CMOS 8-BIT MICROCOMPUTER

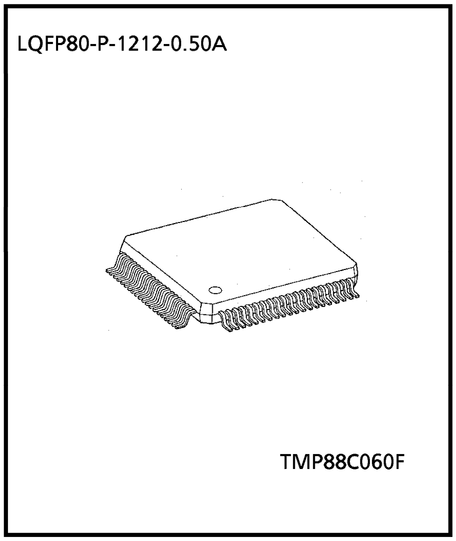
TMP88C060F

The 88C060 is the high-speed and high-performance 8-bit microcomputer, including eight multiple timer / counters, a 10-bit A/D converter, serial interfaces (UART, I^2C bus, and SIO). It can externally expand large program memory / data memory (up to 1 Mbytes linear address space).

PART No.	ROM	RAM	Package
TMP88C060F	ROM less	512 bytes	LQFP80-P-1212-0.50A

FEATURES

- 8-bit microcomputer TLCS-870 / X Series.
- Minimum instruction execution time : 0.32 μs (at 12.5 MHz)
  - Instruction execution time can be changed to reduce power consumption. min. 0.32 μs, 0.64 μs,1.28 μs, 2.56 μs, 5.12 μs, 122 μs at 12.5 MHz / 32.768 kHz
- External memory expansion
  - Expanded up to 1M bytes (for both programs and data)
  - Non-multiplexed bus (20 bits of address and 8 bits of data)
  - Wait control
  - Bus arbitration control
- 18 interrupt sources (External : 6, Internal : 12)
- Input / Output ports (42 pins)
  - High current output : 8 pins (typ. 20 mA), LED direct drive
- Two 16-bit Timer / Counters
  - TC1 : Timer, Event counter, Programmable pulse generator output, Pulse width measurement, External trigger timer, and Window modes.
  - TC2 : Timer, Event counter, and Window modes.
- Four 8-bit Timer / Counters
  - TC3 : Timer, Event counter, and Capture for Remote control signal decoding (Pulse width / duty measurement) modes.
  - TC4 : Timer, Event counter, PWM outputs, and programmable divider output modes.
  - TC5 : Timer, PWM output, and programmable divider output modes
  - TC6 : Timer and Baud-rate generation for UART modes
- Time Base Timer (Interrupt frequency : 1 kHz to 16384 kHz)
- Watchdog Timer
- Divider output (frequency : 1 kHz to 8 kHz)
- Two 8-bit Serial Interfaces
  - 8-bit UART (Parity. framing. overrun error detection)
  - 8-bit Serial Bus (I^2C-Bus for multi-master system and SIO)

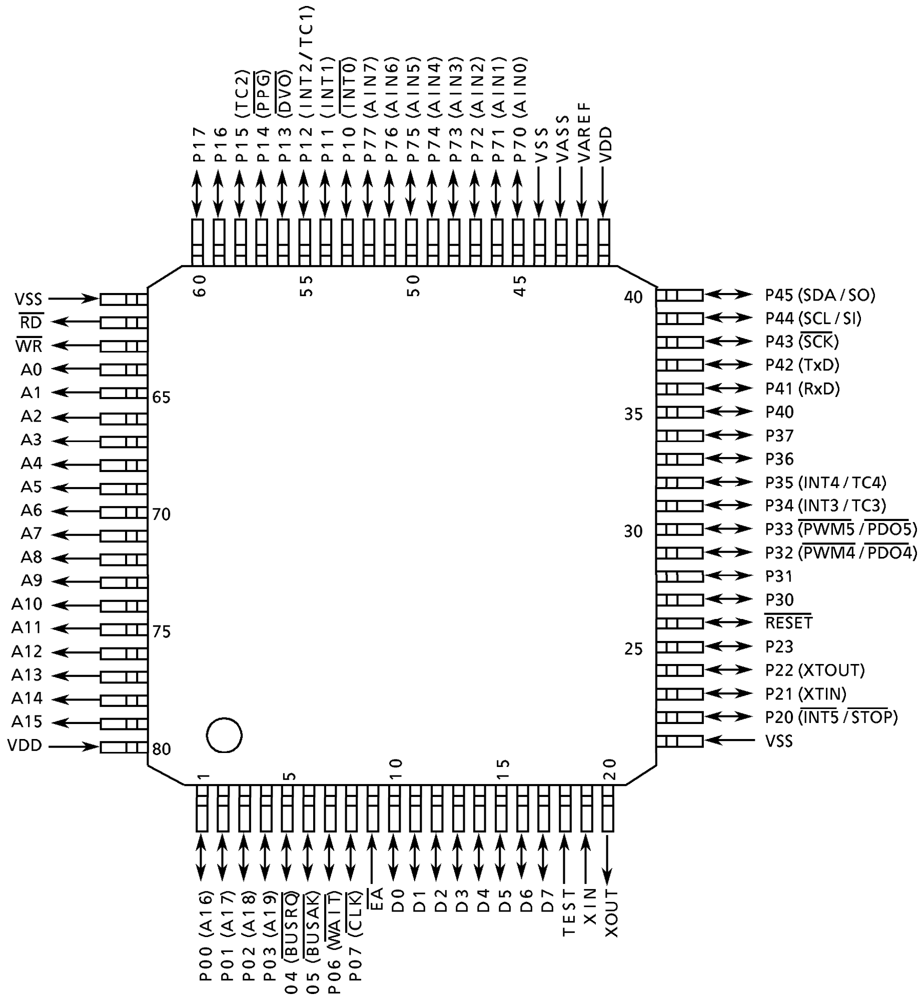


Purchase of TOSHIBA I^2C components conveys a license under the Philips I^2C Patent Rights to use these components in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

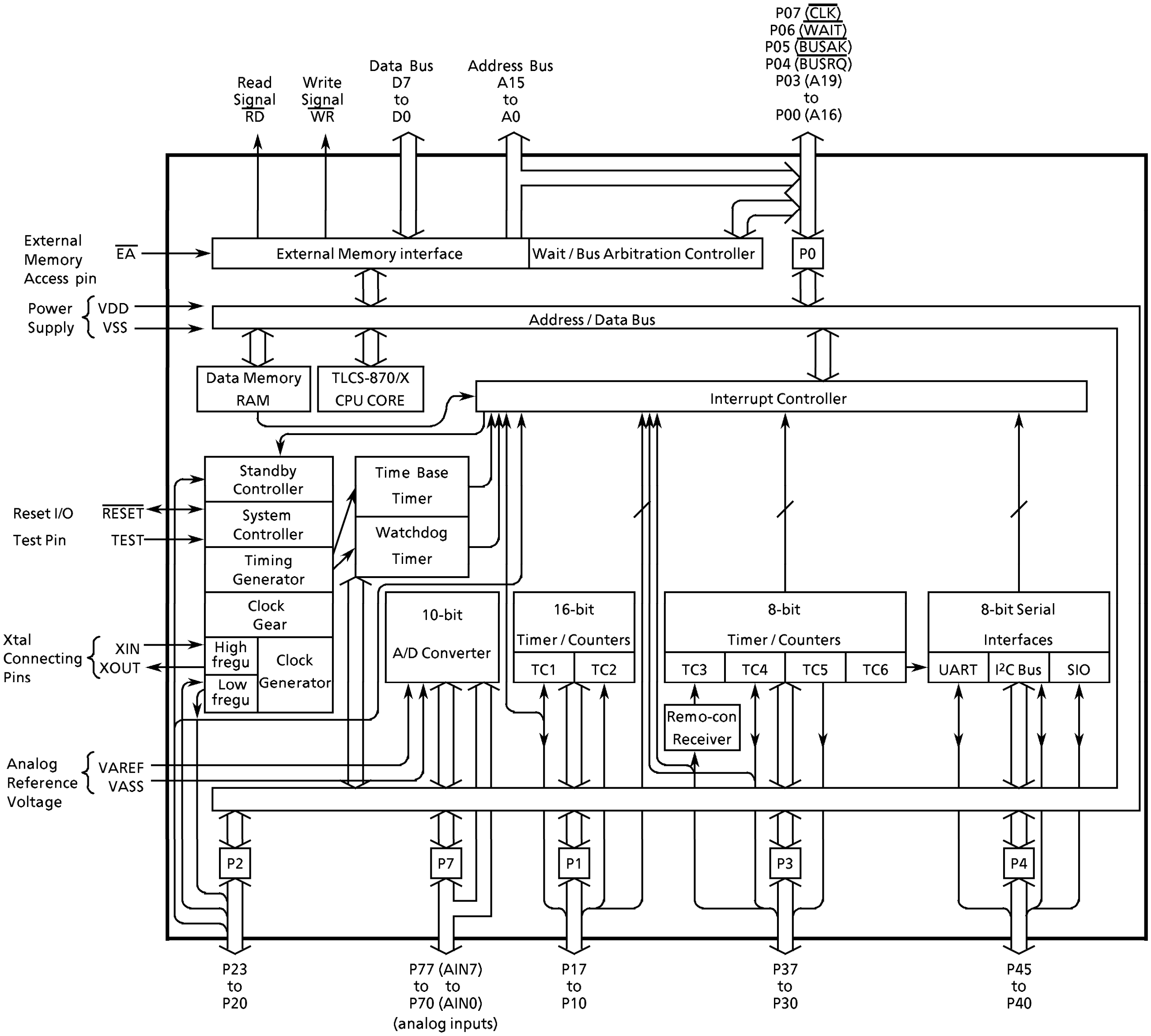
- ◆ 10-bit successive approximate type A/D converter
  - ● 8 analog inputs
  - ● Conversion time : 59 μs at 12.5 MHz, 44 μs at 4.2 MHz
- ◆ Dual clock operation
- ◆ Five Power saving operating modes
  - ● STOP mode : Oscillation stops. Battery / Capacitor back-up. Release by stop pin input.
  - ● SLOW mode : Low power consumption operation using low-frequency clock (32.8 kHz)
  - ● IDEL1 mode : CPU stops, and Peripherals operate using high-frequency clock. Release by interrupts. (CPU restarts)
  - ● IDEL2 mode : CPU stops, and Peripherals operate using high and low frequency clock. Release by interrupts.
  - ● SLEEP mode : CPU stops, and Peripherals operate using low-frequency clock. Release by interrupts.
- ◆ Wide operating voltage : 2.7 to 5.5 V at 4.2 MHz / 32.8 kHz, 4.5 to 5.5 V at 12.5 MHz / 32.768 kHz
- ◆ Emulation Pod : BM88C060F0A

PIN ASSIGNMENTS (TOP VIEW)

LQFP80-P-1212-0.50A



BLOCK DIAGRAM



PIN FUNCTION

Pin name	Input / Output	Function	
P07 ($\overline{\text{CLK}}$)	I/O (Output)	8-bit programmable input / output ports (tri-state). Each bit of these ports can be individually configured as an input or an output. When used as a wait request input, a bus release request input, an external interrupt input, or a timer counter input, corresponding bit must be configured as input . When used as a divided-by-4 clock output, a bus acknowledge output, PPG output, or a divider output, the output latch must be set to "1" and corresponding bit must be configured as output. After reset, P03 to P00 are address buses. When used as a port, these ports must be set to the ports by EXPCR.	Divided-by-4 clock output
P06 ($\overline{\text{WAIT}}$)	I/O (Input)		Wait request input
P05 ($\overline{\text{BUSAK}}$)	I/O (Output)		Bus acknowledge output
P04 ($\overline{\text{BUSRQ}}$)	I/O (Input)		Bus request input
P03 (A19) to P00 (A16)	I/O (Output)		Upper address bus (external memory connect)
P17, P16	I/O		
P15 (TC2)	I/O (Input)		Timer / Counter 2 input
P14 ($\overline{\text{PPG}}$)	I/O (Output)		Programmable pulse generator output
P13 ($\overline{\text{DVO}}$)			Divider output
P12 (INT2/TC1)	I/O (Input)		External interrupt input 2 or Timer / Counter 1 input
P11 (INT1)			External interrupt input 1
P10 ($\overline{\text{INT0}}$)			External interrupt input 0
P23	I/O	4-bit input / output port with latch. When used as an input port, a resonator connecting pin, an external interrupt input, or a STOP mode release input, the output latch must be set to "1".	
P22 (XTOUT)	I/O (Output)		Xtal connecting pins (32.8 kHz). For inputting external clock, XTIN is used and XTOUT is opened.
P21 (XTIN)			
P20 ($\overline{\text{INT5}}$ / $\overline{\text{STOP}}$)	I/O (Input)		External interrupt input 5 or STOP mode release signal input
P37, P36	I/O	8-bit input / output port (large current output) with latch. When used as an input port, PWM output, an external interrupt input, or a timer counter input, the output latch must be set to "1".	
P35 (INT4 / TC4)	I/O (Input)		External interrupt input 4 or Timer / Counter 4 input
P34 (INT3 / TC3)			External interrupt input 3 or Timer / Counter 3 input
P33 ($\overline{\text{PWM5}}$ / $\overline{\text{PDO5}}$)	I/O (Output)		8-bit PWM output 5 or, 8-bit programmable divider output 5
P32 ($\overline{\text{PWM4}}$ / $\overline{\text{PDO4}}$)			8-bit PWM output 4 or, 8-bit programmable divider output 4
P31, P30	I/O		
P45 (SDA / SO)	I/O (I/O)	6-bit input / output port with latch. When used as an input port or a serial interface pin, the output latch must be set to "1".	SIO data output I^2C bus data I/O
P44 (SCL / SI)			SIO data input I^2C bus clock I/O
P43 ($\overline{\text{SCK}}$)			SIO clock input / output
P42 (TxD)			UART data output
P41 (RxD)			UART data input
P40			
P77 (AIN7) to P70 (AIN0)	I/O (Input)	8-bit programmable input / output port (tri-state). Each bit of these ports can be individually configured as an input or an output. When used as an analog input, these ports must be set to the analog input mode by P7CR and select the channel in ADCCR.	A/D converter analog input (ch 7 to ch 0)

Pin name	Input / Output	Function
A15 to A0	Output	Lower address bus (external memory connect)
D7 to D0	I/O	Data bus (external memory connect)
$\overline{RD}$	Output	Read strobe to an extrnal memory
$\overline{WR}$		Write strobe to an extrnal memory
$\overline{EA}$	Input	External memory access input. Be tied to low.
XIN, XOUT	Input, Output	Xtal connecting pins for high-frequency clock. For inputting external clock, XIN is used and XOUT is opened.
$\overline{RESET}$	I/O	Reset signal input or watchdog timer output / address-trap-reset output / system-clock-reset output.
TEST	Input	Test pin for out-going test. Be tied to low.
VDD, VSS	Power Supply	+ 5 V, 0 V (GND)
VAREF, VASS		Analog reference voltage for A/D converter (High, Low)

OPERATIONAL DESCRIPTION

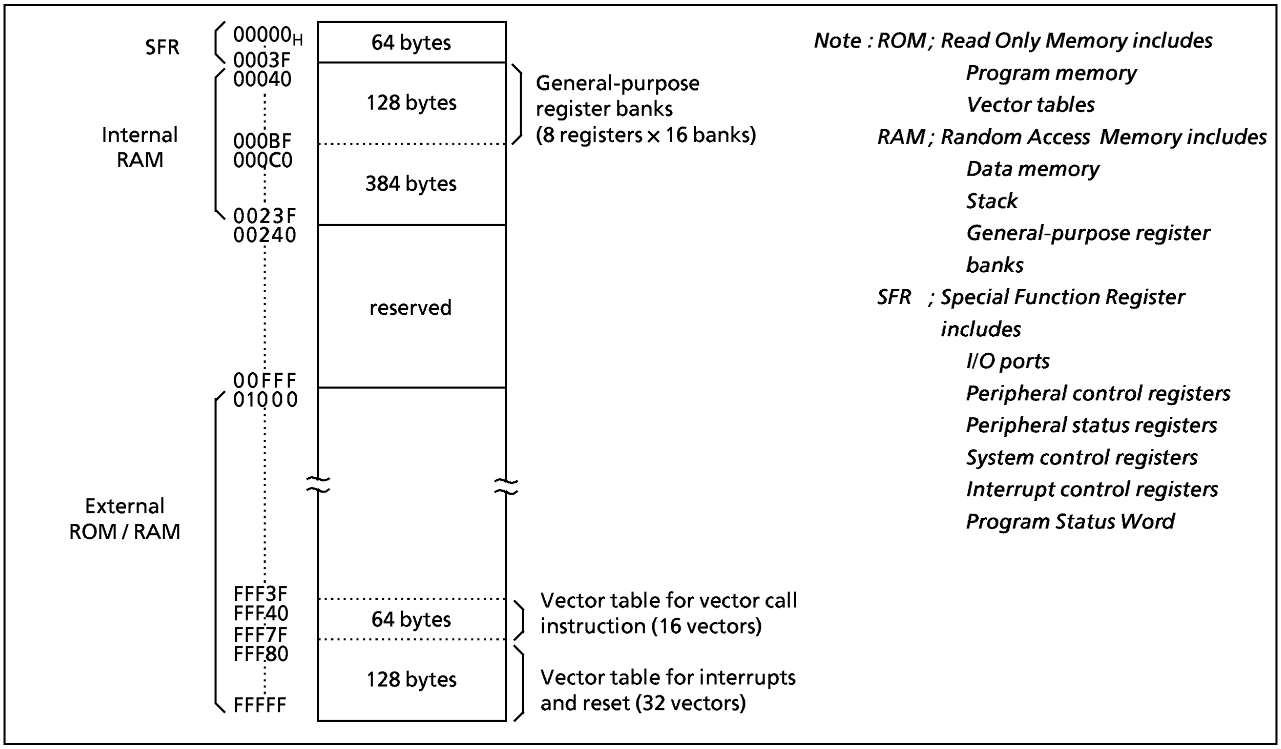
1. CPU CORE FUNCTIONS

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

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

1.1 Memory Address Map

The TLCS-870 / X Series is capable of addressing 1M bytes of memory. Figure 1-1 shows the memory address map of the 88C060. The memory of the 88C060 is organized with 3 address spaces such as ROM, RAM SFR (Special Function Register). It uses a memory mapped I/O system, and all I/O registers are mapped in the SFR address space. There are 16 banks of the general-purpose register. The register banks are also assigned to the first 128 bytes of the RAM address space.



Note : ROM; Read Only Memory includes
Program memory
Vector tables
RAM; Random Access Memory includes
Data memory
Stack
General-purpose register banks
SFR ; Special Function Register includes
I/O ports
Peripheral control registers
Peripheral status registers
System control registers
Interrupt control registers
Program Status Word

Figure 1-1. Memory Address Map

1.2 Program Memory (ROM)

The 88C060 can address up to 1M bytes of external program memory space except the first 4K bytes space (00000_H to $00FFF_H$).

The 88C060 does not have internal ROM. An external program memory must be connected.

1.3 Data Memory (ROM)

The 88C060 can address up to 1M bytes of data memory space. Data memory consists of internal data memory (on-chip RAM) and external data memory (RAM and / or ROM). The 88C060 has 512 bytes of static RAM. The first 128 bytes (00040_H to $000BF_H$) of the internal RAM are also used as general-purpose register banks.

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

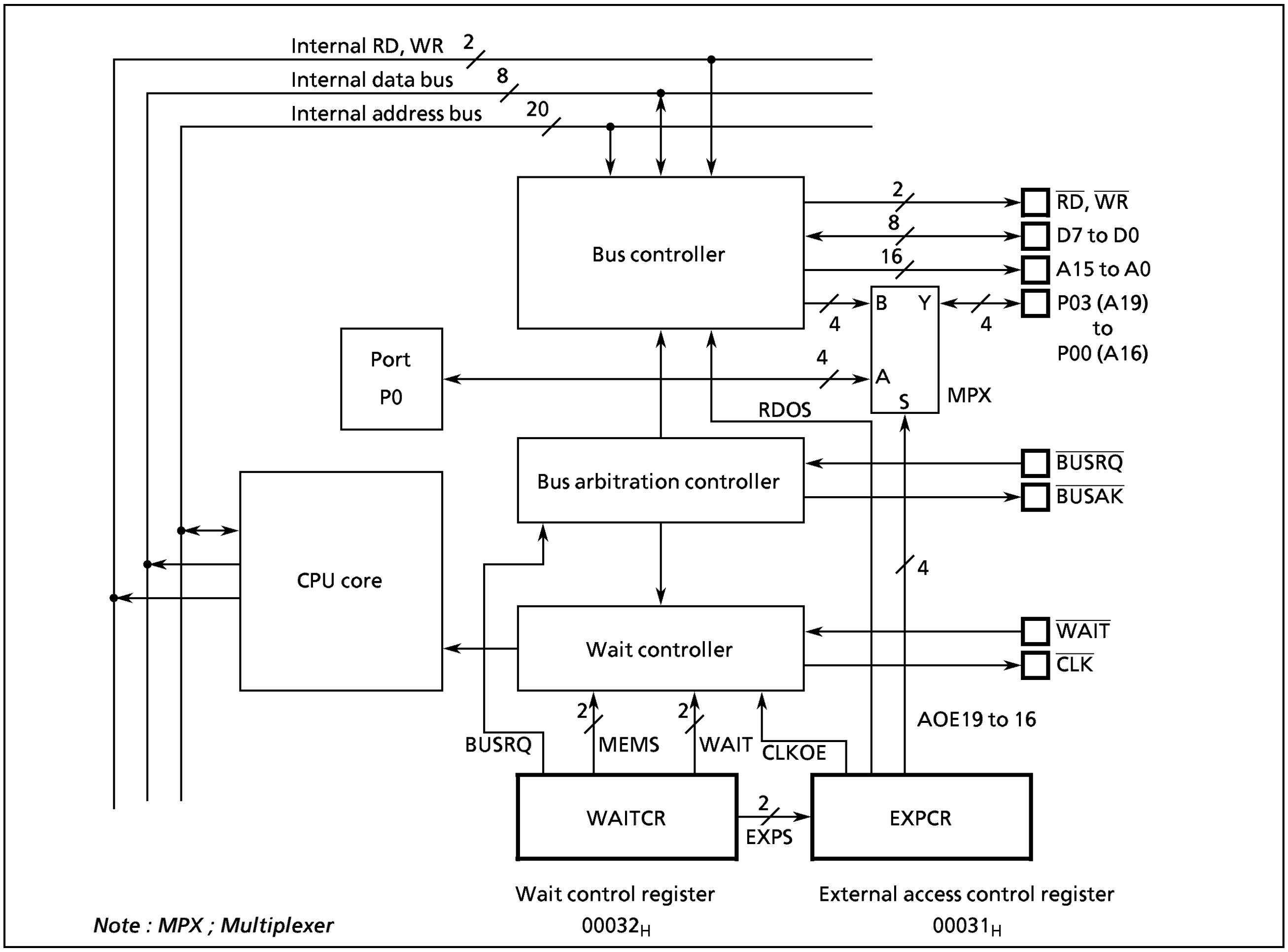
Example : Clears RAM to "00_H" except the bank 0.

```
           LD     HL, 0048H    ; Sets start address to HL register pair
           LD     A, H         ; Sets initial data (00H) to A register
           LD     BC, 01F7H    ; Sets number of byte to BC register pair
SRAMCLR :  LD     (HL + ), A
           DEC    BC
           JRS    F, SRAMCLR
```

> *Note : The general-purpose registers are mapped in the RAM ; therefore, do not clear RAM at the current bank addresses.*

1.4 External Memory Interfaces

The 88C060 is interfaced to the external memory (ROM / RAM) via an address bus, a data bus and a control bus. The address and data buses are separated, that is a non-multiplexed method.



Note : MPX ; Multiplexer

Figure 1-2. External Memory Interface

1.4.1 Control

The external memory interfaces are controlled by wait control register (WAITCR), external access control register (EXP0, EXP3).

The type of waits is selected by WAIT (bit 1, 0 in WAITCR), and the target memory area is selected by MEMS (bits 3 and 2 in WAITCR) for wait control. Bus arbitration is controlled by BUSRQ (bit 4 in WAITCR).

Writing a data to the external access control register (EXP0, EXP3), change EXP0, EXP3 using EXPS (bits 7 and 6 in WAITCR). The EXP0 controls the divided-by-4 cock output and $\overline{\text{RD}}$ output. The EXP3 controls whether address A19-16 pin that is also used as port P03-00 is used as a port or an address for 1 bit each. In the respective bits of EXP0 and EXP3, only CLKOE (bit 7 in EXP0) can be rewritten. The other bits are written at once. Only CLKOE (bit 7 in EXP0) is read in EXP control register, regardless of the value of EXPS (bits 7 and 6 of WAITCR).

WAIT Control Register

WAITCR (00032_H)	7	6	5	4	3	2	1	0
	EXPS			BUSRQ	MEMS		WAIT	

(Initial value : 00*0 0000)

WAIT	Wait mode selection	00 : Wait disable 01 : 1-cycle wait (Regardless the $\overline{\text{WAIT}}$ status) 10 : 1-cycle wait (Sense the $\overline{\text{WAIT}}$ status) 11 : n-cycle wait (During the $\overline{\text{WAIT}}$ status "L")	R/W
MEMS	Wait target memory selection	00 : 10000 to FFFFF_H 01 : 80000 to FFFFF_H 10 : 00240 to FFFFF_H 11 : reserved	
BUSRQ	Bus arbitration control	0 : Disable 1 : Enable	
EXPS	EXPCR selection	00 : EXP0 01 : reserved 10 : reserved 11 : EXP3	

Note 1 : The wait must not be used in SLOW mode.
*Note 2 : * ; don't care*

External Access Control Register

EXP0 (00031_H)	7	6	5	4	3	2	1	0
	CLKOE	"1"	"1"	RDOS	"1"	"1"	"1"	"1"

(Initial value : 0110 1111)

CLKOE	Divided-by-4 clock output enable / disable	0 : Disable 1 : Enable	R/W
RDOS	$\overline{\text{RD}}$ output selection	0 : $\overline{\text{RD}}$ output only for external memory access 1 : Always $\overline{\text{RD}}$ output	Write only

Note : As the divided-by-4 clock output is enabled during n cycle wait that WAIT (bit 1, 0 in WAITCR) = 11, the divided-by-4 clock can be out in wait cycle. As the divided-by-4-clock output is enabled at other that wait cycle, the divided-by-4 clock is always out.

EXP3 (00031_H)	7	6	5	4	3	2	1	0
					AOE19	AOE18	AOE17	AOE16

(Initial value : **** 1111)

AOE19	P03 / A19 pin configuration	0 : P03 input / output port 1 : A19 output	Write only
AOE18	P02 / A18 pin configuration	0 : P02 input / output port 1 : A18 output	
AOE17	P01 / A17 pin configuration	0 : P01 input / output port 1 : A17 output	
AOE16	P00 / A16 pin configuration	0 : P00 input / output port 1 : A16 output	

*Note : * : don't care*

Figure 1-3. Wait Control Register, External Access Control Register

1.4.2 Bus Controller

The 88C060 is interfaced to the external memory via the following buses.

(1) **Address bus** : A19 to A0

The 20-bit address bus provides a 1M byte memory address space. A19 to A16 are also used as port P03 to P00. It can select the ports for 1 bit each, using EXP control register.

(2) **Data bus** : D7 to D0

This is an 8-bit bidirectional data bus. During a write cycle, the CPU sends the write data on the data bus. During an external memory read cycle (instruction fetch or data read), the CPU receives the data from the external memory.

(3) **Control bus** : $\overline{\text{RD}}$ and $\overline{\text{WR}}$

The $\overline{\text{RD}}$ is the read signal output, the $\overline{\text{WR}}$ is the write signal output. In an external memory read cycle, the $\overline{\text{RD}}$ is level "L". In an external memory write cycle, the $\overline{\text{WR}}$ is level "L". In a dummy cycle or an internal memory read / write cycle, both of $\overline{\text{RD}}$ and $\overline{\text{WR}}$ remain to be level "H".
The $\overline{\text{RD}}$ can output level "L" by EXP control register at an external / internal memory read.

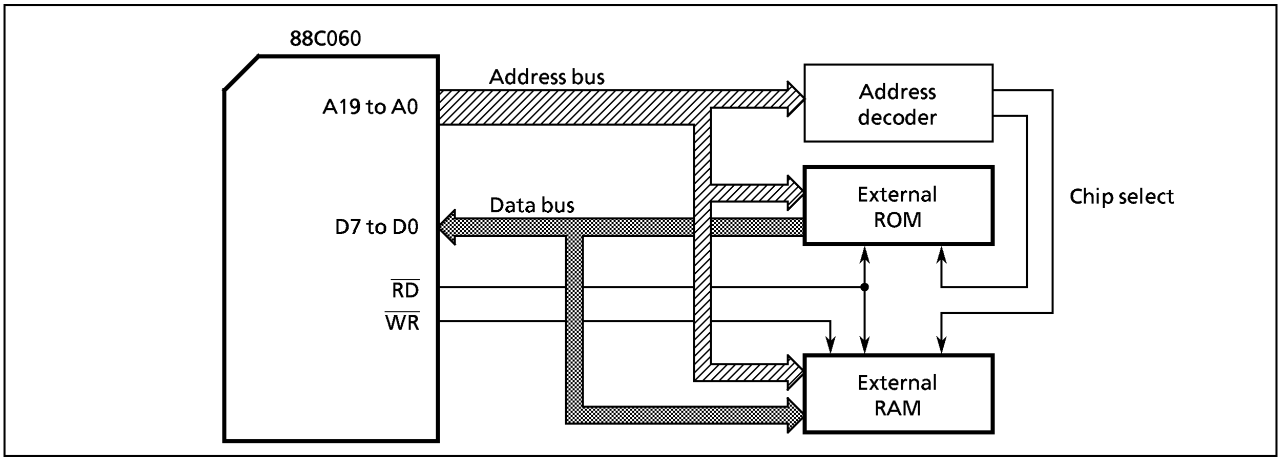


Figure 1-4. External ROM / RAM Interface

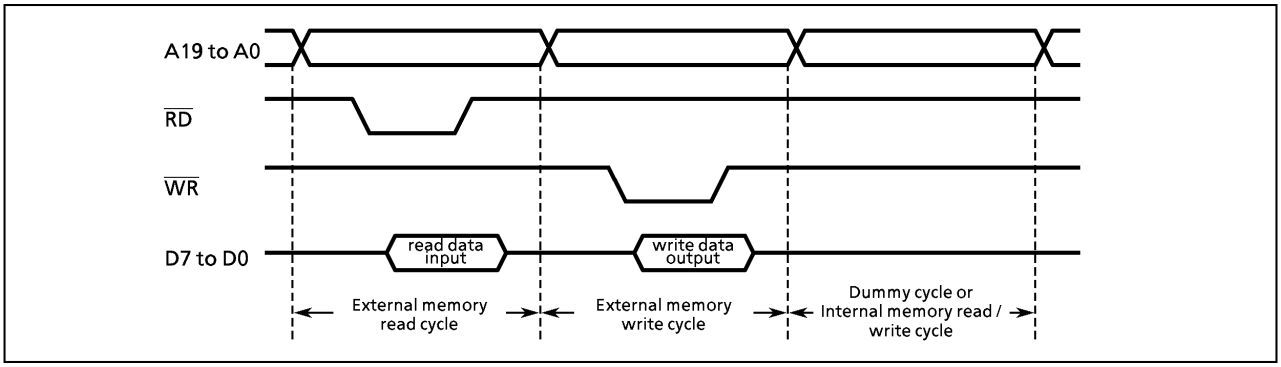

A19 to A0
$\overline{\text{RD}}$
$\overline{\text{WR}}$
D7 to D0
read data input
write data output
External memory read cycle
External memory write cycle
Dummy cycle or Internal memory read / write cycle

Figure 1-5. Timing Chart for External Memory Interface

1.4.3 Wait Controller

For an external memory access, wait mode can be selected from no wait, 1-cycle wait (regardless the $\overline{\text{WAIT}}$ pin status / sense) and n-cycle wait mode. Additionally, it can select the memory of the target wait.

(1) 1-cycle wait mode

Setting WAIT (bit 1, 0 in WAITCR) to 01 inserts a 1-cycle wait regardless of the $\overline{\text{WAIT}}$ pin status. Setting these bits to 10 samples the $\overline{\text{WAIT}}$ pin status and inserts a 1-cycle wait as the $\overline{\text{WAIT}}$ pin status is "L", and no wait as the $\overline{\text{WAIT}}$ pin status is "H".
The timing chart of 1-cycle wait is shown in figure 1-6.

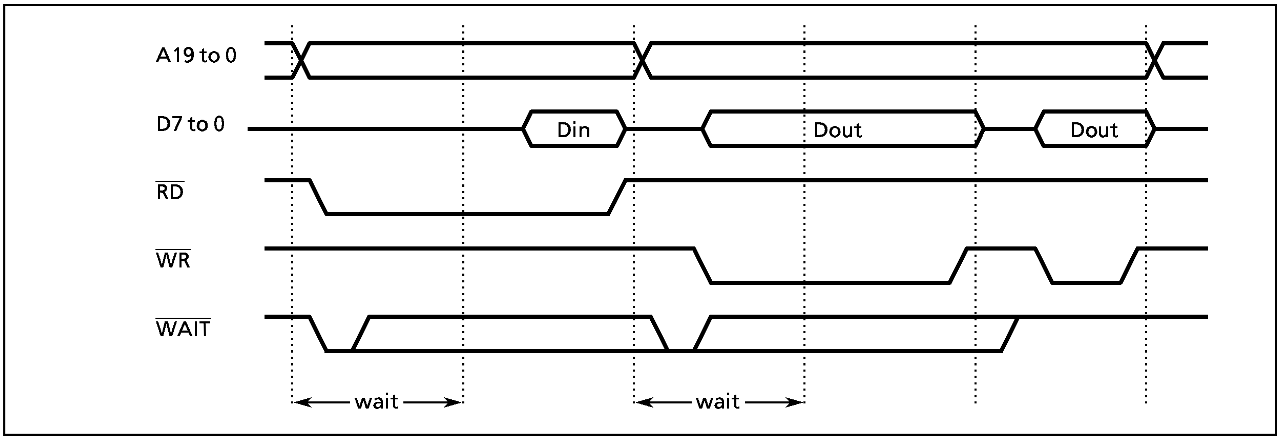


Figure 1-6. 1-cycle Wait

(2) n-cycle wait mode

Setting WAIT to "11" samples the $\overline{\text{WAIT}}$ pin status and inserts a n-cycle wait while the $\overline{\text{WAIT}}$ pin status is "L". The timing chart of n-cycle wait is shown in figure 1-7.

