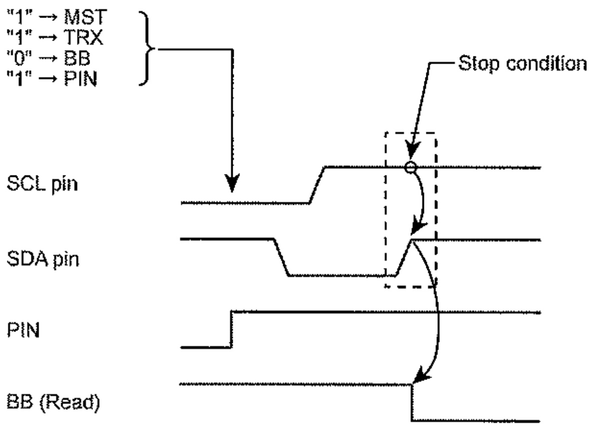


Figure 16-13 Stop Condition Generation

16.6.5 Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart a serial bus interface circuit.

Clear "0" to the MST, TRX and BB and set "1" to the PIN. The SDA pin retains the high-level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Test the BB until it becomes "0" to check that the SCL pin of a serial bus interface circuit is released. Test the LRB until it becomes "1" to check that the SCL line on a bus is not pulled-down to the low level by other devices. After confirming that a bus stays in a free state, generate a start condition with procedure 16.6.2.

In order to meet setup time when restarting, take at least 4.7 µs of waiting time by software from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When restarting after receiving in master recever mode, because the device doesn't send an acknowledgment as a last data, the level of SCL line can not be confirmed by reading LRB. Therefore, confirm the status of SCL line by checking a status of port.

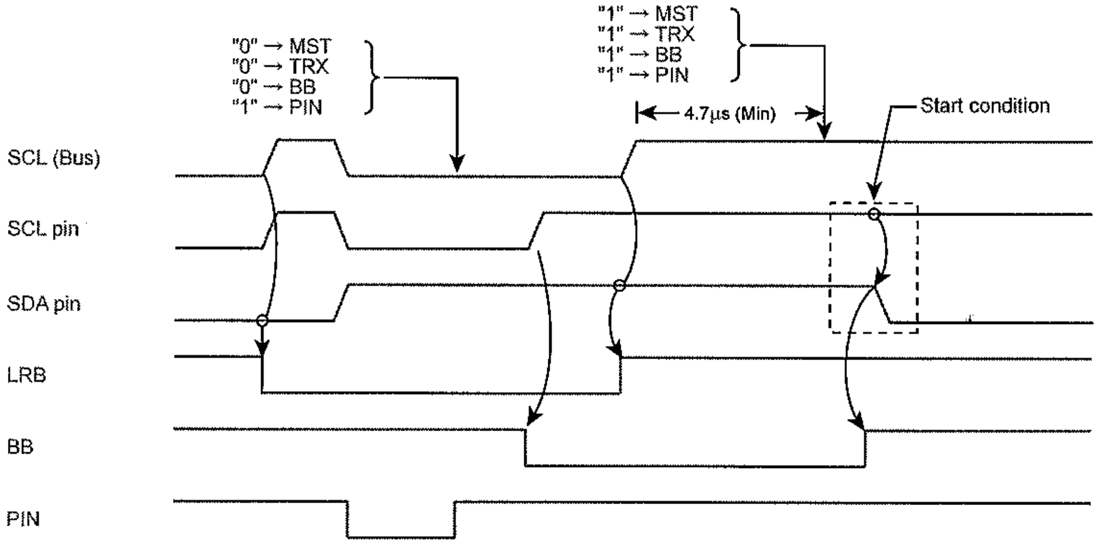


Figure 16-14 Timing Diagram when Restarting

17. 10-Bit AD Converter (ADC)

The TMP86FS49UG have a 10-bit successive approximation type AD converter.

17.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 17-1.

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

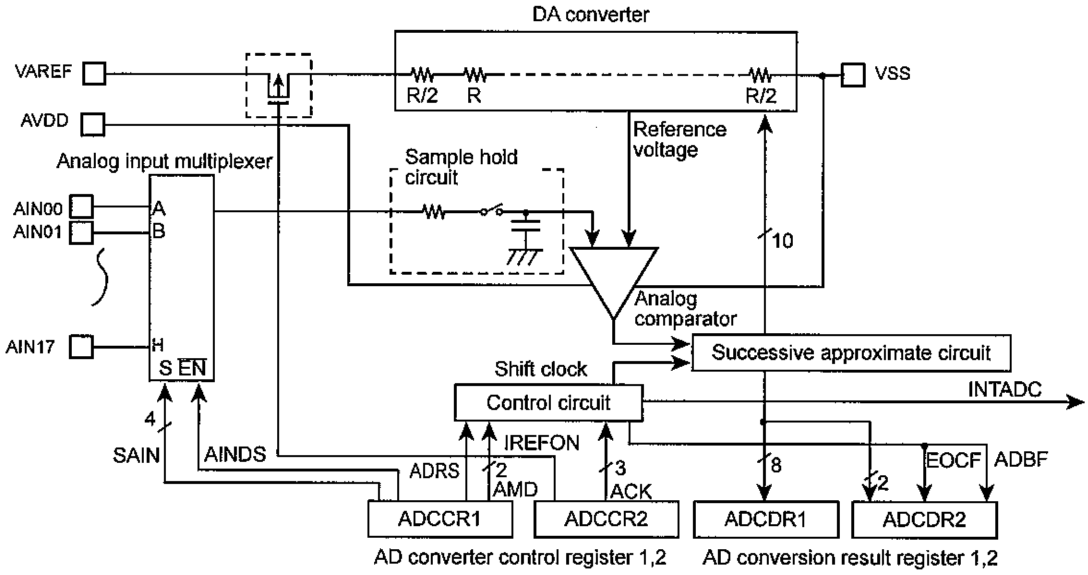


Figure 17-1 AD Converter (ADC)

17.2 Register Configuration

The AD converter consists of the following four registers:

- AD converter control register 1 (ADCCR1)
- AD converter control register 2 (ADCCR2)
- AD converted value register 1/2 (ADCDR1/ADCDR2)

17.2.1 AD converter control register 1 (ADCCR1)

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

17.2.2 AD converter control register 2 (ADCCR2)

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

17.2.3 AD converted value register (ADCDR1)

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

17.2.4 AD converted value register (ADCDR2)

This register monitors the operating status of the AD converter.

The AD converter control register configurations are shown in as follows.

AD Converter Control Register 1

ADCCR1	7	6	5	4	3	2	1	0	
(001CH)	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: Repeat mode	
AINDS	Analog input control	0: Analog input enable 1: Analog input disable	
SAIN	Analog input channel select	0000: AIN00 0001: AIN01 0010: AIN02 0011: AIN03 0100: AIN04 0101: AIN05 0110: AIN06 0111: AIN07 1000: AIN10 1001: AIN11 1010: AIN12 1011: AIN13 1100: AIN14 1101: AIN15 1110: AIN16 1111: AIN17	

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 EOCF (ADCDR bit5) 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 is 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 (001DH)	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	ACK	Conversion time	fc = 16 MHz	fc = 8 MHz	fc = 4 MHz	fc = 2 MHz	
		000	39/fc	–	–	–	19.5 µs	
		001	Reserved					
		010	78/fc	–	–	19.5 µs	39.0 µs	
		011	156/fc	–	19.5 µs	39.0 µs	78.0 µs	
		100	312/fc	19.5 µs	39.0 µs	78.0 µs	156.0 µs	
		101	624/fc	39.0 µs	78.0 µs	156.0 µs	–	
		110	1248/fc	78.0 µs	156.0 µs	–	–	
		111	Reserved					

Note 1: Settings for "–" in the above table are inhibited.

Note 2: Set conversion time by analog reference voltage (VAREF) as follows.
VAREF = 4.5 to 5.5 V (15.6 µs, or more)
VAREF = 3.0 to 3.6 V (31.2 µs, or more)

Note 3: Always set bit 0 in ADCCR2 to "0" and set bit4 in ADCCR2 to "1".

Note 4: When a read instruction for ADCCR2, bit6 to 7 in ADCCR2 read in as undefined data.

Note 5: fc: High-frequency clock [Hz]

Note 6: After STOP or SLOW mode is started, AD converter control register 2 (ADCCR2) is all initialized.
Therefore, set the ADCCR2 newly again after exiting these modes.

AD Conversion Result Register

ADCDR1 (001FH)	7	6	5	4	3	2	1	0	
	AD09	AD08	AD07	AD06	AD05	AD04	AD03	AD02	(Initial value: 0000 000*)

ADCDR2 (001EH)	7	6	5	4	3	2	1	0	
	AD01	AD00	EOCF	ADBF					(Initial value: 0000 ****)

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

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

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

Note 3: If a read instruction is executed for ADCDR2, read data of bit3 to 0 are unstable.

17.3 AD Converter Operation

1. Set up the AD converter control register 1 (ADCCR1) as follows:
   - Choose the channel to AD convert using AD input channel select (SAIN).
   - Specify analog input enable for analog input control (AINDS).
   - Specify AMD for the AD converter control operation mode (software or repeat mode).
2. 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". If software start mode has been selected, AD conversion starts immediately.
4. After an elapse of the specified AD conversion time, the AD converted value is stored in AD converted value register 1 (ADCDR1) and the AD conversion finished flag (EOCF) of AD converted value 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.

17.4 AD Converter Operation Modes

There are following two AD converter operation modes:

- Software start: AD conversion is performed once by setting AMD to "01" and ADRS to "1" .
- Repeat mode: AD conversion is performed repeatedly by setting AMD to "11" and ADRS to "1" .

17.4.1 Software start mode

After setting AMD (ADCCR1 bits 6, 5) to "01" (software start mode), set ADRS (ADCCR1 bit7) to "1". AD conversion of the voltage at the analog input pin specified by SAIN (ADCCR1 bits 0 to 3) is thereby started.

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time EOCF (ADCDR2 bit5) is set to 1, the AD conversion finished interrupt (INTADC) is generated.

ADRS is automatically cleared after AD conversion has started. Do not set ADRS (ADCCR1 bit7) newly again (Restart) during AD conversion. Before setting ADRS newly again, check EOCF (ADCDR bit5) to see that the conversion is completed or wait until the interrupt signal (INTADC) is generated (e.g., interrupt handling routine).

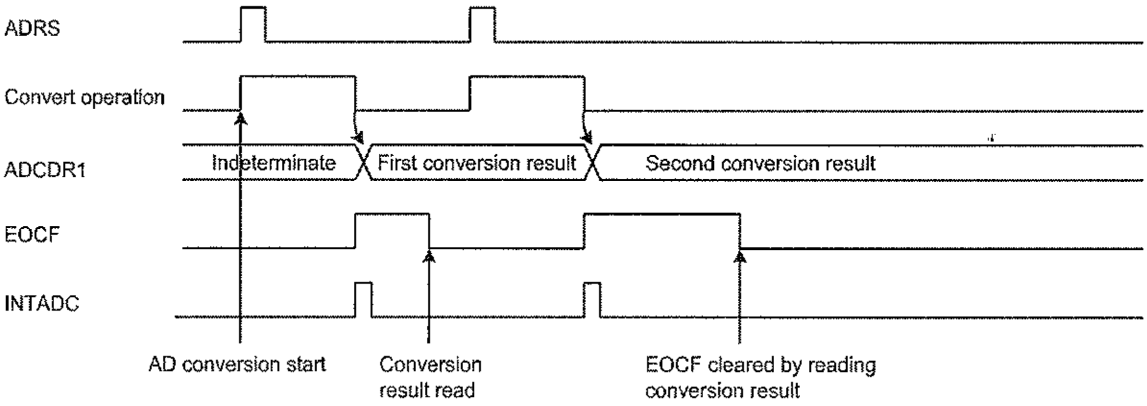


Figure 17-2 Operation in Software Start Mode

17.4.2 Repeat mode

AD conversion of the voltage at the analog input pin specified by SAIN (ADCCR1 bits 0 to 3) is performed repeatedly. In this mode, AD conversion is started by setting ADRS (ADCCR1 bit7) to "1" after setting AMD (ADCCR1 bits 5, 6) to "11".

After completion of the AD conversion, the conversion result is stored in AD converted value registers (ADCDR1, ADCDR2) and at the same time EOCF (ADCDR2 bit5) is set to 1, the AD conversion finished interrupt (INTADC) is generated.

In repeat mode, each time one AD conversion is completed, the next AD conversion is started. To stop AD conversion, set AMD (ADCCR1 bits 5, 6) to "00" (Disable mode) by writing 0s. The AD convert operation is stopped immediately. The converted value at this time is not stored in the AD converted value register.

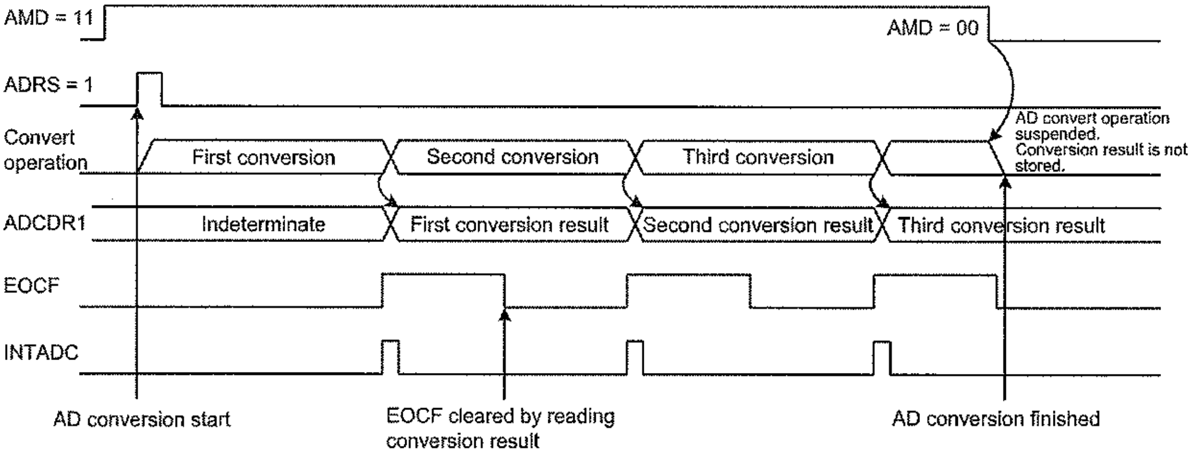


Figure 17-3 Operation in Repeat Mode

17.5 STOP and SLOW Modes during AD Conversion

When standby mode (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 standby mode.) When restored from standby mode, AD conversion is not automatically restarted, so it is necessary to restart AD conversion. Note that since the analog reference voltage is automatically disconnected, there is no possibility of current flowing into the analog reference voltage.