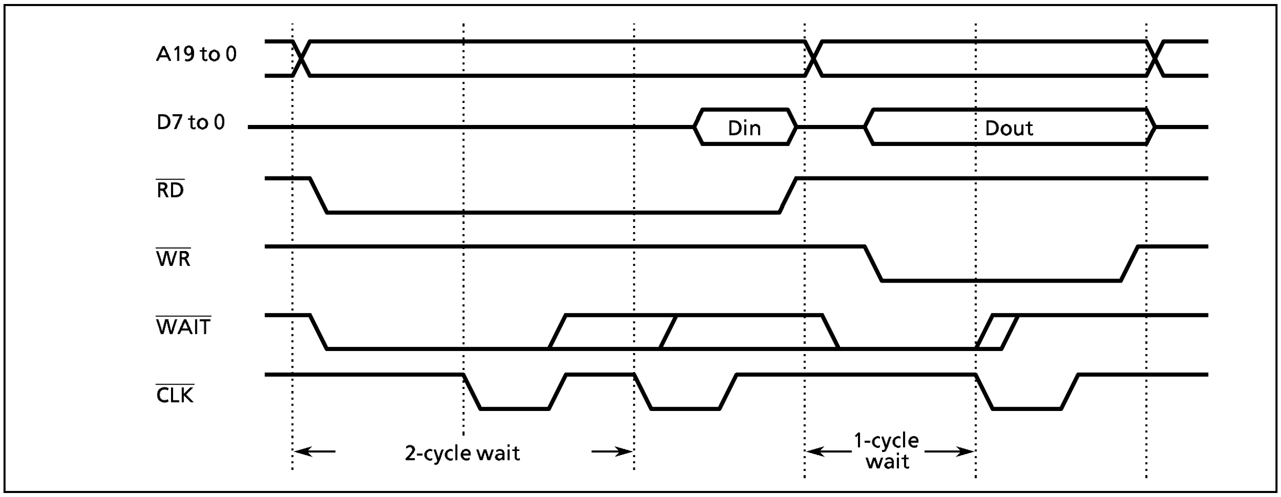
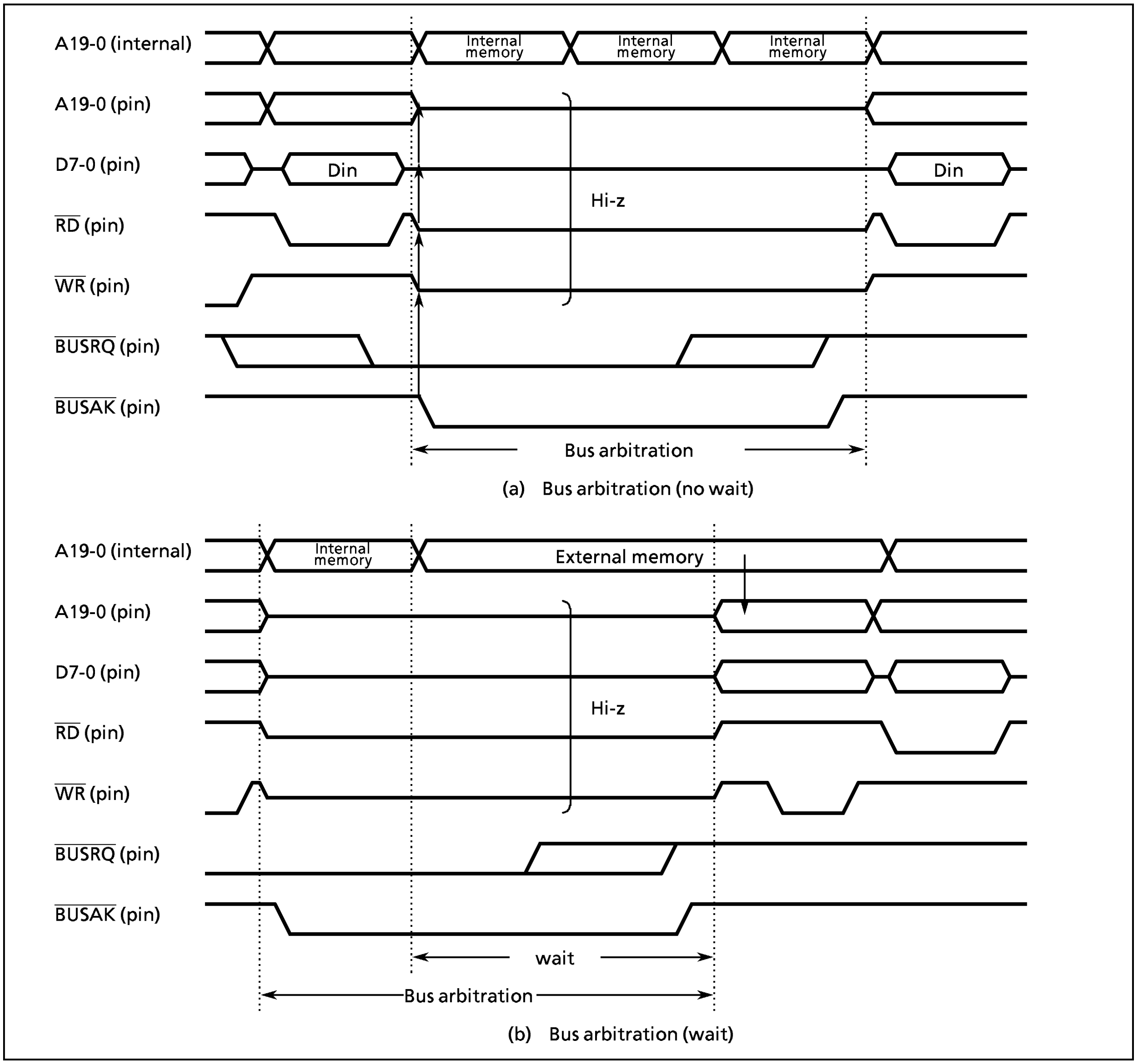


Figure1-7. n-cycle Wait

1.4.4 Bus Arbitration Controller

When the bus arbitration control is enabled and the bus arbitration request input from $\overline{BUSRQ}$ pin is level "L", buses are arbitrated. During bus arbitration, the 88C060 sets Hi-z for A19 to A0, D7 to D0, $\overline{RD}$, and $\overline{WR}$ pins and level "L" for $\overline{BUSAK}$ pin. The access for an internal memory inserts no wait cycle. The external access inserts wait cycles until the bus arbitration is ended.



(a) Bus arbitration (no wait)

(b) Bus arbitration (wait)

Figure 1-8. Bus Arbitration

1.5 System Clock Controller

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

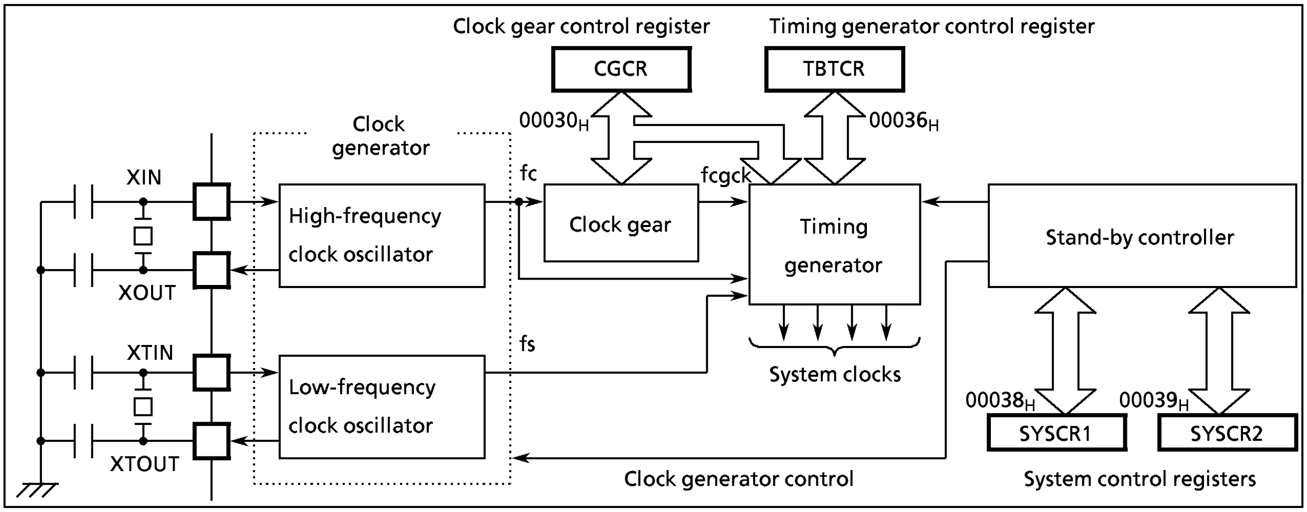


Figure1-9. System Clock Control

1.5.1 Clock Generator

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

The high-frequency (fc) and low-frequency (fs) clocks can 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 / XOUT pin not connected. The 88C060 are not provided a RC oscillation.

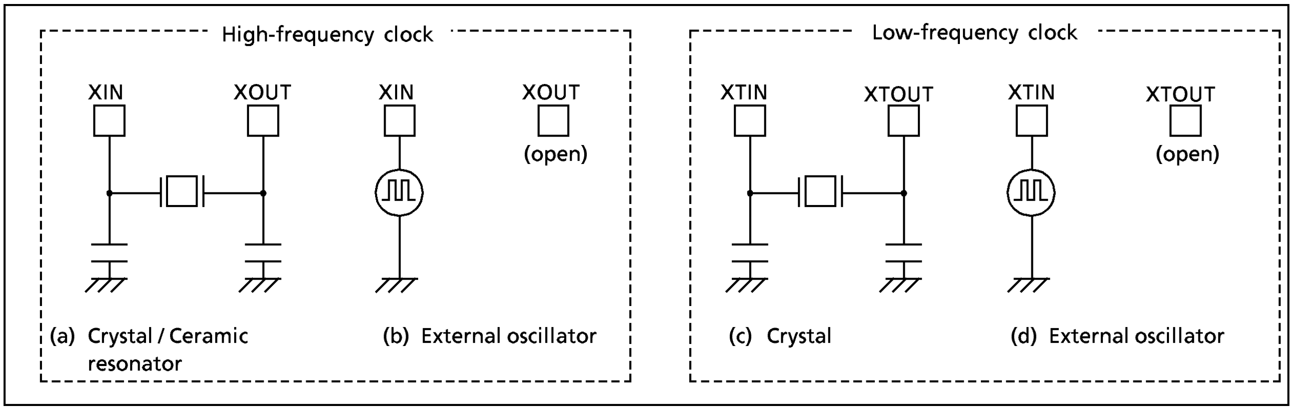


Figure 1-10. Examples of Resonator Connection

Note : Accurate Adjustment of the Oscillation Frequency

Although no hardware to externally and directly monitor the basic clock pulse is not provided, the oscillation frequency can be adjusted by making the program to output fixed frequency pulse (ex.$\overline{\text{CLK}}$ output) to the port with disabling all interrupts and watchdog timers, and monitoring this pulse. With a system requiring adjustment of the oscillation frequency, the adjusting program must be created beforehand.

1.5.2 Clock gear

The clock gear is a circuit to select a gear clock fcgck, that is the basis of the main system clock supplied to the timing generator, from high-frequency clock fc, or the divided clock fc/2, fc/4, fc/8, or fc/16.
Power consumption can be reduced by switching of the high-frequency from fc to fc/2, fc/4, fc/8, and fc/16 with the clock gear using.
The clock gear consists of a 4-stage prescaler with a multiplexer.

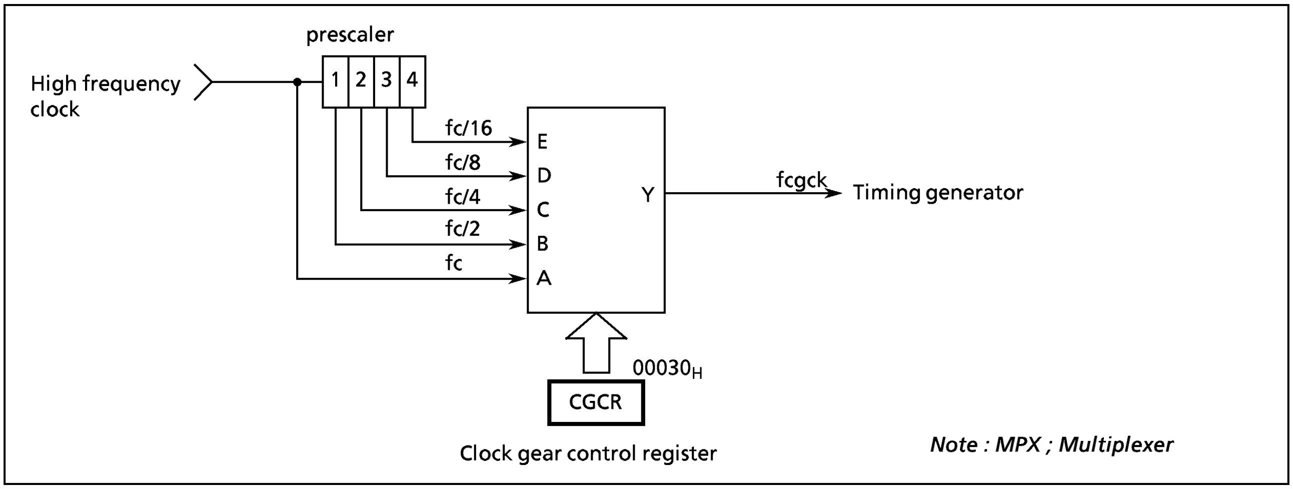
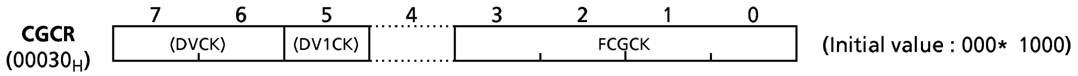


Figure 1-11. Configuration of Clock Gear



FCGCK	Gear clock selection (write) / gear clock monitor (read)	0*** : reserved 1000 : fc 1001 : fc/2 1010 : fc/4 1011 : fc/8 1100 : fc/16 1101 : reserved 111* : reserved	R/W

*Note 1 : fc ; High-frequency clock, * ; don't care*
Note 2 : Bit 4 in CGCR is always read in as "1" when a read instruction is executed.

Figure 1-12. Clock Gear Control Register

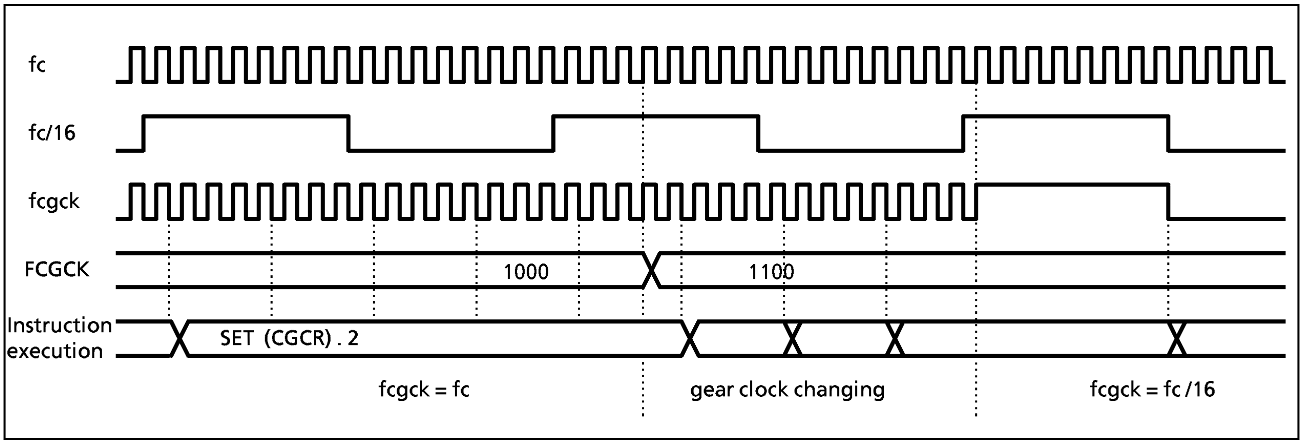


Figure 1-13. Example of Clock Exchangeable Timing by Clock Gear

1.5.3 Timing Generator

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

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

(1) Configuration of timing generator

The timing generator consists of a 3-stage prescaler, a 21-stage divider, a main system clock generator, and machine cycle counters.

The clock fc/4 or fc/8, that is output from the 2nd stage or the 3rd stage of the prescaler, can be selected as the clock to input to the 1st stage of the divider by DV1CK (bit 5 in CGCR). Inputting fc/8 to the 1st stage of the divider operates the peripheral circuit without the setting change when the operation clock is multiplied by 2. (Example : 8 MHz to 16 MHz)

Even if the main system clock is changed by the clock gear, the output from the divider is not changed. The peripheral circuit using high-speed divider output (1st to 4th output) can not be used when the main system clock slows down. In this case, setting DVCK (bits 7 and 6 in CGCR) can change high-speed divider output to low-speed divider output. The DVCK should be set according to the lowest speed of the clock gear and divider output used for the peripheral circuit prior to starting the peripheral circuits. Do not change the set value after setting.

An input clock to the 7th stage of the divider depends on the operating mode, DV1CK (bit 5 in CGCR), and DV7CK (bit 4 in TBTCR), that is shown in table 1-2. As reset and STOP mode started / canceled, the prescaler and the divider are cleared to "0".

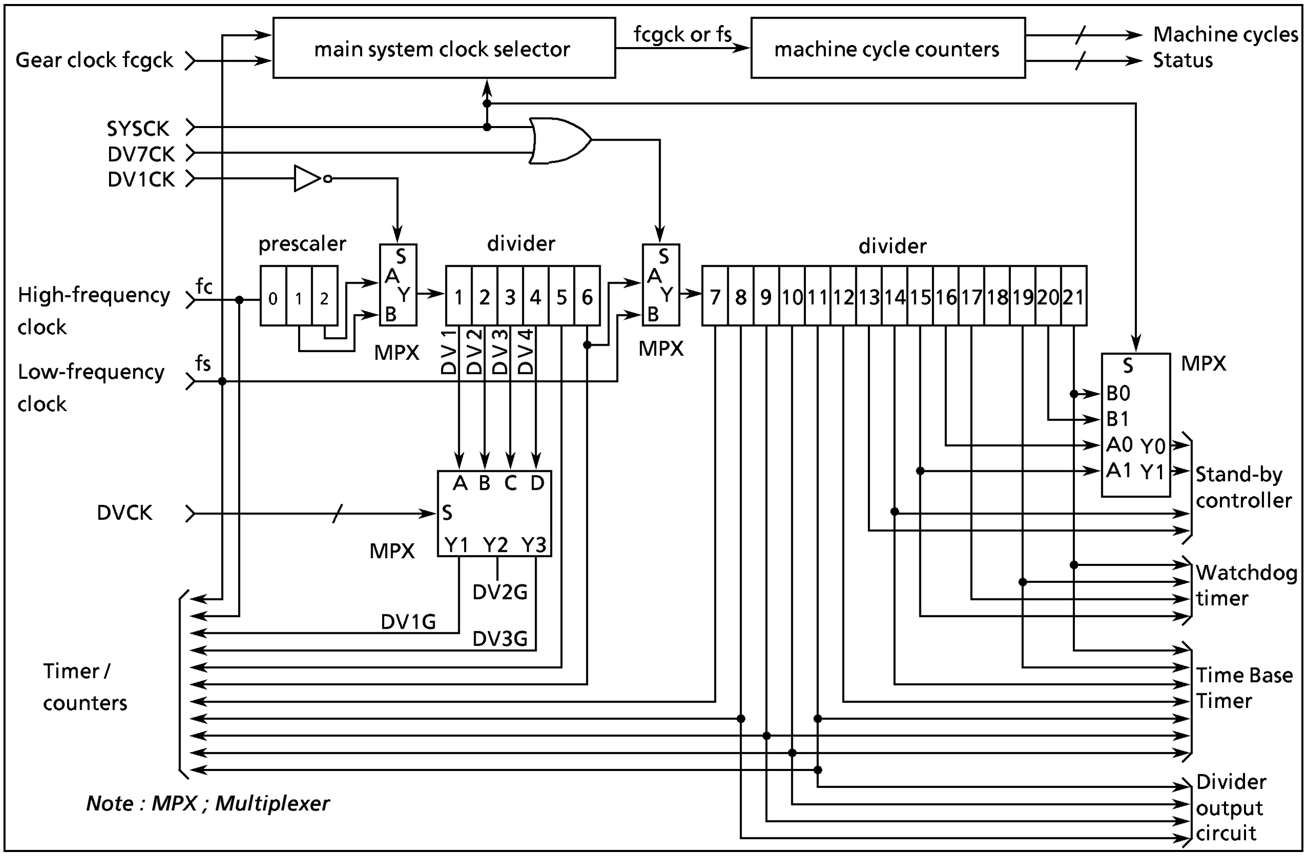


Figure 1-14. Configuration of Timing Generator

CGCR
(00030$_H$)

7	6	5	4	3	2	1	0
DVCK		DV1CK		(FCGCK)			

(Initial value : 000* 1000)

DVCK	Selection of divider output DV1G to DV3G	DV1G DV2G DV3G 00 : DV1 DV2 DV3 01 : DV2 DV2 DV3 10 : DV3 DV3 DV3 11 : DV4 DV4 DV4	R/W
DV1CK	Selection of input clock to the 1st stage of the divider	0 : fc/4 1 : fc/8	

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

Note 2 : Bit 4 in CGCR is always read as "1" when a read instruction is executed.

Figure 1-15. Clock Gear Control Register

TBTCR (00036$_H$)	7	6	5	4	3	2	1	0	
	(DVOEN)	(DVOCK)		DV7CK	(TBTEN)	(TBTCK)			(Initial value : 0**0 0***)

DV7CK	Selection of input clock to the 7 th stage of the divider	0 : fc/2^8 or fc/2^9 [Hz] 1 : fs	R/W

*Note 1 : fc : High-frequency clock [Hz], fs : Low-frequency clock [Hz], *: don't care*
Note 2 : Do not set DV7CK to "1" in the single clock mode.
Note 3 : Do not set DV7CK to "1" before low-frequency clock is stable in the dual-clock mode.

Figure 1-16. Timing Generator Control Register

Table 1-1. Divider Output Capability

DVCK	Gear clock frequency	Divider output capability									
		DV1CK = 0					DV1CK = 1				
		DV1G	DV2G	DV3G	DV4	DV5	DV1G	DV2G	DV3G	DV4	DV5
00	fcgck = fc	×	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	×	×	○	○	○	×	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	×	×	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
01	fcgck = fc	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	×	×	○	○	○	○	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	×	×	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
10	fcgck = fc	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/4	×	×	×	○	○	○	○	○	○	○
	fcgck = fc/8	×	×	×	×	○	×	×	×	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○
11	fcgck = fc	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/2	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/4	○	○	○	○	○	○	○	○	○	○
	fcgck = fc/8	×	×	×	×	○	○	○	○	○	○
	fcgck = fc/16	×	×	×	×	×	×	×	×	×	○

Table 1-2. Input Clock to 7th Stage of The Divider

Single-clock mode		Dual-clock mode			
NORMAL1, IDLE1 mode		NORMAL2, IDLE2 mode (SYSCK = 0)			SLOW, SLEEP mode (SYSCK = 1)
DV1CK = 0	DV1CK = 1	DV7CK = 0		DV7CK = 1	
		DV1CK = 0	DV1CK = 1		
fc/2^8	fc/2^9	fc/2^8	fc/2^9	fs	fs

Note 1 : Do not set DV7CK to "1" in the single clock mode.
Note 2 : In SLOW and SLEEP mode, the input clock to the 1st stage of the divider is stopped; output from the 1st to 6th stages is also stopped.

(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 15 different types of instructions for the TLCS-870/X Series : ranging from 1-cycle instructions which require one machine cycle for execution to 15-cycle instructions which require 15 machine cycles for execution.

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

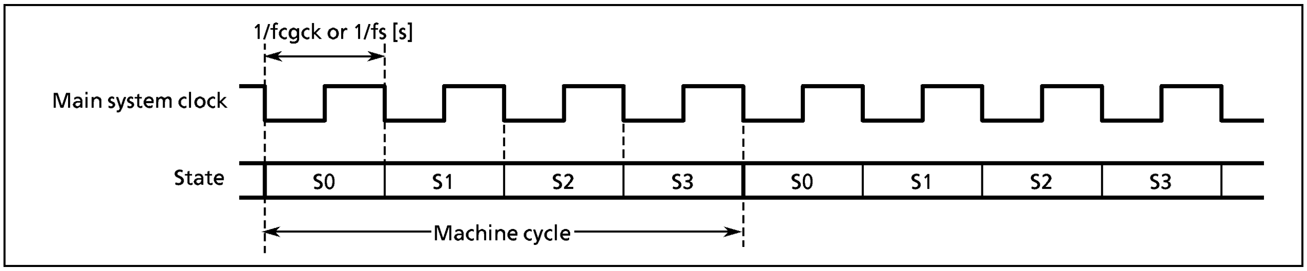


Figure 1-17. Machine Cycle

Table 1-3. Example of Machine Cycle

Frequency		Machine cycle				
		fcgck = fc	fcgck = fc/2	fcgck = fc/4	fcgck = fc/8	fcgck = fc/16
High-frequency clock	fc = 12.5 MHz	0.32 μs	0.64 μs	1.28 μs	2.56 μs	5.12 μs
	fc = 4.2 MHz	0.95 μs	1.9 μs	3.8 μs	7.6 μs	–
Low-frequency clock	fs = 32.768 kHz	122 μs				

1.5.4 Stand-by Controller

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

Figure 1-18 shows the operating mode transition diagram and Figure 1-19 shows the system control registers.

(1) **Single-clock mode**

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

① NORMAL1 mode

In this mode, both the CPU core and on-chip peripherals operate using the high-frequency clock. The 88C060 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 the system control register 2 (SYSCR2), and IDLE1 mode is released to NORMAL1 mode by an interrupt request from the on-chip peripherals or external interrupt inputs. When 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 STOP1 mode

In this mode, the internal oscillation circuit is turned off, causing all system operations to be halted. The internal status immediately prior to the halt is held with a lowest power consumption during STOP1 mode.

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

(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/fcgck [s] in the NORMAL2 and IDLE2 modes, and 4/fs [s] (122 μs at fs = 32.768 kHz) in the SLOW and SLEEP modes.

The TLCS-870/X is placed in the signal-clock mode during reset. To use the dual-clock mode, the low-frequency oscillator should be turned on by executing [SET (SYSCR2), XTEN] instruction.

① NORMAL2 mode

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

② SLOW mode

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

Switching back and forth between NORMAL2 and SLOW modes are performed by the system control register 2 (SYSCR2).

③ IDLE2 mode

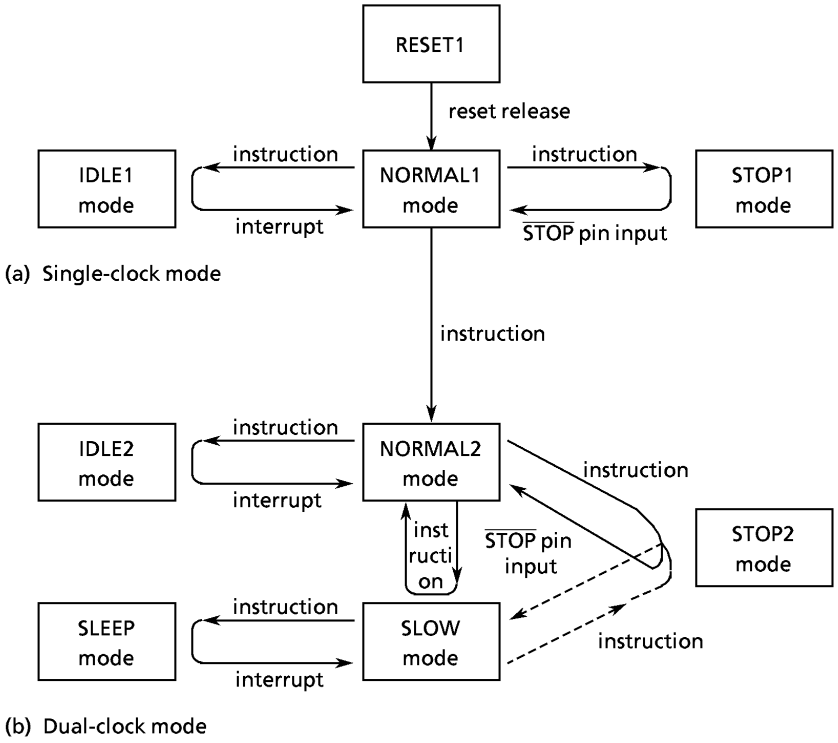
In this mode, the internal oscillation circuit remain active. The CPU and the watchdog timer are halted ; however, on-chip peripherals remain active (operate using the high-frequency clock and / or the low-frequency clock). Starting and releasing of IDLE2 mode are the same as for IDLE1 mode, except that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

④ SLEEP mode

In this mode, the internal oscillation circuit of the low-frequency clock remains active. The CPU, the watchdog timer, and the internal oscillation circuit of the high-frequency clock are halted ; however, on-ship 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.

⑤ STOP2 mode

As in STOP1 mode, all system operations are halted in this mode. As in NORMAL2 mode at the start, the operating mode returns to NORMAL2 mode, and as in SLOW mode at the start, it returns to SLOW mode after release.



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

Operating mode		Frequency		CPU core	On-chip Peripherals	Machine cycle time
		High-frequency	Low-frequency			
Single-Clock	RESET	turning on oscillation	turning off oscillation	reset	reset	4/fcgck [s]
	NORMAL1			operate	operate	
	IDLE1			halt		
	STOP1	turning off oscillation			halt	—
Dual-Clock	NORMAL2	turning on oscillation	turning on oscillation	High-frequency	operate (High and / or Low)	4/fcgck [s]
	IDLE2			halt		
	SLOW	turning off oscillation		Low-frequency	Low-frequency	4/fs [s]
	SLEEP			halt		
	STOP2		turning off oscillation		halt	—

Figure 1-18. Operating Mode Transition Diagram

System Control Register 1

SYSCR1
(00038_H)

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

(Initial value : 0000 00**)

<table>
<tr><td>STOP</td><td>STOP mode start</td><td colspan="4">0 : CPU core and peripherals remain active
1 : CPU core and peripherals are halted (start STOP mode)</td><td rowspan="6">R/W</td></tr>
<tr><td>RELM</td><td>Release method for STOP mode</td><td colspan="4">0 : Edge-sensitive release
1 : Level-sensitive release</td></tr>
<tr><td>RETM</td><td>Operating mode after STOP mode</td><td colspan="4">0 : Return to NORMAL mode
1 : return to SLOW mode</td></tr>
<tr><td rowspan="3">WUT</td><td rowspan="3">Warming-up time at releasing STOP mode</td><td rowspan="2"></td><td colspan="2">Return to NORMAL mode</td><td rowspan="2">Return to SLOW mode</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00
01
10
11</td><td>3×2^{16}/fc
2^{16}/fc
3×2^{14}/fc
2^{14}/fc</td><td>3×2^{17}/fc
2^{17}/fc
3×2^{15}/fc
2^{15}/fc</td><td>3×2^{13}/fs
2^{13}/fs</td></tr>
</table>

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 NORMAL 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 : Always set bit 4 in SYSCR1 to "1" when STOP mode is started.

System Control Register 2

SYSCR2
(00039_H)

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

(Initial value: 1000 ****)

XEN	High-frequency oscillator control	0 : Turn off oscillation 1 : Turn on oscillation	R/W
XTEN	Low-frequency oscillator control	0 : Turn off oscillation 1 : Turn on oscillation	
SYSCK	Main system clock select (write) / main system clock monitor (read)	0 : High-frequency clock 1 : Low-frequency clock	
IDLE	IDLE mode start	0 : CPU and watchdog timer remain active 1 : CPU and watchdog timer are stopped (start IDLE1 mode)	

Note 1 : XEN and SYSCK are automatically overwritten in accordance with the contents of RETM (bit 5 in SYSCR1) when STOP mode is released.

RETM	operating mode after STOP mode	XEN	SYSCK
0	NORMAL 1/2 mode	1	0
1	SLOW mode	0	1

Note 2 : A reset is applied ($\overline{\text{RESET}}$ pin output goes low) 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 3 : * ; don't care*

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

Figure 1-19. System Control Registers

1.5.5 Operating Mode Control

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

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

① Oscillations are turned off, and all internal operations are halted.
② The data memory (except for DBR), registers, 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 of the instruction but one to 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 (bit 6 in SYSCR1).

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

In this mode, STOP mode is released by setting the $\overline{\text{STOP}}$ pin high. This mode is used for capacitor back-up when the main power supply is cut off and long term battery back-up.
When the $\overline{\text{STOP}}$ pin input is high, executing an instruction which starts STOP mode will not place in STOP mode but instead will immediately start the release sequence (warm-up). Thus, to start STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low. The following 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 mode
SSTOPH:   TEST   (P2). 0              ;  Wait until the STOP pin input goes low level
          JRS    F, SSTOPH
          SET    (SYSCR1). 7          ;  Starts STOP mode
```

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

```
PINT5 :   TEST   (P2). 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 mode.
          SET    (SYSCR1). 7          ;  Starts STOP mode
SINT5 :   RETI
```

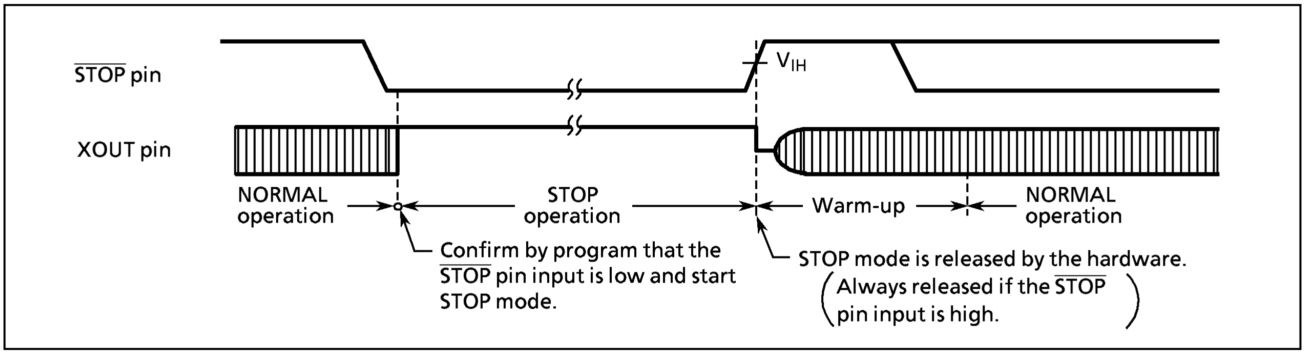

STOP pin
XOUT pin
V_IH
NORMAL operation
STOP operation
Warm-up
NORMAL operation
Confirm by program that the STOP pin input is low and start STOP mode.
STOP mode is released by the hardware.
(Always released if the STOP pin input is high.)

Figure 1-20. Level-sensitive Mode

Note 1	:	*Even if the STOP pin input is low after warming up start, the* ***STOP*** *mode is not restarted.*
Note 2	:	*In this case of changing to the level-sensitive mode from the edge-sensitive mode, the release mode is not switched until a rising edge of the STOP pin input is detected.*

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

In this mode, STOP mode is released by a rising edge of the 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 STOP pin. In the edge-sensitive mode, STOP mode is started even when the STOP pin input is high level.

Example : Starting STOP mode from NORMAL mode

LD (SYSCR1), 10010000B ; Starts after specified to the edge-sensitive mode

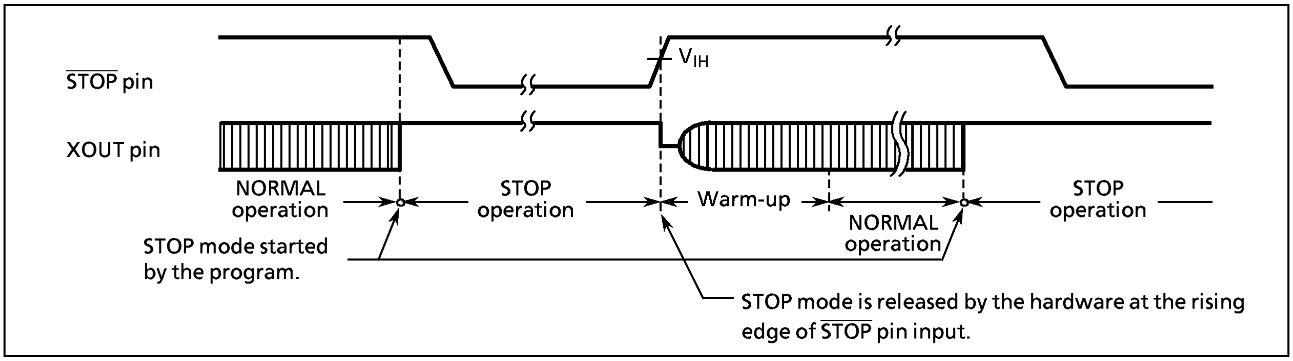

STOP pin
XOUT pin
V_IH
NORMAL operation
STOP operation
Warm-up
NORMAL operation
STOP operation
STOP mode started by the program.
STOP mode is released by the hardware at the rising edge of STOP pin input.

Figure 1-21. Edge-sensitive 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 SLOW mode, only the low-frequency clock oscillator is turned on. In the signal-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 warming-up times can be selected with the WUT (bits 2 and 3 in SYSCR1) in accordance with the resonator characteristics.

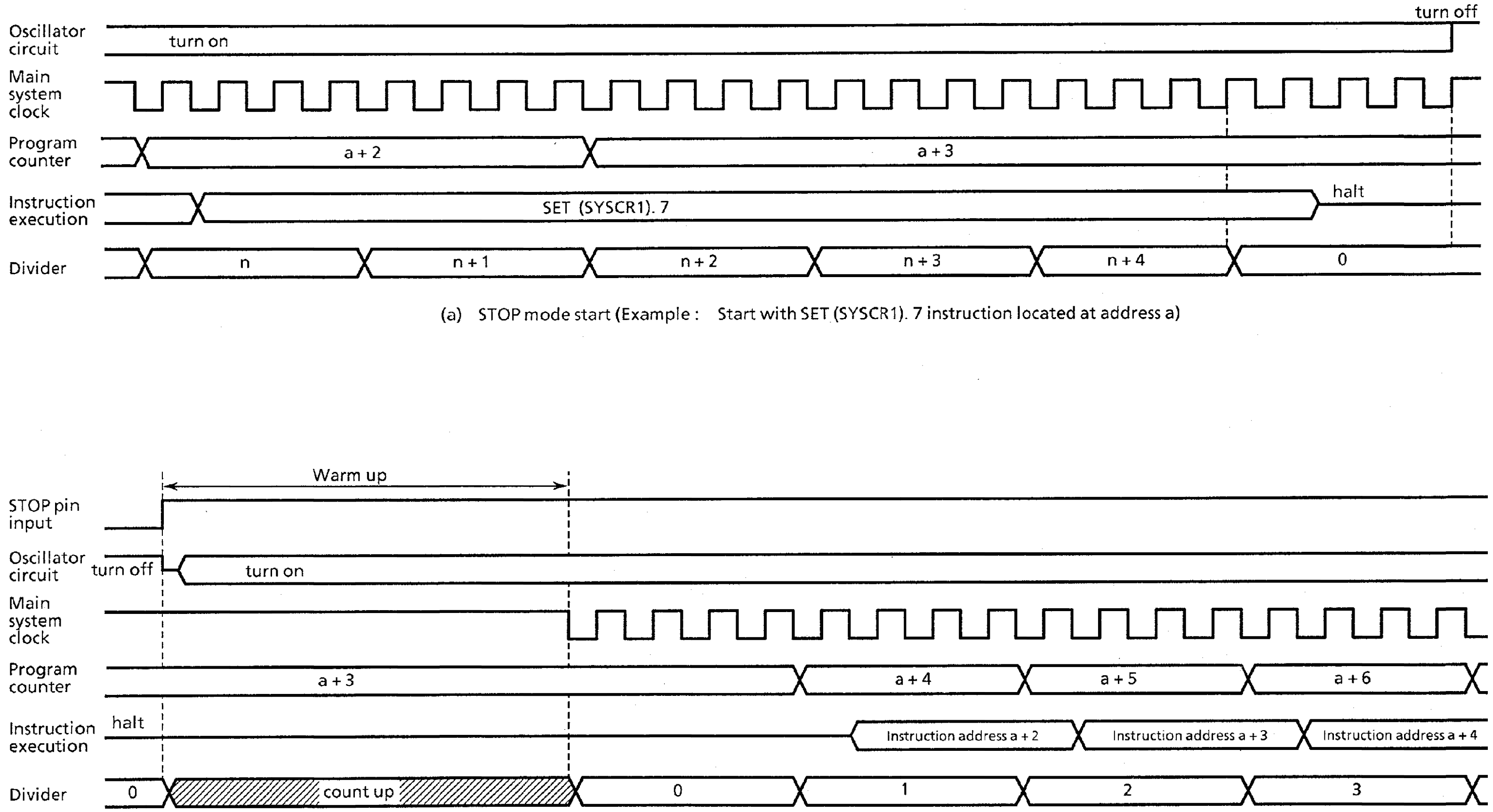
③ When the warm-up time has elapsed, normal operation resumes with the instruction following the STOP mode start instruction (e.g. [SET (SYSCR1). 7]). The start is made after the prescaler and the divider of the timing generator are cleared to "0".

Table 1-4. Warm-up Time Example (at fc = 12.5 MHz, fs = 32.768 kHz)

WUT	Warm-up Time [ms]		
	Return to NORMAL mode		Return to SLOW mode
	DV1CK = 0	DV1CK = 1	
00	15.729	31.457	750
01	5.243	10.486	250
10	3.932	7.864	—
11	1.311	2.621	—

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

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



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

(b) STOP mode release

Figure 1-22. STOP Mode Start / Release

*Note : When **STOP** mode is released with a low hold voltage, the following cautions must be observed.*
*The power supply voltage must be at the operating voltage level before releasing **STOP** mode. The $\overline{\text{RESET}}$ 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).*

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

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

① Operation of the CPU and watchdog timer (WDT) is halted. On-ship 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 IDLE mode was entered.
③ The program counter holds the address of the instruction but one to the instruction which starts IDLE mode.

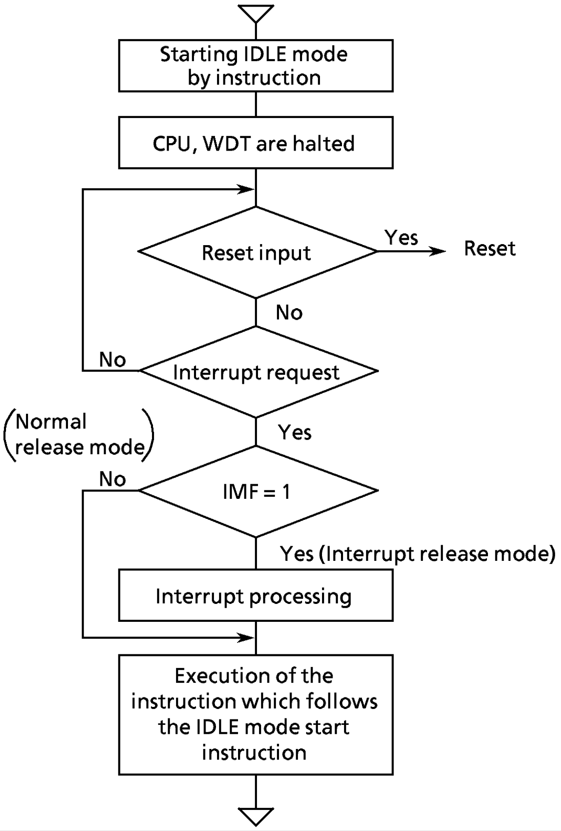

Starting IDLE mode by instruction
CPU, WDT are halted
Reset input
Yes
Reset
No
Interrupt request
No
Yes
(Normal release mode)
IMF = 1
No
Yes (Interrupt release mode)
Interrupt processing
Execution of the instruction which follows the IDLE mode start instruction

Figure 1-23. IDLE Mode

```
Example :   Starting IDLE mode.
                 SET     (SYSCR2). 4           ;   IDLE←1
```

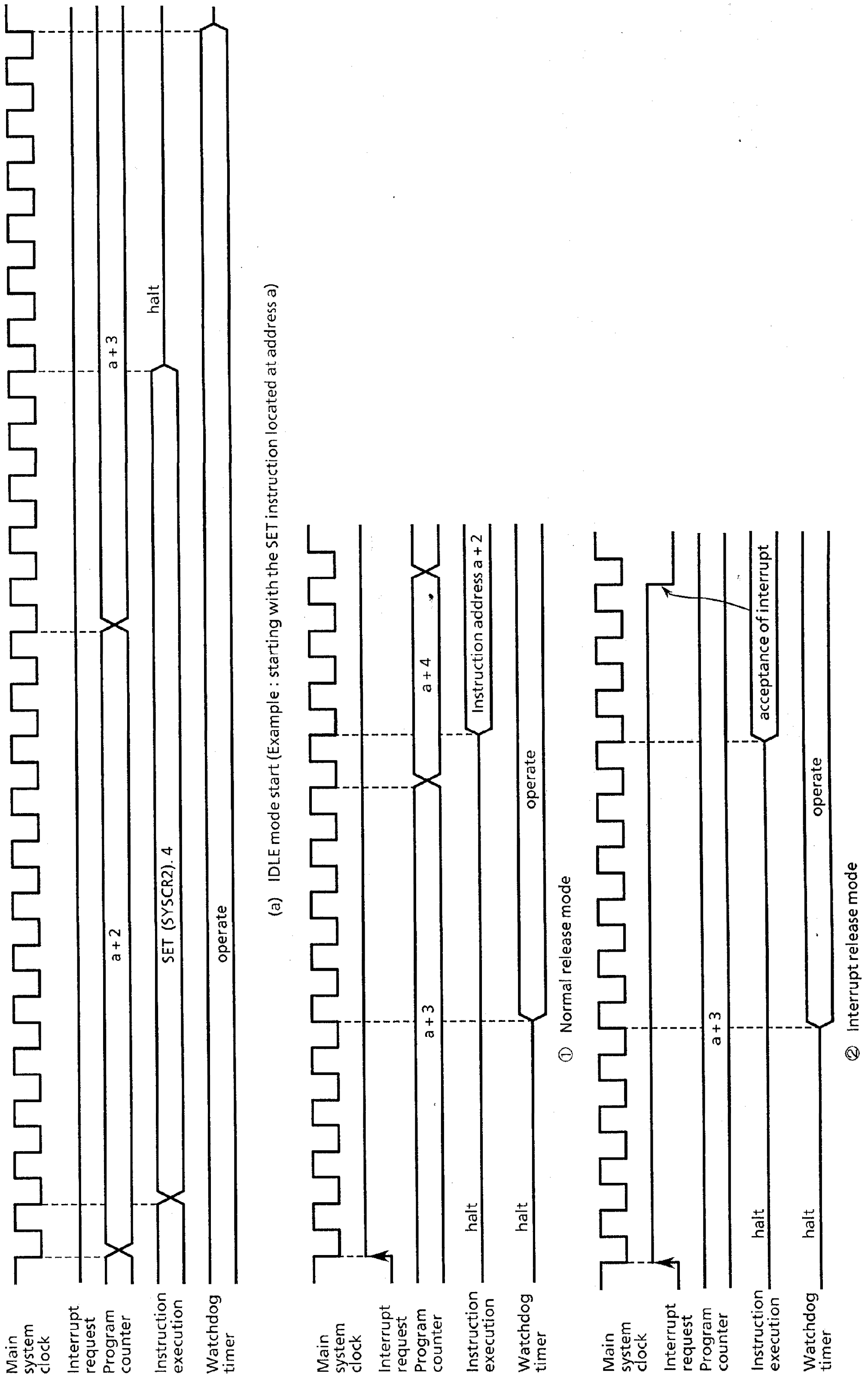
IDLE mode includes a normal release mode and an interrupt release mode. Selection is made with the interrupt master enable flag (IMF). Releasing IDLE mode returns from IDLE1 to NORMAL1, from IDLE2 to NORMAL2, and from SLEEP to SLOW mode.

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

b. Interrupt release mode (IMF = "1")
IDLE mode is released and interrupt processing is started by any interrupt source enabled with the individual interrupt enable flag (EF) or an external interrupt 0 ($\overline{\text{INT0}}$ pin) request. After the interrupt is processed, the execution resumes from the instruction following the instruction which starts IDLE mode.

IDLE mode can also be released by inputting low level on the $\overline{\text{RESET}}$ pin, which immediately performs the reset operation. After reset, the 88C060 is placed in NORMAL 1 mode.

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



(a) IDLE mode start (Example : starting with the SET instruction located at address a)

① Normal release mode

② Interrupt release mode

(b) IDLE mode release

Figure 1-24. IDLE Mode Start / Release

(3) **SLOW** mode

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

a. Switching from NORMAL2 mode to SLOW mode

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

> *Note : The high-frequency clock oscillation can be continued to return quickly to NORMAL2 mode.*

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

Example1 : Switching from NORMAL2 mode to SLOW mode.

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

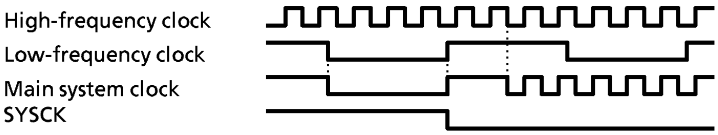
Example2 : Switching to the SLOW mode after low-frequency clock oscillation has stabilized.

```
          LD      (TC2CR), 14H     ;  Sets TC2 mode
                                      (timer mode, source clock : fs)
          LDW     (TREG2), 8000H   ;  Sets warming-up time
                                      (according to Xtal characteristics)
          SET     (EIRH). EF14     ;  Enables INTTC2
          LD      (TC2CR), 34H     ;  Starts TC2
           ⋮
PINTTC2 : LD      (TC2CR), 10H     ;  Stops TC2
          SET     (SYSCR2). 5      ;  SYSCK←1
                                      (switches the main system clock to the low-
                                      frequency clock)
          CLR     (SYSCR2). 7      ;  XEN←0
                                      (turns off high-frequency oscillation)
          RETI
           ⋮
VINTTC2 : DL      PINTTC2          ;  INTTC2 vector table
```

b. Switching from SLOW mode to NORMAL2 mode

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

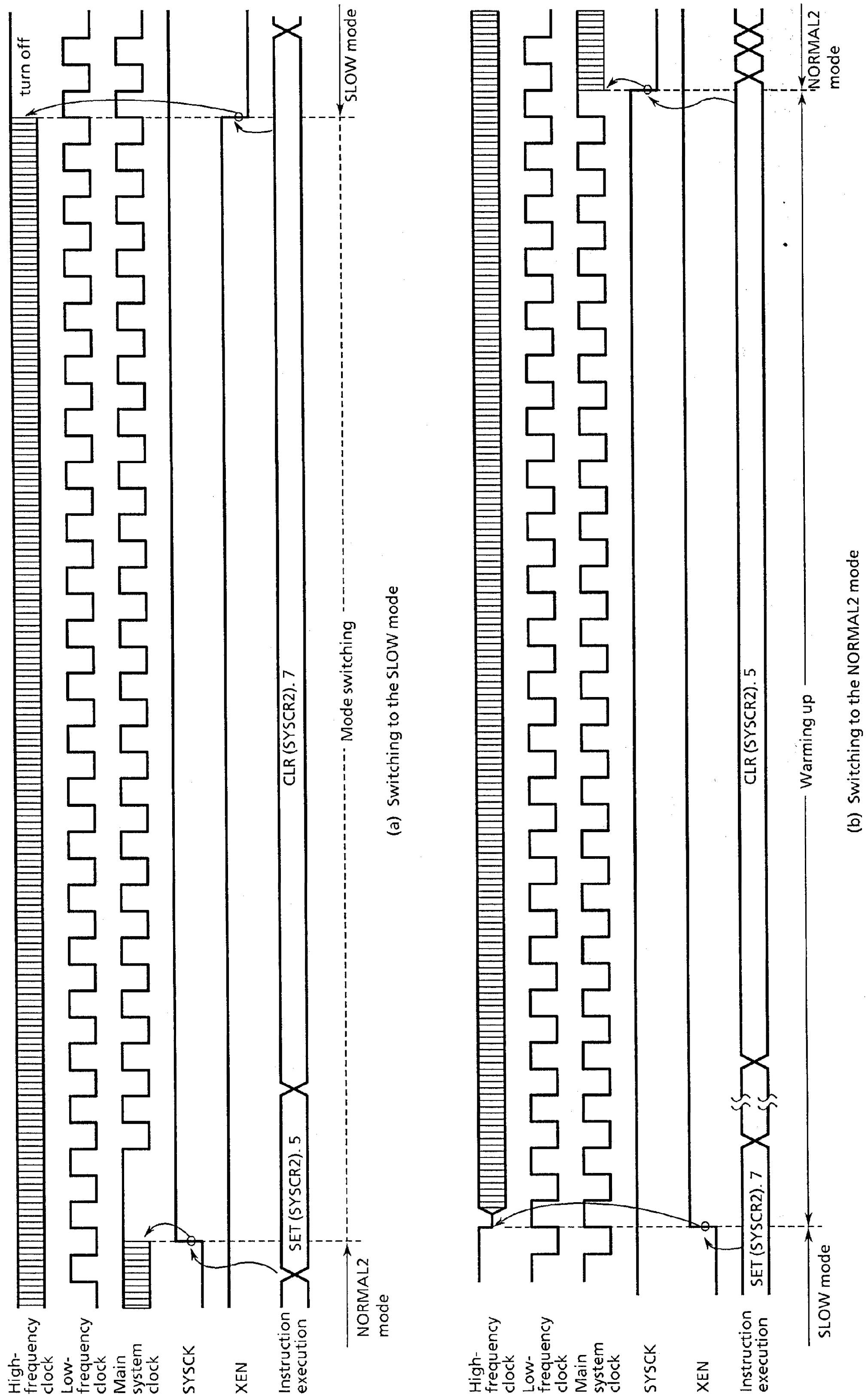
*Note 1 : After **SYSCK** is cleared to "0", executing the instructions is continued by the low-frequency clock for the period synchronized with low-frequency and high-frequency clocks.*



*Note 2 : **SLOW** mode can also be released by inputting low level on the **RESET** pin, which immediately performs the reset operation. After reset, the **88C060** is placed in **NORMAL1** mode.*

Example : Switching from SLOW mode to NORMAL2 mode (fc = 16 MHz, fcgck-fc, warming-up time is 4.0 ms).

```
           SET    (SYSCR2). 7        ; XEN←1 (turns on high-frequency oscillation)
           LD     (TC2CR), 10H       ; Sets TC2 mode
                                       (timer mode, source clock : fcgck)
           LD     (TREG2 + 1), 0F8H  ; Sets the warming-up time
                                       (according to frequency and Xtal characteristics)
           SET    (EIRH). EF14       ; Enables INTTC2
           LD     (TC2CR), 30H       ; Starts TC2
            ⋮
PINTTC2 :  LD     (TC2CR), 10H       ; Stops TC2
           CLR    (SYSCR2). 5        ; SYSCK←0
                                       (switches the main system clock to the high-
                                       frequency clock)
           RETI
            ⋮
VINTTC2 :  DL     PINTTC2            ; INTTC2 vector table
```



(a) Switching to the SLOW mode

(b) Switching to the NORMAL2 mode

Figure 1-25. Switching between the NORMAL2 and SLOW Modes

1.6 Interrupt Controller

The 88C060 has a total of 18 interrupt sources : 6 externals and 12 internals. Nested interrupts control with priorities are also possible. Two of the internal sources are pseudo non-maskable interrupts; the remainder are all maskable interrupts.

Table 1-5. Interrupt Sources

Interrupt source			Enable condition	Interrupt latch	Vector table address	Priority
Internal / External	(Reset)		Non-Maskable	—	FFFFC$_H$	High 0
Internal	INTSW	(Software interrupt)	Pseudo non-maskable	—	FFFF8$_H$	1
Internal	INTWDT	(Watchdog timer interrupt)		IL$_2$	FFFF4$_H$	2
External	INT0	(External interrupt 0)	IMF = 1, INT0EN = 1	IL$_3$	FFFF0$_H$	3
Internal	INTTC1	(16-bit TC1 interrupt)	IMF · EF$_4$ = 1	IL$_4$	FFFEC$_H$	4
External	INT1	(External interrupt 1)	IMF · EF$_5$ = 1	IL$_5$	FFFE8$_H$	5
Internal	INTTBT	(Time base timer interrupt)	IMF · EF$_6$ = 1	IL$_6$	FFFE4$_H$	6
External	INT2	(External interrupt 2)	IMF · EF$_7$ = 1	IL$_7$	FFFE0$_H$	7
Internal	INTTC3	(8-bit TC3 interrupt)	IMF · EF$_8$ = 1	IL$_8$	FFFDC$_H$	8
Internal	INTTSBI	(SBI interrupt)	IMF · EF$_9$ = 1	IL$_9$	FFFD8$_H$	9
Internal	INTTC4	(8-bit TC4 interrupt)	IMF · EF$_{10}$ = 1	IL$_{10}$	FFFD4$_H$	10
External	INT3	(External interrupt 3)	IMF · EF$_{11}$ = 1	IL$_{11}$	FFFD0$_H$	11
External	INT4	(External interrupt 4)	IMF · EF$_{12}$ = 1	IL$_{12}$	FFFCC$_H$	12
Internal	INTTC5	(8-bit TC5 interrupt)	IMF · EF$_{13}$ = 1	IL$_{13}$	FFFC8$_H$	13
Internal	INTTC2	(16-bit TC2 interrupt)	IMF · EF$_{14}$ = 1	IL$_{14}$	FFFC4$_H$	14
External	INT5	(External interrupt 5)	IMF · EF$_{15}$ = 1	IL$_{15}$	FFFC0$_H$	15
Internal	INTRX	(UART receive interrupt)	IMF · EF$_{16}$ = 1	IL$_{16}$	FFFBC$_H$	16
Internal	INTTX	(UART transmit interrupt)	IMF · EF$_{17}$ = 1	IL$_{17}$	FFFB8$_H$	17
Internal	INTTC6	(8-bit TC6 interrupt)	IMF · EF$_{18}$ = 1	IL$_{18}$	FFFB4$_H$	18
		reserved	IMF · EF$_{19}$ = 1	IL$_{19}$	FFFB0$_H$	19
		reserved	IMF · EF$_{20}$ = 1	IL$_{20}$	FFFAC$_H$	20
		reserved	IMF · EF$_{21}$ = 1	IL$_{21}$	FFFA8$_H$	21
		reserved	IMF · EF$_{22}$ = 1	IL$_{22}$	FFFA4$_H$	22
		reserved	IMF · EF$_{23}$ = 1	IL$_{23}$	FFFA0$_H$	23
		reserved	IMF · EF$_{24}$ = 1	IL$_{24}$	FFF9C$_H$	24
		reserved	IMF · EF$_{25}$ = 1	IL$_{25}$	FFF98$_H$	25
		reserved	IMF · EF$_{26}$ = 1	IL$_{26}$	FFF94$_H$	26
		reserved	IMF · EF$_{27}$ = 1	IL$_{27}$	FFF90$_H$	27
		reserved	IMF · EF$_{28}$ = 1	IL$_{28}$	FFF8C$_H$	28
		reserved	IMF · EF$_{29}$ = 1	IL$_{29}$	FFF88$_H$	29
		reserved	IMF · EF$_{30}$ = 1	IL$_{30}$	FFF84$_H$	30
		reserved	IMF · EF$_{31}$ = 1	IL$_{31}$	FFF80$_H$	Low 31

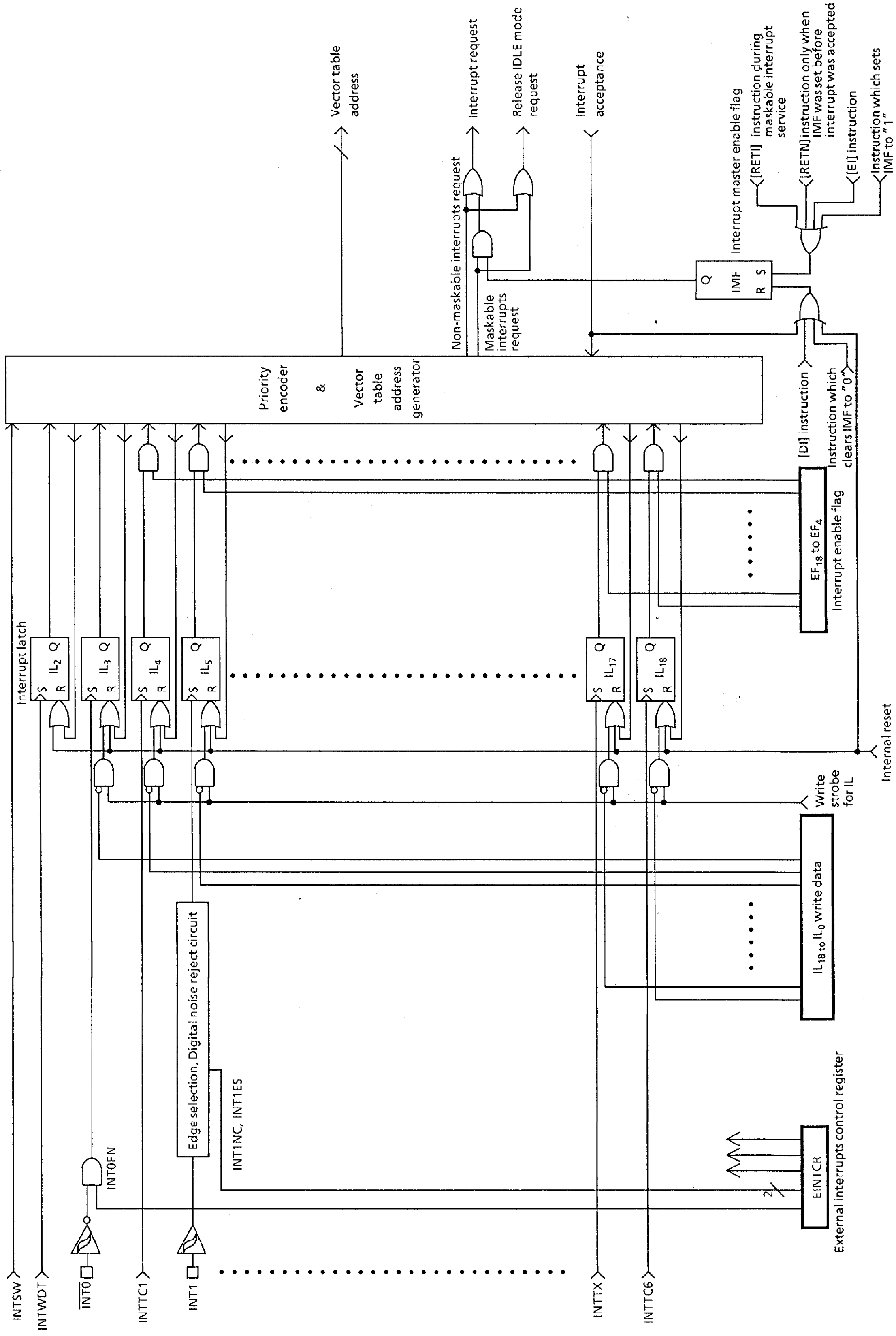

INTSW
INTWDT
INT0
INT0EN
INTTC1
INT1
Edge selection, Digital noise reject circuit
INT1NC, INT1ES
INTTX
INTTC6
EINTCR
External interrupts control register
2
Interrupt latch
S
R
Q
IL2
IL3
IL4
IL5
IL17
IL18
IL18 to IL0 write data
Write strobe for IL
Internal reset
EF18 to EF4
Interrupt enable flag
[DI] instruction
Instruction which clears IMF to "0"
Priority encoder & Vector table address generator
Vector table address
Non-maskable interrupts request
Maskable interrupts request
Interrupt request
Release IDLE mode request
Interrupt acceptance
IMF
Interrupt master enable flag
[RETI] instruction during maskable interrupt service
[RETN] instruction only when IMF was set before interrupt was accepted
[EI] instruction
Instruction which sets IMF to "1"

Figure 1-26. Interrupt Controller Block Diagram

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

The interrupt latch is set to "1" when an interrupt request is generated and requests the CPU to accept the interrupt. The acceptance of maskable interrupts can be selectively enabled and disabled by program using the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). When two or more interrupts are generated simultaneously, the interrupt is accepted in the highest priority order as determined by the hardware. Figure 1-26 shows the interrupt controller.

(1) **Interrupt Latches** (IL_{31} to IL_2)

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

The interrupt latches are assigned to addresses $0003C_H$, $0003D_H$, $0002E_H$ and $0002F_H$ in the SFR. Except for IL_2, each latch can be cleared to "0" individually by an instruction ; *however, the read-modify-write instruction such as bit manipulation or operation instructions cannot be used*. Thus, interrupt requests can be canceled and initialized by the program. Note that request the interrupt latches cannot be set to "1" by an instruction.

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

```
Example 1 :  Clears interrupt latches
                 LDW      (ILL), 1110100000111111B    ;  IL12, IL10 to IL6←0
                 LD       (ILE), 11111000B            ;  IL18 to IL16←0

Example 2 :  Reads interrupt latches
                 LD       WA, (ILL)                   ;  W←ILH, A←ILL
                 LD       BC, (ILE)                   ;  B←ILD, C←ILE

Example 3 :  Tests an interrupt latch
                 TEST     (ILL). 7                    ;  if IL7 = 1 then jump
                 JR       F, SSET
```

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

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

The EIR consists of an interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are assigned to addresses $0003A_H$, $0003B_H$, $0002C_H$ and $0002D_H$ in the SFR, and can be read and written by an instruction (including read-modify-write instruction such as bit manipulation instructions).

① **Interrupt Master enable Flag** (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all maskable interrupts. Clearing this flag to "0" disables the acceptance of all maskable interrupts. Setting to "1" enables the acceptance of interrupts.

When an interrupt is accepted, this flag is cleared to "0" to temporarily disable the acceptance of other maskable interrupts. After execution of the interrupt service program, this flag is set to "1" by the maskable interrupt return instruction [RETI] to again enable the acceptance of interrupts. If an interrupt request has already been occurred, interrupt service starts immediately after execution of the [RETI] instruction.

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

The IMF is assigned to bit 0 at address $0003A_H$ in the SFR, and can be read and written by an instruction. The IMF is normally set and cleared by the [EI] and [DI] instructions, and the IMF is initialized to "0" during reset.

② **Individual interrupt Enable Flags** (EF_{31} to EF_4)

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

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

```
LD      (EIRE), 00000001B           ; EF16←1
LDW     (EIRL), 1110100010100001B   ; EF15 to EF13, EF11, EF7, EF5, IMF←1
```

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

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

Interrupt Latches (IL)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IL_{31}	IL_{30}	IL_{29}	IL_{28}	IL_{27}	IL_{26}	IL_{25}	IL_{24}	IL_{23}	IL_{22}	IL_{21}	IL_{20}	IL_{19}	IL_{18}	IL_{17}	IL_{16}
IL_D ($0002F_H$)								IL_E ($0002E_H$)							
IL_{15}	IL_{14}	IL_{13}	IL_{12}	IL_{11}	IL_{10}	IL_9	IL_8	IL_7	IL_6	IL_5	IL_4	IL_3	IL_2		
IL_H ($0003D_H$)								IL_L ($0003C_H$)							

(Initial value : 00000000 00000000 00000000 000000**)

Interrupt Enable Registers (EIR)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EF_{31}	EF_{30}	EF_{29}	EF_{28}	EF_{27}	EF_{26}	EF_{25}	EF_{24}	EF_{23}	EF_{22}	EF_{21}	EF_{20}	EF_{19}	EF_{18}	EF_{17}	EF_{16}
EIR_D ($0002D_H$)								EIR_E ($0002C_H$)							
EF_{15}	EF_{14}	EF_{13}	EF_{12}	EF_{11}	EF_{10}	EF_9	EF_8	EF_7	EF_6	EF_5	EF_4				IMF
EIR_H ($0003B_H$)								EIR_L ($0003A_H$)							

(Initial value : 00000000 00000000 00000000 0000***0)

Note 1 : Do not clear IL with read-modify-write instructions such as bit operations.

Note 2 : Do not clear IL_2 to "0" by an instruction

Note 3 : Do not set IMF to "1" during non-maskable interrupt service program.

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

Note 5 : ** ; don't care*

Figure 1-27. Interrupt Latches (IL) and Interrupt Enable Registers (EIR)

1.6.1 Interrupt Sequence

An interrupt request is held until the interrupt is accepted or the interrupt latch is cleared to "0" by a reset or an instruction. Interrupt acceptance sequence requires 12 machine cycles (3 μs at fc = 16 MHz in the NORMAL mode) after the completion of the current instruction execution. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for pseudo non-maskable interrupts). Figure 1-28 shows the timing chart of interrupt acceptance processing.

(1) Interrupt acceptance

Interrupt acceptance processing is as follows.

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

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

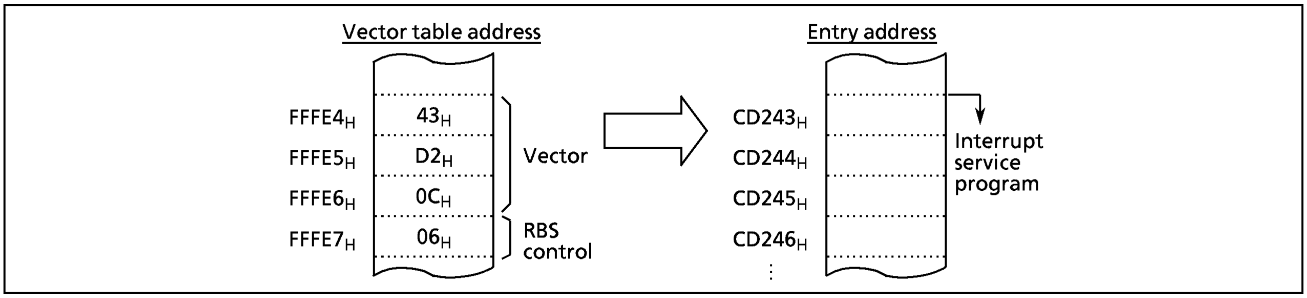
③ The contents of the program counter (return address) and the program status word (PSW) are saved (pushed) on the stack in sequence of PSWH, PSWL, PCE, PCH, PCL. The stack pointer (SP) is decremented five times.

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

⑤ The RBS control code is read from the vector table. The lower 4-bit of this code is added to the RBS.

⑥ The instruction stored at the entry address of the interrupt service program is executed.

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



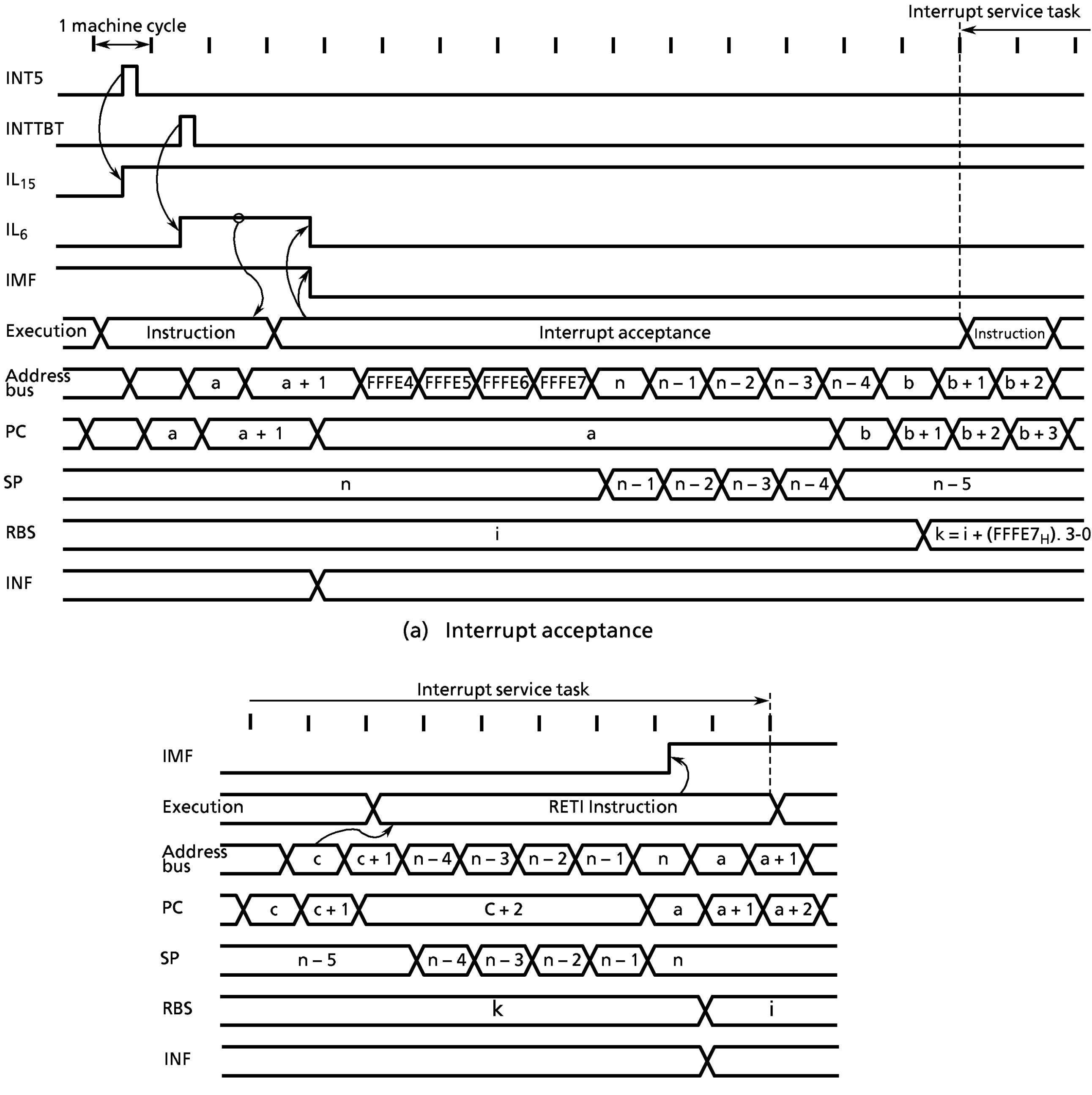
A maskable interrupt is not accepted until the IMF is set to "1" even if the maskable interrupt higher than the level of current servicing interrupt is occurred.
When nested interrupt service is necessary, the IMF is set to "1" in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags. However, an acceptance of external interrupt 0 cannot be disabled by the EF; therefore, if disable is necessary, either the external interrupt function of the $\overline{\text{INT0}}$ pin must be disabled with the INT0EN in the external interrupt control register (EINTCR) (the interrupt latch IL3 is not set at INT0EN = 0, therefore, the rising edge of $\overline{\text{INT0}}$ pin input can not be detected.) or an interrupt processing must be avoided by the program.

Example 1 : Disables an external interrupt 0 using the INT0EN

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

Example 2 : Disables the processing of external interrupt 0 under the software control (using bit 0 at address 000F0H as the interrupt processing disable switch)

```
PINT0 :  TEST  (000F0H). 0     ;   Return without interrupt processing if (000F0H) 0 = 1
         JRS   T, SINT0
         RETI
SINT0 :  Interrupt processing
         RETI
           ⋮
VINT0 :  DL    PINT0
```



(a) Interrupt acceptance

(b) Return from interrupt instruction

Note 1 : a ; return address, b ; entry address, c ; address which the RETI instruction is stored

Note 2 : The maximum response time from when an IL is set until an interrupt acceptance processing starts is 62/fcgck [s] or 62/fs [s] with interrupt enabled..

Figure 1-28. Timing Chart of Interrupt Acceptance and Interrupt Return Instruction

(2) Saving / Restoring general-purpose registers

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

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

① General-purpose register save / restore by automatic register bank changeover

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

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

```
Example :       Register bank changeover
    PINTxx  :     interrupt processing
                RETI
                 :
    VINTxx  :   DP          PINTxx
                DB          1                 ;  RBS←RBS + 1
```

② General-purpose register save / restore by register bank changeover

The general-purpose registers can be saved at high-speed by switching to a register bank that is not in use. Normally, the bank 0 is used for the main tank and the banks 1 to 15 are assigned to interrupt service tasks.