17.6 Analog Input Voltage and AD Conversion Result

Example :After selecting the conversion time of 19.5 ms at 16 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, read the converted value, store the lower 2 bits in address 009EH and store the upper 8 bits in address 009FH on RAM. The operation mode is software start mode.

```
                                            ; AIN SELECT
          LD      (P6CR2), 00000000B        ; P6CR2 bit3 = 0
          LD      (P6CR1), 00000000B        ; P6CR1 bit3 = 0
          LD      (ADCCR1), 00100011B       ; Select AIN3
          LD      (ADCCR2), 11011000B       ; Select conversion time (312/fc) and operation mode
                                            ; AD CONVERT START
          SET     (ADCCR1). 7               ; ADRS = 1
SLOOP     TEST    (ADCDR2). 5               ; EOCF = 1 ?
          JRS     T, SLOOP
                                            ; RESULT DATA READ
          LD      A, (ADCDR2)
          LD      (9EH), A
          LD      A, (ADCDR1)
          LD      (9FH), A
```

The analog input voltage is corresponded to the 10-bit digital value converted by the AD as shown in Figure 17-4.

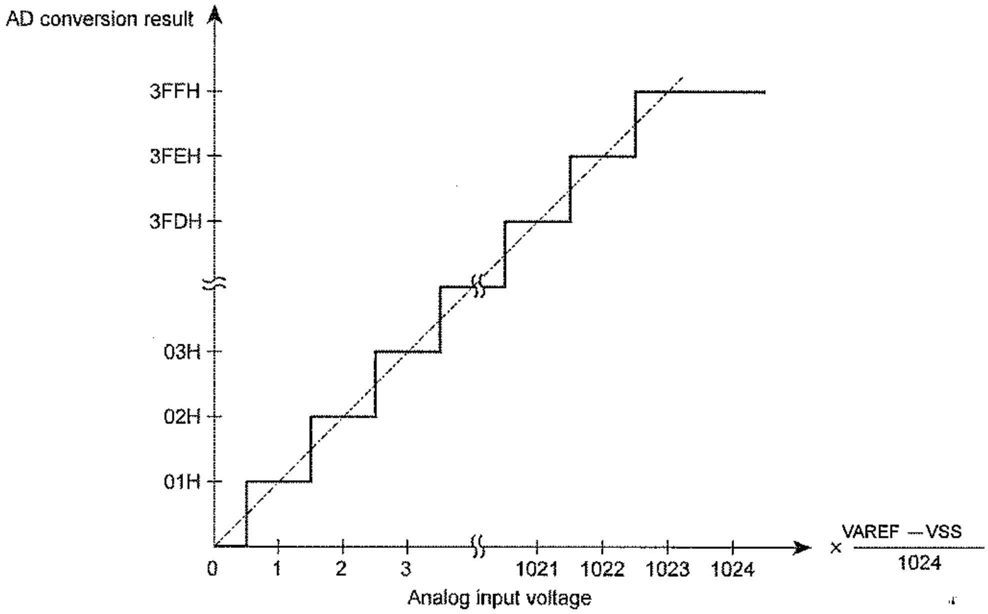


Figure 17-4 Analog Input Voltage and AD Conversion Result (Typ.)

17.7 Precautions about AD Converter

17.7.1 Analog input pin voltage range

Make sure the analog input pins (AIN00 to AIN17) 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.

17.7.2 Analog input shared pins

The analog input pins (AIN00 to AIN17) 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.

17.7.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 17-5. 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.

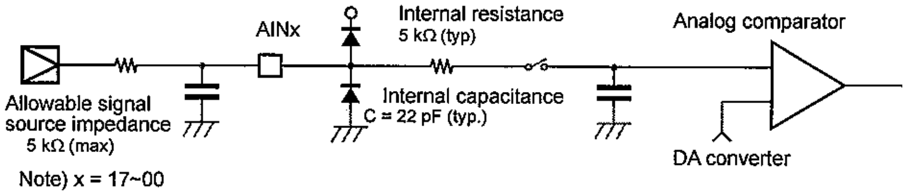
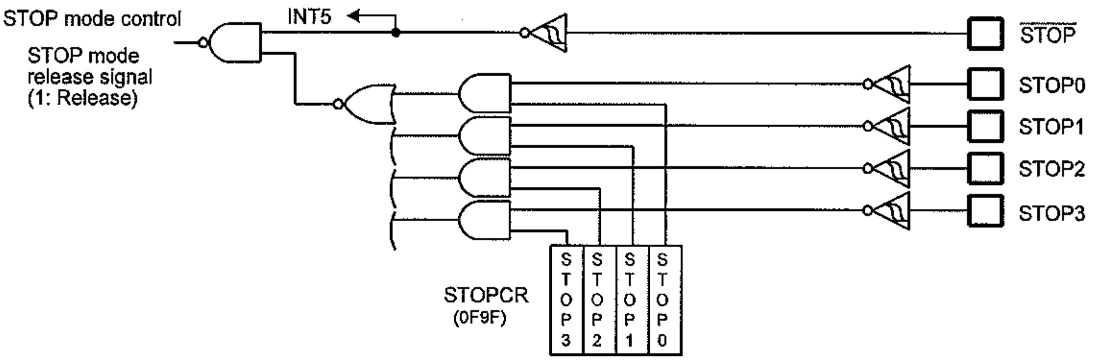


Figure 17-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

18. Key-on Wakeup (KWU)

In the TMP86FS49UG, the STOP mode must be released by not only $\overline{\text{STOP}}$ pin but also STOP0 to STOP3 pins. When the STOP mode is released by STOP0 to STOP3 pins, the $\overline{\text{STOP}}$ pin needs to be used.

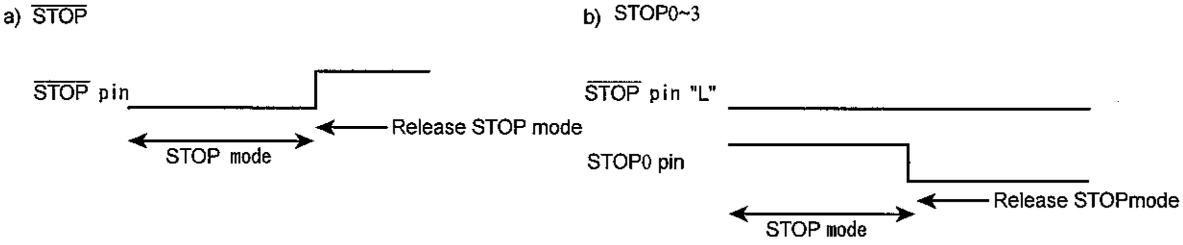
18.1 Configuration



Note: To use the Key-on Wake up function, corresponding shared I/Opins must be input mode. Refer to "I/O Ports" for detail.

Figure 18-1 STOP Mode Control Circuit

Note 1: When the STOP mode is released by STOP0 to 3 pin, the level of $\overline{\text{STOP}}$ pin should hold "L".



a) $\overline{\text{STOP}}$
$\overline{\text{STOP}}$ pin
Release STOP mode
STOP mode
b) STOP0~3
$\overline{\text{STOP}}$ pin "L"
STOP0 pin
Release STOPmode
STOP mode

Note: $\overline{\text{STOP}}$ pin doesn't have the control register such as STOPCR, so when STOP mode is relesed by STOPx (x: 0 to 3), $\overline{\text{STOP}}$ pin should be used as STOP function.

18.2 Control

STOP0 to STOP3 pins can be controlled by key-on wakeup control register (STOPCR). It can be configured as enable/disable in 1-bit unit. When those pins are used for STOP mode release, configure corresponding I/O pins to input mode beforehand.

STOP mode can be entered by setting up the system control registerl (SYSCR1), and can be exited by detecting the falling edge on STOP0 to 3 pins, which are enabled by STOPCR, for releasing STOP mode (Note 1). Also, each level of the STOP0 to 3 can be confirmed by reading corresponding Port Data Register, check all STOP0 to 3 pins that is enabled by STOPCR before the STOP mode is started (Note 2).

Note 1: When the STOP mode release by edge mode (SYSCR1<RELM> = "0"), inhibit input from STOP0 to STOP3 or must be set "1" level into STOP0 to STOP3 pins.

Note 2: When the $\overline{\text{STOP}}$ pin input is high or STOP0 to STOP3 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).

Key-on Wakeup Control Register

STOPCR (0F9FH)	7	6	5	4	3	2	1	0	
	STOP3	STOP2	STOP1	STOP0	–	–	–	–	(Initial value: 0000 ****)

STOP3	STOP mode released by STOP3 pin	0: Disable 1: Enable	Write only
STOP2	STOP mode released by STOP2 pin	0: Disable 1: Enable	
STOP1	STOP mode released by STOP1 pin	0: Disable 1: Enable	
STOP0	STOP mode released by STOP0 pin	0: Disable 1: Enable	

19. Flash Memory

The TMP86FS49UG has built-in 60-Kbyte flash memory (addresses: 1000H to FFFFH). In the MCU mode and the serial PROM mode, the write operations to the flash memory are controlled via the flash memory control register (FLSCR).

19.1 Control

The flash memory is controlled via the flash memory control register (FLSCR).

Flash Memory Control Register

	7	6	5	4	3	2	1	0	
FLSCR (0FFFH)	FLSMD				BANKSEL				(Initial value : 1100 1***)

FLSMD	Flash memory write enable control	1100: Disable flash memory write 0011: Enable flash memory write Others: Reserved	R/W
BANKSEL	Flash memory bank select control (Serial PROM mode only)	0: Select BANK0(1000H-7FFFH) 1: Select BANK1(8000H-FFFFH)	

Note 1: The flash memory can be written only when FLSMD="0011B". In other cases, any attempts to write to the flash memory will be ineffective.

Note 2: FLSMD must be set to either "1100B" or "0011B". Do not set these bits to any other values.

Note 3: BANKSEL is effective only in the serial PROM mode. In the MCU mode, the flash memory is always accessed with actual addresses at 1000H-FFFFH regardless of BANKSEL.

Note 4: Bits 2 to 0 in FLSCR are always read as undefined data.

19.2 Flash Memory Write Enable Control (FLSCR<FLSMD>)

The flash memory can be protected from inadvertent write due to program error or microcontroller misoperation. This write protection feature is realized by disabling flash memory write via the flash memory control register (FLSCR). To write to the flash memory, set FLSCR<FLSMD> to "0011B". To disable flash memory write, set FLSCR<FLSMD> to "1100B". After reset, FLSCR<FLSMD> is initialized to "1100B" to enable the write protection feature. Normally, FLSCR<FLSMD> should be set to "1100B" except when the flash memory needs to be written.

19.3 Flash Memory Bank Select Control (FLSCR<BANKSEL>)

In the serial PROM mode, a 2-Kbyte BOOT-ROM is mapped to addresses 7800H-7FFFH and the flash memory is mapped into 2 banks at 8000H-FFFFH. Flash memory addressee 1000H-7FFFH are mapped to 9000H-FFFFH as BANK0 and flash memory addresses 8000H-FFFFH are mapped to 8000H-FFFFH as BANK1. FLSCR<BANKSEL> is used to switch between these banks. For example, to access flash memory address 7000H, set FLSCR<BANKSEL> to "0" and then access F000H. To access flash memory address 9000H, set FLSCR<BANKSEL> to "1" and then access 9000H.

In the MCU mode, the flash memory is accessed with actual addresses at 1000H-FFFFH. In this case, FLSCR<BANKSEL> is ineffective (i.e., its value has no effect on other operations).

Table 19-1 Flash Memory Access

Operating Mode	FLSCR <BANKSEL>	Access Area	Specified Address
MCU mode	Don't care	1000H~FFFFH	
Serial PROM mode	0 (BANK0)	1000H~7FFFH	9000H~FFFFH
	1 (BANK1)	8000H~FFFFH	

19.4 Accessing the Flash Memory Area

When the flash memory is to be written, Read and Fetch operations cannot be performed on the entire flash memory area. Therefore, to write to the flash memory, the flash memory writing mode or RAM loader mode of the serial PROM mode is used. (For details about the flash memory writing mode, see "20. Serial PROM Mode".) It is also possible to write to the flash memory by transferring a user write control program to the RAM area and executing it there in the MCU mode. The flash memory can be written or read in units of 1 byte. Erase operations can be performed either on the entire area (60Kbytes) or in units of 4 Kbytes.

Read operations can be performed by a 1-byte transfer instruction. However, the command sequence method is adopted for Write and Erase operations, requiring several-byte transfer instructions for each operation.

19.5 Command Sequence

The MCU mode and the serial PROM mode have six types of command sequence (JEDEC compatible), as shown in Table 19-2. Addresses specified in each command sequence are identified with the low-order 12 bits (excluding BA, SA, and FF7FH in Read-Protect). The high-order 4 bits are used to specify the flash memory area, as shown in Table 19-3.