```
Example :    Register bank changeover
    PINTxx  :   LD          RBS, n
                 interrupt processing
                RETI                          ;  Restores bank and Returns
                 :
    VINTxx  :   DP          PINTxx            ;  Interrupt service routine entry address
                DB          0
```

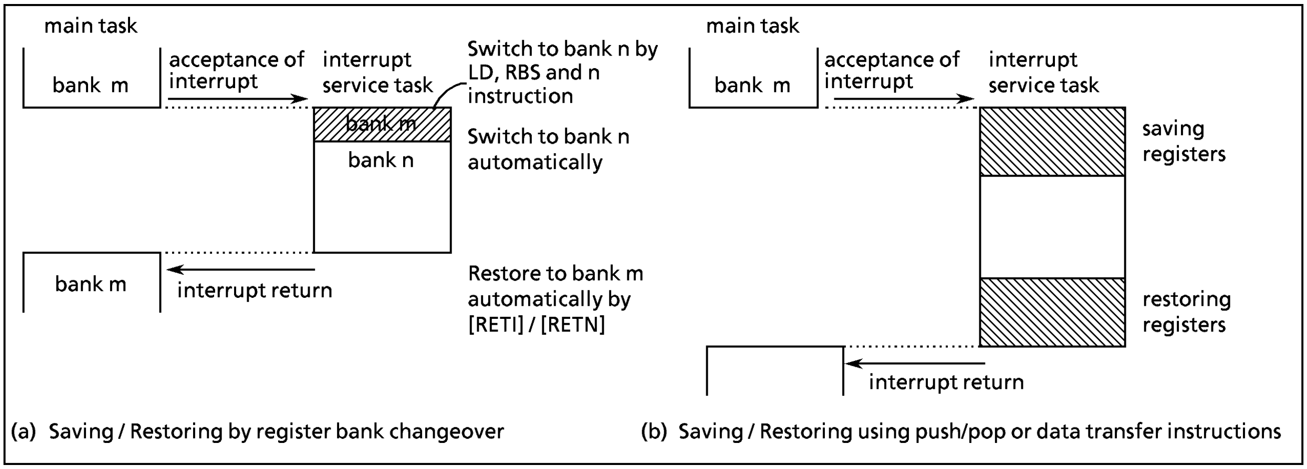

main task
bank m
acceptance of interrupt
interrupt service task
Switch to bank n by LD, RBS and n instruction
bank m
bank n
Switch to bank n automatically
bank m
interrupt return
Restore to bank m automatically by [RETI] / [RETN]
(a) Saving / Restoring by register bank changeover
main task
bank m
acceptance of interrupt
interrupt service task
saving registers
restoring registers
interrupt return
(b) Saving / Restoring using push/pop or data transfer instructions

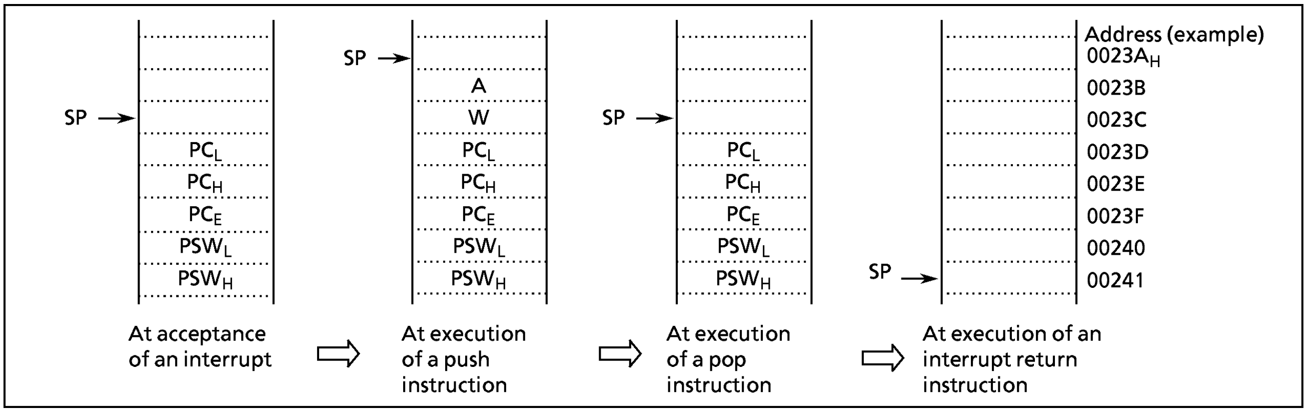
Figure 1-29. Saving / Restoring General-purpose Registers

③ General-purpose registers save / restore using **push and pop instructions**

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

Example : Register save / restore using push and pop instructions

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



SP
PCL
PCH
PCE
PSWL
PSWH
At acceptance of an interrupt
SP
A
W
PCL
PCH
PCE
PSWL
PSWH
At execution of a push instruction
SP
PCL
PCH
PCE
PSWL
PSWH
At execution of a pop instruction
Address (example)
0023AH
0023B
0023C
0023D
0023E
0023F
00240
00241
SP
At execution of an interrupt return instruction

④ General-purpose registers save / restore using **data transfer instructions**

Data transfer instruction can be used to save only a specific general-purpose register during processing of single interrupt.

Example : Saving / restoring a register using data transfer instructions

```
PINTxx :    LD        (GSAVA), A       ;  Save A register
            interrupt processing
            LD        A, (GSAVA)       ;  Restore A register
            RETI                       ;  Return
```

(3) Interrupt return

The interrupt return instructions [RETI] / [RETN] perform the following operations.

[RETI] Maskable interrupt return	[RETN] Non-maskable interrupt return
① The contents of the program counter and the program status word are restored from the stack.	① The contents of the program counter and program status word are restored from the stack.
② The stack pointer is incremented 5 times.	② The stack pointer is incremented 5 times.
③ The interrupt master enable flag is set to "1".	③ The interrupt master enable flag is set to "1" only when a non-maskable interrupt is accepted in interrupt enable status. However, the interrupt master enable flag remains at "0" when so clear by an interrupt service program.
④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.	④ The interrupt nesting counter is decremented, and the interrupt nesting flag is changed.

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

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

1.6.2 Software Interrupt (INTSW)

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

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

① Address error detection

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

② Debugging

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

1.6.3 External Interrupts

The 88C060 has six external interrupt inputs ($\overline{\text{INT0}}$, INT1, INT2, INT3, INT4, $\overline{\text{INT5}}$). Four of these are equipped with digital noise reject circuits (pulse inputs of less than a certain time are eliminated as noise).

Edge selection is also possible with INT1 to INT4. The $\overline{\text{INT0}}$ / P10 pin can be configured as either an external interrupt input pin or an input / output port, and is configured as an input port during reset.

Edge selection, noise reject control and INT0 / P10 pin function selection are performed by the external interrupt control register (EINTCR).

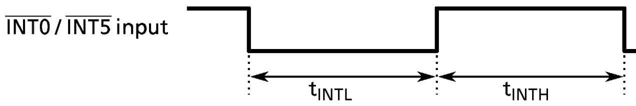
Table 1-6. External Interrupts

Source	Pin	Secondary function pin	Enable conditions	Edge	Digital noise reject
INT0	$\overline{\text{INT0}}$	P10	IMF = 1, INT0EN = 1	falling edge	— (hysteresis input)
INT1	INT1	P11	IMF · EF_5 = 1	falling edge or rising edge	Pulses of less than 15/fc or 63/fc [s] are eliminated as noise. Pulses of 48/fc or 192/fc [s] or more are considered to be signals.
INT2	INT2	P12/TC1	IMF · EF_7 = 1		pulses of less than 7/fc [s] are eliminated as noise. Pulses of 24/fc [s] or more are considered to be signals.
INT3	INT3	P34/TC3	IMF · EF_{11} = 1		
INT4	INT4	P35/TC4	IMF · EF_{12} = 1		
INT5	$\overline{\text{INT5}}$	P20/$\overline{\text{STOP}}$	IMF · EF_{15} = 1	rising edge	— (hysteresis input)

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

Note 2 : The noise reject function is also affected for timer / counter input (TC1 pin, TC3 pin).

Note 3 : The pulse width (both "H"and "L" level) for input to the $\overline{\text{INT0}}$ and $\overline{\text{INT5}}$ pins must be over 2 machine cycle.



t_{INTL}, $t_{INTH} \geqq$ 2tcyc *(Note : tcyc = 4 / fm [s])*
Note : tcyc = 4/fcgck [s] (in NORMAL1/2, IDLE1/2 mode)
4/fs (in SLOW, SLEEP mode)

Note 4 : If a noiseless signal is input to the external interrupt pin in the NORMAL 1/2 or IDLE 1/2 mode, the maximum time from the edge of input signal until the IL is set is as follows :
① INT1 pin 49/fc [s] (INT1NC = 1), 193/fc [s] (INT1NC = 0)
② INT2 pin 25/fc [s]

Note 5 : Even if the falling edge of $\overline{\text{INT0}}$ pin input is detected at INT0EN = 0, the interrupt latch IL_3 is not set.

EINTCR (00037_H)

7	6	5	4	3	2	1	0
INT1 NC	INT0 EN		INT4 ES	INT3 ES	INT2 ES	INT1 ES	

(Initial value : 00*0 000*)

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	P10/$\overline{\text{INT0}}$ pin configuration	0 : P10 input / output port 1 : $\overline{\text{INT0}}$ pin (Port P10 should be set to an input mode)	
INT4 ES INT3 ES INT2 ES INT1 ES	INT4 to INT1 edge select	0 : Rising edge 1 : Falling edge	

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

Figure 1-30. External Interrupt Control Register

1.7 Reset Circuit

The 88C060 has four types of reset generation procedures : an external reset input, an address trap reset output, a watchdog timer reset output and a system clock reset output. Table 1-7 shows on-chip hardware initialization by reset action.

The malfunction reset output 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 output level "L" at the maximum 24/fcgck[s] (1.92 μs at 12.5 MHz) when power is turned on.

Table 1-7. Initializing Internal Status by Reset Action

On-chip hardware	Initial value	On-chip hardware	Initial value
Program counter (PC)	(FFFFE$_H$ to FFFFC$_H$)	Prescaler and Divider of timing generator	0
Stack pointer (SP)	not initialized		
General-purpose registers (W, A, B, C, D, E, H, L)	not initialized		
Register bank selector (RBS)	0	Watchdog timer	Enable
Jump status flag (JF)	1		
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		
Interrupt nesting flag (INF)	0	RAM	not initialized

1.7.1 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/fcgck [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 FFFFC$_H$ to FFFFE$_H$.

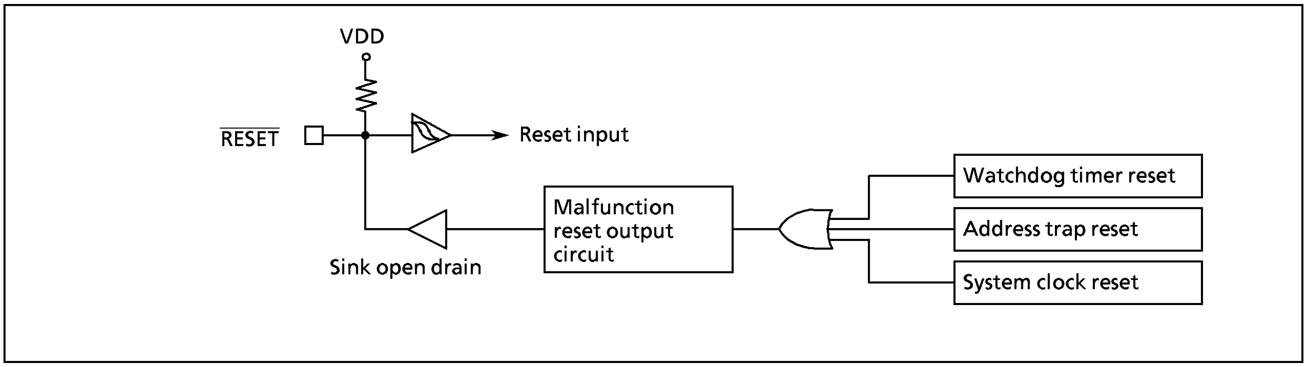
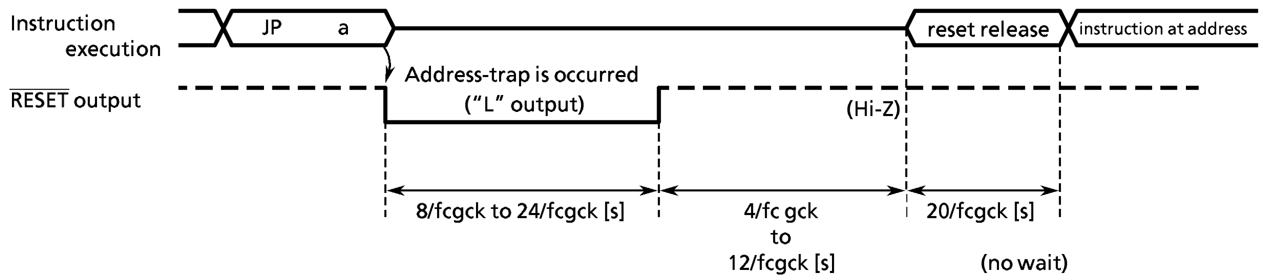


Figure 1-31. Reset Circuit

1.7.2 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 or the SFR area, address-trap-reset will be generated. Then, the $\overline{\text{RESET}}$ pin output will go low. The reset time is about 8/fcgck to 24/fcgck [s] (0.64 to 1.92 μs at 12.5 MHz).



Note 1 : Address "a" is in the SFR or on-chip RAM space.
Note 2 : During reset release, reset vector "r" is read out, and an instruction at address "r" is fetched and decoded.

Figure 1-32. Address-Trap-Reset

1.7.3 Watchdog Timer Reset

Refer to Section "2.4 Watchdog Timer".

1.7.4 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 XEN 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, the $\overline{\text{RESET}}$ pin output goes low from high-impedance. The reset time is about 8/fcgck to 24/fcgck [s] (0.64 to 1.92 μs at 12.5 MHz).

2. ON-CHIP PERIPHERAL FUNCTIONS

2.1 Special Function Registers (SFR)

The TLCS-870/X Series uses the memory mapped I/O system, and all peripheral control and data transfers are performed through the special function registers (SFR).

The SFR are mapped to addresses 00000$_H$ to 0003F$_H$.

Figure 2-1. shows the 88C060 SFRs.

<table>
<tr><th>Address</th><th>Read</th><th>Write</th><th>Address</th><th>Read</th><th>Write</th></tr>
<tr><td>00000$_H$</td><td colspan="2">P0 port</td><td>00020$_H$</td><td>–</td><td>SBICR1 (SBI control 1)</td></tr>
<tr><td>01</td><td colspan="2">P1 port</td><td>21</td><td colspan="2">SBIBUF (SBI data buffer)</td></tr>
<tr><td>02</td><td colspan="2">P2 port</td><td>22</td><td>–</td><td>I2CAR (I^{2}C bus address)</td></tr>
<tr><td>03</td><td colspan="2">P3 port</td><td>23</td><td>SBISR (SBI status)</td><td>SBICR2 (SBI control 2)</td></tr>
<tr><td>04</td><td colspan="2">P4 port</td><td>24</td><td>ADCDR2L (The lower 8 bits of A/D conv. result)</td><td>–</td></tr>
<tr><td>05</td><td colspan="2">reserved</td><td>25</td><td>ADCDR2H (The upper 2 bits of A/D conv. result)</td><td>–</td></tr>
<tr><td>06</td><td colspan="2">reserved</td><td>26</td><td colspan="2">reserved</td></tr>
<tr><td>07</td><td colspan="2">P7 port</td><td>27</td><td colspan="2">reserved</td></tr>
<tr><td>08</td><td>–</td><td>TREG6 (Timer register 6)</td><td>28</td><td colspan="2">reserved</td></tr>
<tr><td>09</td><td>–</td><td>TC6CR (TC 6 control)</td><td>29</td><td>RDBUF (UART receive data buffer)</td><td>TDBUF (UART transmit data buffer)</td></tr>
<tr><td>0A</td><td>–</td><td>P0CR (P0 I/O control)</td><td>2A</td><td>UARTSR (UART status)</td><td>UARTCR1 (UART control 1)</td></tr>
<tr><td>0B</td><td>–</td><td>P1CR (P1 I/O control)</td><td>2B</td><td>–</td><td>UARTCR2 (UART control 2)</td></tr>
<tr><td>0C</td><td colspan="2">reserved</td><td>2C</td><td colspan="2">EIR$_E$ (Extended interrupt enable register)</td></tr>
<tr><td>0D</td><td>–</td><td>P7CR (P7 I/O control)</td><td>2D</td><td colspan="2">EIR$_D$</td></tr>
<tr><td>0E</td><td colspan="2">ADCCR (A/D converter control)</td><td>2E</td><td colspan="2">IL$_E$ (Extended interrupt latch)</td></tr>
<tr><td>0F</td><td>ADCDR1 (The upper 8 bits of A/D conv. result)</td><td>–</td><td>2F</td><td colspan="2">IL$_D$</td></tr>
<tr><td>10</td><td>–</td><td>TREG1A$_L$ (Timer register 1A)</td><td>30</td><td colspan="2">CGCR (Clock gear control)</td></tr>
<tr><td>11</td><td>–</td><td>TREG1A$_H$</td><td>31</td><td colspan="2">EXPCR (External access control)</td></tr>
<tr><td>12</td><td colspan="2">TREG1B$_L$ (Timer register 1B)</td><td>32</td><td colspan="2">WAITCR (Wait control)</td></tr>
<tr><td>13</td><td colspan="2">TREG1B$_H$</td><td>33</td><td colspan="2">reserved</td></tr>
<tr><td>14</td><td>–</td><td>TC1CR (TC 1 control)</td><td>34</td><td>–</td><td>WDTCR1 (WDT control)</td></tr>
<tr><td>15</td><td>–</td><td>TC2CR (TC 2 control)</td><td>35</td><td>–</td><td>WDTCR2</td></tr>
<tr><td>16</td><td>–</td><td>TREG2$_L$ (Timer register 2)</td><td>36</td><td colspan="2">TBTCR (TBT / TG / DVO control)</td></tr>
<tr><td>17</td><td>–</td><td>TREG2$_H$</td><td>37</td><td colspan="2">EINTCR (External interrupt control)</td></tr>
<tr><td>18</td><td colspan="2">TREG3A (Timer register 3A)</td><td>38</td><td colspan="2">SYSCR1 (System control)</td></tr>
<tr><td>19</td><td>TREG3B (Timer register 3B)</td><td>–</td><td>39</td><td colspan="2">SYSCR2</td></tr>
<tr><td>1A</td><td>–</td><td>TC3CR(TC 3 control)</td><td>3A</td><td colspan="2">EIR$_L$ (Interrupt enable register)</td></tr>
<tr><td>1B</td><td>–</td><td>TREG4 (Timer register 4)</td><td>3B</td><td colspan="2">EIR$_H$</td></tr>
<tr><td>1C</td><td>–</td><td>TC4CR (TC 4 control)</td><td>3C</td><td colspan="2">IL$_L$ (Interrupt latch)</td></tr>
<tr><td>1D</td><td>–</td><td>TREG5 (Timer register 5)</td><td>3D</td><td colspan="2">IL$_H$</td></tr>
<tr><td>1E</td><td>–</td><td>TC5CR (TC 5 control)</td><td>3E</td><td colspan="2">PSW$_L$ (Program status word)</td></tr>
<tr><td>1F</td><td colspan="2">RCCR (Remote control receive control)</td><td>3F</td><td colspan="2">PSW$_H$</td></tr>
</table>

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

Note 2 : - ; Cannot be accessed.

Note 3 : When defining address 0003F$_H$ with assembler symbols, use GRBS.
Address 0003E$_H$ must be GPW / GFLAG.

Note 4 : Write-only registers and interrupt latches cannot use the read-modify-write instructions (bit manipulation instructions such as SET, CLR, etc. and logical operation instructions such as AND, OR, etc.).

Figure 2-1. Special Function Registers (SFR)

2.2 I/O Ports

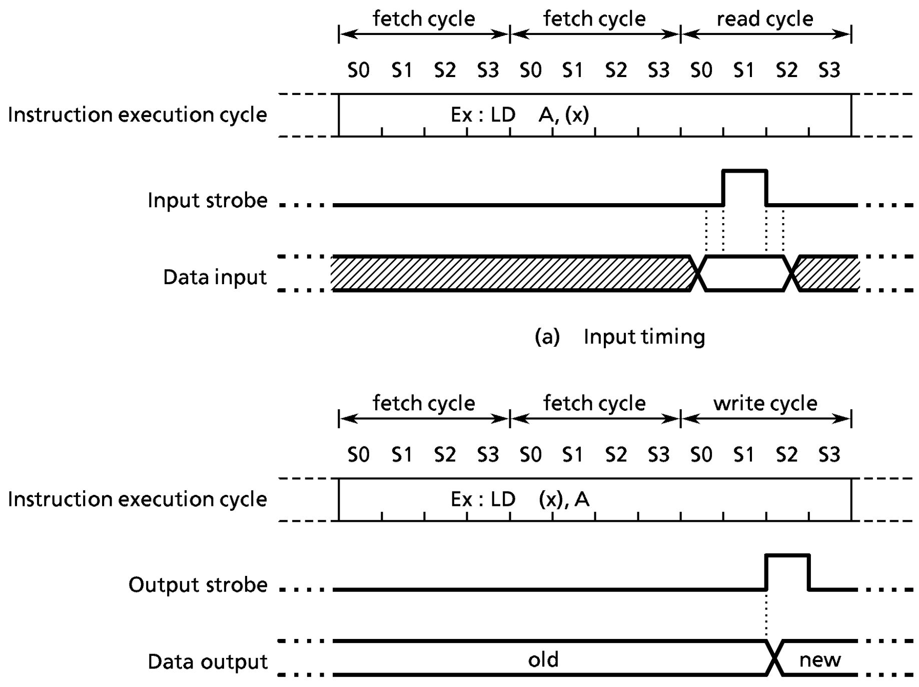
The 88C060 has 6 parallel input / output ports (42 pins) as follows.

	Primary Function	Secondary Function
Port P0	8-bit I/O port	external memory interface input / output
Port P1	8-bit I/O port	external interrupt input, timer counter input / output, and divider output
Port P2	4-bit I/O port	low frequency resonator connections, external interrupt input, and STOP mode release signal input
Port P3	8-bit I/O port	external interrupt input and timer counter input / output
Port P4	6-bit I/O port	serial interface input / output
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 either be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 2-2 shows input / output timing examples.

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

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



(a) Input timing

(b) Output timing

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

Figure 2-2. Input / Output Timing (Example)

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

(1) Instructions that read the output latch contents

① XCH r, (src)
② SET / CLR / CPL (src). b
③ SET / CLR / CPL (pp). g
④ LD (src). b, CF
⑤ LD (pp). b, CF
⑥ XCH CF, (src), b
⑦ ADD / ADDC / SUB / SUBB / AND / OR / XOR (src), n
⑧ (src) side of ADD / ADDC / SUB / SUBB / AND / OR / XOR (src), (HL)
⑨ MXOR (src), m

(2) Instructions that read the pin input data

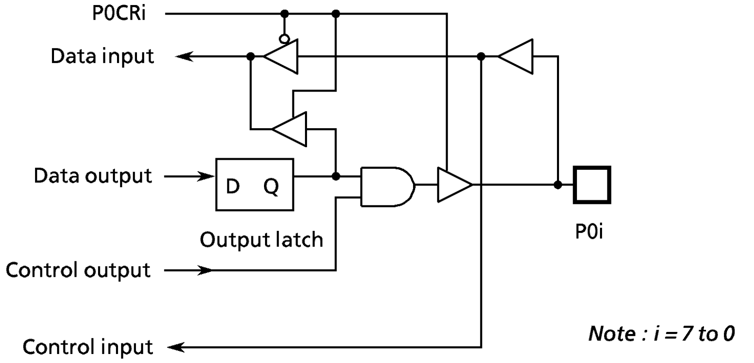
① Instructions other than the above (1)
② (HL) side of ADD / ADDC / SUB / SUBB / AND / OR / XOR (src), (HL)

2.2.1 Port P0 (P07 to P00)

Port P0 is an 8-bit input / output port which can be configured as an input or an output in one-bit unit. Input / output mode is specified by the port P0 input / output control register (P0CR). During reset, P0CR is initialized to "0", which configures port P0 as an input. The P0 output latches are also initialized to "0".

Port P0 is also used as the upper address buses A19 to A16, a bus release request input, a bus release acknowledge output, a wait request input, and a divided-by-4-clock output. When used as a bus release request input or a wait request input, the pin should be set to the input mode. When used as a bus release acknowledge output or a divided-by-4-clock output, the pin should be set to the output mode and beforehand the output latch should be set to "1". Pins P03 to P00 can be configured as either I/O ports or upper address outputs with EXP3 register. When used as the upper address output, the bit corresponding to P0CR must not be changed from the initialized value "0".

Note : Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode may be overwritten by executing bit manipulation instructions.



Note : i = 7 to 0

	7	6	5	4	3	2	1	0	
P0 (00000$_H$)	P07 $\overline{\text{CLK}}$	P06 $\overline{\text{WAIT}}$	P05 $\overline{\text{BUSAK}}$	P04 $\overline{\text{BUSRQ}}$	P03 A19	P02 A18	P01 A17	P00 A16	(Initial value : 0000 0000)

	7	6	5	4	3	2	1	0	
P0CR (0000A$_H$)									(Initial value : 0000 0000)

P0CR	I/O control for port P0 (specified for each bit)	0 : Input mode 1 : Output mode	Write only

Note : P0CR is a write-only register and must not be used with any of the read-modify-write instructions.

Figure 2-3. Port P0 and P0CR

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

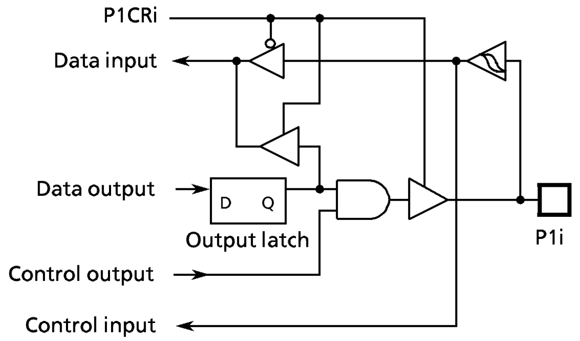
```
LD    (P0), 00001010B      ;   Sets initial data to P0 output latches
LD    (P0CR), 00001111B    ;   Sets the port P0 input / output mode
```

Note : Ports set to the input mode read the pin states. When input pin and output in exist port P0 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

2.2.2 Port P1 (P17 to P10)

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

Port P1 is also used as an external interrupt input, a timer counter input / output, and a divider output. When used as a secondary function pin, the input pin should be set to the input mode, and the output pin should be set to the output mode and beforehand the output latch should be set to "1". It is recommended that pins P11 and P12 should be used as external interrupt inputs, timer counter input, or input ports. The interrupt latch is set on the rising or falling edge of the output when used as output ports. Pin P10 can be configured as either an I/O port or an external interrupt input with INT0EN (bit 6 in EINTCR). During reset, pin P10 ($\overline{\text{INT0}}$) is configured as an input port P10.



Note : i = 7 to 0

	7	6	5	4	3	2	1	0	
P1 (00001$_H$)	P17	P16	P15 TC2	P14 $\overline{\text{PPG}}$	P13 $\overline{\text{DVO}}$	P12 INT2 TC1	P11 INT1	P10 $\overline{\text{INT0}}$	(Initial value : 0000 0000)

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

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

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

Figure 2-4. Port P1 and P1CR

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

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

Note : Input mode port reads the state of input pin. When input / output mode is used to mixed, the contents of output latch setting to the input mode port may be overwritten by executing bit manipulation instructions.

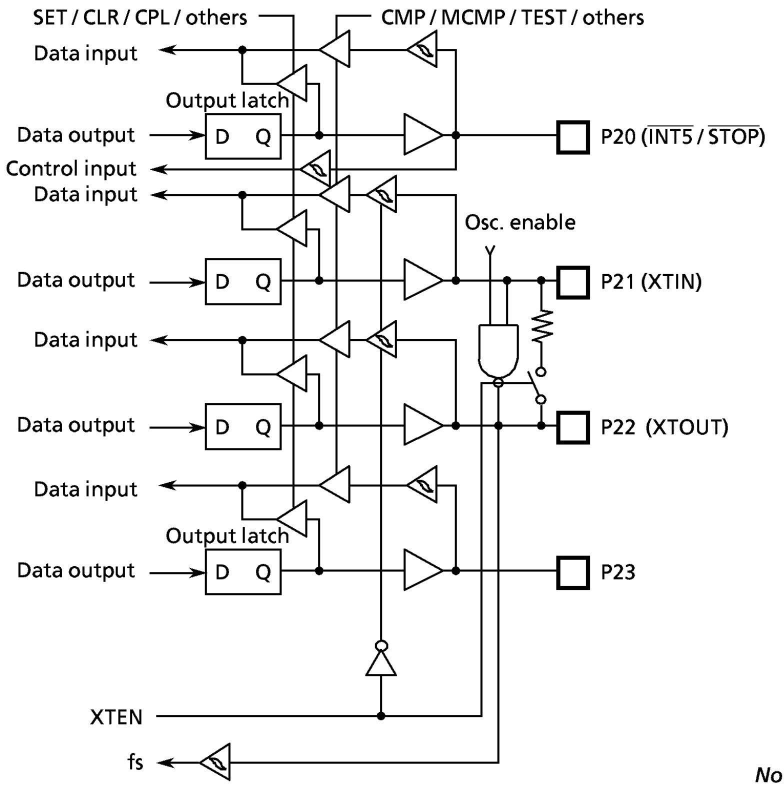
2.2.3 Port P2 (P23 to P20)

Port P2 is a 4-bit input / output port. It is also used as an external interrupt input, a STOP mode release signal input, and low-frequency Xtal connection pins. When use as an input port or a secondary function pin, the output latch should be set to "1". During reset, the output latches are initialized to "1".

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

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

When a read instruction is executed for port P2, bits 7 to 4 in P2 are read in as "1".



*Note 1 : * ; don't care*

Note 2 : XTEN is bit 6 in SYSCR2

	7	6	5	4	3	2	1	0
P2 (00002_H)					P23	P22 XTOUT	P21 XTIN	P20 $\overline{\text{INT5}}$ $\overline{\text{STOP}}$

(Initial value : **** 1111)

Figure 2-5. Port P2

2.2.4 Port P3 (P37 to P30)

Port P3 is an 8-bit input / output port. It is also used as an external interrupt input and a timer counter input / output. High current output is available, so LEDs can be driven directly. When used as an input port, the output latch should be set to "1". The output latches are initialized to "1" during reset.

Example 1 : Output the immediate data "5A" to the P3 port.

```
          LD     (P3), 5AH          ;   P3←5AH
```

Example 2 : Inverts the output of the upper 4 bits (P37 to P34) of the P3 port.

XOR (P3), 11110000B ; P37 to P34←$\overline{P37}$ to $\overline{P34}$

2.2.5 Port P4 (P45 to P40)

Port P4 is a 6-bit input / output port. It is also used as a serial interface input / output. When used as an input port or a serial interface input / output, the output latch should be set to "1". The output latches are initialized to "1" during reset. Bits 7 to 6 are read in as "1" when a read instruction is executed for the port P4.

Example : Clear the P43 port (outputs "L").