Table 19-2 Command Sequence

	Command Sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
		Add	Data	Add	Data	Add	Data	Add	Data	Add	Data	Add	Data
1	Byte-Program	555H	AAH	AAAH	55H	555H	A0H	BA (Note 1)	Data (Note 1)	-	-	-	-
2	Sector-Erase (4-Kbyte Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	SA (Note 2)	30H
3	Chip-Erase (All Erase)	555H	AAH	AAAH	55H	555H	80H	555H	AAH	AAAH	55H	555H	10H
4	Product ID Entry	555H	AAH	AAAH	55H	555H	90H	-	-	-	-	-	-
5	Product ID Exit	XXH	F0H	-	-	-	-	-	-	-	-	-	-
	Product ID Exit	555H	AAH	AAAH	55H	555H	F0H	-	-	-	-	-	-
6	Read-Protect	555H	AAH	AAAH	55H	555H	A5H	FF7FH	00H	-	-	-	-

Note 1: Set the address and data to be written.

Note 2: The area to be erased is determined by the high-order 4 bits of the address.

Table 19-3 Address Specification in Command Sequence

Operating Mode	FLSCR <BANKSEL>	Specified Address
MCU mode	Don't care	1***H~F***H
Serial PROM mode	0 (BANK0)	9***H~F***H
	1 (BANK1)	8***H~F***H

19.5.1 Byte-Program

This command sequence programs the flash memory byte by byte. The address and data to be written are specified in the 4th bus write cycle. Each byte can be programmed in approximately 30 µs. A next command sequence cannot be executed until the Write operation has completed. To check for the completion of the Write operation, perform Read operations repeatedly until the same data is read twice from the same address in the flash memory. During the Write operation, any consecutive attempts to read from the same address will produce alternating "0"s and "1"s (toggling between 0 and 1) on bit 6 of the data.

19.5.2 Sector-Erase (4-Kbyte Erase)

This command sequence erases the flash memory in units of 4 Kbytes. The flash memory area to be erased is specified by the high-order 4 bits of the 6th bus write cycle address. For example, in the MCU mode, to erase 4 Kbytes from 7000H to 7FFFH, specify one of the addresses in 7000H-7FFFH in the 6th bus write cycle. In the serial PROM mode, to erase 4 Kbytes from 7000H to 7FFFH, set FLSCR<BANKSEL> to "0" and then specify one of the addresses in F000H-FFFFH in the 6th bus write cycle. Sector-Erase is effective only in the MCU mode and the serial PROM mode, and it cannot be used in the parallel PROM mode.

It takes approximately 15 ms to erase 4 Kbytes. A next command sequence cannot be executed until the Erase operation has completed. To check for the completion of the Erase operation, perform Read operations repeatedly until the same data is read twice from the same address in the flash memory. During the Erase operation, any consecutive attempts to read from the same address will produce alternating "0"s and "1"s (toggling between 0 and 1) on bit 6 of the data. The Erase operation clears data to FFH.

19.5.3 Chip-Erase (All Erase)

This command sequence erases the entire flash memory in approximately 30 ms. A next command sequence cannot be executed until the Erase operation has completed. To check for the completion of the Erase operation, perform Read operations repeatedly until the same data is read twice from the same address in the flash memory. During the Erase operation, any consecutive attempts to read from the same address will produce alternating "0"s and "1"s (toggling between 0 and 1) on bit 6 of the data. After the chip has been erased, all bytes contain FFH.

19.5.4 Product ID Entry

This command sequence activates the Product ID mode. In the Product ID mode, the vendor ID, the flash ID, and the read protection status can be read from the flash memory.

Table 19-4 Values To Be Read in the Product ID Mode

Address	Meaning	Read Value
0000H	Vendor ID	98H
0001H	Flash macro ID	41H
0002H	Flash size	0EH: 60 Kbytes 0BH: 48 Kbytes 07H: 32 Kbytes 05H: 24 Kbytes 03H: 16 Kbytes 01H: 8 Kbytes 00H: 4 Kbytes
FF7FH	Read protection status	FFH: Read protection OFF Other than FFH: Read protection ON

Note: The value at address 0002H (Flash size) is determined by the size of flash memory incorporated in each product. For example, if the product has 60-Kbyte flash memory, "0EH" is read from address 0002H.

19.5.5 Product ID Exit

This command sequence is used to exit the Product ID mode.

19.5.6 Read-Protect

This command sequence enables the read protection setting on the flash memory. When read protection is enabled, the flash memory cannot be read in the parallel PROM mode. In the serial PROM mode, the flash write and RAM loader commands cannot be executed.

To enable the read protection setting in the serial PROM mode, set FLSCR<BANKSEL> to "1" before executing the Read-Protect command sequence. To disable the read protection setting, it is necessary to execute the Chip-Erase command sequence. Whether or not read protection is enabled can be checked by reading FF7FH in the Product ID mode. For details, see Table 19-4.

It takes approximately 30 µs to set read protection on the flash memory. A next command sequence cannot be executed until this operation has completed. To check for the completion of the Read-Protect operation, perform Read operations repeatedly until the same data is read twice from the same address in the flash memory. During the Read-Protect operation, any attempts to read from the same address will produce alternating "0"s and "1"s (toggling between 0 and 1) on bit 6 of the data.

19.6 Toggle Bit (D6)

After the Byte-Program, Sector-Erase, Chip-Erase, or Read-Protect command sequence is executed, any consecutive attempts to read from the same address will produce alternating "0"s and "1"s (toggling between 0 and 1) on bit 6 (D6) of the data until the corresponding operation has completed. Thus, this Toggle Bit provides a software mechanism to check for the completion of each operation. After the Byte-Program, Sector-Erase, Chip-Erase, or Read-Protect command sequence is executed, perform Read operations repeatedly until the same data is read twice from the same address in the flash memory. The initial read of the Toggle Bit will always produce a "1".

19.7 Writing to the Flash Memory

To write to the flash memory, a write control program should be transferred to the RAM area and executed there using the RAM loader mode in the serial PROM mode. It is also possible to write to the flash memory by transferring a user write control program to the RAM area and executing it there in the MCU mode. The procedures for writing to the flash memory by these methods are explained below.

Note 1: Before writing to the flash memory via the RAM area, be sure to disable interrupts by setting the interrupt master enable flag (IMF) to "0". This is normally done by executing a DI instruction at the head of the write control program to be executed in the RAM area.

Note 2: When writing to the flash memory, do not intentionally use non-maskable interrupts (the watchdog timer must be disabled if it is used). If a non-maskable interrupt occurs while the flash memory is being written, unexpected data will be read from the flash memory (interrupt vector) and this may cause the microcontroller to misoperate.

19.7.1 How to write to the flash memory by executing a write control program in the RAM area (RAM loader mode in the serial PROM mode)

(Steps 1 and 2 are controlled by the BOOT-ROM, and steps 3 through 10 are controlled by the write control program executed in the RAM area.)

1. Transfer a write control program to the RAM area by using the RAM loader mode.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Set FLSCR<FLSMD> to "0011B" (to enable flash memory write).
5. Execute the Chip-Erase or Sector-Erase command sequence.
6. Read the same flash memory address twice.

   (Repeat step 6 until the same data is read by two consecutive reads.)
7. Specify the bank to be written in FLSCR<BANKSEL>.
8. Execute the Byte-Program command sequence.
9. Read the same flash memory address twice.

   (Repeat step 9 until the same data is read by two consecutive reads.)
10. Set FLSCR<FLSMD> to "1100B" (to disable flash memory write).

Note: In the case of using the RAM loader mode, the watchdog timer is disabled by the BOOT-ROM. It is therefore not necessary to disable the watchdog timer in the RAM loader program.

Example :This write control program executes Chip-Erase and then writes data 3FH to address F000H.

```
            DI                                  ; Disable interrupts (IMF←"0")
            LD      (FLSCR),0011_1000B          ; Enable flash memory write.
            LD      IX,0F555H
            LD      IY,0FAAAH
            LD      HL,0F000H
; #### Flash Memory Chip-Erase Process ####
            LD      (IX),0AAH                   ; 1st Bus Write Cycle
            LD      (IY),55H                    ; 2nd Bus Write Cycle
            LD      (IX),80H                    ; 3rd Bus Write Cycle
            LD      (IX),0AAH                   ; 4th Bus Write Cycle
            LD      (IY),55H                    ; 5th Bus Write Cycle
            LD      (IX),10H                    ; 6th Bus Write Cycle
sLOOP1:     LD      W,(IX)
            CMP     W,(IY)
            JR      NZ,sLOOP1                   ; Loop until the same value is read.
            SET     (FLSCR).3                   ; Set BANK1.
; #### Flash Memory Byte-Program Process ####
            LD      (IX),0AAH                   ; 1st Bus Write Cycle
            LD      (IY),55H                    ; 2nd Bus Write Cycle
            LD      (IX),0A0H                   ; 3rd Bus Write Cycle
            LD      (HL),3FH                    ; 4th Bus Write Cycle, (F000H)=3FH
sLOOP2:     LD      W,(HL)
            CMP     W,(HL)
            JR      NZ,sLOOP2                   ; Loop until the same value is read.
            LD      (FLSCR),1100_1000B          ; Disable flash memory write.
sLOOP3:     JP      sLOOP3
```

19.7.2 How to write to the flash memory by executing a user write control program in the RAM area (MCU mode)

(Steps 1 and 2 are controlled by a program on the flash memory, and steps 3 through 11 are controlled by the write control program executed in the RAM area.)

1. Transfer a write control program to the RAM area.
2. Jump to the RAM area.
3. Disable (DI) the interrupt master enable flag (IMF←"0").
4. Disable the watchdog timer if it is used.
5. Set FLSCR<FLSMD> to "0011B" (to enable flash memory write).
6. Execute the Chip-Erase or Sector-Erase command sequence.
7. Read the same flash memory address twice.

(Repeat step 7 until the same data is read by two consecutive reads.)

8. Execute the Byte-Program command sequence. (There is no need to specify the bank to be written.)
9. Read the same flash memory address twice.

   (Repeat step 9 until the same data is read by two consecutive reads.)
10. Set FLSCR<FLSMD> to "1100B" (to disable flash memory write).
11. Jump to the flash area.

Example :This write control program executes Sector-Erase (1000H~1FFFH) and then writes data 3FH to address 1000H.

```
            DI                                   ; Disable interrupts (IMF←"0")
            LD      (WDTCR2),4EH                 ; Clear the WDT binary counter.
            LDW     (WDTCR1),0B101H              ; Disable the WDT.
            LD      (FLSCR),0011_1000B           ; Enable flash memory write.
            LD      IX,0F555H
            LD      IY,0FAAAH
            LD      HL,1000H
; #### Flash Memory Sector-Erase Process ####
            LD      (IX),0AAH                    ; 1st Bus Write Cycle
            LD      (IY),55H                     ; 2nd Bus Write Cycle
            LD      (IX),80H                     ; 3rd Bus Write Cycle
            LD      (IX),0AAH                    ; 4th Bus Write Cycle
            LD      (IY),55H                     ; 5th Bus Write Cycle
            LD      (HL),30H                     ; 6th Bus Write Cycle
sLOOP1:     LD      W,(IX)
            CMP     W,(IY)
            JR      NZ,sLOOP1                    ; Loop until the same value is read.
; #### Flash Memory Byte-Program Process ####
            LD      (IX),0AAH                    ; 1st Bus Write Cycle
            LD      (IY),55H                     ; 2nd Bus Write Cycle
            LD      (IX),0A0H                    ; 3rd Bus Write Cycle
            LD      (HL),3FH                     ; 4th Bus Write Cycle, (1000H)=3FH
sLOOP2:     LD      W,(HL)
            CMP     W,(HL)
            JR      NZ,sLOOP2                    ; Loop until the same value is read.
            LD      (FLSCR),1100_1000B           ; Disable flash memory write.
            JP      8000H                        ; Jump to the flash area.
```

20. Serial PROM Mode

20.1 Outline

The TMP86FS49UG has a boot-ROM for programming to flash memory. This boot-ROM is a mask ROM that contains a program to write the flash memory on-board. The boot-ROM is available in a serial PROM mode and it is controlled by TEST pin, BOOT pin and $\overline{\text{RESET}}$ pin, and is communicated with UART. There are 7 operation modes in a serial PROM mode: Flash memory writing mode, RAM loader mode, Flash memory SUM output mode, Product discrimination code output mode, Flash memory status output mode, Flash memory erasing mode and Read protection setting mode. Operating area of serial PROM mode differs from that of MCU mode. The operating area of serial PROM mode shows in " Table 20-1 Operating Area of Serial PROM Mode ".

Table 20-1 Operating Area of Serial PROM Mode

Parameter	Min	Max	Unit
Operating voltage	4.5	5.5	V
High frequency (Note)	2	16	MHz

Note: Even though included in above operating area, part of frequency can not be supported in serial PROM mode. For details, refer to " Table 20-4 Operating Frequency and Baud Rate in Serial PROM Mode " on Page 231.

20.2 Memory Mapping

The Figure 20-1 shows a memory mapping of Serial PROM mode.

In serial PROM mode, the BOOT-ROM (Mask ROM) is mapped in address 7800H to 7FFFH and then the Flash memory is mapped to divide into two banks. Therefore, when the RAM loader mode (60H) is used, it is necessary to specify Flash memory address according to Figure 20-1 (For detail of banks and control register, reffer to the chapter of "Flash memory control register".)

However, when the Flash writing mode (30H) is used, it is necessary to specify Flash memory address from 1000H to FFFFH mode (the same mapping as MCU mode) because BOOT-ROM changes the Flash memory address.

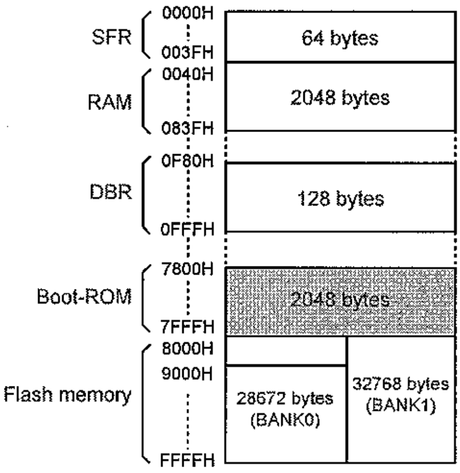


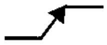
Figure 20-1 Memory Address Maps

20.3 Serial PROM Mode Setting

20.3.1 Serial PROM Mode Control Pins

To execute on-board programming, start the TMP86FS49UG in serial PROM mode. Setting of a serial PROM mode is shown in Table 20-2.

Table 20-2 Serial PROM Mode Setting

Pin	Setting
TEST pin	High
BOOT/RXD1 pin	High
$\overline{\text{RESET}}$ pin	

Note: BOOT pin is RXD1 pin during a serial PROM mode.

20.3.2 Pin Function

In the serial PROM mode, TXD1 (P02) and RXD1 (P01) pins are used as a serial interface pin.

Pin Name (Serial PROM Mode)	Input/ Output	Function		Pin Name (MCU Mode)
TXD1	Output	Serial data output	(Note 1)	P02
BOOT/RXD1	Input	Serial PROM mode control/Serial data input		P01
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
TEST	Input	Fixed to "High"		TEST
VDD, AVDD	Power Supply	4.5 V to 5.5 V		
VSS		0V		
VAREF		Input reference voltage or open.		
P00 , P03 to P07	I/O	Placed in High-Z state during serial PROM mode.		
P10 to P17				
P20 to P22				
P30 to P37				
P40 to P47				
P50 to P54				
P60 to P67				
P70 to P77				
XIN	Input	Resonator connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.		(Note 2)
XOUT	Output			

Note 1: When the device is used as on-board writing and other parts are already mounted in place, be careful no to affect these communication control pins.
Note 2: Operating area of high frequency in serial PROM mode is from 2 MHz to 16 MHz.

To set a serial PROM mode, connect device pins as shown in Figure 20-2.

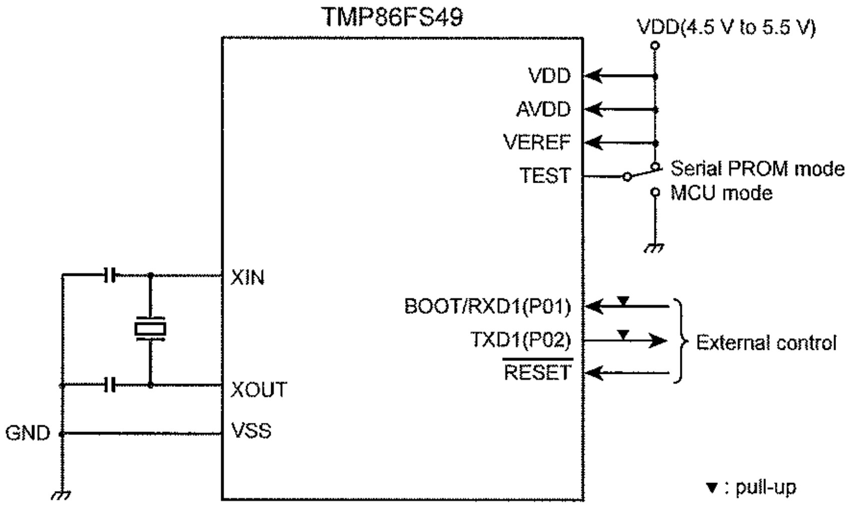


Figure 20-2 Serial PROM Mode Port Setting

20.3.3 Activating Serial PROM Mode

The following is a procedure of setting of serial PROM mode. " Figure 20-3 Serial PROM Mode Timing " shows a serial PROM mode timing.

1. Turn on the power to the VDD pin.
2. Set the $\overline{\text{RESET}}$ pin to low level.
3. Set the TEST pin to high level and pulled-up the BOOT/RXD1 pin.
4. Wait until the power supply and clock sufficiently stabilize.
5. Release the $\overline{\text{RESET}}$. (Set to high level)
6. Input a matching data (5AH) to BOOT/RXD1 pin after waiting for setup sequence. For details of the setup timing, refer to " 20.15 UART Timing " on Page 250 .

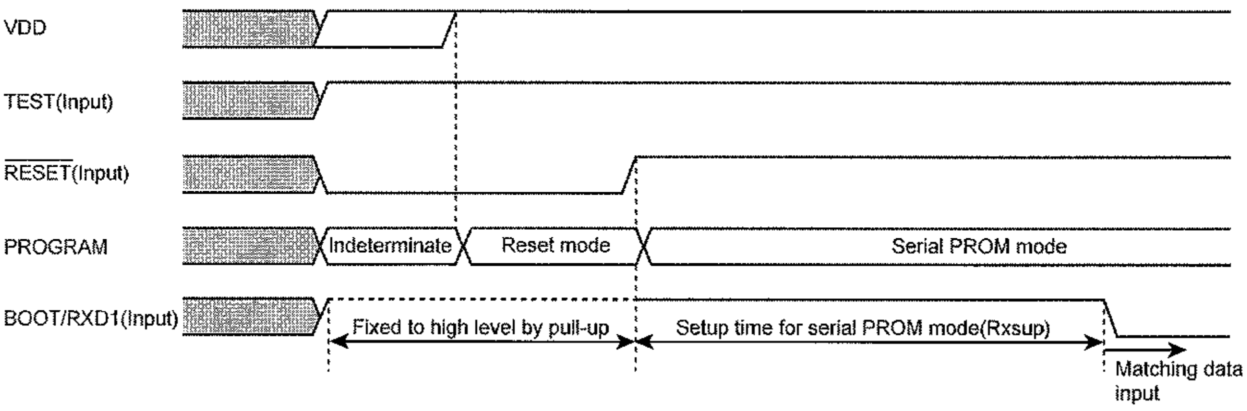


Figure 20-3 Serial PROM Mode Timing

20.4 Interface Specifications for UART

The following shows the UART communication format used in serial PROM mode.

Before on-board programming can be executed, the communication format on the external controller side must also be set up in the same way as for this product.

Note that although the default baud rate is 9600 bps, it can be changed to other values as shown in Table 20-3. The Table 20-4 shows an operating frequency and baud rate in serial PROM mode. Except frequency which is not described in Table 20-4 can not use in serial PROM mode.

- Baud rate (Default): 9600 bps
- Data length: 8 bits
- Parity addition: None
- Stop bit length: 1 bit

Table 20-3 Baud Rate Modification Data

Baud rate modification data	04H	05H	06H	07H	0AH	18H	28H
Baud rate (bps)	76800	62500	57600	38400	31250	19200	9600

Table 20-4 Operating Frequency and Baud Rate in Serial PROM Mode

(Note 3)	Reference Baud Rate (bps)		76800		62500		57600		38400		31250		19200		9600	
	Baud Rate Modification Data		04H		05H		06H		07H		0AH		18H		28H	
	Ref. Fre-quency (MHz)	Area (MHz)	Baud rate (bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)	(bps)	(%)
1	2	1.91~2.10	-	-	-	-	-	-	-	-	-	-	-	-	9615	+0.16
2	4	3.82~4.19	-	-	-	-	-	-	-	-	31250	0.00	19231	+0.16	9615	+0.16
	4.19	3.82~4.19	-	-	-	-	-	-	-	-	32734	+4.75	20144	+4.92	10072	+4.92
3	4.9152	4.70~5.16	-	-	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00
	5	4.70~5.16	-	-	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73
4	6	5.87~6.45	-	-	-	-	-	-	-	-	-	-	-	-	9375	-2.34
	6.144	5.87~6.45	-	-	-	-	-	-	-	-	-	-	-	-	9600	0.00
5	7.3728	7.05~7.74	-	-		-	57600	0.00	-	-	-	-	19200	0.00	9600	0.00
6	8	7.64~8.39	-	-	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16
7	9.8304	9.40~10.32	76800	0.00	-	-	-	-	38400	0.00	-	-	19200	0.00	9600	0.00
	10	9.40~10.32	78125	+1.73	-	-	-	-	39063	+1.73	-	-	19531	+1.73	9766	+1.73
8	12	11.75~12.90	-	-	-	-	57692	+0.16	-	-	31250	0.00	18750	-2.34	9375	-2.34
	12.288	11.75~12.90	-	-	-	-	59077	+2.56	-	-	32000	+2.40	19200	0.00	9600	0.00
	12.5	11.75~12.90	-	-	60096	-3.85	60096	+4.33	-	-	30048	-3.85	19531	+1.73	9766	+1.73
9	14.7456	14.10~15.48	-	-	-	-	57600	0.00	38400	0.00	-	-	19200	0.00	9600	0.00
10	16	15.27~16.77	76923	+0.16	62500	0.00	-	-	38462	+0.16	31250	0.00	19231	+0.16	9615	+0.16

Note 1: "Ref.Frequency" and "Area" show the high frequency area supported in serial PROM mode. Except the above frequency can not be supported in serial PROM mode even though the high frequency is included in area from 2 MHz to 16 MHz.

Note 2: The total error of frequency must be kept within +/-3% so that the auto-detection of frequency is executed correctly.

Note 3: An external controller should transmit a matching data repeatedly till the TMP86FS49UG transmit an echo back data. Above number indicates a transmission number of times of matching data till transmission of echo back data.

20.5 Command

There are eight commands in serial PROM mode. After reset release, the TMP86FS49UG waits a matching data (5AH).

Table 20-5 Command in Serial PROM Mode

Command Data	Operation Mode	Remarks
5AH	Setup	Matching data. Always start with this command after reset release.
30H	Flash memory writing	Writing to area from 1000H to FFFFH is enable.
60H	RAM loader	Writing to area from 0050H to 082FH is enable.
90H	Flash memory SUM output	The checksum of entire flash area (from 1000H to FFFFH) is output in order of the upper byte and the lower byte.
C0H	Product discrimination code output	Product discrimination code, that is expressed by 13 bytes data, is output.
C3H	Flash memory status output	The status of read protection and flash memory, those are expressed by 7 bytes data, are output.
F0H	Flash memory erasing	Erasing from 1000H to FFFFH is enable.
FAH	Read protection setting	Setting a read protection.

20.6 Operation Mode

There are seven operating modes in serial PROM mode: Flash memory writing mode, RAM loader mode, Flash memory SUM output mode, Product discrimination code output mode, Flash memory status output mode, Flash memory erasing mode and Read protection setting mode. For details about these modes, refer to "(1) Flash memory writing mode" through "(7) Read protection setting mode".

1. Flash memory writing mode

   The data are written to the specified flash memory addresses. The controller should send the write data in the Intel Hex format (Binary).

   If no errors are encountered till the end record, the SUM of 60 Kbytes of flash memory is calculated and the result is returned to the controller.

   When the read protection is enabled, the flash memory writting mode can not execute. Therefore, excute chip erase (erasing of all flash area) of the flash memory erasing mode for releasing the read protection beforehand.

   To execute the flash memory writing mode, the TMP86FS49UG checks the passwords except a blank product. If the passwords did not match, the program is not executed.

   Note : Before writing data, the flash memory should be erased.

2. RAM loader mode

   The RAM loader transfers the data into the internal RAM that has been sent from the controller in Intel Hex format. When the transfer has terminated normally, the RAM loader calculates the SUM and sends the result to the controller before it starts executing the user program. After sending of SUM, the program jumps to the start address of RAM in which the first transferred data has been written. This RAM loader function provides the user's own way to control on-board programming.

   When the read protection is enabled, the RAM loader mode can not execute. Therefore, execute chip erase (erasing of all flash area) of the flash memory erasing mode for releasing the read protection beforehand.

   To execute the RAM loader mode, the TMP86FS49UG checks the passwords except a blank product. If the passwords did not match, the program is not executed.

3. Flash memory SUM output mode

   The SUM of 60 Kbytes of flash memory is calculated and the result is returned to the controller.

   The boot ROM does not support the reading function of the flash memory. Instead, it has this SUM command to use. By reading the SUM, it is possible to manage Revisions of application programs.

4. Product discrimination code output mode

   The product discrimination code is output as a 13-byte data, that includes the start address and the end address of ROM (In case of TMP86FS49UG, the start address is 1000H and the end address is FFFFH). Therefore, the controller can recognize the device information by using this function.

5. Flash memory status output mode
   The status of an area from FFE0H to FFFFH and a read protection are output. Those are expressed by 7 bytes data. Therefore, the controller can recognize the flash memory status.
6. Flash memory erasing mode
   The flash memory can be erased by chip erase (erasing of all flash area) or sector erase (erasing of 4Kbyte units). The data of an erased area is filled with FFH.
   The read protection is released by executing chip erase.
   To execute the flash memory erasing mode, the TMP86FS49UG checks the passwords except a blank product. If the password did not match, the erasing is not executed.
7. Read protection setting mode
   Disable the reading data from flash memory in parallel mode. And in serial PROM mode, the flash memory writing mode and RAM loader mode are disabled, too. To release read protection, execute chip erase by the flash memory erasing mode.

20.6.1 Flash Writing Mode (Operation command: 30H)

Table 20-6 shows flash memory writing mode process.

Table 20-6 Flash Writing Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller
Boot ROM	1st byte 2nd byte	Matching data (5Ah) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-3) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (30H) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (30H) Error: A1H ¥ 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted)
	15th byte : m'th byte	Password string (Note 5) -	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (binary) (Note 2)	Changed new baud rate	-
	n'th - 1 byte	-	Changed new baud rate	OK: SUM (High) (Note 3) Error: Nothing transmitted
	n'th byte	-	Changed new baud rate	OK: SUM (Low) (Note 3) Error: Nothing transmitted
	n'th + 1 byte	(Wait for the next operation) (command data)	Changed new baud rate	-

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code " on Page 242.