```
          CLR    (P4). 3 ;   P43←0
```

	7	6	5	4	3	2	1	0
P3 (00003$_H$)	P37	P36	P35 INT4 TC4	P34 INT3 TC3	P33 $\overline{PWM5}$ $\overline{PDO5}$	P32 $\overline{PWM4}$ $\overline{PDO4}$	P31	P30

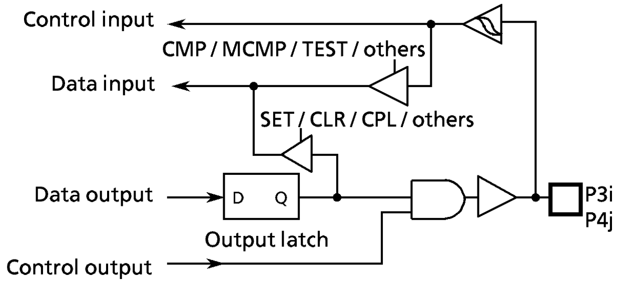
(Initial value : 1111 1111)

(a) Port P3

	7	6	5	4	3	2	1	0
P4 (00004$_H$)	"1"	"1"	P45 SDA SO	P44 SCL SI	P43 $\overline{SCK}$	P42 TxD	P41 RxD	P40

(Initial value : **11 1111)

(b) Port P4



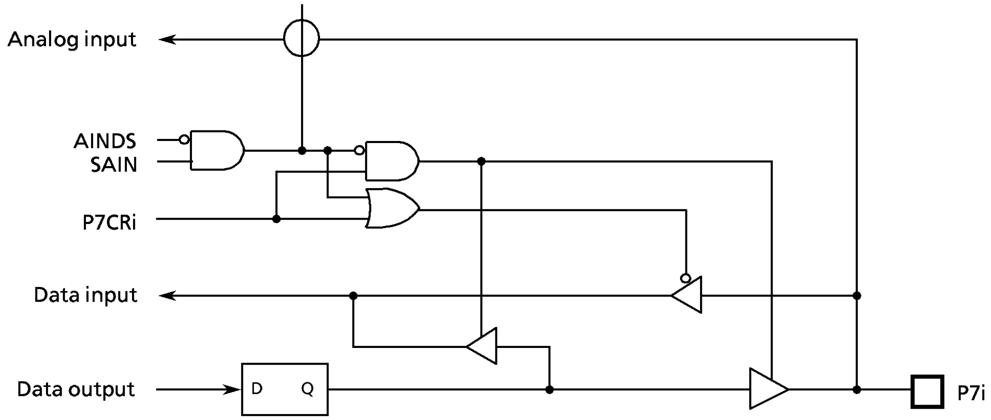
(c) Ports P3, P4

*Note : i = 7 to 0, j = 5 to 0, * ; don't care*

Figure 2-6. Ports P3, P4

2.2.6 Port P7 (P77 to P70)

Port P7 is an 8-bit input / output port which can be configured as an input or an output in one-bit unit. It is also used as an analog input for the A/D converter. Input / output is specified by the port P7 input / output control register (P7CR) and AINDS (bit 4 in ADCCR). During reset, P7CR is initialized to "0" and AINDS is initialized to "0". At the same time, P70 becomes an analog input port, and P77 to P71 become input ports. The P7 output latches are also initialized to "0". Pins not used for analog input can be used as I/O ports. However, an output instruction should not be executed on port P7 to keep a precision during A/D conversion. In addition, a variable signal should not be input to a port adjacent to the analog input during A/D conversion. When the A/D converter is in use (AINDS = 0), pins set for analog input are read in as "1", and pins not set for analog input are read in as "1" or "0", depending on the pin input level.



Note : i = 7 to 0

P7 (00007$_H$)

7	6	5	4	3	2	1	0
P77 AIN7	P76 AIN6	P75 AIN5	P74 AIN4	P73 AIN3	P72 AIN2	P71 AIN1	P70 AIN0

(Initial value : 0000 0000)

P7CR (0000C$_H$)

7	6	5	4	3	2	1	0

(Initial value : 0000 0000)

P7CR	I/O control for port P7 (specified for each bit)	0 : Input mode 1 : Output mode	Write only

Note 1 : P7CR is a write-only register and must not be used with any of read-modify-write instructions.
Note 2 : Pins to be used for analog input must be set to input mode.

Figure 2-7. Port 7 and P7CR

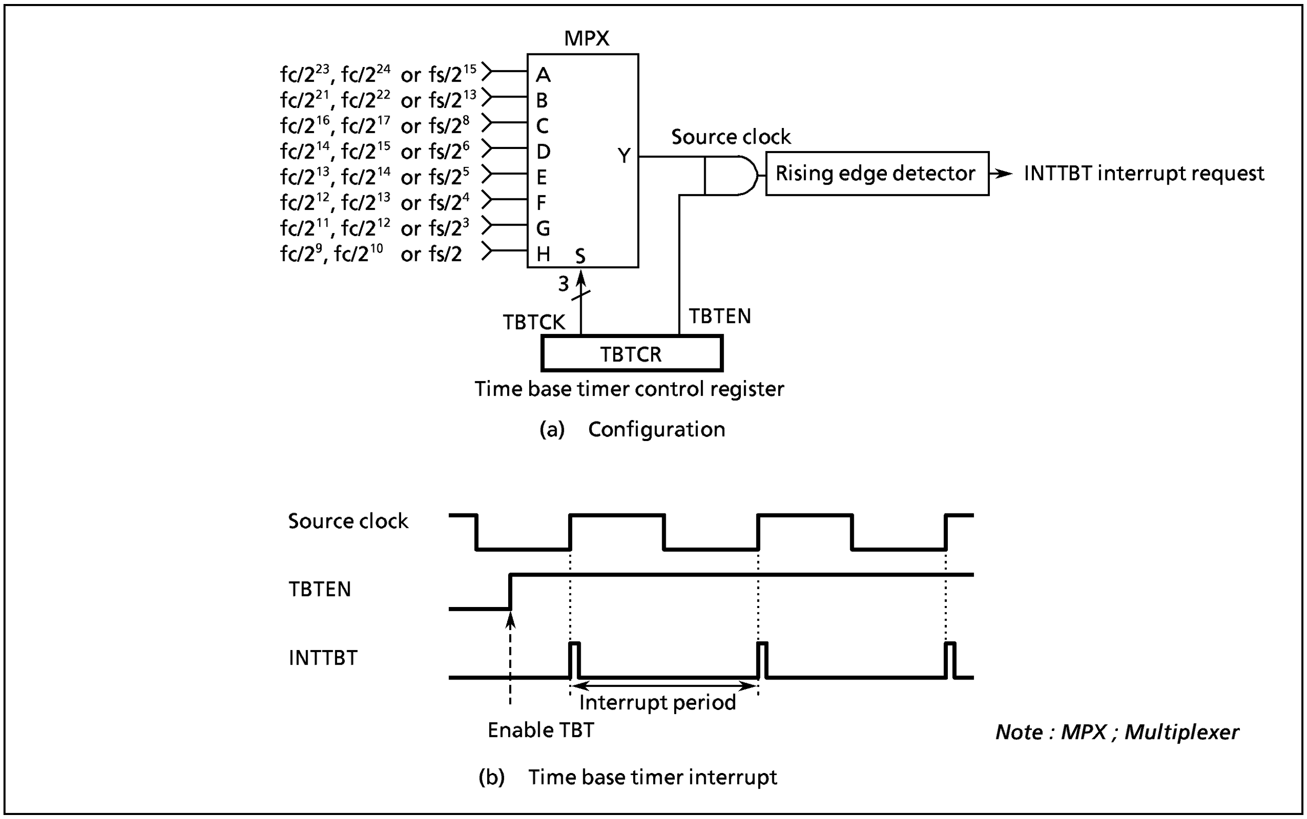
Note : Ports set to the input mode read the pin states. When input pin and output in exist in port P1 together, the contents of the output latch of ports set to the input mode may be rewritten by executing the bit manipulation instructions. Pins set to the output mode read a value of the output latch.

2.3 Time Base timer (TBT)

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

This interrupt is generated beginning with the first rising edge of the source clock (the timing generator's divider output selected by TBTCK) after the time-base timer is enabled. Note that since the divider cannot be cleared by a program, the first interrupt only may occur earlier than the set interrupt period (See Figure 2-8, (b)).

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


MPX
$fc/2^{23}$, $fc/2^{24}$ or $fs/2^{15}$
$fc/2^{21}$, $fc/2^{22}$ or $fs/2^{13}$
$fc/2^{16}$, $fc/2^{17}$ or $fs/2^{8}$
$fc/2^{14}$, $fc/2^{15}$ or $fs/2^{6}$
$fc/2^{13}$, $fc/2^{14}$ or $fs/2^{5}$
$fc/2^{12}$, $fc/2^{13}$ or $fs/2^{4}$
$fc/2^{11}$, $fc/2^{12}$ or $fs/2^{3}$
$fc/2^{9}$, $fc/2^{10}$ or $fs/2$
A
B
C
D
E
F
G
H
S
Y
3
Source clock
Rising edge detector
INTTBT interrupt request
TBTCK
TBTEN
TBTCR
Time base timer control register

(a) Configuration

Source clock
TBTEN
INTTBT
Interrupt period
Enable TBT

Note : MPX ; Multiplexer

(b) Time base timer interrupt

Figure 2-8. Time Base Timer

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

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

	7	6	5	4	3	2	1	0	
TBTCR (00036_H)	(DVOEN)	(DVOCK)		(DV7CK)	TBTEN	TBTCK			(Initial value : 0**0 0***)

TBTEN	Time base timer enable / disable	0 : Disable 1 : Enable						R/W
TBTCK	Time base timer interrupt frequency select [Hz]		NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		000	$fc/2^{23}$	$fc/2^{24}$	$fs/2^{15}$	$fs/2^{15}$	$fs/2^{15}$	
		001	$fc/2^{21}$	$fc/2^{22}$	$fs/2^{13}$	$fs/2^{13}$	$fs/2^{13}$	
		010	$fc/2^{16}$	$fc/2^{17}$	$fs/2^{8}$	$fs/2^{8}$	–	
		011	$fc/2^{14}$	$fc/2^{15}$	$fs/2^{6}$	$fs/2^{6}$	–	
		100	$fc/2^{13}$	$fc/2^{14}$	$fs/2^{5}$	$fs/2^{5}$	–	
		101	$fc/2^{12}$	$fc/2^{13}$	$fs/2^{4}$	$fs/2^{4}$	–	
		110	$fc/2^{11}$	$fc/2^{12}$	$fs/2^{3}$	$fs/2^{3}$	–	
		111	$fc/2^{9}$	$fc/2^{10}$	fs/2	fs/2	–	

*Note ; fc ; High-frequency clock [Hz] , fs ; Low-frequency clock [Hz] , * ; don’t care*

Figure 2-9. Time Base Timer Control Register

Table 2-1. Time Base Timer Interrupt Frequency (Example ; fc = 12.5 MHz, fs = 32.768 kHz)

TBTCK	Time base timer interrupt frequency [Hz]				
	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
000	1.49	0.75	1	1	1
001	5.96	2.98	4	4	4
010	190.73	95.37	128	128	–
011	762.94	381.47	512	512	–
100	1525.88	762.94	1024	1024	–
101	3051.76	1525.88	2048	2048	–
110	6103.52	3051.76	4096	4096	–
111	24414.06	12207.03	16384	16384	–

2.4 Watchdog Timer (WDT)

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

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

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

2.4.1 Watchdog Timer Configuration

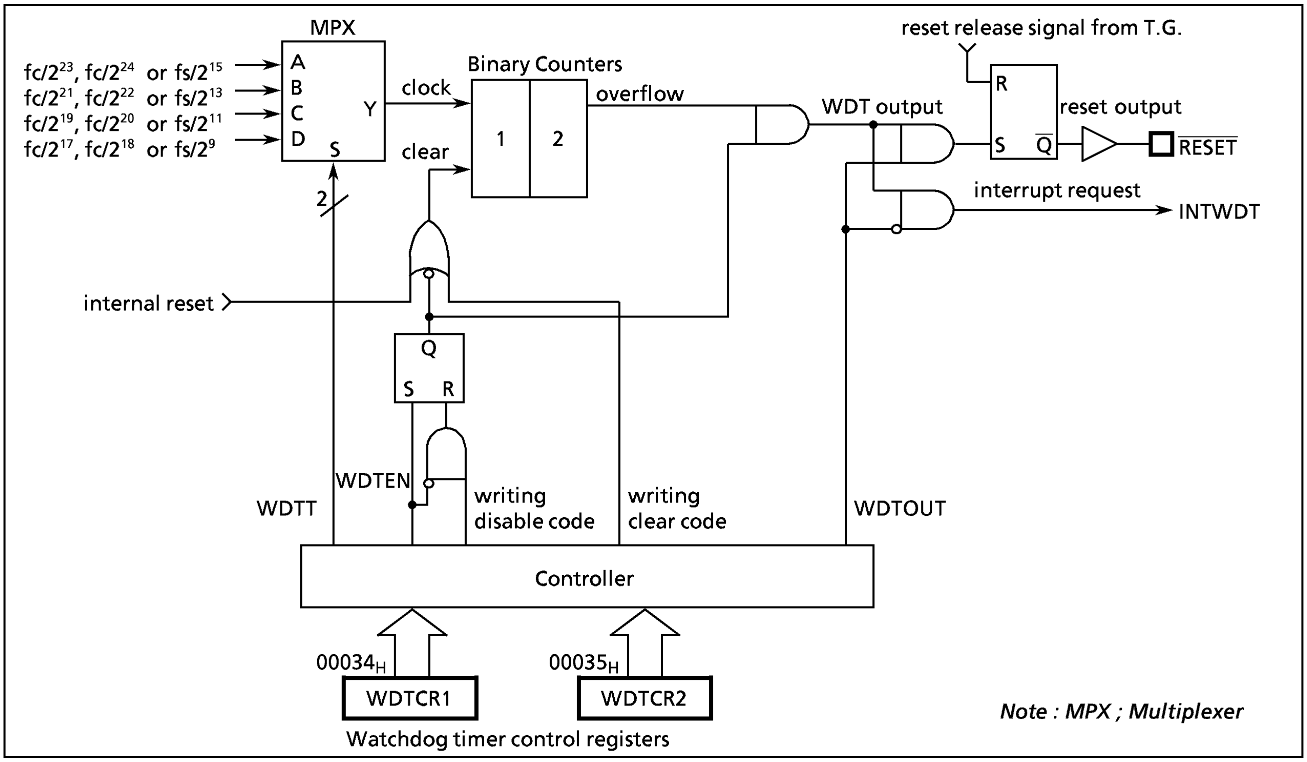


Figure 2-10. Watchdog Timer Configuration

2.4.2 Watchdog Timer Control

Figure 2-11 shows the watchdog timer control registers (WDTCR1, WDTCR2). The watchdog timer is automatically enabled after reset.

(1) Malfunction detection methods using the watchdog timer

The CPU malfunction is detected at 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, which drivers the $\overline{\text{RESET}}$ pin low to reset the internal hardware and the external circuit. 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.

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

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

Watchdog Timer Register 1

WDTCR1 (00034$_H$)

7	6	5	4	3	2	1	0
				WDTEN	WDTT		WDTOUT

(Initial value : **** 1001)

WDTEN	Watchdog timer enable / disable	0 : Disable (It is necessary to write the disable code to WDTCR2) 1 : Enable						Write only
WDTT	Watchdog timer detection time [s]		NORMAL1/2 mode				SLOW mode	
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		00	2^{25}/fc	2^{26}/fc	2^{17}/fs	2^{17}/fs	2^{17}/fs	
		01	2^{23}/fc	2^{24}/fc	2^{15}/fs	2^{15}/fs	2^{15}/fs	
		10	2^{21}/fc	2^{22}/fc	2^{13}/fs	2^{13}/fs	2^{13}/fs	
		11	2^{19}/fc	2^{20}/fc	2^{11}/fs	2^{11}/fs	2^{11}/fs	
WDTOUT	Watchdog timer output select	0 : Interrupt request 1 : Reset output						

Note 1 : WDTOUT cannot be set to "1" by program after clearing WDTOUT to "0".
*Note 2 : fc ; High-frequency clock [Hz], fs ; Low-frequency clock [Hz], *; don't care*
Note 3 : WDTCR1 is a write-only register and must not be used with any of read-modify-write instructions.

Watchdog Timer Register 2

WDTCR2 (00035$_H$)

7	6	5	4	3	2	1	0

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

WDTCR2	Watchdog timer control code write register	4E$_H$: Watchdog timer binary counter clear (clear code) B1$_H$: Watchdog timer disable (disable code) Others : Invalid	Write only

Note 1 : The disable code is invalid unless written when WDTEN = 0.
*Note 2 : * ; don't care*
Note 3 : The binary counter of the watchdog timer must not be cleared by the interrupt task.
Note 4 : Since WDTCR2 is a write-only resister, read-modify-write instructions (e.g., bit manipulating instructions such as SET or CLR and arithmertic instructions such as AND or OR) cannot be used for read/write to this register.

Figure 2-11. Watchdog Timer Control Registers

(2) Watchdog timer enable

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

Example : Enables watchdog timer

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

(3) Watchdog timer disable

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

Example : Disables watchdog timer

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

Table 2-2. Watchdog Timer Detection Time (Example : fc = 12.5 MHz, fs = 32.768 kHz)

WDTT	Watchdog timer detection time [s]				
	NORMAL1/2 mode				SLOW mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	2.684	5.369	4	4	4
01	671.089 m	1.342	1	1	1
10	167.772 m	335.544 m	250 m	250 m	250 m
11	41.943 m	83.886 m	62.5 m	62.5 m	62.5 m

2.4.3 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, 0023FH               ;   Sets the stack pointer
LD    (WDTCR1), 00001000B      ;   WDTOUT←0
```

2.4.4 Watchdog Timer Reset

If the watchdog timer output becomes active, a reset is generated, which drivers the $\overline{\text{RESET}}$ pin (sink open drain input / output with pull-up) low to reset the internal hardware and external circuits. The reset output time is about 8/fcgck to 24/fcgck [s] (0.64 to 1.92 μs at fc = 12.5 MHz, fcgck = fc).

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/fcgck to 24/fcgck [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.

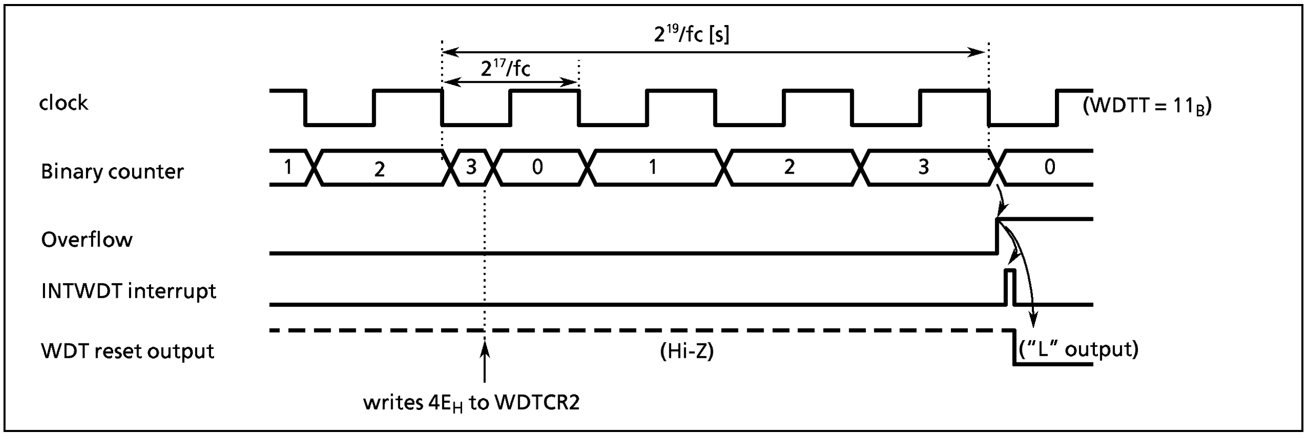


Figure 2-12. Watchdog Timer Interrupt / Reset

2.5 Divider Output ($\overline{\text{DVO}}$)

Approximately 50 % duty pulse can be output using the divider output circuit, which is useful for piezoelectric buzzer drive. Divider output is from pin P13 ($\overline{\text{DVO}}$). The P13 output latch should be set to "1" and then the P13 should be configured as an output mode.
Divider output circuit is controlled by the control register (TBTCR) shown in Figure 2-13.

	7	6	5	4	3	2	1	0	
TBTCR (00036$_H$)	DVOEN	DVOCK		(DV7CK)	(TBTEN)	(TBTCK)			(Initial value : 0**0 0***)

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	
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		00	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	
		01	$fc/2^{12}$	$fc/2^{13}$	$fs/2^4$	$fs/2^4$	$fs/2^4$	
		10	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	
		11	$fc/2^{10}$	$fc/2^{11}$	$fs/2^2$	$fs/2^2$	$fs/2^2$	

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

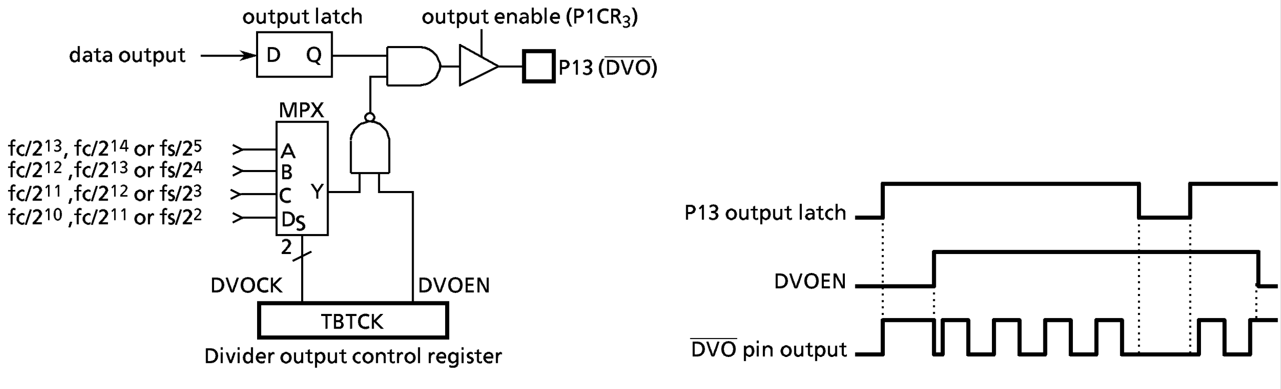
Figure 2-13. Divider Output Control Register

Example : 1 kHz pulse output (at fc = 16 MHz, DV1CK = 1)

```
SET     (P1). 3            ;  P13 output latch←1
LD (P1CR), 00001000B       ;  Configures P13 as an output mode
LD (TBTCR), 10000000B      ;  DVOEN←1, DVOCK←00
```

Table 2-3. Divider Output Frequency (Example : at fc = 12.5 MHz, fs = 32.768 kHz)

DVOCK	Divider output frequency [kHz]				
	NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode
	DV7CK = 0		DV7CK = 1		
	DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	
00	1.526 k	0.763 k	1.024	1.024	1.024
01	3.502	1.526	2.048	2.048	2.048
10	6.104	3.502	4.096	4.096	4.096
11	12.207	6.104	8.192	8.192	8.192


output latch
output enable (P1CR3)
data output
D Q
P13 ($\overline{\text{DVO}}$)
MPX
fc/2^13, fc/2^14 or fs/2^5
fc/2^12 ,fc/2^13 or fs/2^4
fc/2^11 ,fc/2^12 or fs/2^3
fc/2^10 ,fc/2^11 or fs/2^2
A
B
C
Y
Ds
2
DVOCK
DVOEN
TBTCK
Divider output control register
P13 output latch
DVOEN
$\overline{\text{DVO}}$ pin output

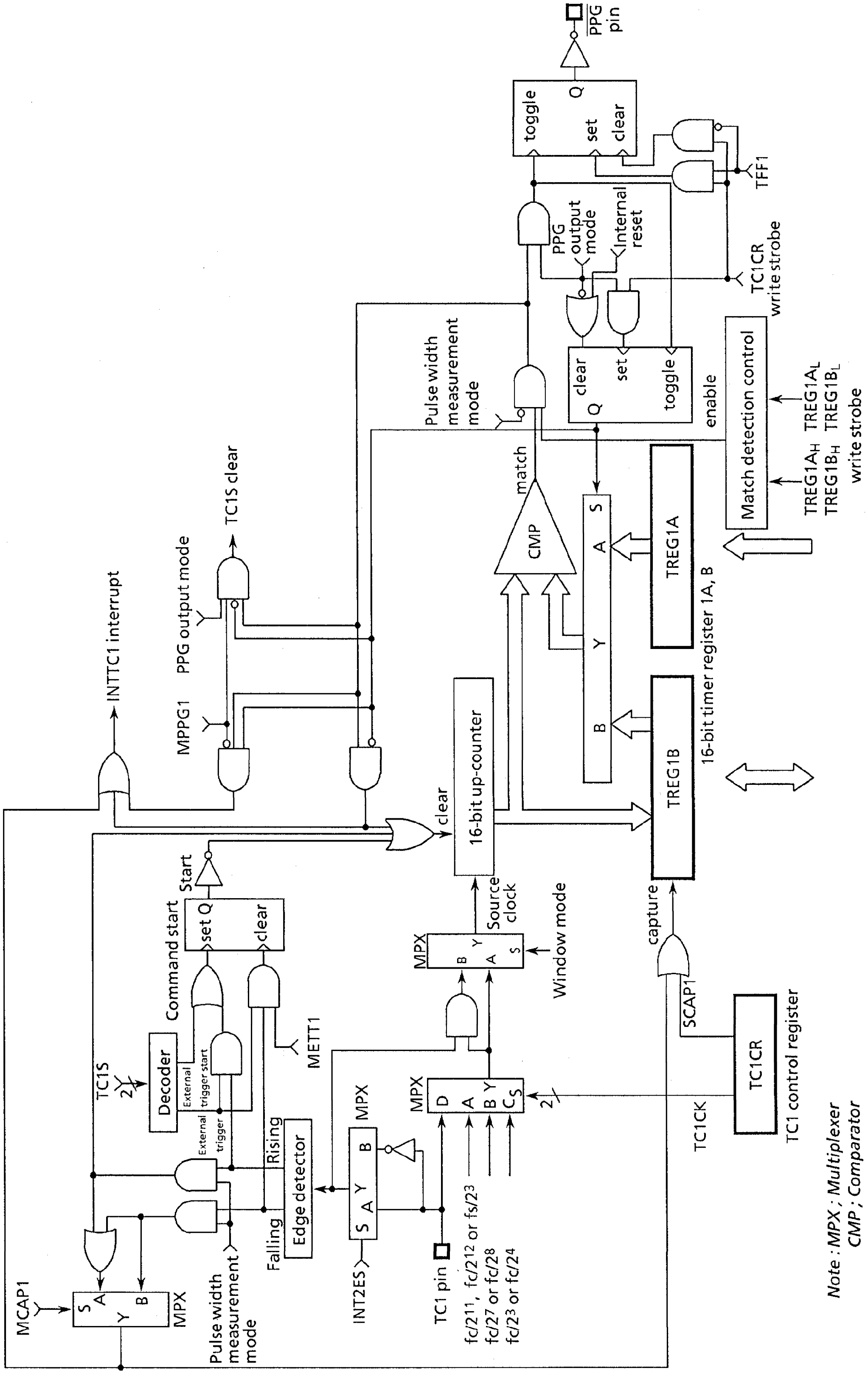
(a) Configuration (b) Timing Chart

Note : MPX ; Multiplexer

Figure 2-14. Divider Output

2.6 16-bit Timer / Counter 1 (TC1)

2.6.1 Configuration



Note : MPX ; Multiplexer
CMP ; Comparator

Figure 2-15. Timer / Counter 1 (TC1)

2.6.2 Control

The timer / counter 1 is controlled by a timer / counter 1 control register (TC1CR) and two 16-bit timer registers (TREG1A and TREG1B). Reset does not affect TREG1A and TREG1B.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TREG1A (00010, 00011$_H$)	TREG1A$_H$ (00011$_H$)								TREG1A$_L$ (00010$_H$)							
									Write only							
TREG1B (00012, 00013$_H$)	TREG1B$_H$ (00013$_H$)								TREG1B$_L$ (00012$_H$)							
									Read / Write (Writing is capable only when PPG output mode)							

	7	6	5	4	3	2	1	0	
TC1CR (00014$_H$)	TFF1	SCAP1 MCAP1 METT1 MPPG1	TC1S		TC1CK		TC1M		(Initial value : 0000 0000)

<table>
<tr><td>TC1M</td><td>TC1 operating mode select</td><td colspan="7">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="14">Write only</td></tr>
<tr><td rowspan="7">TC1CK</td><td rowspan="7">TC1 source clock select [Hz]</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00</td><td>$fc/2^{11}$</td><td>$fc/2^{12}$</td><td>$fs/2^3$</td><td>$fs/2^3$</td><td>$fs/2^3$</td></tr>
<tr><td>01</td><td>$fc/2^7$ (DV5)</td><td>$fc/2^8$ (DV5)</td><td>$fc/2^7$ (DV5)</td><td>$fc/2^8$ (DV5)</td><td>–</td></tr>
<tr><td>10</td><td>$fc/2^3$ to 2^6(DV1G)</td><td>$fc/2^4$ to 2^7(DV1G)</td><td>$fc/2^3$ to 2^6(DV1G)</td><td>$fc/2^4$ to 2^7(DV1G)</td><td>–</td></tr>
<tr><td>11</td><td colspan="5">External clock (TC1 pin input)</td></tr>
<tr><td>TC1S</td><td>TC1 start control</td><td colspan="7">00 : Stop &amp; counter clear
01 : Command start
10 : Reserved
11 : External trigger start</td></tr>
<tr><td>SCAP1</td><td>Software capture control</td><td colspan="7">0 : – 1 : Software capture trigger (Note 4)</td></tr>
<tr><td>MCAP16</td><td>Pulse width measurement mode control</td><td colspan="7">0 : Double edge capture 1 : Single edge capture</td></tr>
<tr><td>METT1</td><td>External trigger timer mode control</td><td colspan="7">0 : Trigger start 1 : Trigger start &amp; stop</td></tr>
<tr><td>MPPG1</td><td>PPG output control</td><td colspan="7">0 : Pulse 1 : Single</td></tr>
<tr><td>TFF1</td><td>Time F/F1 control for PPG output mode</td><td colspan="7">0 : Clear 1 : Set</td></tr>
</table>

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

Note 2 : Writing to the lower byte of the timer registers (TREG1A$_L$, TREG1B$_L$), the comparison is inhibited until the upper byte (TREG1A$_H$, TREG1B$_H$) is written. Only the lower byte of the timer registers can not be changed. After writing to the upper byte, any match during 1 machine cycle (instruction execution cycle) is ignored.

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

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

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

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

Note 7 : TC1R and TREG1A are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

Note 8 : Writing to the TREG1B is not possible unless TC1 is set to the PPG output mode.

Figure 2-16. Timer registers and TC1 Control Register

2.6.3 Function

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

(1)**Timer** mode

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

Table 2-4. Source Clock (internal clock) for Timer / Counter 1 (Example : at fc = 12.5 MHz, fs = 32.768 kHz)

TC1CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting	Resolution [μs]	Maximum time setting
00	163.84	10.8 s	327.68	21.5 s	244.14	16.0 s	244.14	16.0 s
01	10.24	0.64 s	20.48	1.28 s	10.24	0.64 s	20.48	1.28 s
10	0.64	41.92 ms	1.28	83.84 ms	0.64	41.92 m	1.28	83.84 ms

TC1CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum time setting
00	244.14	16.0
01	–	–
10	–	–

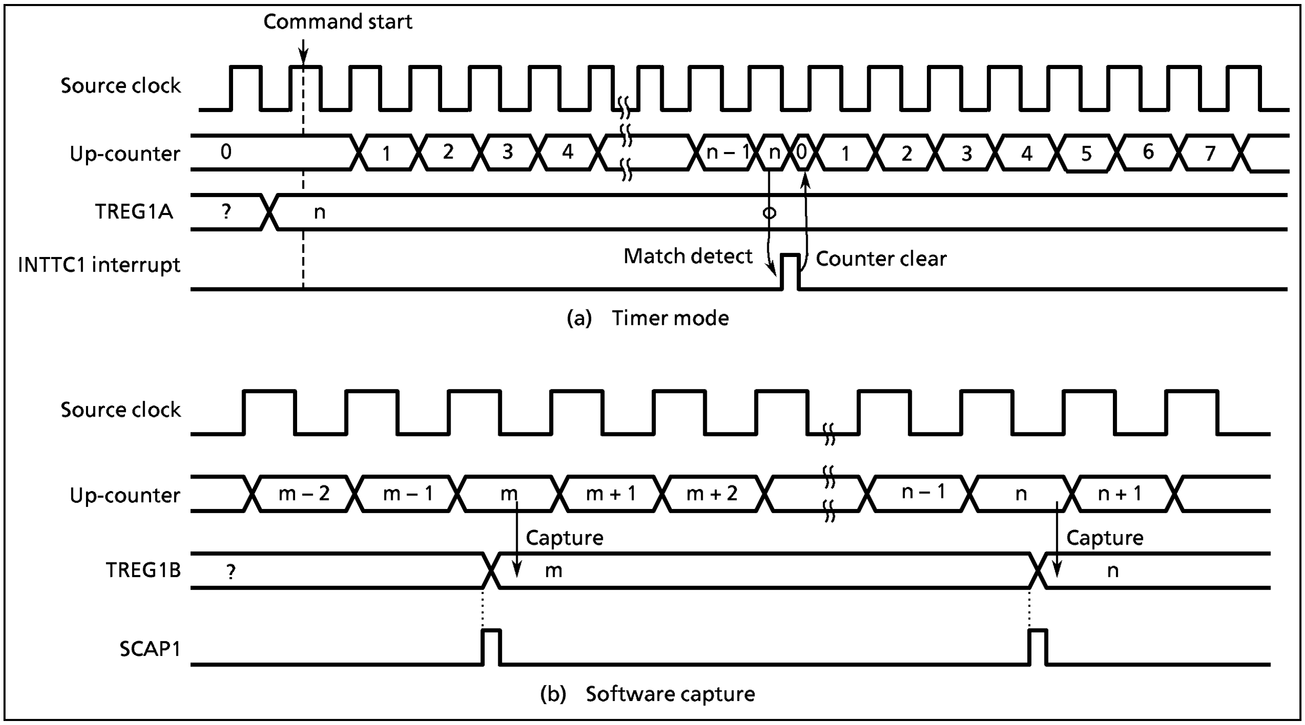
Example 1 : Sets the timer mode with source clock fs/2^3 [Hz] and generates an interrupt 1 later (at fs = 32.8 kHz)

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

Note : TC1CR is a wire-only register and must not be used with [SET (TC1CR). 4] instruction.

Example 2 : Software capture

```
LD (TC1CR), 01010000B   ;   SCAP1←1 (Capture)
LD WA, (TREG1B)         ;   Reads the capture value
```



(a) Timer mode

(b) Software capture

Figure 2-17. Timer Mode Timing Chart

(2) **External trigger timer** mode

In this mode, counting up is started by an external trigger. This trigger is the edge of the TC1 pin input. Either the rising or falling edge can be selected with IBT2ES in EINTCR. Edge selection is the same as for INT2 pin. Source clock is an internal clock selected with TC1CK. The contents of TREG1A is compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared to "0" and halted. The counter is restarted by the selected edge of the TC1 pin input.

When METT1 (bit 6 in TC1CR) 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 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 same noise rejection as the INT2 pin; therefore, pulses of 7/fc [s] or less are rejected as noise. A pulse width of 24/fc [s] or more is required for edge detection in NORMAL1, 2 or IDLE1, 2 mode. The noise rejection circuit is turned off in SLOW and SLEEP modes. But, a pulse width of one machine cycle or more is required.