Note 2: Refer to "" 20.9 Intel Hex Format (Binary) " on Page 244 .

Note 3: Refer to "" 20.8 Checksum (SUM) " on Page 242 .

Note 4: Refer to "" 20.10 Passwords " on Page 244 .

Note 5: If all data of addresses from FFE0H to FFFFH are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. However, it is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. If a password error occurs, the UART function of TMP86FS49UG stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FS49UG should be reset by $\overline{\text{RESET}}$ pin input.

Note 6: If the detecting a read protection or occuring a password error, the UART function of TMP86FS49UG stops without returning error code to the controller.

Note 7: If the error occurs during transferring a password address or a password string, the UART function of TMP86FS49UG stops without returning error code to the controller.

Description of flash memory writing mode

1. The receive data in the 1st byte is the matching data. When the boot program starts in serial PROM mode, This device waits for the matching data (5AH) to receive. Upon receiving the matching data, it automatically adjusts the UART's initial baud rate to 9,600bps.
2. When the device has received the matching data, the device transmits the data "5AH" as an echo back to the controller. If the device can not receive the matching data, the device does not transmit the echo back data and waits for the matching data again with changing baud rate. Therefore, the controller should send the matching data continuously until the device transmits the echo back data. An external controller should transmit a matching data repeatedly till the device transmit an echo back data. The transmission number of times of matching data varies by the frequency of device. For details, refer to Table 20-4.
3. The receive data in the 3rd byte is the baud rate modification data. The seven kinds of baud rate modification data shown in Table 20-3 are available. Even if baud rate changing is no need, be sure to send the initial baud rate data (28H: 9,600 bps).
4. When the 3rd byte data is one of the baud rate modification data corresponding to the device's operating frequency, the device sends the echo back data which is the same as received baud rate modification data. Then the baud rate is changed. If the 3rd byte data does not correspond to the baud rate modification data, the device stops UART function after sending 3 bytes of baud rate modification error code: (62H). The changing of baud rate is executed after transmitting the echo back data.
5. The receive data in the 5th byte is the command data (30H) to write the flash memory.
6. When the 5th byte is one of the operation command data shown in Table 20-52, the device sends the echo back data which is the same as received operation command data (in this case, 30H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).
7. The 7th byte is used as an upper bit (Bit15 to bit8) of the password count storage address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occurs, the device does not sent any data and UART function.
8. The 9th byte is used as a lower bit (Bit7 to bit0) of the password count storage address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occurs, the device does not sent any data and UART function.
9. The 11th byte is used as an upper bit (Bit15 to bit8) of the password comparison start address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occurs, the device does not sent any data and UART function.
10. The 13th byte is used as a lower bit (Bit7 to bit0) of the password comparison start address. When the receiving is executed correctly (No error), the device does not send any data. If the receiving error or password error occurs, the device does not sent any data and UART function.
11. The 15th through the m'th bytes are the password data. The number of passwords is the data (N) indicated by the password count storage address. The password data are compared for N entries beginning with the password comparison start address. The controller should send N bytes of password data to the device. If the passwords do not match, the device stops UART function without returning error code to the controller. If the data of addresses from FFE0H to FFFFH are all "FFH", the comparison of passwords is not executed because the device is considered as a blank product.
12. The receive data in the m'th + 1 through n'th - 2 byte are received as binary data in Intel Hex format. No received data are echoed back to the controller. The data which is not the start mark (3AH for ":") in Intel Hex format is ignored and does not send an error code to the controller until the device receives the start mark. After receiving the start mark, the device receives the data record, that con-

sists of length of data, address, record type, writing data and checksum. After receiving the checksum of data record, the device waits the start mark data (3AH) again. The data of data record is temporarily stored to RAM and then, is written to specified flash memory. Since after receiving an end record, the device starts to calculate the SUM, the controller should wait the SUM after sending the end record. If receive error or Intel Hex format error occurs, the device stops UART function without returning error code to the controller.

13. The n'th - 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to "" 20.8 Checksum (SUM) " on Page 242. The SUM calculation is performed after detecting the end record, but the calculation is not executed when receive error or Intel Hex format error has occurred. The time required to calculate the SUM of the 60 Kbytes of flash memory area is approximately 500 ms at fc = 16 MHz. After the SUM calculation, the device sends the SUM data to the controller. After sending the end record, the controller can judge that the transmission has been terminated correctly by receiving the checksum.
14. After sending the SUM, the device waits for the next operation command data.

Note: Do not write only the address from FFE0H to FFFFH when all data of flash memory are the same data. If these area are only written, the next operation can not be executed because of password error.

20.6.2 RAM Loader Mode (Operation Command: 60H)

Table 20-7 shows RAM loader mode process.

Table 20-7 RAM Loader Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller
Boot ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See " 20.3 Serial PROM Mode Setting " on Page 228) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (60H) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (60H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store Password count (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start Password comparison (Note 4)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 5) -	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	m'th + 1 byte : n'th - 2 byte	Intel Hex format (Binary) (Note 2)	Changed new baud rate	-
	n'th - 1 byte	-	Changed new baud rate	OK: SUM (High) (Note 3) Error: Nothing transmitted
	n'th byte	-	Changed new baud rate	OK: SUM (Low) (Note 3) Error: Nothing transmitted
RAM	-	The program jumps to the start address of RAM in which the first transferred data has been written.		

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code " on Page 242.

Note 2: Refer to " 20.9 Intel Hex Format (Binary) " on Page 244 .

Note 3: Refer to " 20.8 Checksum (SUM) " on Page 242 .

Note 4: Refer to " 20.10 Passwords " on Page 244 .

Note 5: If all data of addresses from FFE0H to FFFFH are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. However, it is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. If a password error occurs, the UART function of TMP86FS49UG stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FS49UG should be reset by $\overline{\text{RESET}}$ pin input.

Note 6: If the detecting a read protection or occurring a password error, the UART function of TMP86FS49UG stops without returning error code to the controller.

Note 7: If the error occurs during transferring a password address or a password string, the UART function of TMP86FS49UG stops without returning error code to the controller.

Description of RAM loader mode

1. The process of the 1st byte through the 4th byte are the same as flash memory writing mode.
2. The receive data in the 5th byte is the RAM loader command data (60H) to write the user's program to RAM.
3. When the 5th byte is one of the operation command data shown in Table 20-5, the device sends the echo back data which is the same as received operation command data (in this case, 60H). If the 5th

byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The process of the 7th byte through the m'th byte are the same as flash memory writing mode.
5. The receive data in the m'th + 1 through n'th - 2byte are received as binary data in Intel Hex format. No received data are echoed back to the controller.
   The data which is not the start mark (3AH for ":") in Intel Hex format is ignored and does not send an error code to the controller until the device receives the start mark. After receiving the start mark, the device receives the data record, that consists of length of data, address, record type, writing data and checksum. After receiving the checksum of data record, the device waits the start mark data (3AH) again. The data of data record is written to specified RAM by the receiving data. Since after receiving an end record, the device starts to calculate the SUM, the controller should wait the SUM after sending the end record. If receive error or Intel Hex format error occurs, the UART function of TMP86FS49UG stops without returning error code to the controller.
6. The n'th - 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to " 20.8 Checksum (SUM) " on Page 242. The SUM calculation is performed after detecting the end record, but the calculation is not executed when receive error or Intel Hex format error has occurred.
   The SUM is calculated by the data written to RAM, but the length of data, address, record type and checksum in Intel Hex format are not included in SUM.
7. The boot program jumps to the first address that is received as data in Intel Hex format after sending the SUM to the controller.

20.6.3 Flash Memory SUM Output Mode (Operation Command: 90H)

Table 20-8 shows flash memory SUM output mode process.

Table 20-8 Flash Memory SUM Output Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller
Boot ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-3) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (90H) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (90H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte	-	Changed new baud rate	OK: SUM (High) (Note 2) Error: Nothing transmitted
	8th byte	-	Changed new baud rate	OK: SUM (Low) (Note 2) Error: Nothing transmitted
	9th byte	(Wait for the next operation) (Command data)	Changed new baud rate	-

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code " on Page 242 ".
Note 2: Refer to "" 20.8 Checksum (SUM) " on Page 242.

Description of flash memory SUM output mode

1. The process of the 1st byte through the 4th byte are the same as flash memory writing mode.
2. The receive data in the 5th byte is the flash memory SUM command data (90H) to calculate the entire flash memory.
3. When the 5th byte is one of the operation command data shown in Table 20-5, the device sends the echo back data which is the same as received operation command data (in this case, 90H). If the 5th

byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The 7th and the 8th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to "" 20.8 Checksum (SUM) " on Page 242.
5. After sending the SUM, the device waits for the next operation command data.

20.6.4 Product Discrimination Code Output Mode (Operation Command: C0H)

Table 20-9 shows product discrimination code output mode process.

Table 20-9 Product Discrimination Code Output Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller	
Boot ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte 4th byte	Baud rate modification data (See Table 20-3) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H x 3, A3H x 3, 62H x 3 (Note 1)	
	5th byte 6th byte	Operation command data (C0H) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (C0H) Error: A1H x 3, A3H x 3, 63H x 3 (Note 1)	
	7th byte		Changed new baud rate	3AH	Start mark
	8th byte		Changed new baud rate	0AH	The number of transfer data (from 9th to 18th byte)
	9th byte		Changed new baud rate	02H	Length of address (2 bytes)
	10th byte		Changed new baud rate	1DH	Reserved data
	11th byte		Changed new baud rate	00H	Reserved data
	12th byte		Changed new baud rate	00H	Reserved data
	13th byte		Changed new baud rate	00H	Reserved data
	14th byte		Changed new baud rate	01H	The number of ROM block (1 block)
	15th byte		Changed new baud rate	10H	First address of ROM
	16th byte		Changed new baud rate	00H	
	17th byte		Changed new baud rate	FFH	End address of ROM
	18th byte		Changed new baud rate	FFH	
	19th byte		Changed new baud rate	D2H	Checksum of transferred data (from 9th to 18th byte)
	20th byte	(Wait for the next operation) (Command data)	Changed new baud rate	-	

Note: "xxH x 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code " on Page 242 .

Description of product discrimination code output mode

1. The process of the 1st byte through the 4th byte are the same as flash memory writing mode.
2. The receive data in the 5th byte is the product discrimination code output command data (C0H).
3. When the 5th byte is one of the operation command data shown in " 20.5 Command ", the device sends the echo back data which is the same as received operation command data (in this case, C0H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The bytes from the 9th to 19th are the product discrimination code. For details, refer to " 20.11 Product Discrimination Code " on Page 246 .
5. After sending the SUM, the device waits for the next operation command data.

20.6.5 Flash Memory Status Output Mode (Operation Command : C3H)

Table 20-10 shows Flash memory status output mode process.

Table 20-10 Flash Memory Status Output Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller	
BOOT ROM	1st byte	Matching data (5AH)	9600 bps	- (Baud rate auto set)	
	2nd byte	-	9600 bps	OK: Echo back data (5AH) Error: Nothing transmitted	
	3rd byte	Baud rate modification data (See Table 20-3)	9600 bps	-	
	4th byte	-	9600 bps	OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)	
	5th byte	Operation command data (C3H)	Changed new baud rate	-	
	6th byte	-	Changed new baud rate	OK: Echo back data (C3H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)	
	7th byte		Changed new baud rate	3AH	Start mark
	8th byte		Changed new baud rate	04H	The number of transfer data (from 9th to 14th byte)
	9th byte		Changed new baud rate	00H to 03H	Status Code 1
	10th byte		Changed new baud rate	00H	Reserved data
	11th byte		Changed new baud rate	00H	Reserved data
	12th byte		Changed new baud rate	00H	Reserved data
	13th byte		Changed new baud rate	00H or FFH or FEH or FDH	Checksum of transferred data (from 9th to 12th byte) Status Code 1 Checksum 00H: 00H 01H: FFH 02H: FEH 03H: FDH
	14th byte	(Wait for the next operation) (Command data)	Changed new baud rate	-	

Note: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Description of Flash memory Status output mode

1. The process of the 1st byte through the 4th byte are the same as Flash memory writing mode.
2. The receive data in the 5th byte is the flash memory status output command data (C3H) to get the status of flash memory.
3. When the 5th byte is one of the operation command data shown in Table 20-5, the device sends the echo back data which is the same as received operation command data (in this case, C3H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).
4. The bytes from 9th to 13th are the status code. For details on the status code, refer to " 20.12 Flash Memory Status Code ".
5. After sending the status code, the device waits for the next operation command data.

20.6.6 Flash Memory Erasing Mode (Operation Command : F0H)

Table 20-11 shows Flash memory erasing mode process.

Table 20-11 Flash Erasing Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-3) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (F0H) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (F0H) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store Password count (Note 4, 5)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store Password count (Note 4, 5)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start Password comparison (Note 4, 5)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start Password comparison (Note 4, 5)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 4, 5) -	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th-2 byte	Erase area select data (Note 2)	Changed new baud rate	-
	n'th-1 byte	-	Changed new baud rate	OK: SUM (High) (Note 3) Error: Nothing transmitted
	n'th byte	-	Changed new baud rate	OK: SUM (Low) (Note 3) Error: Nothing transmitted
	n'th+1 byte	(Wait for the next operation) (command data)	Changed new baud rate	-

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.13 Erasing Flash Memory Area ".

Note 3: Refer to " 20.8 Checksum (SUM) ".

Note 4: Refer to " 20.10 Passwords ".

Note 5: The password strings should not send for the blank device.

Note 6: If the occurring a password error, the UART function of TMP86FS49UG stops without returning error code to the controller.

Note 7: If the error occurs during transferring a password address or a password string, the UART function of TMP86FS49UG stops without returning error code to the controller.

Description of Flash memory erasing mode

1. The process of the 1st byte through the 4th byte are the same as Flash memory writing mode.
2. The receive data in the 5th byte is the command data (F0H) to erase the flash memory.
3. When the 5th byte is one of the operation command data shown in Table 20-5, the device sends the echo back data which is the same as received operation command data (in this case, F0H). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).

4. The process of the 7th byte through the m'th byte are the same as flash memory writting mode. However, the password strings should not send for the blank device. (Do not send the password strings of dummy.)
5. The receive data in the n'th - 2 byte is the erase area select data. The upper 4-bit of this byte specifies the start address of the erasing area. And the lower 4-bit of this byte specifies the end address of the erasing area. For details, refer to " 20.13 Erasing Flash Memory Area ".
6. The n'th - 1 and the n'th bytes are the SUM value that is sent to the controller in order of the upper byte and the lower byte. For details on how to calculate the SUM, refer to " 20.8 Checksum (SUM) ". The SUM calculation is performed after detecting the end record, but the calculation is not executed when receive error or Intel Hex format error has occurred. The time required to calculate the SUM of the 60 Kbytes of flash memory area is approximately 500 ms at fc = 16 MHz. After the SUM calculation, the device sends the SUM data to the controller. After sending the end record, the controller can judge that the transmission has been terminated correctly by receiving the checksum.
7. After sending the SUM, the device waits for the next operation command data.

20.6.7 Read Protection Setting Mode (Operation Command : FAH)

Table 20-12 shows Read protection setting mode process.

Table 20-12 Read Protection Setting Mode Process

	Number of Bytes Transferred	Transfer Data from External Controller to TMP86FS49UG	Baud Rate	Transfer Data from TMP86FS49UG to External Controller
BOOT ROM	1st byte 2nd byte	Matching data (5AH) -	9600 bps 9600 bps	- (Baud rate auto set) OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte 4th byte	Baud rate modification data (See Table 20-3) -	9600 bps 9600 bps	- OK: Echo back data Error: A1H × 3, A3H × 3, 62H × 3 (Note 1)
	5th byte 6th byte	Operation command data (FAH) -	Changed new baud rate Changed new baud rate	- OK: Echo back data (FAH) Error: A1H × 3, A3H × 3, 63H × 3 (Note 1)
	7th byte 8th byte	Address 15 to 08 in which to store password count (Note 2, 3)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	9th byte 10th byte	Address 07 to 00 in which to store password count (Note 2, 3)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	11th byte 12th byte	Address 15 to 08 in which to start password comparison (Note 2, 3)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	13th byte 14th byte	Address 07 to 00 in which to start password comparison (Note 2, 3)	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	15th byte : m'th byte	Password string (Note 2, 3) -	Changed new baud rate Changed new baud rate	- OK: Nothing transmitted Error: Nothing transmitted
	n'th byte	-	Changed new baud rate	OK: FBH Error: Nothing transmitted
	n'+1th byte	(Wait for the next operation) (Command data)	Changed new baud rate	-

Note 1: "xxH × 3" denotes that operation stops after sending 3 bytes of xxH. For details, refer to " 20.7 Error Code ".

Note 2: Refer to " 20.10 Passwords ".

Note 3: If the read protection setting mode is executed for the blank product, the UART function of TMP86FS49UG stops without returning error code to the controller. Then, if the password error occurs for the non blank product, the UART function stops similarly.

Note 4: If the error occurs during transferring a password address or a password string, the UART function of TMP86FS49UG stops without returning error code to the controller.

Description of Read protection setting mode

1. The process of the 1st byte through the 4th byte are the same as Flash memory writing mode.
2. The receive data in the 5th byte is the command data (FAH) to set the read protection.
3. When the 5th byte is one of the operation command data shown in Table 20-5, the device sends the echo back data which is the same as received operation command data (in this case, FAH). If the 5th byte data does not correspond to the operation command data, the device stops UART function after sending 3 bytes of operation command error code: (63H).
4. The process of the 7th byte through the m'th byte are the same as flash memory writting mode.
5. The receive data in the n'th byte is the status that is sent to the controller in order of the succeeding read protection.
6. After sending the status, the device waits for the next operation command data.

20.7 Error Code

When the device detects an error, the error codes are sent to the controller.

Table 20-13 Error Code

Transmit Data	Meaning of Transmit Data
62H, 62H, 62H	Baud rate modification error occurred.
63H, 63H, 63H	Operating command error occurred.
A1H, A1H, A1H	Framing error in received data occurred.
A3H, A3H, A3H	Overrun error in received data occurred.

Note: If password error occurs, the TMP86FS49UG doesn't send error codes.

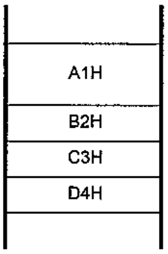
20.8 Checksum (SUM)

20.8.1 Calculation method

SUM consists of byte + byte... + byte, the checksum of which is returned in word as the result.

Namely, data is read out in byte and checksum of which is calculated, with the result returned in word.

Example:



If the data to be calculated consists of the four bytes shown to the left, SUM of the data is

A1H + B2H + C3H + D4H = 02EAH
SUM (HIGH)= 02h
SUM (LOW)= EAH

The SUM returned when executing the flash memory write command, RAM loader command, or flash memory SUM command is calculated in the manner shown above.

20.8.2 Calculation data

The data from which SUM is calculated are listed in Table 20-14 below.

Table 20-14 Checksum Calculation Data

Operating Mode	Calculation Data	Remarks
Flash memory writing mode	Data in the entire area (60 Kbytes) of flash memory	Even when written to part of the flash area, data in the entire memory area (60 Kbytes) is calculated. The length of data, address, record type and checksum in Intel Hex format are not included in SUM.
Flash memory SUM output mode		
RAM loader mode	Data written to RAM	The length of data, address, record type and checksum in Intel Hex format are not included in SUM.
Product Discrimination Code Output mode	Checksum of transferred data (from 9th to 18th byte)	For details, refer to " 20.11 Product Discrimination Code " on Page 246.
Flash Memory Status Output mode	Checksum of transferred data (from 9th to 12th byte)	For details, refer to " Table 20-10 Flash Memory Status Output Process "
Flash Memory Erasing mode	Data in the erased area of flash memory (the whole or part of memory)	For details, refer to " Table 20-11 Flash Erasing Mode Process "

20.9 Intel Hex Format (Binary)

1. After receiving the checksum of a record, the device waits for the start mark data (3AH for ":") of the next record. Therefore, the device ignores the data, which does not match the start mark data after receiving the checksum of a record.
2. Make sure that once the controller program has finished sending the checksum of the end record, it does not send anything and waits for two bytes of data to be received (Upper and lower bytes of checksum). This is because after receiving the checksum of the end record, the boot program calculates the checksum and returns the calculated checksum in two bytes to the controller.
3. If a receive error or Intel Hex format error occurs, the UART function of TMP86FS49UG stops without returning error code to the controller. In the following cases, an Intel Hex format error occurs:

   When the record type is not 00H, 01H, or 02H

   When a SUM error occurred

   When the data length of an extended record (Type = 02H) is not 02H

   When the address of an extended record (Type = 02H) is larger than 1000H and after that, receives the data record

   When the data length of the end record (Type = 01H) is not 00H

20.10Passwords

The eight or more bytes consecutive data in flash memory area can be used as password. In password check, TMP86FS49UG compares these data with data which are transmitted from the external controller. The area in which passwords can be specified is located at addresses 1000H to FF9FH. The area from FFA0H to FFFFH can not be specified as passwords area. The device compares the stored passwords with the passwords, which are received from the controller. If all data of addresses from FFE0H to FFFFH are "00H" or "FFH", the passwords comparison is not executed because the device is considered as blank product. It is necessary to specify the password count storage addresses and the password comparison start address even though it is a blank product. Table 20-15 shows the password setting in the blank product and non blank product.

Table 20-15 Password Setting in the Blank Product and Non Blank Product

Password	Blank Product (Note 1)	Non Blank Product
PNSA (Password count storage addresses)	1000H ≤ PNSA ≤ FF9FH	1000H ≤ PNSA ≤ FF9FH
PCSA (Password comparison start address)	1000H ≤ PCSA ≤ FF9FH	1000H ≤ PCSA ≤ FFA0 - N
N (Password count)	*	8 ≤ N
Setting of password	No need (Note 5)	Need (Note 2)

Note 1: When all data of addresses from FFE0H to FFFFH area are "00H" or "FFH", the device is judged as blank product.

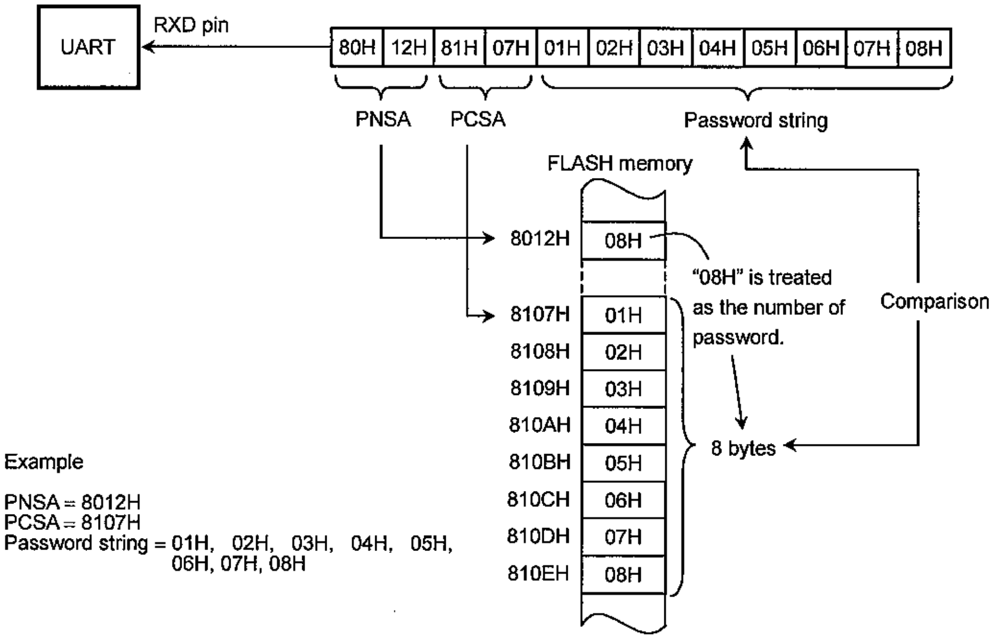
Note 2: The same three or more bytes consecutive data can not be used as password. When the password includes the same consecutive data (three or more bytes), the password error occurs. If the password error occurred, the UART function of device stops without returning error code.

Note 3: *: Don't care.

Note 4: When the password doesn't match the above condition, the password error occurs. If the password error occurred, the UART function of device stops without returning error code.

Note 5: In case of flash writing mode or as for RAM loader mode, the blank device receives Intel Hex format immediately after receiving PCSA without receiving password strings. In this time, because the blank device ignores the data except the start mark data (3AH for ":") as Intel Hex format data, even if external controller transmitted dummy password settings, process operates correctly.
However, if the dummy password string contain data "3AH", the blank device detects it as start mark data mistakenly, and blank device stops process without returning error code. Therefore, if these process becomes issue, the external controller should not transmit the dummy password strings.

Note 6: As for flash erasing mode, the password count address (PNSA), the password comparison address (PCSA) and the password strings should not send for the blank device.



20.10.1Password String

A string of passwords in the received data are compared with the data in the flash memory. In the following cases, a password error occurs:

When the received data does not match the data in the flash memory

20.10.2Handling of Password Error

If a password error occurs, the UART function of TMP86FS49UG stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FS49UG should be reset by $\overline{\text{RESET}}$.

20.10.3Password Management during Program Development

If a program is modified many times while it is being developed, confusion may arise as to its password. Therefore, we recommended that a fixed password be used in the program development stage.

Example :Using 8 bytes from address F000H for the password

```
Password Section code abs = 0F000H

          DB          "CODE1234"
```

20.11Product Discrimination Code

The product discrimination code is a 13-byte data, that includes the start address and the end address of ROM. Table 20-16 shows the product discrimination code format.

Table 20-16 Product Discrimination Code Format

Data	The Meaning of Data	In Case of TMP86FS49UG
1st	Start Mark (3AH)	3AH
2nd	The number of transfer data (from 3rd to 12th byte)	0AH
3rd	Length of address	02H
4th	Reserved data	1DH
5th	Reserved data	00H
6th	Reserved data	00H
7th	Reserved data	00H
8th	The number of ROM block	01H
9th	The upper byte of the first address of ROM	10H
10th	The lower byte of the first address of ROM	00H
11th	The upper byte of the end address of ROM	FFH
12th	The lower byte of the end address of ROM	FFH
13th	Checksum of transferred data (from 3rd to 12th byte)	D2H

20.12Flash Memory Status Code

The flash memory status code is a 7-byte data, that includes the read protection status and the status of the data from FFE0H to FFFFH. Table 20-17 shows the flash memory status code.

Table 20-17 Flash Memory Status Code

Data	The Meaning of Data	In Case of TMP86FS49UG
1st	Start mark	3AH
2nd	The number of transfer data (from 3rd to 6th byte)	04H
3rd	Flash memory status code 1	Refer to below
4th	Flash memory status code 2	Reserved
5th	Flash memory status code 3	Reserved
6th	Flash memory status code 4	Reserved
7th	Checksum of transferred data (from 3rd to 6th byte)	FDH to FFH or 00H

Flash Memory Status Code 1

7	6	5	4	3	2	1	0	
						RPENA	BLANK	(Initial Value: 0000 00**)

RPENA	The status of read protection	0: 1:	Read protect is disabled. Read protect is enabled.
BLANK	The status of the data from FFE0H to FFFFH.	0: 1:	All data are 00H or FFH from FFE0H to FFFFH. The data includes the value except 00H or FFH from FFE0H to FFFFH.