Example 1 : Detects rising edge in TC1 pin input and generates an interrupt 100 μs later. (at fc = 12.5 MHz, DV1CK = 1)

```
LD (EINTCR), 00000000B   ;   INT2ES←0 (rising edge)
LDW  (TREG1A), 004EH     ;   100 μs ÷ 2^4/fc = 4E_H
SET   (EIRL). EF4        ;   INTTC1 interrupt enable
EI
LD (TC1CR),  00111000B   ;   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 = 12.5 MHz, DV1CK = 1)

```
LD (EINTCR), 00000100B   ;   INT2ES←1 ("L" level)
LDW  (TREG1A), 00C3H     ;   4 ms ÷ 2^8/fc = C3_H
SET   (EIRL). EF4        ;   INTTC1 interrupt enable
EI
LD (TC1CR), 01110100B    ;   TC1 external trigger start, METT1 = 1
```

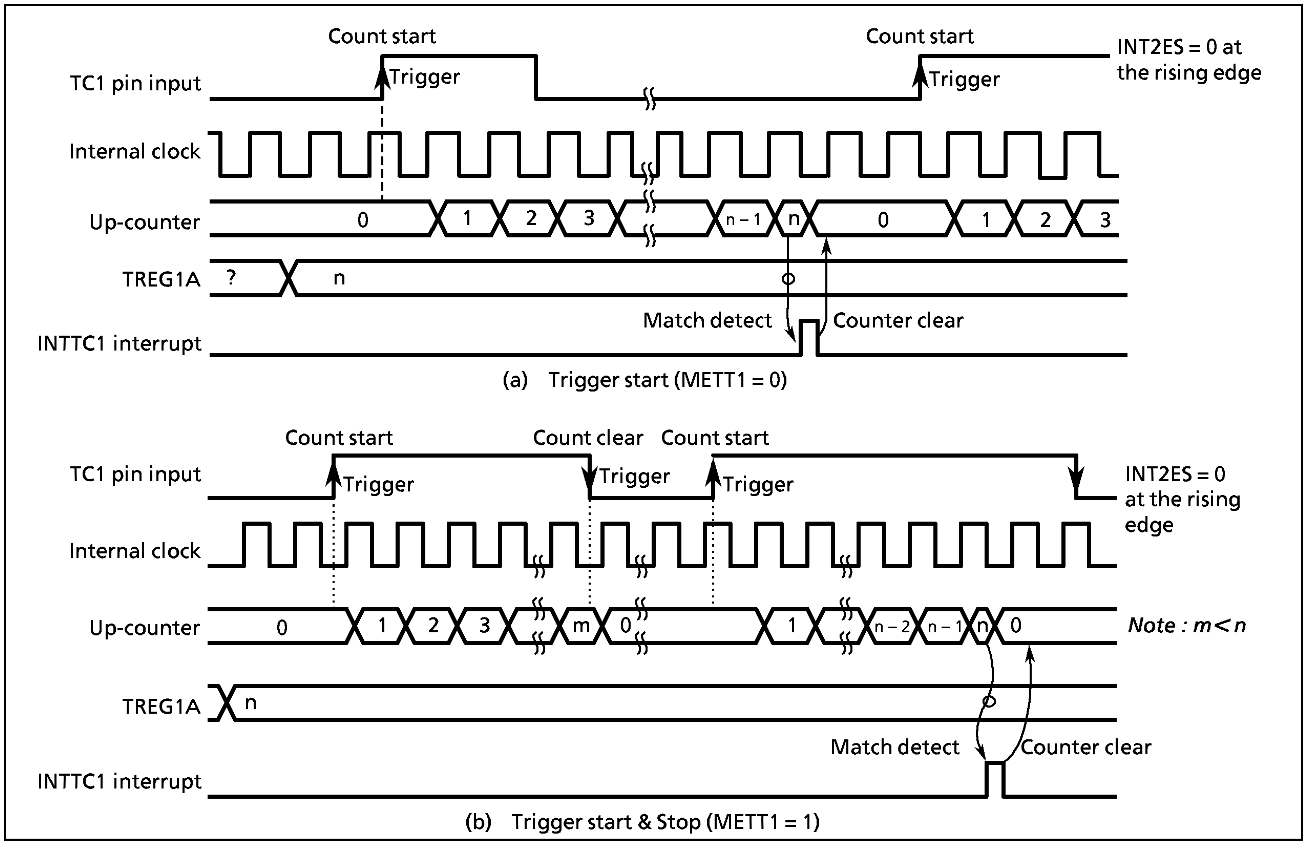


Figure 2-18. External Trigger Timer Mode Timing Chart

(3) **Event counter** mode

In this mode, events are counted at the edge of the TC1 pin input. Either the rising or falling edge can be selected with INT2ES in EINTCR. Edge selection is the same as for INT2 pin. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. This maximum applied frequency is shown in table 2-5.

Setting SCAP1 to "1" transfers the current contents of up-counter to TREG1B (software capture function). SCAP1 is automatically cleared after capturing.

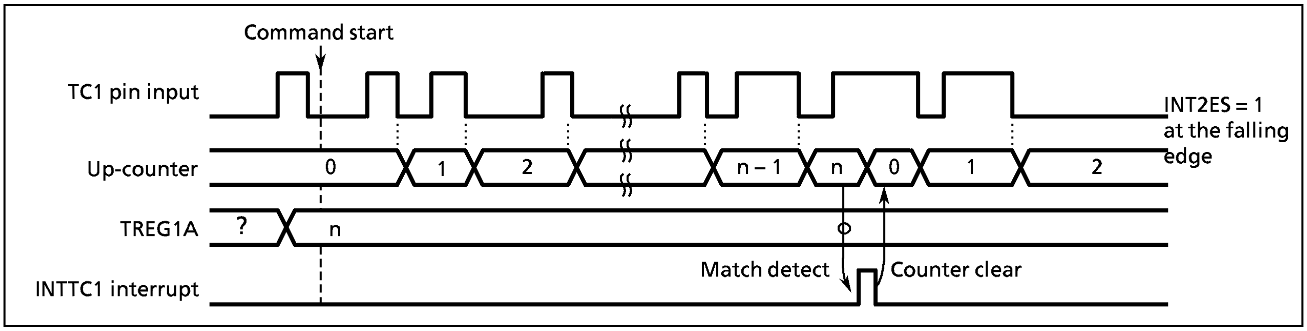


Figure 2-19. Event Counter Mode Timing Chart

Table 2-5. Timer / Counter 1 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode
$fcgck/2^4$	$fs/2^4$

(4) **Window** mode

Counting up is performed on the rising edge of the pulse that is the logical AND-ed product of the TC1 pin input (window pulse) and an internal clock. The contents of TREG1A are compared with the contents of up-counter. If a match is found, an INTTC1 interrupt is generated, and the counter is cleared. Positive or negative logic for the TC1 pin input can be selected. Edge selection is the same as for INT2 pin. Setting SCAP to "1" transfers the current contents of up-counter to TREG1B.

It is necessary that the maximum applied frequency be such that the counter value can be analyzed by the program. That is ; the frequency must be considerably slower than the selected internal clock.

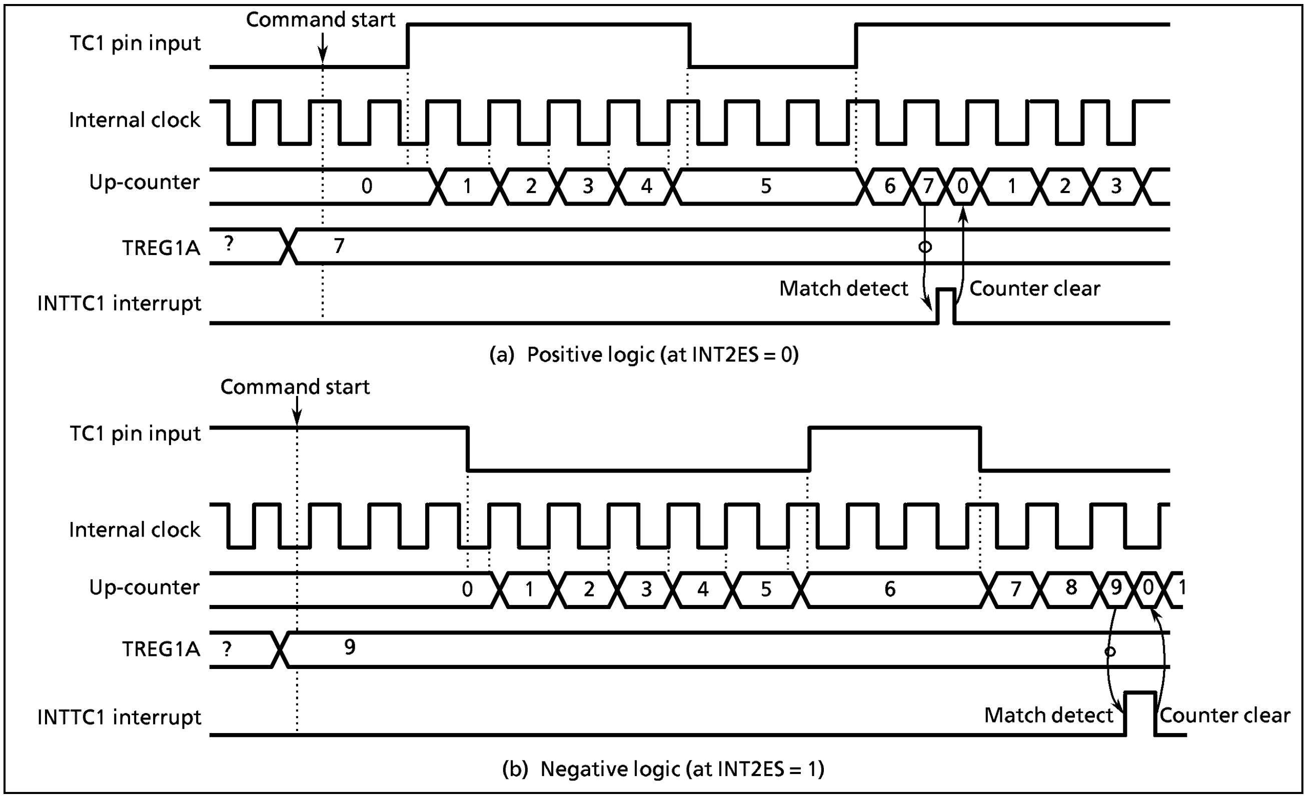


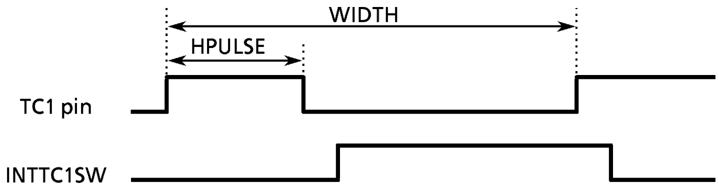
Figure 2-20. Window Mode Timing Chart

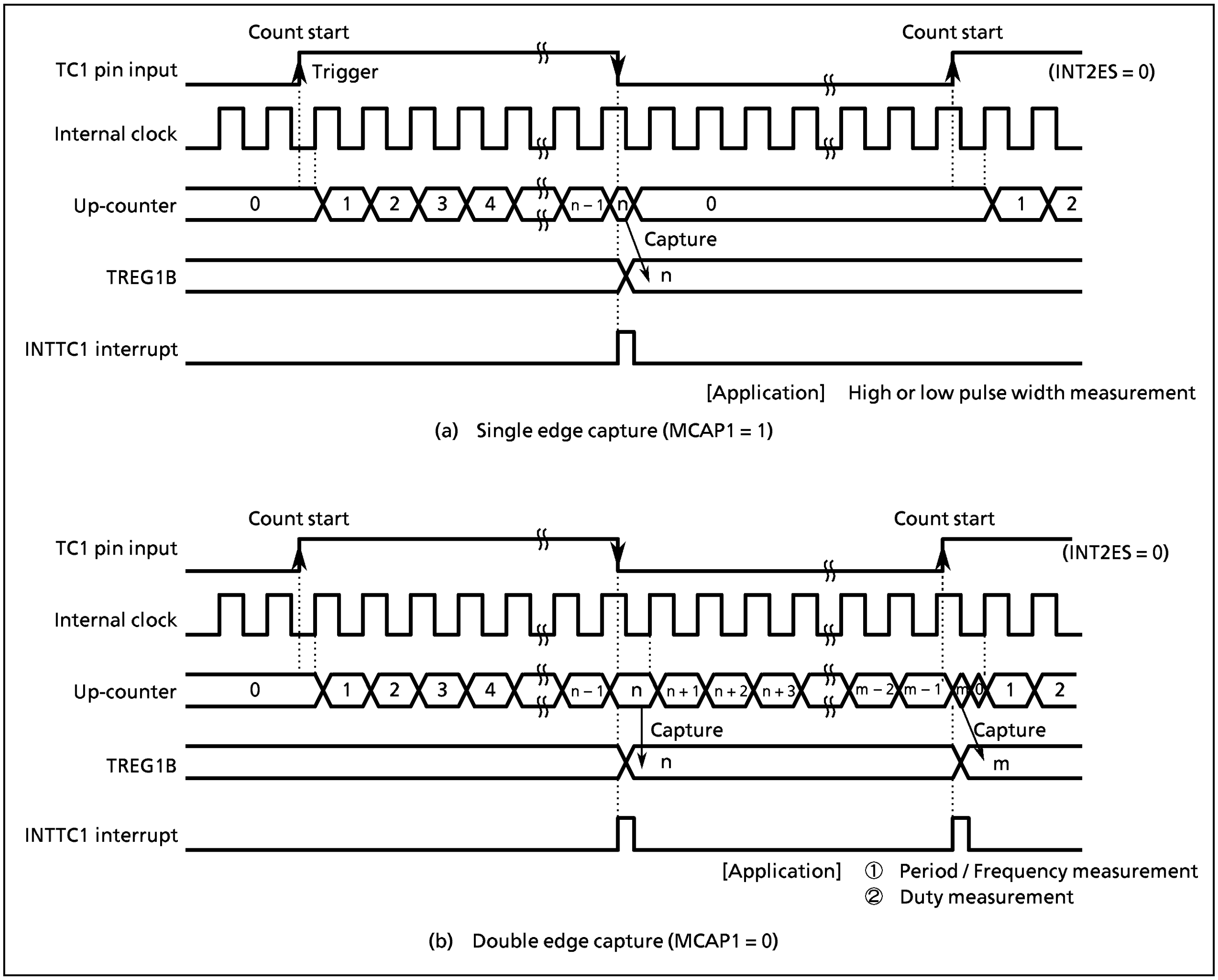
(5) **Pulse width measurement** mode

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

Example : Duty measurement (resolution fc/2^7 [Hz] DV1CK = 0)

```
           CLR    (INTTC1SW). 0         ;  INTTC1 service switch initial setting
           LD (EINTCR), 00000000B       ;  Sets the rise edge at the INT2 edge
           LD (TC1CR), 00000110B        ;  Sets the TC1 mode and source clock
           SET    (EIRL). EF4           ;  Enables INTTC1
           EI
           LD (TC1CR), 00110110B        ;  Starts TC1 with an external trigger at MCAP1 = 0
                    ⋮
PINTTC1 :  CPL    (INTTC1SW). 0         ;  Complements INTTC1 service switch
           JRS    F, SINTTC1
           LD (HPULSE), (TREG1BL)       ;  Reads TREG1B ("H" level pulse width)
           LD (HPULSE + 1), (TREG1BH)
           RETI
SINTTC1 :  LD (WIDTH), (TREG1BL)        ;  Reads TREG1B (Period)
           LD (WIDTH + 1), (TREG1BH)
                    ⋮                   ;  Duty calculation
                  RETI
                    ⋮
VINTTC1 :  DW     PINTTC1
```





[Application] High or low pulse width measurement

(a) Single edge capture (MCAP1 = 1)

[Application] ① Period / Frequency measurement
② Duty measurement

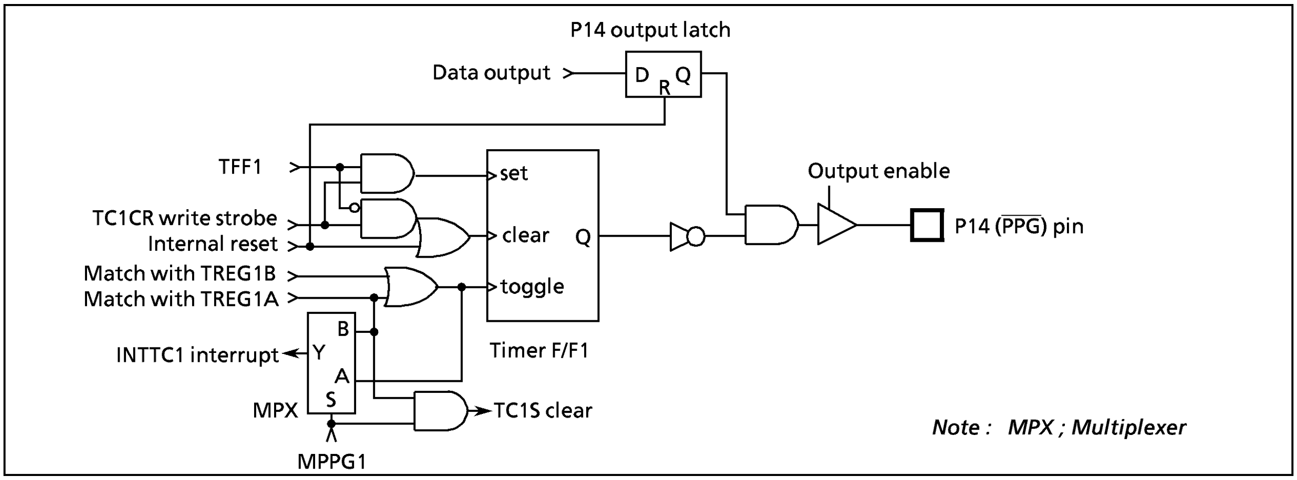
(b) Double edge capture (MCAP1 = 0)

Figure 2-21. Pulse Measurement Mode Timing Chart

(6) **Programmable Pulse Generate** (PPG) **output** mode

Counting is started by an edge of the TC1 pin input (either the rising or falling edge can be selected) or by a command. The source clock is used an internal clock. First, the contents of TREG1B are compared with the contents of the up-counter. If a match is found, timer F/F1 output is toggled. When MPPG1 = 0, an INTTC1 interrupt is generated. Next, timer F/F is again toggled and the counter is cleared by matching with TREG1A. An INTTC1 interrupt is generated at this time. Timer F/F output is connected to the P14 ($\overline{\text{PPG}}$) pin. In the case of $\overline{\text{PPG}}$ output, set the P14 output latch to "1" and configure as an output with $P1CR_4$. timer F/F1 is cleared to "0" during reset. The timer F/F1 value can also be set by TFF1 (bit 7 in TC1CR) and either a positive or negative logic pulse output is available. Also, writing to the TREG1B is not possible unless the timer / counter 1 is set to the PPG output mode with TC1M.

```
Example   :  Pulse output "H" level 800 μs, "L" level 200 μs (at fc = 12.5 MHz, DV1CK = 1)
             SET   (P1). 4               ;  P14 output latch←1
             LD    (P1CR), 00010000B     ;  Sets the P14 output mode
             LD    (TC1CR), 10001011B    ;  Sets the PPG output mode
             LDW   (TREG1A), 03E8H       ;  Sets the period (1 ms ÷ 1 μs = 03E8H)
             LDW   (TREG1B), 00C8H       ;  Sets "L" level pulse width
                                            (200 μs ÷ 1 μs = 00C8H)
             LD (TC1CR), 10011011B       ;  Starts
```



Note : MPX ; Multiplexer

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

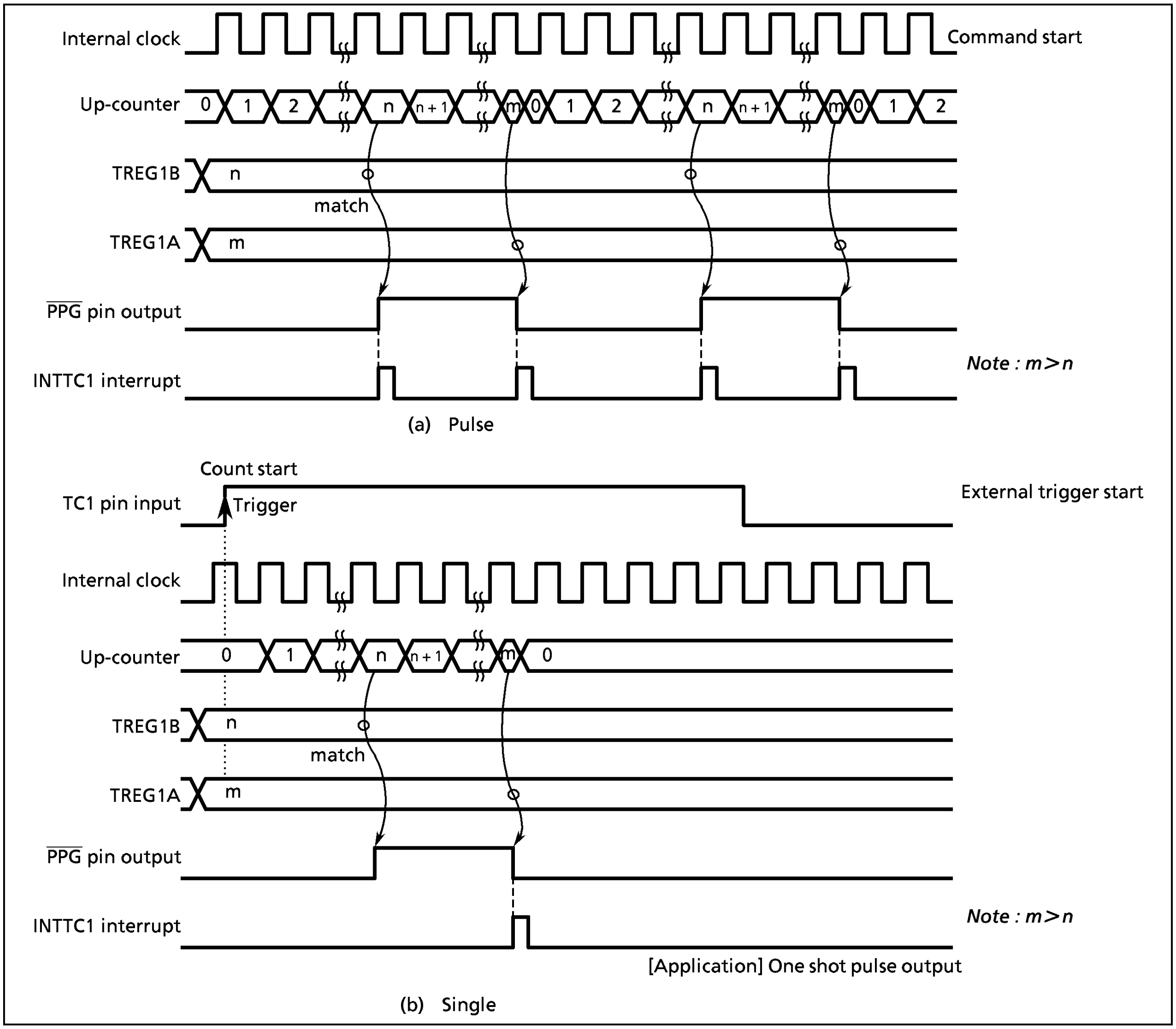


Figure 2-23. PPG Output Mode Timing Chart

2.7 16-bit Timer / Counter 2 (TC2)

2.7.1 Configuration

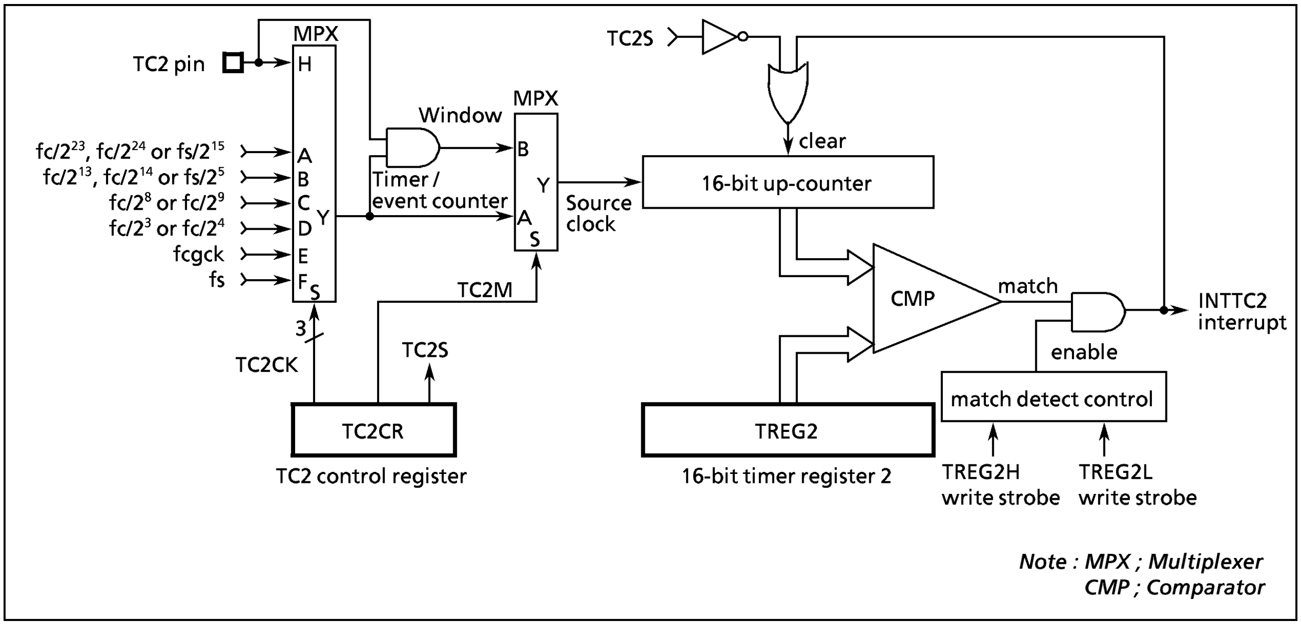


Figure 2-24. Timer / Counter 2 (TC2)

2.7.2 Control

The timer / counter 2 is controlled by a timer / counter 2 control register (TC2CR) and a 16-bit timer register 2 (TREG2). Reset does not affect TREG2.

TREG2 (00016, 00017$_H$)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TREG2$_H$ (00017$_H$)								TREG2L (00016$_H$)							

Write only

TC2CR (00015$_H$)	7	6	5	4	3	2	1	0
			TC2S	TC2CK				TC2M

(Initial value : **00 00*0)

			NORMAL1/2, IDLE1/2 mode				SLOW mode	SLEEP mode	
			DV7CK = 0		DV7CK = 1				
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1			
TC2M	TC2 operating mode select	0 : Timer / event counter mode 1 : Windou mode							Write only
TC2CK	TC2 source clock select [Hz]	000	fc/2^{23}	fc/2^{24}	fs/2^{15}	fs/2^{15}	fs/2^{15}	fs/2^{15}	
		001	fc/2^{13}	fc/2^{14}	fs/2^{5}	fs/2^{5}	fs/2^{5}	fs/2^{5}	
		010	fc/2^{8}	fc/2^{9}	fc/2^{8}	fc/2^{9}	–	–	
		011	fc/2^3 to 2^6 (DV1G)	fc/2^4 to2^7 (DV1G)	fc/2^3 to 2^6 (DV1G)	fc/2^4 to 2^7 (DV1G)	–	–	
		100	–	–	–	–	fcgck	–	
		101	fs	fs	fs	fs	–	–	
		110	reserved						
		111	External clock (TC2 pin input)						
TC2S	TC2 start control	0 : Stop and counter clear 1 : Start							

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

Note 2 : Writing to the lower byte of timer register 2 (TREG2$_L$), the comparison is inhibited until the upper byte (TREG2$_H$) is written. After writing to the upper byte, any match during 1 machine cycle (instructon execution cycle) is ignored.

Note 3 : Set the mode and source clock when the TC2 stops (TC2S = 0).

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

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

Note 6 : TC2CR and TREG2 are write-only registers and must not be used with any of the read-modify-write instructions such as SET, CLR, etc.

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

2.7.3 Function

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

(1) **Timer** mode

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

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

Table 2-6. Source Clock (internal clock) for Timer / Counter 2 (at fc = 12.5 MHz, fs = 32.768 kHz)

TC2CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution	Maximum time setting	Resolution	Maximum time setting	Resolution	Maximum time setting	Resolution	Maximum time setting
000	671 ms	12.2 h	1.34 s	24.4 h	1 s	18.2 h	1 s	18.2 h
001	655.36 μs	43.0 s	1.31 ms	1.4 min	0.98 ms	1.07 min	0.98 ms	1.07 min
010	20.48 μs	1.34 s	40.96 μs	2.7 s	20.48 μs	1.34 s	40.96 μs	2.7 s
011	0.64 μs	41.92 ms	1.28 μs	83.8 ms	0.64 μs	41.92 ms	1.28 μs	83.84 ms
100	–	–	–	–	–	–	–	–
101	30.5 μs	2 s	30.5 μs	2 s	30.5 μs	2 s	30.5 μs	2 s

TC2CK	SLOW mode		SLEEP mode	
	Resolution (s)	Maximum time setting	Resolution (s)	Maximum time setting
000	1 s	18.2 h	1 s	18.2 h
001	0.98 ms	1.07 min	0.98 ms	1.07 min
01*	–	–	–	–
100	80.0 ns (Note)	–	–	–
101	–	–	–	–

Note : "fcgck" can be used only in the timer mode. It is used for warm-up when switching from SLOW mode to NORMAL2 mode. (80.0 ns at fcgck = fc)

Example : Sets the timer mode with source clock fc/2^4 [Hz] and generates an interrupt every 25 ms (at fc = 12.5 MHz, DV1CK = 1).

```
LD (TC2CR), 00101100B   ;   Sets the TC2 mede and source clock.
LDW (TREG2), 4C4BH      ;   Sets TREG2 (25 ms ÷ 2⁴/fc = 4C4BH)
SET  (EIRH). EF14       ;   Enables INTTC2 interrupt
EI
LD (TC2CR), 00101100B   ;   Starts TC2
```

(2) **Event counter** mode

In this mode, events are counted on the rising edge of the TC2 pin input. The contents of TREG2 are compared with the contents of the up-counter. If a match is found, an INTTC2 interrupt is generated, and the counter is cleared. The maximum frequency applied to the TC2 pin is shown in table 2-7. Two or more machine cycles are required for both the "H" and "L" levels of the pulse width.

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

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

Table 2-7. Timer / Counter 2 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode
fcgck/ 2^4	fs/2^4

(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 internal clock is selected with TC2CK. The contents of TREG2 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.

Example : Generates an interrupt, inputting "H" level pulse width of 120 ms or more. (at fc = 12.5 MHz, DV1CK = 1)