Some operation commands are limited by flash memory status code 1. If a read protection is enabled, flash memory writing mode command and RAM loader mode command can not executed. It is necessary to erase all flash memory for executing these command.

RPENA	BLANK	Flash Memory Writing Mode	RAM Loader Mode	Checksum Output Mode	Product Discrimination Code Output Mode	Product Status Output Mode	Flash Memory Erasing Mode	Read Protection Setting Mode
0	0	○	○	○	○	○	○	×
0	1	Pass	Pass	○	○	○	Pass	Pass
1	0	×	×	○	○	○	○	×
1	1	×	×	○	○	○	Pass	Pass

Note: ○ : The command can be executed without a password.
Pass: The command can be executed with a password.
×: The command can not be executed.

20.13Erasing Flash Memory Area

In the erasing flash memory mode, the area of erased flash memory is specified by n–2 byte data.

The start address of a erased flash memory area is specified by ERASTA and the end address is specified by ERAEND.

If ERASTA is smaller than or equal to ERAEND, sector erase (erasing of 4Kbyte units) is executed.

If ERASTA is bigger than ERAEND, chip erase (erasing of all flash area) is executed and a read protection is released. Therefore, execute chip erase (not sector erase) for releasing the read protection.

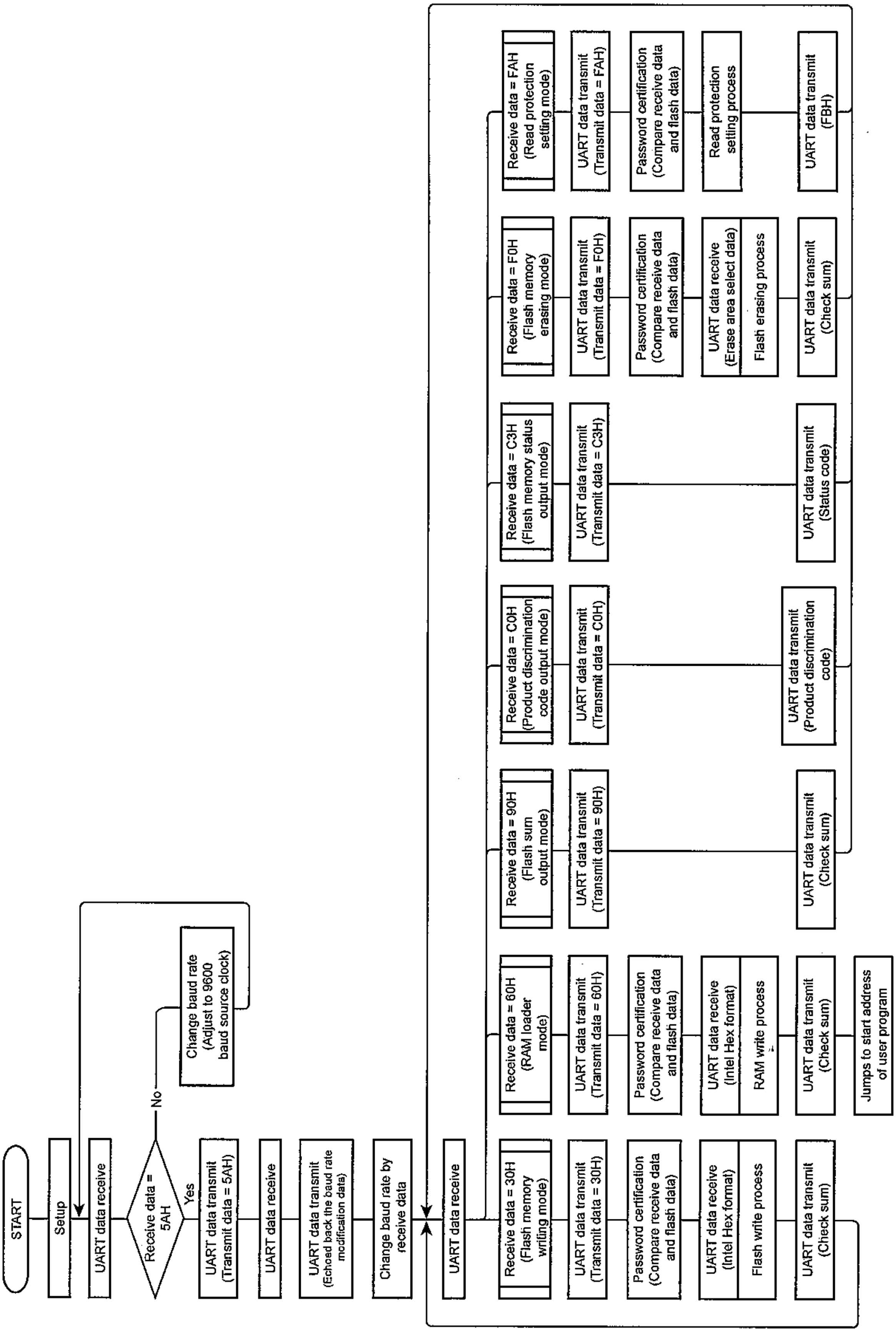
Erase Area Select Data (n–2 byte data)

7	6	5	4	3	2	1	0
ERASTA				ERAEND			

ERASTA	The start addresses of a erased flash memory area	0000:	from 0000H
		0001:	from 1000H
		0010:	from 2000H
		0011:	from 3000H
		0100:	from 4000H
		0101:	from 5000H
		0110:	from 6000H
		0111:	from 7000H
		1000:	from 8000H
		1001:	from 9000H
		1010:	from A000H
		1011:	from B000H
		1100:	from C000H
		1101:	from D000H
		1110:	from E000H
		1111:	from F000H
ERAEND	The end address of a erased flash memory area	0000:	to 0FFFH
		0001:	to 1FFFH
		0010:	to 2FFFH
		0011:	to 3FFFH
		0100:	to 4FFFH
		0101:	to 5FFFH
		0110:	to 6FFFH
		0111:	to 7FFFH
		1000:	to 8FFFH
		1001:	to 9FFFH
		1010:	to AFFFH
		1011:	to BFFFH
		1100:	to CFFFH
		1101:	to DFFFH
		1110:	to EFFFH
		1111:	to FFFFH

Note: When the area which the MCU does not have is specified, the UART function of TMP86FS49UG stops without returning error code to the controller. Therefore, when a password error occurs, the TMP86FS49UG should be reset by $\overline{\text{RESET}}$ pin input.

20.14Flowchart


START
Setup
UART data receive
Receive data = 5AH
Yes
No
Change baud rate (Adjust to 9600 baud source clock)
UART data transmit (Transmit data = 5AH)
UART data receive
UART data transmit (Echoed back the baud rate modification data)
Change baud rate by receive data
UART data receive
Receive data = 30H (Flash memory writing mode)
UART data transmit (Transmit data = 30H)
Password certification (Compare receive data and flash data)
UART data receive (Intel Hex format)
Flash write process
UART data transmit (Check sum)
Receive data = 60H (RAM loader mode)
UART data transmit (Transmit data = 60H)
Password certification (Compare receive data and flash data)
UART data receive (Intel Hex format)
RAM write process
UART data transmit (Check sum)
Jumps to start address of user program
Receive data = 90H (Flash sum output mode)
UART data transmit (Transmit data = 90H)
UART data transmit (Check sum)
Receive data = C0H (Product discrimination code output mode)
UART data transmit (Transmit data = C0H)
UART data transmit (Product discrimination code)
Receive data = C3H (Flash memory status output mode)
UART data transmit (Transmit data = C3H)
UART data transmit (Status code)
Receive data = F0H (Flash memory erasing mode)
UART data transmit (Transmit data = F0H)
Password certification (Compare receive data and flash data)
UART data receive (Erase area select data)
Flash erasing process
UART data transmit (Check sum)
Receive data = FAH (Read protection setting mode)
UART data transmit (Transmit data = FAH)
Password certification (Compare receive data and flash data)
Read protection setting process
UART data transmit (FBH)

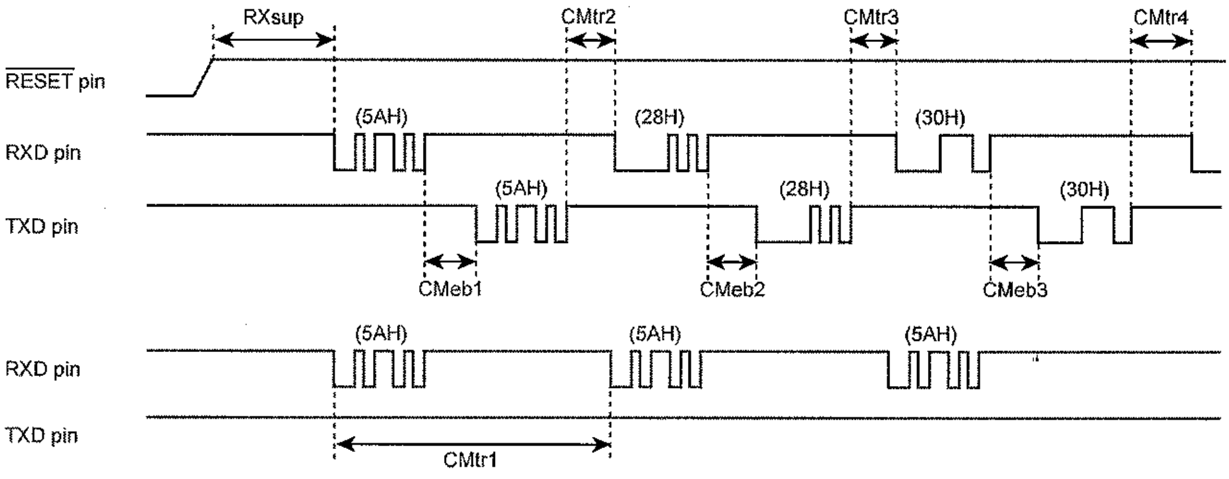
20.15UART Timing

Table 20-18 UART Timing-1 (VDD = 4.5 V to 5.5 V, fc = 2 MHz to 16 MHz, Topr = -40 to 85°C)

Parameter	Symbol	The Number of Clock (fc)	Required Minimum Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from the reception of a matching data until the output of an echo back	CMeb1	Approx. 930	465 µs	58.1 µs
Time from the reception of a baud rate modification data until the output of an echo back	CMeb2	Approx. 980	490 µs	61.3 µs
Time from the reception of an operation command until the output of an echo back	CMeb3	Approx. 800	400 µs	50 µs
Calculation time of checksum	CKsm	Approx. 7864500	3.93 s	491.5 µs
Time erasing all flash memory	CEall	–	30 ms	30 ms
Time erasing sector of flash memory(@4 kbytes)	CEsec	-	15 ms	15 ms

Table 20-19 UART Timing-2 (VDD = 4.5 V to 5.5 V, fc = 2 MHz to 16 MHz, Topr = -40 to 85°C)

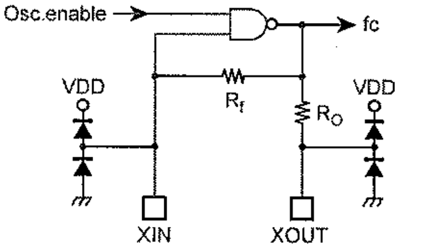
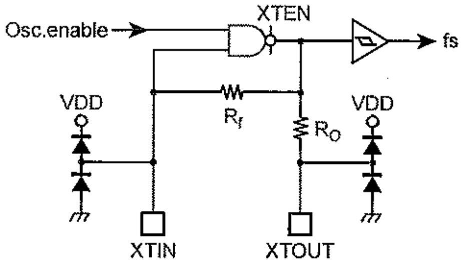
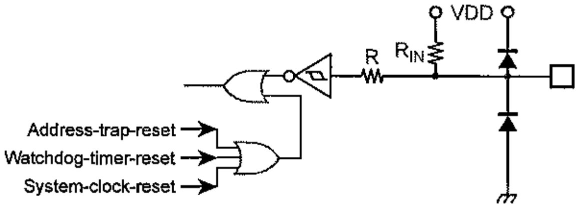
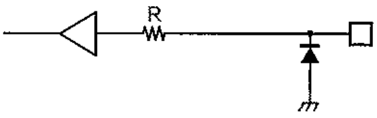
Parameter	Symbol	The Number of Clock (fc)	Required Minimum Time	
			At fc = 2 MHz	At fc = 16 MHz
Time from reset release until acceptance of start bit of RXD pin	RXsup	2100	1.05 ms	131.3 ms
Time between a matching data and the next matching data	CMtr1	28500	14.2 ms	1.78 ms
Time from the echo back of matching data until the acceptance of baud rate modification data	CMtr2	380	190 µs	23.8 µs
Time from the output of echo back of baud rate modification data until the acceptance of an operation command	CMtr3	650	325 µs	40.6 µs
Time from the output of echo back of operation command until the acceptance of Password count storage addresses	CMtr4	800	400 µs	50 µs