```
LD (TC2CR), 00000101B   ;   Sets the TC2 mode and source clock
LDW   (TREG2), 005BH    ;   Sets TREG2 (120 ms ÷ 2^14/fc = 005B_H)
SET   (EIRH). EF14      ;   Enables INTTC2 interrupt
EI
LD (TC2CR), 00100101B   ;   Starts TC2
```

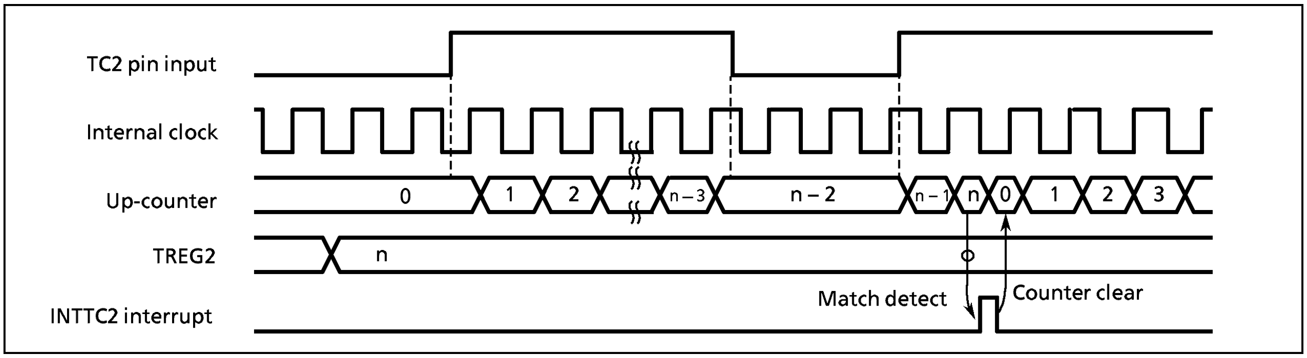
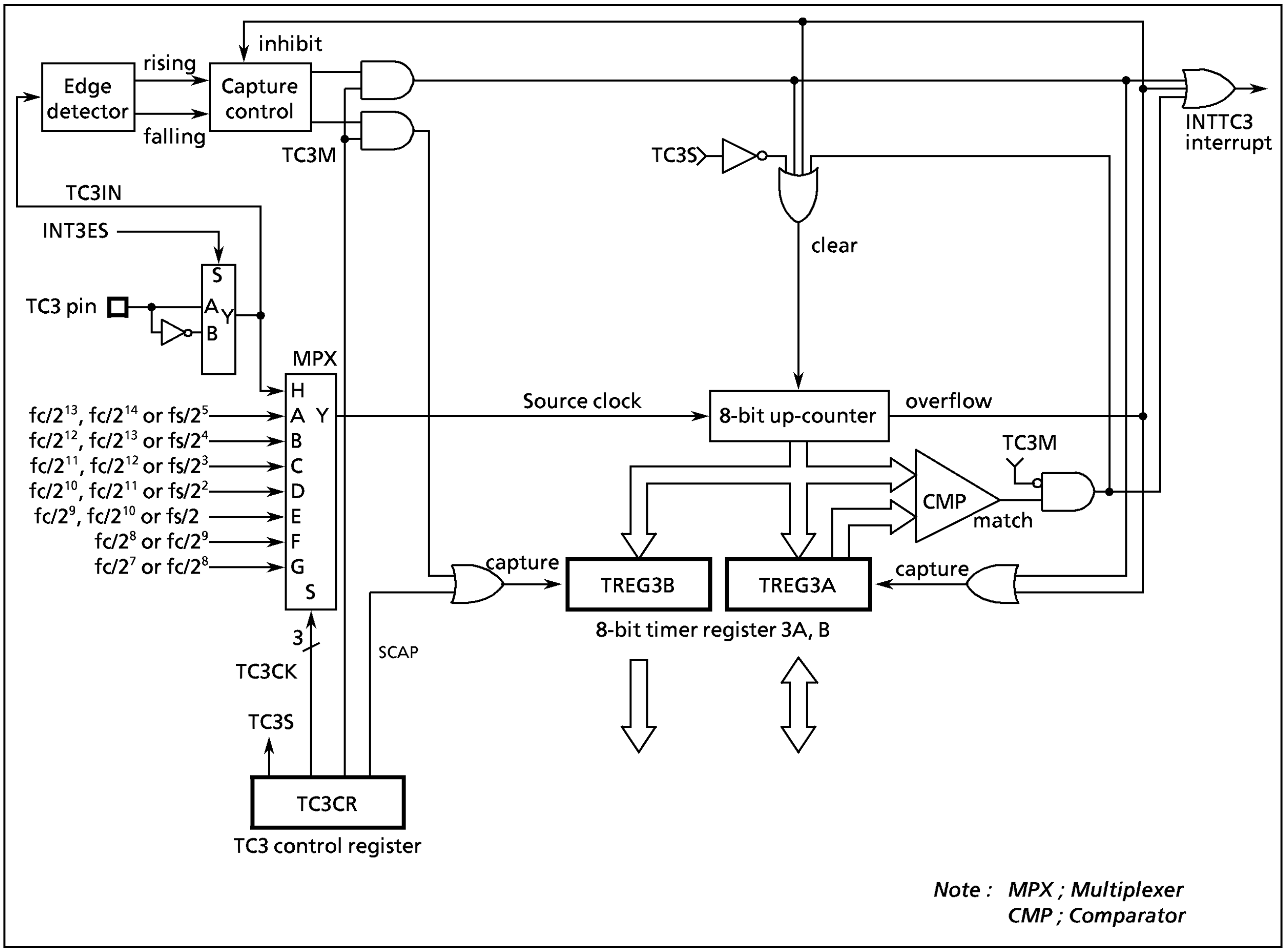


Figure 2-26. Window Mode Timing Chart

2.8 8-bit Timer / Counter 3 (TC3)

2.8.1 Configuration



Note : MPX ; Multiplexer
CMP ; Comparator

Figure 2-27. Timer / Counter 3 (TC3)

2.8.2 Control

The timer / counter 3 is controlled by a timer / counter 3 control register (TC3CR) and two 8-bit timer registers (TREG3A and TREG3B)

	7	6	5	4	3	2	1	0	
TREG3A (00018_H)									Read / Write
TREG3B (00019_H)									Read only
TC3CR ($0001A_H$)		SCAP		TC3S	TC3CK			TC3M	(Initial value : *0*0 0000)

TC3M	TC3 operation mode set	0 : Timer / event counter 1 : Capture						Write only
TC3CK	TC3 source clock select [Hz]		NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		000	$fc/2^{13}$	$fc/2^{14}$	$fs/2^5$	$fs/2^5$	$fs/2^5$	
		001	$fc/2^{12}$	$fc/2^{13}$	$fs/2^4$	$fs/2^4$	$fs/2^4$	
		010	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	–	
		011	$fc/2^{10}$	$fc/2^{11}$	$fs/2^2$	$fs/2^2$	–	
		100	$fc/2^9$	$fc/2^{10}$	fs/2	fs/2	–	
		101	$fc/2^8$	$fc/2^9$	$fc/2^8$	$fc/2^9$	–	
		110	$fc/2^7$ (DV5)	$fc/2^8$ (DV5)	$fc/2^7$ (DV5)	$fc/2^8$ (DV5)	–	
		111	External clock (TC3 pin input)					
TC3S	TC3 start select	0 : Stop & clear 1 : Start						
SCAP	Software capture control	0 : – 1 : Software capture trigger						

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

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

Note 3 : Software capture can be used only in the timer and event counter mode. SCAP is automatically cleared to "0" after capturing.

Note 4 : Values to be loaded into timer register 3A must satisfy the following condition.
TREG3A > 0 (in the timer and event counter mode)

Note 5 : TC3CR is a write-only register and must not be used with any of read -modify-write instructions such as SET, CLR, etc.

Figure 2-28. Timer Register 3 and TC3 Control Register

2.8.3 Function

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

When it is used in the capture mode, the noise rejection time of TC3 pin input can be set by remote control receive control register.

(1) **Timer** mode

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

Table 2-8. Source Clock (internal clock) for Timer / Counter 3 (Example : at fc = 12.5 MHz, fs = 32.768 kHz)

TC3CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum setting time [ms]	Resolution [μs]	Maximum setting time [ms]	Resolution [μs]	Maximum setting time [ms]	Resolution [μs]	Maximum setting time [ms]
000	655.36	167.8	1310.72	334.2	976.56	249.0	976.56	249.0
001	327.68	83.6	655.36	167.8	488.28	124.5	488.28	124.5
010	163.84	41.7	327.68	83.6	244.14	62.3	244.14	62.3
011	81.92	20.9	163.84	41.7	122.07	31.1	122.07	31.1
100	40.96	10.5	81.92	20.9	61.01	15.6	61.01	15.6
101	20.48	5.2	40.96	10.5	20.48	5.2	40.96	10.5
110	10.24	2.6	20.48	5.2	10.24	2.6	20.48	5.2

TC3CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum setting time [ms]
000	976.56	249.0
001	488.28	124.5
01*	–	–
10*	–	–
110	–	–

(2) **Event counter** mode

In this mode, the TC3 pin input pulses are used for counting up. Either the rising or falling edge can be selected. Edge selection is the same as for INT3 pin. The contents of TREG3A are compared with the contents of the up-counter. If a match is found, an INTTC3 interrupt is generated and the counter is cleared.

The maximum applied frequency is shown in table 2-9. Two or more machine cycles are required for both the high and low levels of the pulse width.

The current contents of up-counter are loaded into TREG3B by setting SCAP (bit 6 in TC3CR) to "1". SCAP is automatically cleared to "0" after capturing.

Example : Generates an interrupt every 0.5 s, inputting 50 Hz pulses to the TC3 pin.
LD (TREG3A), 19_H ; 0.5 s ÷ 1/50 = 25 = 19_H
LD (TC3CR), 00011110B ; Starts TC3

Table 2-9. Source Clock (External Clock) for Timer / Counter

Maximum applied frequency [Hz]	
NORMAL1/2, IDLE1/2 mode	SLOW, SLEEP mode
fcgck/2^4	fs/2^4

(3) **Capture** mode

The pulse width, period and duty of the TC3 pin input are measured in this mode, which can be used in decoding the remote control signals or distinguishing AC 50/60 Hz, etc. The counter is free running by the internal clock. Either the rising or falling edge can be selected. Edge selection the same as for INT3 pin. On the rising (falling) edge of the TC3 pin input, the current contents of counter is loaded into TREG3A, then the up-counter is cleared to "0" and an INTTC3 interrupt is generated. On the falling (rising) edge of the TC3 pin input, the current contents of the counter is loaded into TREG3B. In this case, counting continues. On the next rising (falling) edge of the TC3 pin input, the current contents of counter are loaded into TREG3A, then the counter is cleared again and an interrupt is generated. If the counter overflows before the edge is detected. FF$_H$ is set into TREG3A, and the counter is cleared and an INTTC3 interrupt is generated. During interrupt processing, it can be determined whether or not there is an overflow by checking whether or not the TREG3A value is FF$_H$. Also, after an interrupt (capture to TREG3A, or overflow detection) is generated, capture and overflow detection are halted until TREG3A has been read out ; however, the counter continues. As reading out TREG3A resumes capture / overflow detection, TREG3B must be beforehand read out.

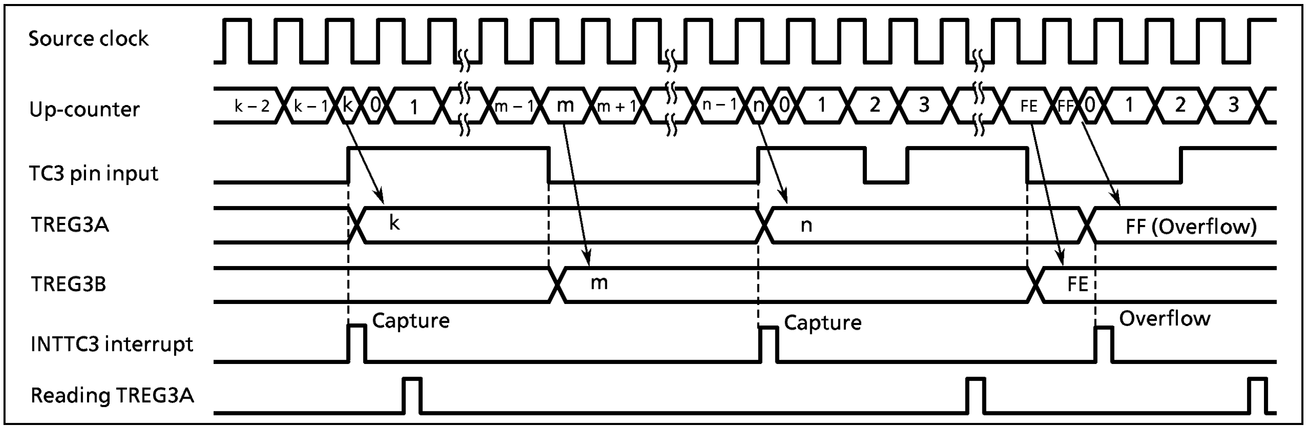


Figure 2-29. Capture Mode Timing Chart (at INT3ES = 0)

The edge of TC3 pin input is detected in the remote control receive circuit with noize rejection. The remote control receive circuit is controlled by the remote control receive control register (RCCR). The romote control receive status register (RCSR) can monitor the porality selection and noize rejection circuit.

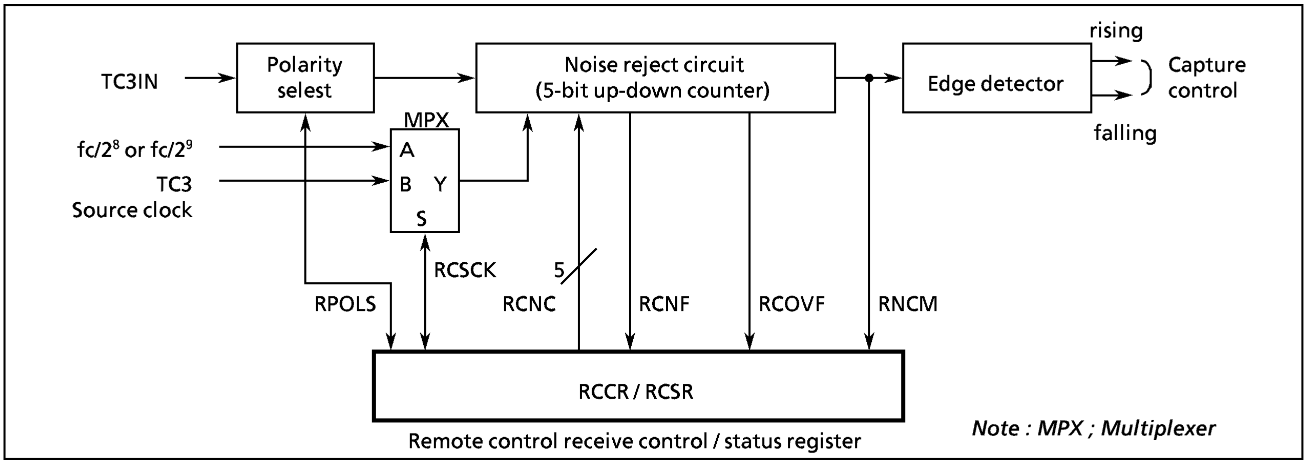


Figure 2-30. Remote Control Receiving Circuit

RCCR
(0001FH)

	RPOLS	RCSCK			RCNC		

(Initial value : *001 1111)

<table>
<tr><td>RPOLS</td><td>Remote control signal polarity select</td><td colspan="6">0 : Positive
1 : Negative</td><td rowspan="6">R/W</td></tr>
<tr><td rowspan="5">RCSCK</td><td rowspan="5">Noise reject circuit
Source clock select</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00</td><td>2^8/fc</td><td>2^9/fc</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>11</td><td colspan="5">TC3CK (Note2)</td></tr>
<tr><td>RCNC</td><td>Noise reject time select
$02_H \leqq$ RCNC $\leqq 1F_H$</td><td colspan="6">(Source clock) × (RCNC–1) [s]</td><td>Write only</td></tr>
</table>

Note 1 : Set RPOLS and RCSCK when the timer / counter stops (TC3S = 0)
Note 2 : Source clock of timer / counter 3
*Note 3 : fc ; High-frequency clock [Hz], * ; don't care*
Note 4 : RCCR include a write-only register and must not be used with any of the read-modify-write instructions.

RCSR
(0001F$_H$)

RCNF	RPOLS	RCSCK	RCOVF	RNCM			

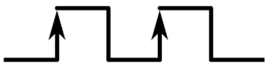
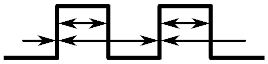
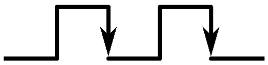
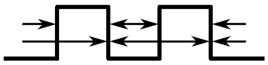
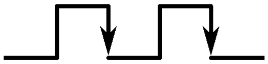
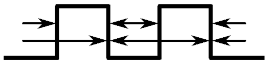
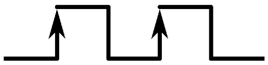
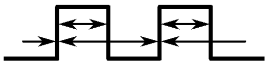
(Initial value : 0000 0***)

<table>
<tr><td>RCNF</td><td>Remote control signal monitor after noise rejecter</td><td colspan="6">0 : Without noise
1 : With noise</td><td>Read only</td></tr>
<tr><td>RPOLS</td><td>Remote control signal polarity select</td><td colspan="6">0 : Positive
1 : Negative</td><td rowspan="6">R/W</td></tr>
<tr><td rowspan="5">RCSCK</td><td rowspan="5">Noise reject circuit
Source clock select</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>00</td><td>2^8/fc</td><td>2^9/fc</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>11</td><td colspan="5">TC3CK Note2</td></tr>
<tr><td>RCOVF</td><td>Noise reject circuit Overflow flag</td><td colspan="6">0 : Signal and definition by overwriting the noise reject time RCNC
1 : Other than above</td><td rowspan="4">Read only</td></tr>
<tr><td rowspan="3">RNCM</td><td rowspan="3">Remote control signal monitor after noise rejecter</td><td></td><td colspan="2">INT3ES = 0</td><td colspan="3">INT3ES = 1</td></tr>
<tr><td>0</td><td colspan="2">0</td><td colspan="3">1</td></tr>
<tr><td>1</td><td colspan="2">1</td><td colspan="3">0</td></tr>
</table>

Note 1 : Reading out the register RCSR resets RCNF and RCOVF.
Note 2 : Source clock of timer / counter 3
Note 3 : When a 5-bit up-down counter counts down to "0" after counting up, the RCNF defines to be noise.
*Note 4 : fc; High-frequency clock [Hz], * ; don't care*

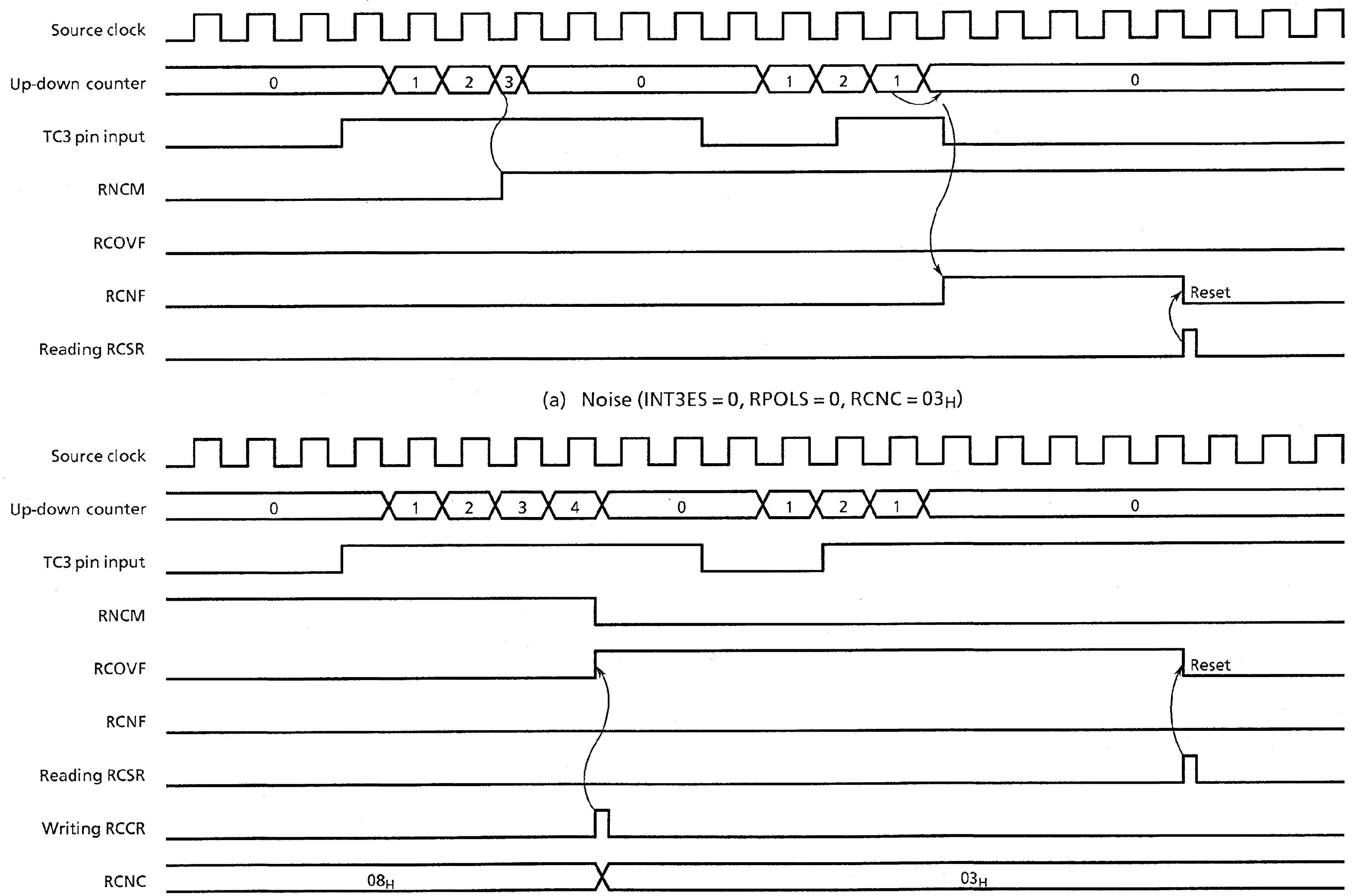
Figure 2-31. Remote Control Receive Control Register and Remote Control Receive Status Register

Table 2-10. Combination between The Porality and The Edge Selection

RPOLS	INT3ES	TC3 pin input pulse (Interrupt occurrence is shown as allow.)	Measuremont
0	0		
	1		
1	0		
	1		

Note 1 : *When TC3CK is used in RCSCK, do not select an external clock to the TC3CK.*

Note 2 : *Starting remote control receive detects an interrupt occurrence edge first.*



(a) Noise (INT3ES = 0, RPOLS = 0, RCNC = 03_H)

(b) Noise rejection circuit overflow flag (INT3ES = 0, RPOLS = 1, RCNC = 08H to 03H)

Figure2-32. Remote Control Receive Circuit Timing Chart

2.9 8-bit Timer / Counter (TC4)

2.9.1 Configuration

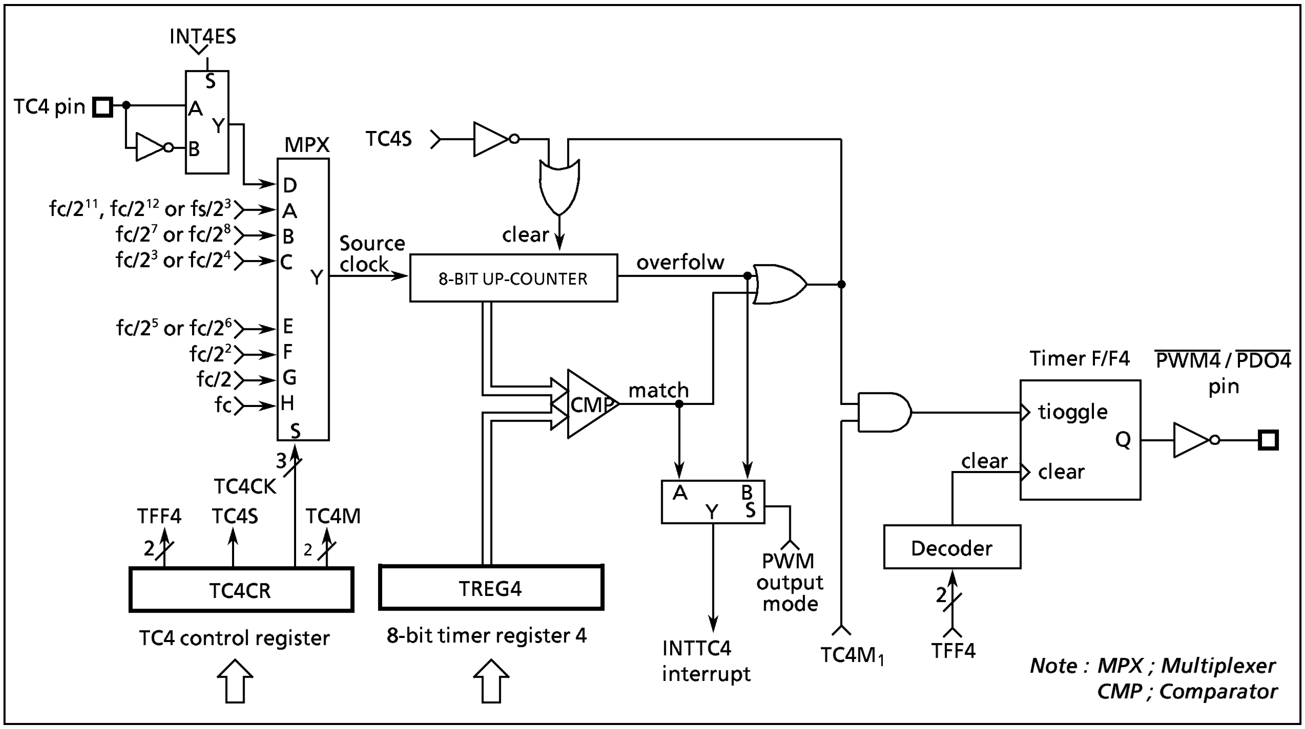


Figure 2-33. Timer / Counter 4 (TC4)

2.9.2 Control

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

	7	6	5	4	3	2	1	0	
TREG4 (0001B$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC4CR (0001C$_H$)	TFF4		TC4CK	TC4S	TC4CK		TC4M		(Initial value : 0000 0000)

TC4M	TC4 operating mode select	00 : Timer / event counter mode 01 : reserved 10 : Programmable divider output (PDO) mode 11 : Pulse width modulation (PWM) output mode						Write only
TC4CK	TC4 source clock select [Hz] (in sequence from bit 5, 3, 2)		NORMAL1/2, IDLE1/2 mode				SLOW, SLEEP mode	
			DV7CK = 0		DV7CK = 1			
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1		
		000	fc/2^{11}	fc/2^{12}	fs/2^3	fs/2^3	fs/2^3	
		001	fc/2^7 (DV5)	fc/2^8 (DV5)	fc/2^7 (DV5)	fc/2^8 (DV5)	–	
		010	fc/2^3 to 2^6 (DV1G)	fc/2^4 to 2^7 (DV1G)	fc/2^3 to 2^6 (DV1G)	fc/2^4 to 2^7 (DV1G)	–	
		011	External clock (TC4 pin input)					
		100	fc/2^5 to 2^6 (DV3G)	fc/2^6 to 2^7 (DV3G)	fc/2^5 to 2^6 (DV3G)	fc/2^6 to 2^7 (DV3G)	–	
		101	fc/2^2	fc/2^2	fc/2^2	fc/2^2	–	
		110	fc/2	fc/2	fc/2	fc/2	–	
		111	fc	fc	fc	fc	–	
TC4S	TC4 start control	0 : Stop & clear clear 1 : Start						
TFF4	Timer F/F4 control	00 : Clear 01 : Toggle 10 : Reset 11 : – Note 3						

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

Note 2 : Set the operating mode, the source clock selection, the edge selection (INT4ES) and timer F/F4 control when the TC4 stops (TC4S = 0)

Note 3 : Set TFF4 to "11" in the timer and event counter mode and PWM mode.

Note 4 : Values to be loaded to the timer register must satisfy the following condition.
(a) 5 · fc /fcgck< TREG4< 251 in PWM output mode
(b) 0< TREG4 in others

Note 5 : The source clock fc/2^2, fc/2 and fc must be used in only PWM output mode.

Note 6 : TC4CR is a write-only register and must not be used with any of the read-modify-write instructions such as SET, CLR. etc.

Figure 2-34. Timer Register 4 and TC4 Control Register

2.9.3 Function

The timer / counter 4 has four operating modes: timer, event counter, programmable divider output, and PWM output mode.

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, an INTTC4 interrupt is generated and the up-counter is cleared to "0". Counting up resumes after the up-counter is cleared.

Table 2-11. Source Clock (internal clock) for Timer / Counter 4 (Example: at fc = 12.5 MHz, fs = 32.768 kHz)

TC4CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]
000	163.84	41.7 m	327.68	83.6 m	244.14	62.2 m	244.14	62.2 m
001	10.24	2.6 m	20.48	5.2 m	10.24	2.6 m	20.48	5.2 m
010	0.64	163.2 μ	1.28	326 μ	0.64	163.2 μ	1.28	326 μ
100	2.56	653 μ	5.12	1306 μ	2.56	653 μ	5.12	1306 μ

TC4CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum setting time [s]
000	244.14	62.2 m
001	—	—
010	—	—
100	—	—

(2) **Event counter** mode

In this mode, the TC4 pin input (external clock) pulse is used for counting up. Either the rising or falling edge can be selected with TC4ES (bit 4 in EINTCR).

The contents of the TREG4 are compared with the contents of the up-counter. If a match is found, an INTTC4 interrupt is generated and the counter is cleared. Counting up resumes after the up-counter is cleared. The maximum applied frequency is shown in table 2-12. Two or more machine cycles are required for both the high and low levels of the pulse width.

Table 2-12. Timer / Counter 4 External Clock Source

Maximum applied frequency [Hz]	
NORMAL1/2, IDEL1/2 mode	SLOW, SLEEP mode
fcgck/2^4	fs/2^4

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

The internal clock is used for counting up. The contents of TREG4 are compared with the contents of the up-counter. If a match is found, the timer F/F4 output is toggled and the counter is cleared. Timer F/F4 output is inverted and output to the P32 ($\overline{\text{PDO4}}$) pin. When programmable divider output is executed, P32 output latch is set to "1". This mode can be used for approximate 50 % duty pulse output. Timer F/F4 can be initialized by program, and it is initialized to "0" during reset. An INTTC4 interrupt is generated each time the $\overline{\text{PDO4}}$ output is toggled.

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

```
SET   (P3).2                ;   P32 output latch←1
LD    (TREG4), 10H          ;   (1/1024 ÷ 2^7/fc) ÷ 2 = 10H
LD    (TC4CR), 00010110B    ;   Stats TC4
```

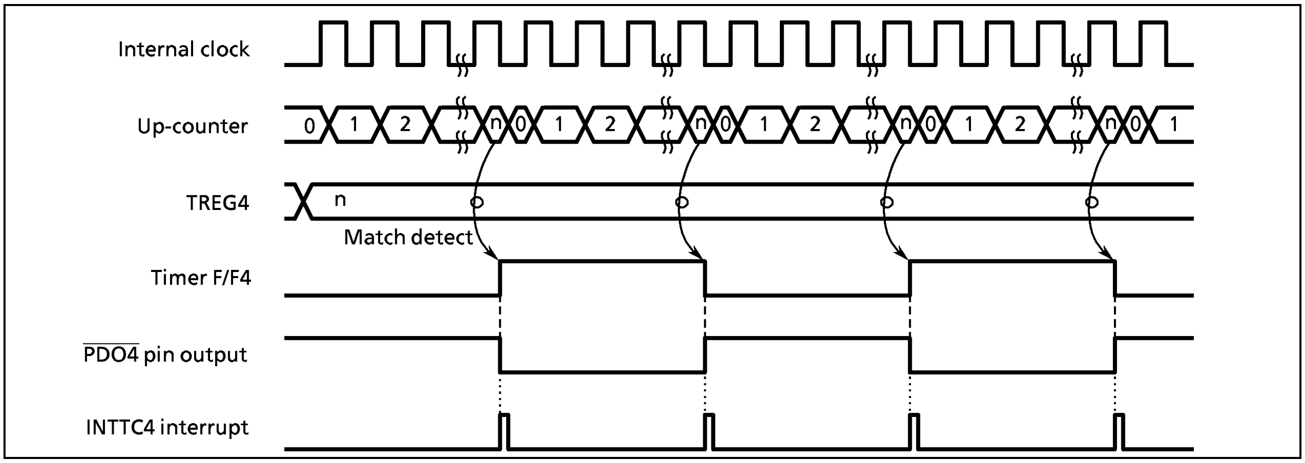


Figure 2-35. PDO Mode Timing Chart

(4) **Pulse Width Modulation** (PWM) **output** mode

PWM output with a resolution of 8 bits is possible. The internal clock is used for counting up. The contents of TREG4 are compared with the contents of up-counter. If a match is found, the timer F/F4 output is toggled. Counting up resumes. 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 P32 ($\overline{\text{PWM4}}$) pin. When programmable divider output is executed, P32 output latch is set to "1". An INTTC4 interrupt is generated when an overflow occurs.

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

*Note : Do not rewrite the contents of **TREG4** at only an **INTTC4** interrupt generation cycle. The contents of **TREG4** is rewritten by the **INTTC4** interrupt service routine.*

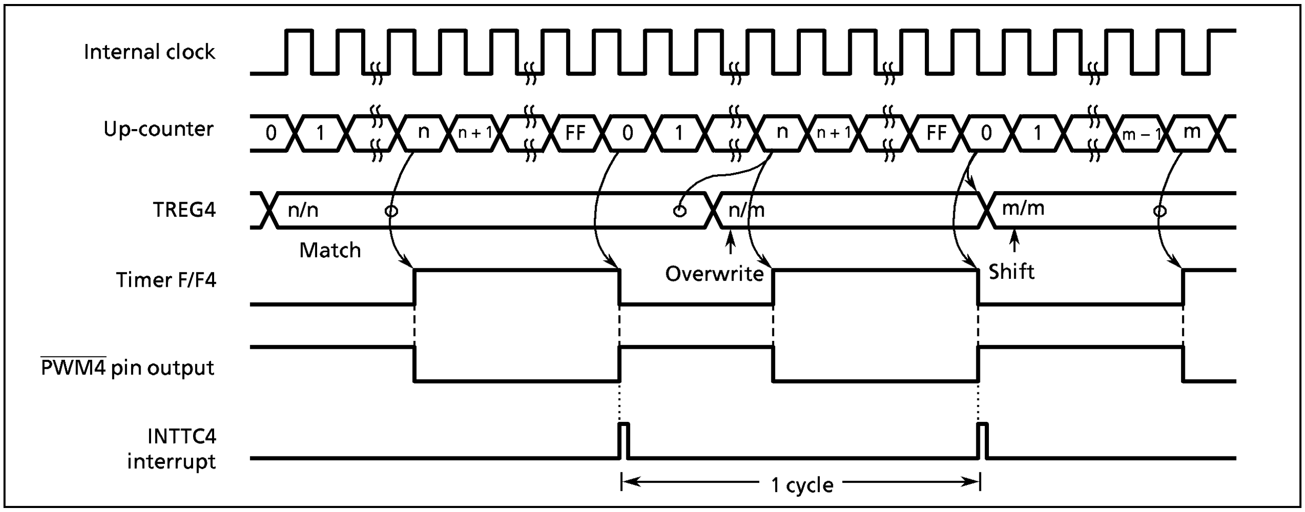


Figure 2-36. PWM Output Mode Timing Chart

Table 2-13. PWM Output Mode (Example: fc = 12.5 MHz, fs = 32.768 kHz)

TC4CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Repeat cycle [μs]
00*	—	—	—	—	—	—	—	—
010	640	163.84	—	—	640	163.84	—	—
100	—	—	—	—	—	—	—	—
101	320	81.92	320	81.92	320	81.92	320	81.92
110	160	40.96	160	40.96	160	40.96	160	40.96
111	80	20.48	80	20.48	80	20.48	80	20.48

2.10 8-bit Timer / Counter 5 (TC5)

2.10.1 Configuration

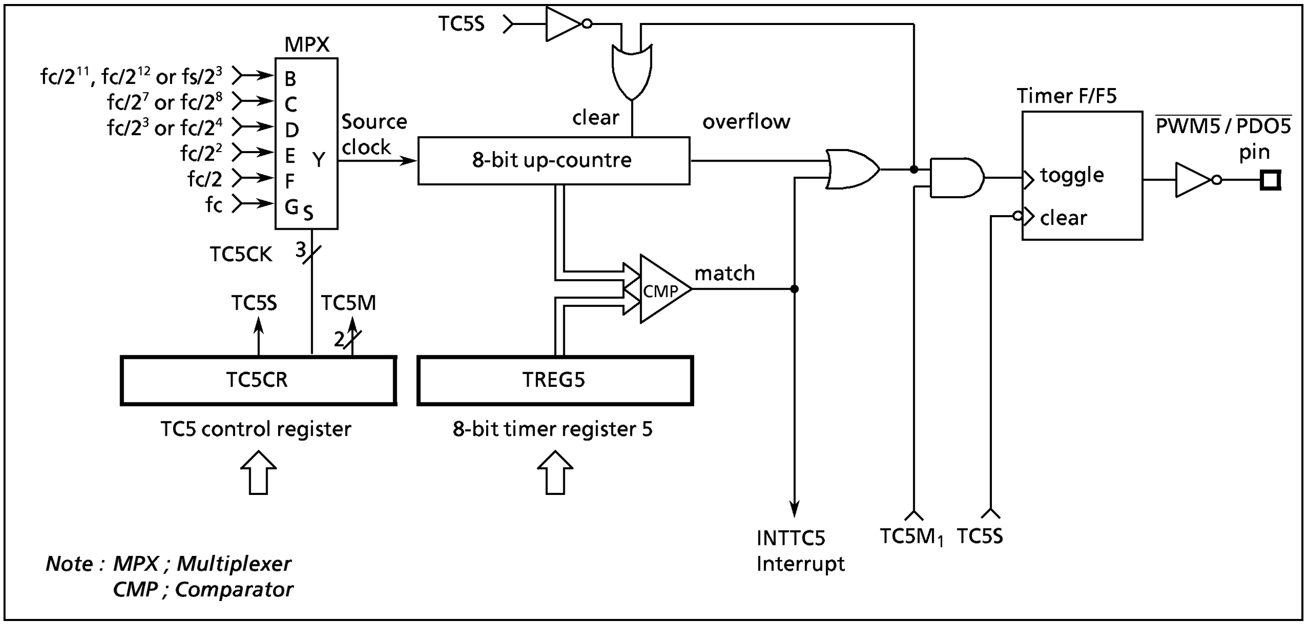


Figure 2-37. Timer / Countre 5 (TC5)

2.10.2 Control

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

	7	6	5	4	3	2	1	0	
TREG5 (0001D$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC5CR (0001E$_H$)	"0"	"0"	TC5S	TC5CK			TC5M		(Initial value : **00 0000)

<table>
<tr><td>TC5M</td><td>TC5 operating mode select</td><td colspan="7">00 : Timer / event counter mode
01 : Reserved
10 : Programmable divider output (PDO) mode
11 : Pulse width modulation (PWM) output mode</td><td rowspan="13">Write only</td></tr>
<tr><td rowspan="11">TC5CK</td><td rowspan="11">TC5 source clock select [Hz]</td><td rowspan="3"></td><td colspan="4">NORMAL1/2, IDLE1/2 mode</td><td rowspan="3">SLOW, SLEEP mode</td></tr>
<tr><td colspan="2">DV7CK = 0</td><td colspan="2">DV7CK = 1</td></tr>
<tr><td>DV1CK = 0</td><td>DV1CK = 1</td><td>DV1CK = 0</td><td>DV1CK = 1</td></tr>
<tr><td>000</td><td colspan="5">reserved</td></tr>
<tr><td>001</td><td>fc/2¹¹</td><td>fc/2¹²</td><td>fs/2³</td><td>fs/2³</td><td>fs/2³</td></tr>
<tr><td>010</td><td>fc/2⁷ (DV5)</td><td>fc/2⁸ (DV5)</td><td>fc/2⁷ (DV5)</td><td>fc/2⁸ (DV5)</td><td>–</td></tr>
<tr><td>011</td><td>fc/2³ to 2⁶(DV1G)</td><td>fc/2⁴ to 2⁷(DV1G)</td><td>fc/2³ to 2⁶(DV1G)</td><td>fc/2⁴ to 2⁷(DV1G)</td><td>–</td></tr>
<tr><td>100</td><td>fc/2²</td><td>fc/2²</td><td>fc/2²</td><td>fc/2²</td><td>–</td></tr>
<tr><td>101</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>fc/2</td><td>–</td></tr>
<tr><td>110</td><td>fc</td><td>fc</td><td>fc</td><td>fc</td><td>–</td></tr>
<tr><td>111</td><td colspan="5">reserved</td></tr>
<tr><td>TC5S</td><td>TC4 start control</td><td colspan="7">0 : Stop &amp; counter clear
1 : Start</td></tr>
</table>

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

Note 2 : Values to be loaded to the timer register must satisfy the followiing condition.
(a) 5 · fc/fcgck< TREG5< 251 in PWM output mode
(b) 0< TREG5 in others

Note 3 : The source clock fc/2^2, fc/2 and fc must be used in only PWM output mode.

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

Figure 2-38. Timer Register 5 and Control Register

2.10.3 Function

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

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG5 are compared with the contents of up-counter. If a match is found, an INTTC5 interrupt is generated and the up-counter is cleared. Counting up resumes after the up-counter is cleared.

Table 2-14. Source Clock (internal clock) for Timer / Counter 5 (Example: at fc = 12.5 MHz, fs = 32.768 kHz)

TC5CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]
001	163.84	41.7 m	327.68	83.6 m	244.14	62.2 m	244.14	62.2 m
010	10.24	2.6 m	20.48	5.2 m	10.24	2.6 m	20.48	5.2 m
011	0.64	163.2 μ	1.28	326 μ	0.64	163.2 μ	1.28	326 μ

TC5CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum setting time [s]
001	244.14	62.2 m

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

The internal clock is used for counting up. The contents of TREG5 are compared with the contents of the up-counter. Timer F/F5 output is toggled and the counter is cleared each timer a match is found. The timer F/F5 output is inverted and output to the P33 ($\overline{\text{PDO5}}$) pin. In this mode, P33 pin output latch must be set to "1". This mode can be used for approximate 50 % duty pulse output. An INTTC5 interrupt is generated each time the $\overline{\text{PDO5}}$ output is toggled.

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