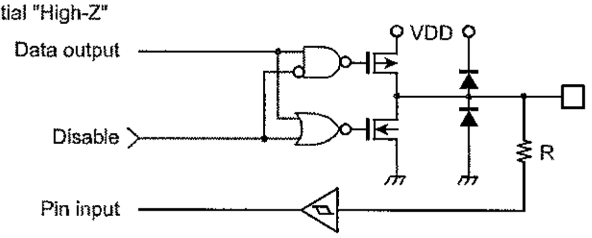
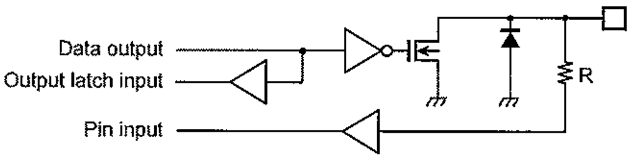
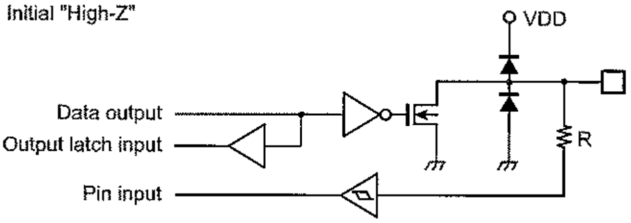
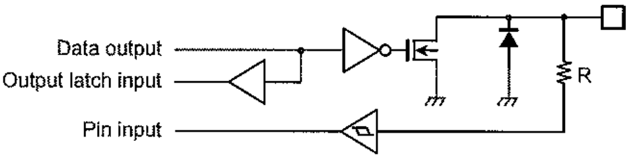
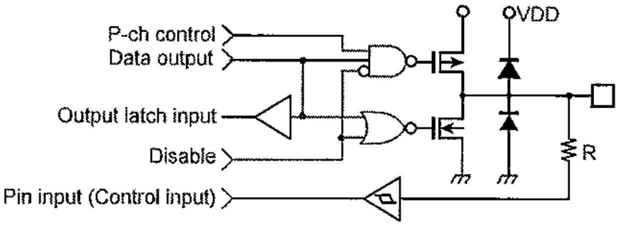
21. Input/Output Circuit

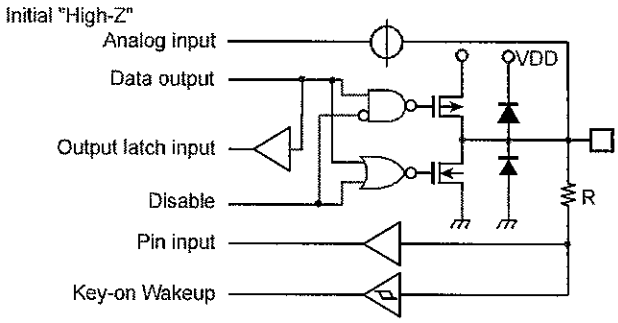
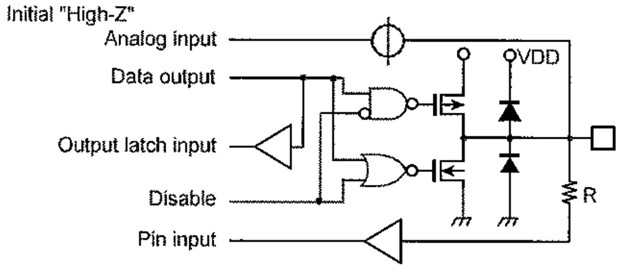
21.1 Control pins

The input/output circuitries of the TMP86FS49UG control pins are shown below.

Control Pin	I/O	Input/Output Circuitry	Remarks
XIN XOUT	Input Output	 Osc.enable fc VDD R_f R_O VDD XIN XOUT	Resonator connecting pins (high frequency) R_f = 1.2 MΩ (typ.) R_O =0.5 kΩ (typ.)
XTIN XTOUT	Input Output	 Osc.enable XTEN fs VDD R_f R_O VDD XTIN XTOUT	Resonator connecting pins (Low frequency) R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
RESET	Input	 VDD R R_{IN} Address-trap-reset Watchdog-timer-reset System-clock-reset	Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 100 Ω (typ.)
TEST	Input		R = 100 Ω (typ.)

21.2 Input/Output Ports

Port	I/O	Input/Output Circuitry	Remarks
P1	I/O	Initial "High-Z"  	Tri-state I/O Hysteresis input R = 100 Ω (typ.)
P3	I/O	Initial "High-Z"  	Sink open drain output High current output R = 100 Ω (typ.)
P2	I/O	Initial "High-Z"  	Sink open drain output Hysteresis input R = 100 Ω (typ.)
P5	I/O	Initial "High-Z"  	Sink open drain output High current output Hysteresis input R = 100 Ω (typ.)
P0 P4	I/O	Initial "High-Z"  VDD P-ch control Data output Output latch input Disable R Pin input (Control input)	Sink open drain output or C-MOS output Hysteresis input R = 100 Ω (typ.)

Port	I/O	Input/Output Circuitry	Remarks
P67 P66 P65 P64	I/O	 Initial "High-Z" Analog input Data output VDD Output latch input Disable R Pin input Key-on Wakeup	Tri-state I/O R = 100 Ω (typ.)
P63 P62 P61 P60 P7	I/O	 Initial "High-Z" Analog input Data output VDD Output latch input Disable R Pin input	Tri-state I/O R = 100 Ω (typ.)

22. Electrical Characteristics

22.1 Absolute Maximum Ratings

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.

(VSS = 0 V)

Parameter	Symbol	Pins	Ratings	Unit
Supply voltage	V_{DD}		-0.3 to 6.5	V
Input voltage	V_{IN}		-0.3 to V_{DD} + 0.3	
Output voltage	V_{OUT1}		-0.3 to V_{DD} + 0.3	
Output current (Per 1 pin)	I_{OUT1}	P0, P1, P4, P6, P7 ports	-1.8	mA
	I_{OUT2}	P0, P1, P4, P6, P7 ports	3.2	
	I_{OUT3}	P3, P5 ports	30	
Output current (Total)	ΣI_{OUT1}	P0, P1, P4, P6, P7 ports	60	
	ΣI_{OUT2}	P3, P5 ports	80	
Power dissipation [Topr = 85 °C]	P_D		250	mW
Soldering temperature (time)	Tsld		260 (10 s)	°C
Storage temperature	Tstg		-55 to 125	
Operating temperature	Topr		-40 to 85	

22.2 Recommended Operating Conditions

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.

22.2.1 MCU mode (Flash Programming or erasing)

(V_{SS} = 0 V, Topr = -10 to 40°C)

Parameter	Symbol	Pins	Ratings	Min	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	4.5	5.5	V
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V	0	$V_{DD} \times 0.30$	
	V_{IL2}	Except hysteresis input			$V_{DD} \times 0.25$	
Clock frequency	fc	XIN, XOUT		1.0	16.0	MHz

22.2.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, Topr = -40 to 85°C)

Parameter	Symbol	Pins	Ratings		Min	Max	Unit
Supply voltage (Condition 1)	V_{DD}		fc = 16 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	4.5	5.5	V
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes			
			STOP mode				
Supply voltage (Condition 2)			fc = 8 MHz	NORMAL1, 2 modes IDLE0, 1, 2 modes	3.0	3.6	
			fs = 32.768 KHz	SLOW1, 2 modes SLEEP0, 1, 2 modes			
			STOP mode				
Input high level	V_{IH1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
Input low level	V_{IL1}	Except hysteresis input	$V_{DD} \geq 4.5$ V		0	$V_{DD} \times 0.30$	
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
Clock frequency	fc	XIN, XOUT	V_{DD} = 3.0 to 3.6 V, 4.5 to 5.5V		1.0	8.0	MHz
	fc	XIN, XOUT	V_{DD} = 4.5 to 5.5 V		1.0	16.0	
	fs	XTIN, XTOUT	V_{DD} = 3.0 to 3.6 V, 4.5 to 5.5V		30.0	34.0	kHz

Note 1: The Supply voltage (V_{DD}) is divided into two different voltage areas. Do not change V_{DD} from Condition 1 to Condition 2 and vice versa while the MCU is operationg. If you wish to use V_{DD} in a continuous range of 3.0V to 5.5V without stopping the MCU, please contact your local Toshiba office.

22.2.3 Serial PROM mode

(V_{SS} = 0 V, Topr = -10 to 40 °C)

Parameter	Symbol	Pins	Condition	Min	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	4.5	5.5	V
Input high voltage	V_{IH1}	Except hysteresis input	V_{DD} ≥ 4.5 V	V_{DD} × 0.70	V_{DD}	
	V_{IH2}	Hysteresis input		V_{DD} × 0.75		
Input low voltage	V_{IL1}	Except hysteresis input	V_{DD} ≥ 4.5 V	0	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input			V_{DD} × 0.25	
Clock frequency	fc	XIN, XOUT		2.0	16.0	MHz

22.3 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} = 5.5 V, V_{IN} = 5.5 V/0 V		–	0.9	–	V
Input current	I_{IN1}	TEST			–	–	±2	µA
	I_{IN2}	Sink open drain, tri–state port						
	I_{IN3}	$\overline{\text{RESET}}$, $\overline{\text{STOP}}$						
Input resistance	R_{IN}	$\overline{\text{RESET}}$ pull–up			100	220	450	kΩ
Output leakage current	I_{LO1}	Sink open drain port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V		–	–	2	µA
	I_{LO2}	Tri–state port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V		–	–	±2	
Output high voltage	V_{OH}	Tri–state port	V_{DD} = 4.5 V, I_{OH} = -0.7 mA		4.1	–	–	V
Output low voltage	V_{OL}	Except XOUT, P3, P5	V_{DD} = 4.5 V, I_{OL} = 1.6 mA		–	–	0.4	
Output low curren	I_{OL}	High current port (P3, P5 Port)	V_{DD} = 4.5 V, V_{OL} = 1.0 V		–	20	–	mA
Supply current in NORMAL1, 2 modes	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V fc = 16 MHz fs = 32.768 kHz	When a program operates on flash memory	–	12.6	20	
Supply current in IDLE0 mode					–	6.3	7.5	
Supply current in IDLE 1, 2 modes					–	7.6	10	
Supply current in SLOW1 mode			V_{DD} = 3.0 V V_{IN} = 2.8 V/0.2 V fs = 32.768 kHz	When a program operates on flash memory	–	0.9	3	
				When a program operates on RAM	–	10	21	µA
Supply current in SLEEP1 mode					–	4.5	16	
Supply current in SLEEP0 mode					–	3.5	12	
Supply current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V		–	0.5	10	

Note 1: Typical values show those at Topr = 25°C and V_{DD} = 5 V.

Note 2: Input current (I_{IN1}, I_{IN3}): The current through pull-up or pull-down resistor is not included.

Note 3: I_{DD} does not include I_{REF}.

Note 4: The supply currents of SLOW2 and SLEEP2 modes are equivalent to those of IDLE1 and IDLE2 modes.

22.4 AD Characteristics

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = -40 to 85 °C)

Paramete	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} - 1.0	–	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note 4)	Δ V_{AREF}		3.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 V V_{SS} = A_{VSS} = 0.0 V	–	0.6	1.0	mA
Non linearity error		V_{DD} = A_{VDD} = 5.0 V, V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 5.0 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
Total error			–	–	±2	

(V_{SS} = 0 V, 3.0 V ≤ V_{DD} ≤ 3.6 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		A_{VDD} - 1.0	–	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note 4)	Δ V_{AREF}		2.5	–	–	
Analog input voltage	V_{AIN}		V_{SS}	–	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} =3.6 V V_{SS} = A_{VSS} = 0.0 V	–	0.5	0.8	mA
Non linearity error		V_{DD} = A_{VDD} = 3.0 V V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 3.0 V	–	–	±2	LSB
Zero point error			–	–	±2	
Full scale error			–	–	±2	
Total error			–	–	±2	

Note 1: The total error includes all errors except a quanitization error, and is defined as a maximum deviation from the ideal conversion line.

Note 2: Conversion time is defferent in recommended value by power supply voltage.

Note 3: The voltage to be input on the AIN input pin must not exceed the range between V_{AREF} and V_{SS}. If a voltage outside this range is input, conversion values will become unstable and conversion values of other channels will also be affected.

Note 4: Analog reference voltage range: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: When AD converter is not used, fix the AVDD and VAREF pin on the V_{DD} level.

22.5 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 V to 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	tcy	NORMAL1, 2 modes	0.25	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 16 MHz	–	31.25	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

(V_{SS} = 0 V, V_{DD} = 3.0 V to 3.6 V, Topr = -40 to 85°C)

Paramete	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.5	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1, 2 modes				
High-level clock pulse width	t_{WCH}	For external clock operation (XIN input) fc = 8 MHz	–	62.5	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

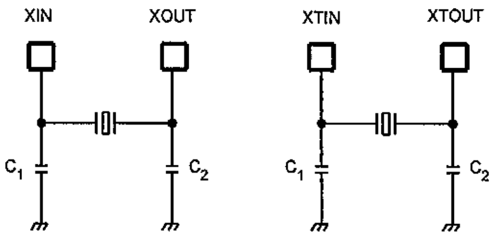
22.6 Flash Characteristics

22.6.1 Write/Retention Characteristics

(V_{SS} = 0 V)

Paramete	Condition	Min	Typ.	Max.	Unit
Number of guaranteed writes to flash memory	V_{SS} = 0 V, Topr = -10 to 40°C	–	–	100	Times

22.7 Recommended Oscillating Conditions



(1) High-frequency Oscillation (2) Low-frequency Oscillation

Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL:
http://www.murata.co.jp/search/index.html

22.8 Handling Precaution

- The solderability test conditions for lead-free products (indicated by the suffix G in product name) are shown below.
  1. When using the Sn-63Pb solder bath
     Solder bath temperature = 230 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used
  2. When using the Sn-3.0Ag-0.5Cu solder bath
     Solder bath temperature = 245 °C
     Dipping time = 5 seconds
     Number of times = once
     R-type flux used

  Note: The pass criteron of the above test is as follows:
  Solderability rate until forming ≥ 95 %
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

This is a technical document that describes the operating functions and electrical specifications of the 8-bit microcontroller series TLCS-870/C (LSI).

Toshiba provides a variety of development tools and basic software to enable efficient software development.

These development tools have specifications that support advances in microcomputer hardware (LSI) and can be used extensively. Both the hardware and software are supported continuously with version updates.

The recent advances in CMOS LSI production technology have been phenomenal and microcomputer systems for LSI design are constantly being improved. The products described in this document may also be revised in the future. Be sure to check the latest specifications before using.

Toshiba is developing highly integrated, high-performance microcomputers using advanced MOS production technology and especially well proven CMOS technology.

We are prepared to meet the requests for custom packaging for a variety of application areas.

We are confident that our products can satisfy your application needs now and in the future.