```
LD    (TC5CR), 00001010B  ;  Sets the PDO mode (TC5M = 10, TC5CK = 010)
SET   (P3). 3             ;  P33 output latch←1
LD    (TREG5), 10H        ;  (1/1024 ÷ 2^7/fc) ÷ 2 = 10_H
LD    (TC5CR), 00101110B  ;  Starts TC5
```

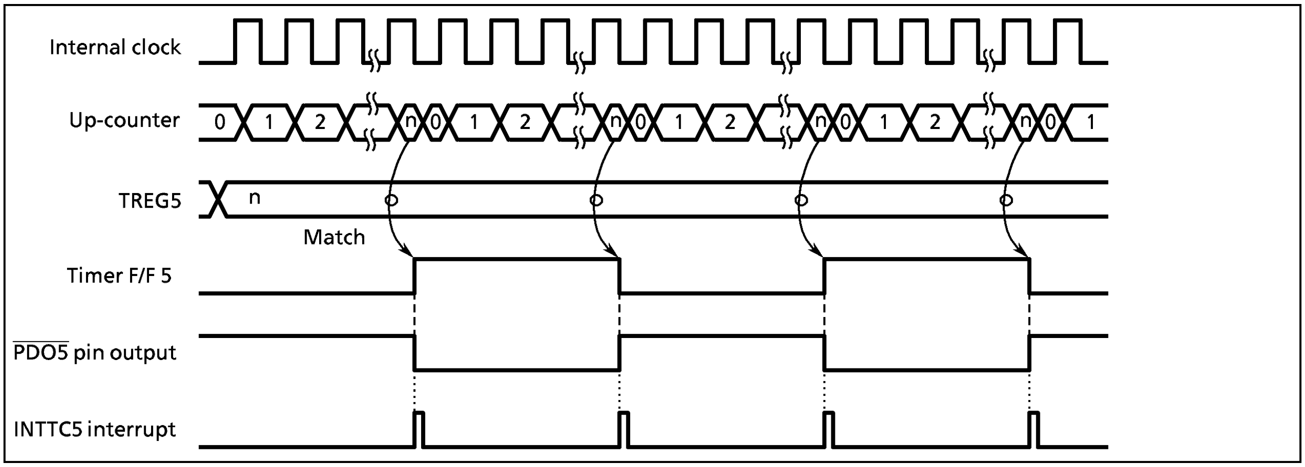


Figure 2-39. PDO Mode Timing Chart

(3) **Pulse Width Modulation** (PWM) **output** mode

PWM output with a resolution of 8-bits is possible. The internal clock is used for counting up. The contents of TREG5 are compared with the contents of up-counter. If a match is found, the timer F/F5 output is toggled. Counting up resumes. And, when an overflow occurs, the timer is again toggled and the counter is cleared. Timer F/F5 output is inverted and output to P33 ($\overline{\text{PWM5}}$) pin. When the PWM output is executed, P33 output latch is set to "1". An INTTC5 interrupt is generated when an overflow occurs. TREG5 is configured a 2-stage shift register and, during output, will not switch until one output cycle is completed even if TREG5 is overwritten, therefore, output can be altered continuously. Also, the first time, TREG5 is shifted at starting by TC5CR after data are loaded to TREG5.

Note : The PWM output mode can be used in NORMAL1, 2 and IDLE1, 2 modes.

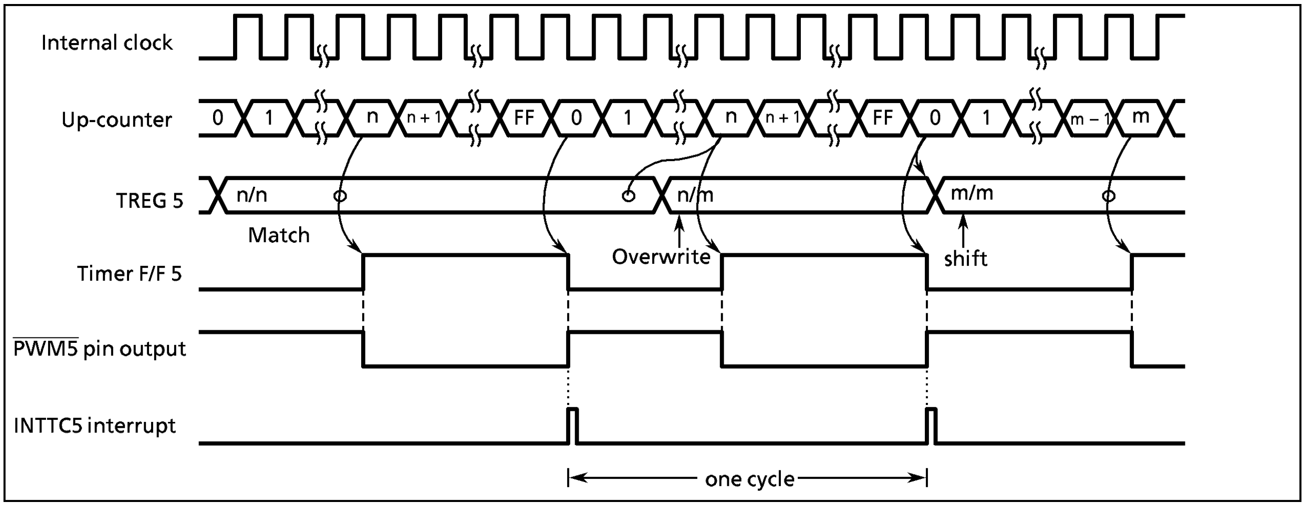


Figure 2-40. PWM Output Mode Timing Chart

Table 2-15. PWM Output Mode (Example: at fc = 12.5 MHz)

TC5CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [ns]	Repeat cycle [μs]	Resolution [ns]	Resolution [μs]	Resolution [ns]	Resolution [μs]	Resolution [ns]	Resolution [μs]
001	—	—	—	—	—	—	—	—
010	—	—	—	—	—	—	—	—
011	640	163.84	—	—	640	163.84	—	—
100	320	81.92	320	81.92	320	81.92	320	81.92
101	160	40.96	160	40.96	160	40.96	160	40.96
110	80	20.48	80	20.48	80	20.48	80	20.48

2.11 8-bit Timer / Counter 6 (TC6)

2.11.1 Configuration

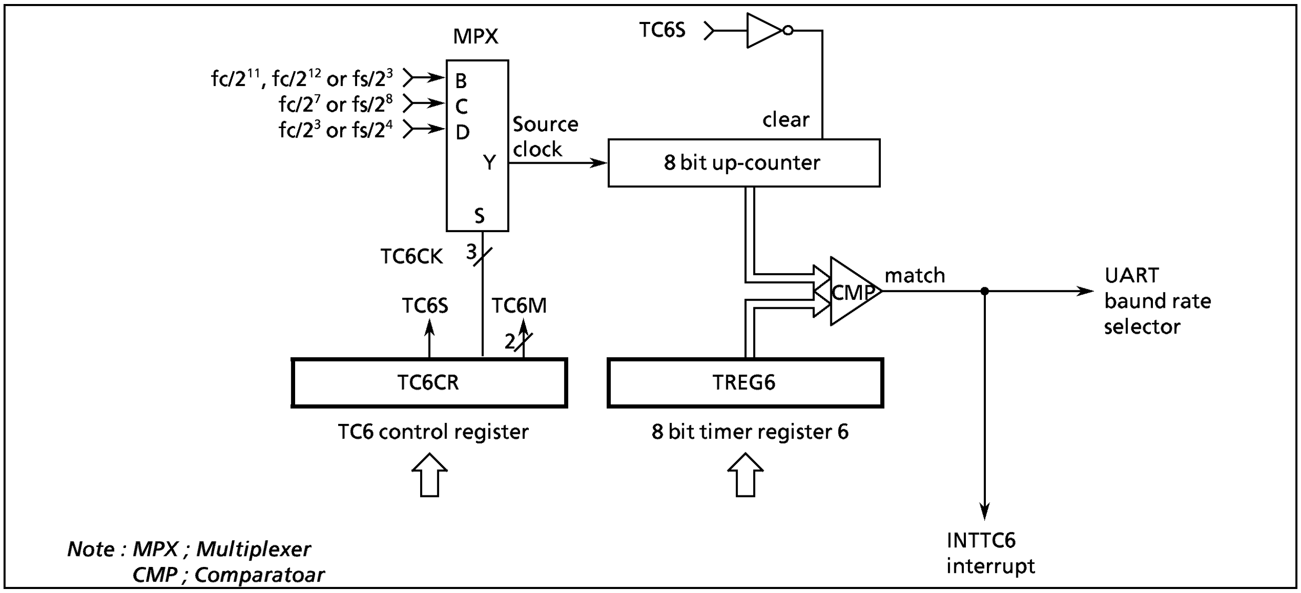


Figure 2-41. Timer / Counter 6 (TC6)

2.11.2 Control

The timer / counter 6 is controlled by a timer / counter 6 control register (TC6CR) and an 8-bit timer register 6 (TREG6)

	7	6	5	4	3	2	1	0	
TREG6 (00008$_H$)									Write only

	7	6	5	4	3	2	1	0	
TC6CR (00009$_H$)	"0"	"0"	TC6S	TC6CK			"0"	"0"	(Initial value : **00 0000)

			NORMAL1/2, IDLE1/2 mode				SLOW	
			DV7CK = 0		DV7CK = 1		SLEEP	
			DV1CK = 0	DV1CK = 1	DV1CK = 0	DV1CK = 1	mode	
TC6CK	TC6 source clock select [Hz]	000	reserved					
		001	$fc/2^{11}$	$fc/2^{12}$	$fs/2^3$	$fs/2^3$	$fs/2^3$	Write only
		010	$fc/2^7$ (DV5)	$fc/2^8$ (DV5)	$fc/2^7$ (DV5)	$fc/2^8$ (DV5)	–	
		011	$fc/2^3$ to 2^6 (DV1G)	$fc/2^4$ to 2^7 (DV1G)	$fc/2^3$ to 2^6 (DV1G)	$fc/2^4$ to 2^7 (DV1G)	–	
		1**	reserved					
TC6S	TC6 starat control	0 : Stop & clear clear 1 : Start						

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

Note 2 : Values to be loaded to the timer register must satisfy the following condition. TREG6>0

Note 3 : Set the source clock when TC6 stops (TC6S = 0).

Figure 2-42. Timer Register 6 and TC6 Cotrol Register

2.11.3 Function

The timer / counter 6 has two operating modes: timer and UART baud rate generator mode.

(1) **Timer** mode

In this mode, the internal clock is used for counting up. The contents of TREG6 are compared with the contents of up-counter. If a match is found, an INTTC6 interrupt is generated and the up-counter is cleared. Counting up resumes after the up-counter is cleared.

Table2-16. Source Clock (internal clock) for Timer / Counter 6 (Example: at fc = 12.5 MHz, fs = 32.768 kHz)

TC6CK	NORMAL1/2, IDLE1/2 mode							
	DV7CK = 0				DV7CK = 1			
	DV1CK = 0		DV1CK = 1		DV1CK = 0		DV1CK = 1	
	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]	Resolution [μs]	Maximum setting time [s]
001	163.84	41.7 m	327.68	83.6 m	244.14	62.2 m	244.14	62.2 m
010	10.24	2.6 m	20.48	5.2 m	10.24	2.6 m	20.48	5.2 m
011	0.64	163.2 μ	1.28	32.6 μ	0.64	163.2μ	1.28	326 μ

TC6CK	SLOW, SLEEP mode	
	Resolution [μs]	Maximum setting time [s]
001	244.14	62.2 m

(2) **UART baud rate generator** mode

In this mode, the internal clock is used for counting up. The contents of TREG6 are compared with the contents of up-counter. A match signal is used as a transfer clock of UART.

When the timer / counter 6 is used in this mode, BRG of UART control register (UARTCR1) must be set to "110". An INTTC6 interrupt is not generated. The transfer clock and the land rate of UART is shown as follows.

$$\text{Transfer clock} = \frac{\text{TC6 source clock}}{\text{TREG6 specified value}}$$

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

2.12 UART (Asynchronous serial interface)

The 88C060 includes 1 channel of UART (Asynchronous serial interface). It is connected to an external device via RxD and TxD. The RxD is also used as P41, and the TxD as P42. Set "1" for the output latch of the respective port of P4 for using RxD as RxD pin and TxD as TxD pin.

2.12.1 Configuration

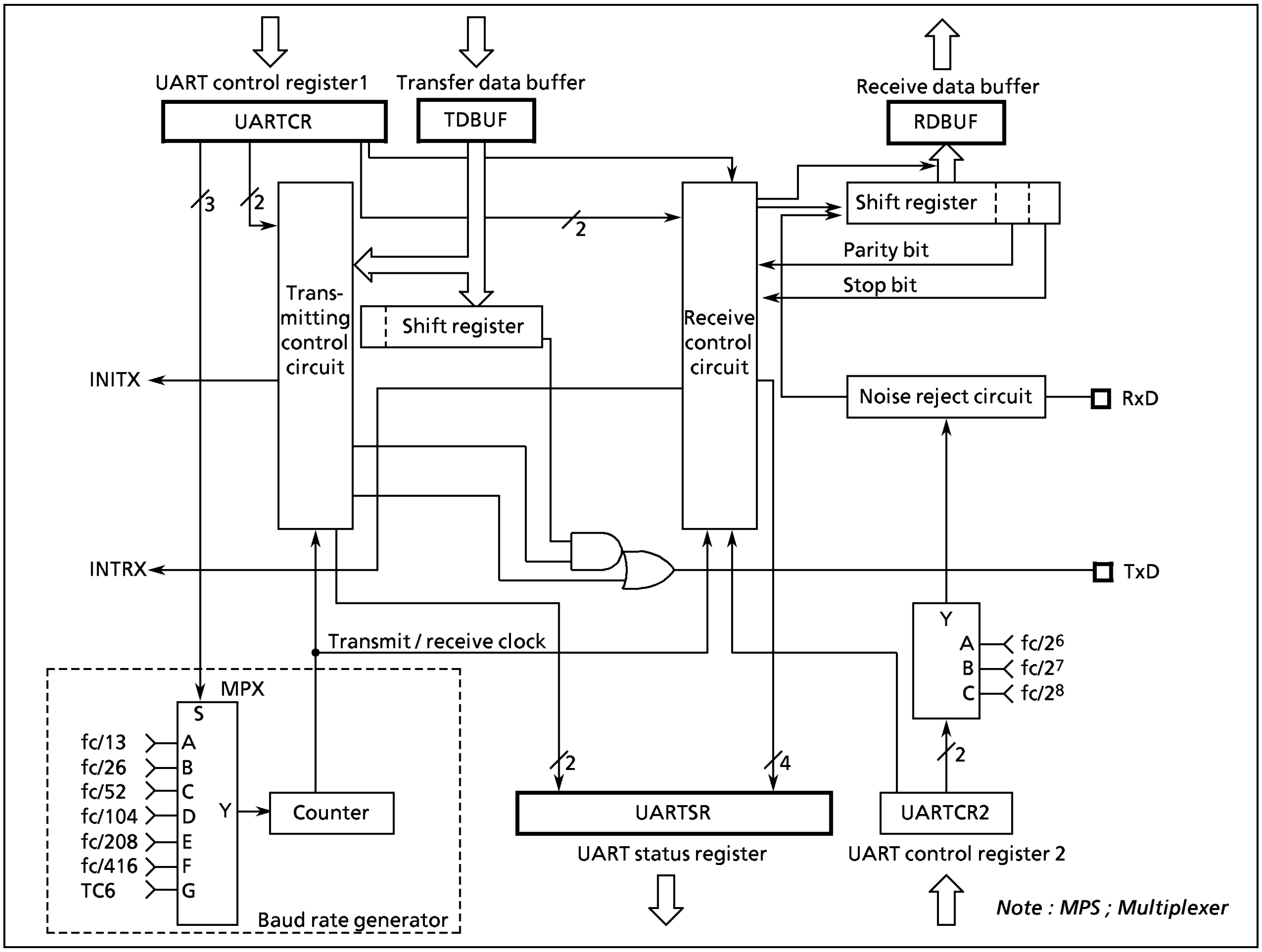


Figure 2-43. UART (Asynchronous serial interface)

2.12.2 Control

UART is controlled and monitored by UARTCR1 (UART control register1) and UARTCR2 (UART control register2) and UARTSR (UART status register).

UART Control Register

	7	6	5	4	3	2	1	0	
UARTCR1 (0002A$_H$)	TXE	RXE	STBT	EVEN	PE	BRG			(Initial value : 0000 0000)

BRG	Transfer clock select	000 : fc/13 [Hz] 001 : fc/26 010 : fc/52 011 : fc/104 100 : fc/208 101 : fc/416 110 : TC6 usable 111 : fc/96	Write only
PE	Parity addition	0 : Without parity 1 : Parity added	
EVEN	Even parity	0 : Odd parity 1 : Even party	
STBT	Transmit stop bit length	0 : 1 bit 1 : 2 bit	
RXE	Receive operation	0 : Disable 1 : Enable	
TXE	Transmit operation	0 : Disable 1 : Enable	

Note 1 : When transmitting or receiving operation is implemented, setting TXE and RXE bit to "0" inhibits the operation. When transmitting is disabled, the transfer data stored in the transfer data buffer is not output. Even if the transmit enable is set, transmitting is not operated until writing new data.

Note 2 : The transfer clock and the parity are used for transmitting and receiving.

	7	6	5	4	3	2	1	0	
UARTCR2 (0002B$_H$)						RXDNC		STOPBR	(Initial value : **** *000)

STOPBR	Receive stop bit length	0 : 1 bit 1 : 2 bits
RXDNC		00 : No noise cancellation (histeresis input) 01 : Cancels less than 31/fc [s] as noise (DV4) 10 : Cancels less than 63/fc [s] as noise (DV5) 11 : Cancels less than 127/fc [s] as noise

Note 3 : 96/fc or more when RXDNC = 01, 192/fc or more when RXDNC = 10, 384/fc or more when RXDNC = 11 are regarded as signals.

Figure 2-44. UART Control Register

UART Status Register

	7	6	5	4	3	2	1	0	
UARTSR (0002A$_H$)	PERR	FERR	OERR	RBFL	TEND	TBEP			(Initial value : 0000 11**)

TBEP	Transmit buffer empty flag	0 : – 1 : Transmit buffer empty	Read only
TEND	Transmit end flag	0 : Under transmission 1 : End of transmission	
RBFL	Receive buffer full flag	0 : – 1 : Receive buffer full	
OERR	Overrun error flag	0 : Without overrun error 1 : Overrun occurrence	
FERR	Framing error flag	0 : Without framing error 1 : Framing error occurrence	
PERR	Parity error flag	0 : Without parity error 1 : Parity error occurrence	

UART Receive Data Buffer

	7	6	5	4	3	2	1	0	
RDBUF (00029$_H$)									Read only (Initial value : 0000 0000)

UART Transfer Data Buffer

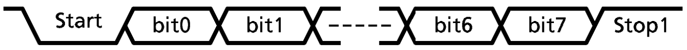
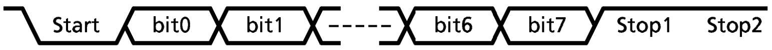
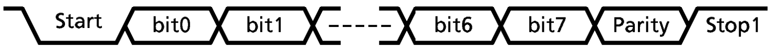
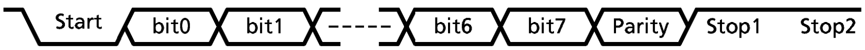
	7	6	5	4	3	2	1	0	
TDBUF (00029$_H$)									Write only (Initial value : 0000 0000)

Figure 2-45. UART Status Register / Data Buffer Register

2.12.3 Transfer Data Format

In UART, 1 start bit (level "L"), send stop bit (STBT can select 1 bit or 2 bit of the bit length Level "H"), and Parity (PE can select whether to be parity or not. EVEN can select whether to be even or odd parity.) are added to the transfer data. The data is fixed to be 8 bit. The transfer data format is shown as follow.

Table2-17. Transfer Data Format

PE	STBT	Frame length | 1 | 2 | 3 |-------| 8 | 9 | 10 | 11 | 12 |
0	0	
0	1	
1	0	
1	1	

2.12.4 Baud Rate

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

Table 2-18. Baud Rate (example)

BRG	Source clock		
	12.5 MHz	8 MHz	4 MHz
000	60000 [baud]	38400 [baud]	19200 [baud]
001	30000	19200	9600
010	15000	9600	4800
011	7500	4800	2400
100	3750	2400	1200
101	1875	1200	600

When the TC6 is used as the baud rate of UART (BRG = 110), the transfer clock and the band rate are shown as follows.

$$\text{Transfer clock} = \frac{\text{TC6 source clock}}{\text{TREG6 specified value}}$$

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

2.12.5 Data Sampling Method

The receiver of UART functions input sampling with the clock selected by BRG (bit 0, 1, and 2 of UARTCR1) until a start bit is found in RxD pin input. When a start bit is found, start bit, data bit, stop bit, and parity bit are sampled at three times of RT7, RT8, and RT9 at an interval of 1 receiver clock (1RT clock), that is shown in the following figure. RT0 is a position where the bit is expected to start. The bit data is decided by majority. (two or three times out of three sampling times)

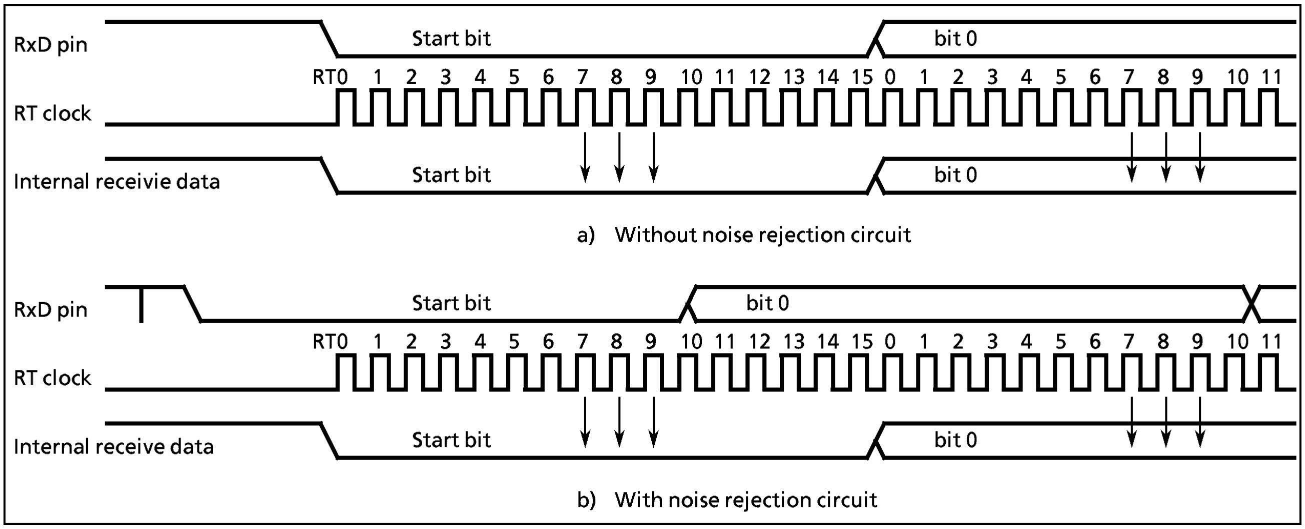

RxD pin
Start bit
bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receivie data
Start bit
bit 0
a) Without noise rejection circuit
RxD pin
Start bit
bit 0
RT0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7 8 9 10 11
RT clock
Internal receive data
Start bit
bit 0
b) With noise rejection circuit

Figure 2-46. Data Sampling Method

2.12.6 STOP Bit Length

STBT (bit 5 of UARTCR1) can select the transmit stop bit length (1 bit / 2 bits).

2.12.7 Parity

PE (bit 3 of UARTCR1) sets whether the parity is added or not, and EVEN (bit 4 of UARTCR1) sets whether the parity is odd or even.

2.12.8 Transmit / Receive Operation

(1) Data transmit operation

Set TXE (bit 7) of UARTCR1 to "1". Read UARTSR, subsequently to TBEP = 1, write the data to TDBUF (transmit data buffer). Writing the data clears TBEP to "0", and transfers the data to the clears TBEP to "0", and transmit shift register. The data is sequentially output from the TxD pin. 1 start bit, a stop bit specified with STBT (bit 5 of UARTCR1), and a parity bit (parity added) are added to the output data. The baud rate of the data transfer is selected among BRGS (bit 0 to 2 of UARTCR1).

During TXE = "0", TXE"1", and non-transfer operation (no data to transfer), TxD pin fixes to level "H".

Reading UARTSR and writing the data to TDBUF transmit the data. If no reading, TBEP is not cleared to "0". Thus the data transmit is not started.

(2) Data receive operation

Set REX (bit 6) of UARTCR1 (UART control register1) to "1". Subsequently, receiving the data from RxD pin transfers the received data to RDBUF (receive data buffer). The transmitted data includes a start bit, a stop bit, and a parity bit (parity added). Receiving the stop bit sets the receive buffer full flag (RBFL), and occurs INTRX interrupt. Only the data is transferred to the receive data buffer (RDBUF). The baud rate of the data transfer is selected among bit 0 to 2 of UARTCR.

When over run error occurs at receiving the data, the data is not transferred to RDBUF. The data is canceled. When over run error occurs, the data is transferred to RDBUF. The data in RDBUF is not effected.

2.12.9 Status Flag

(1) Parity error

When the parity calculated from the data bit of the receive data differs the received parity bit, the parity error flag PERR is set. Reading RDBUF subsequently to UARTSR reading clears the PERR.

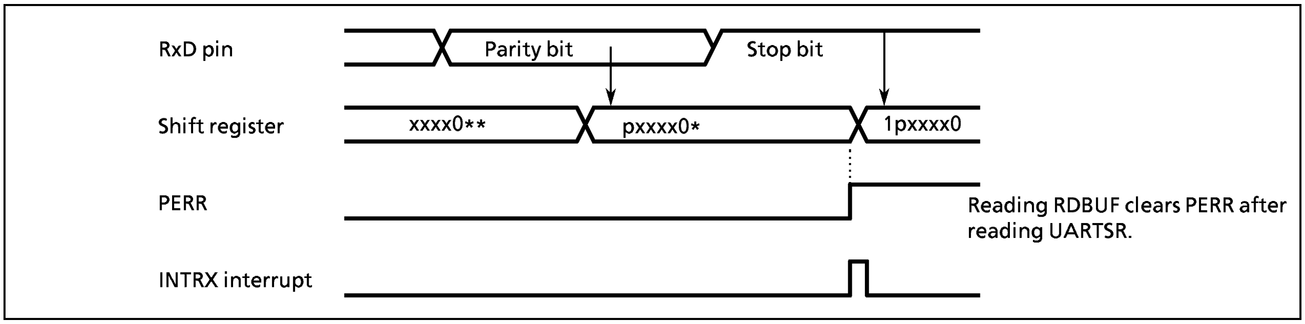


Figure 2-47. Parity Error Occurrence

(2) Framing error

When "0" is sampled as the stop bit of the receive data, the framing error flag FERR is set. Subsequently to UARTSR read, reading RDBUF clears the FERR.

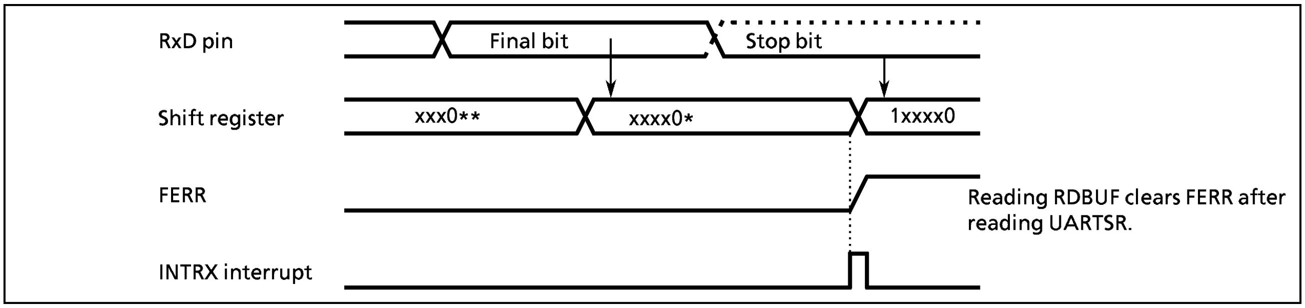


Figure 2-48. Framing Error Occurrence

(3) Over run error

When the data that is not read out to RDBUF is stored, and receiving the next data is completed for all of the bits, over run error flag OERR is set. In this case, the received data is canceled, and the data in the receive data buffer is not effected. Subsequently to UARTSR read, reading RDBUF clears the UARTSR.

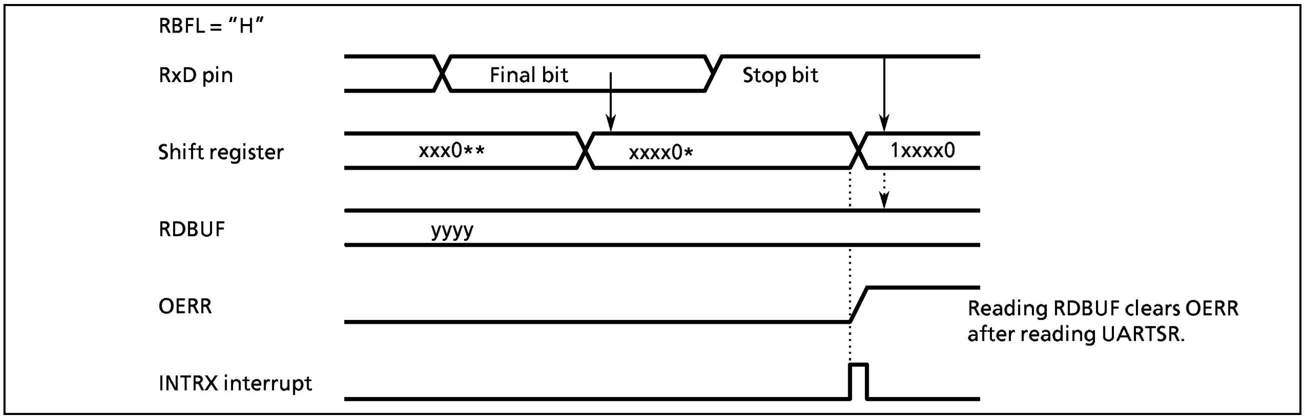


Figure 2-49. Over Run Error Occurrence

(4) Receive buffer full

Fetching the received data to RDBUF sets RBFL. Subsequently to UARTSR read, reading the data from the receive data buffer RDBUF clears the RBFL.

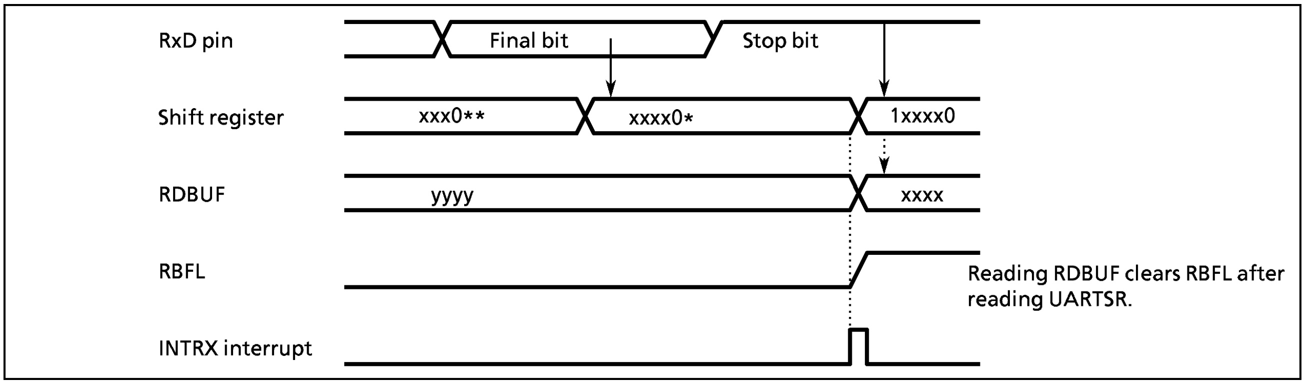


Figure 2-50. Receive Buffer Full Occurrence

(5) Transmit buffer empty

When no data is in the transmit buffer, the data of TDBUF is transferred to the transmit shift register. TBEP starting transmission sets. Subsequently UARTSR read, writing the data to the transmit buffer TDBUF clears the TBEP.

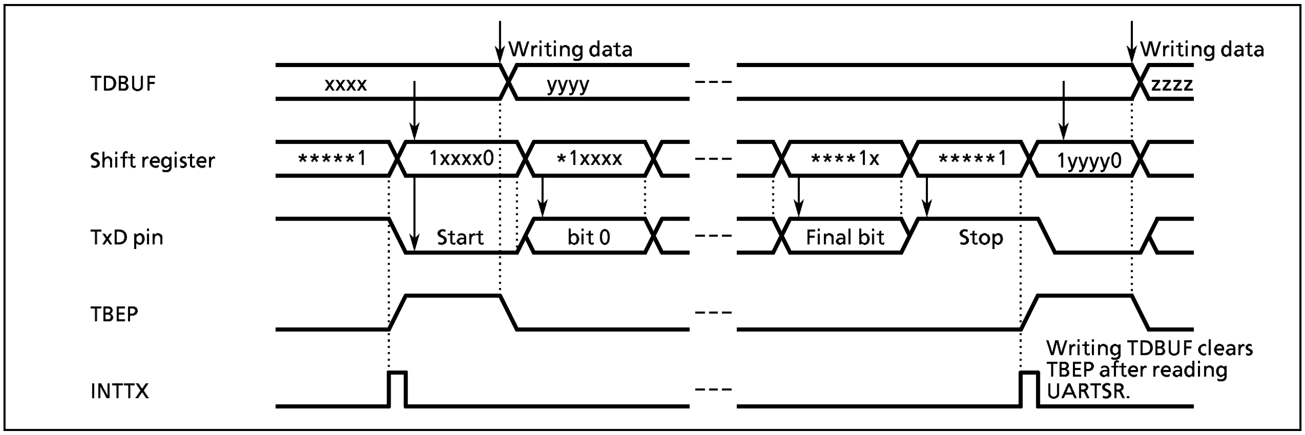


Figure 2-51. Transmit Buffer Empty Occurrence

(6) Transmit end flag

When the transmit ends, and no data is in the transmit buffer (TBEP = 1), TEND is set. Subsequently to data written to the transmit buffer, restarting the transmit clears the TEND.

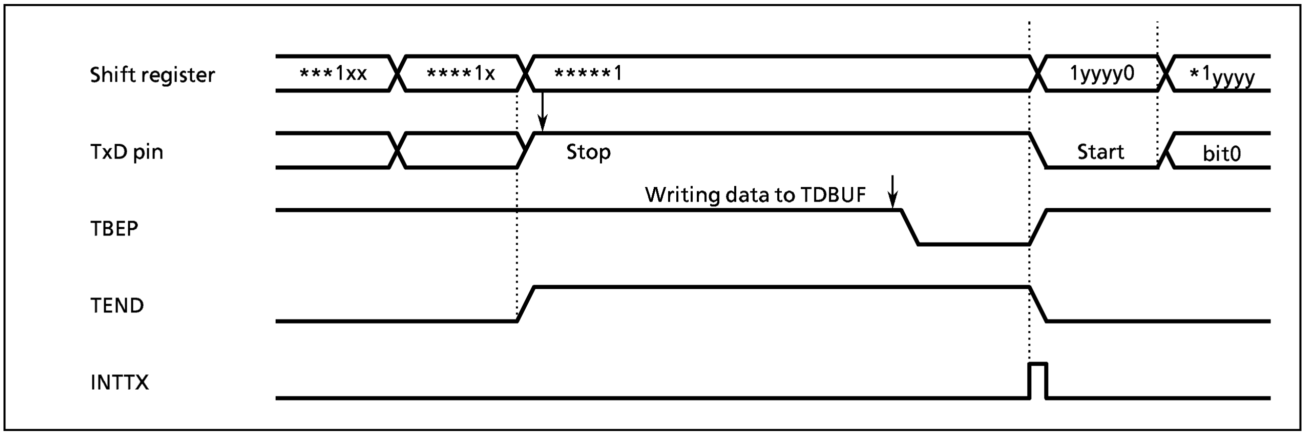


Figure 2-52. Tranmit End Flag

2.13 Serial Bus Interface (SBI)

The 88C060 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit serial bus interface and an I^{2}C bus (a bus system by Philips).

The serial interface is connected to an external device through P45 (SDA) and P44 (SCL) in the I^{2}C bus mode ; and through P43 ($\overline{\text{SCK}}$), P44 (SI) and P45 (SO) in the clocked-synchronous 8-bit SIO mode.

The serial bus interface pins are also used for the P4 port. When used for serial bus interface pins, set the P4 output latches of these pins to "1". When not used for serial bus interface pins, the pin is used as a normal I/O port.

2.13.1 Configuration

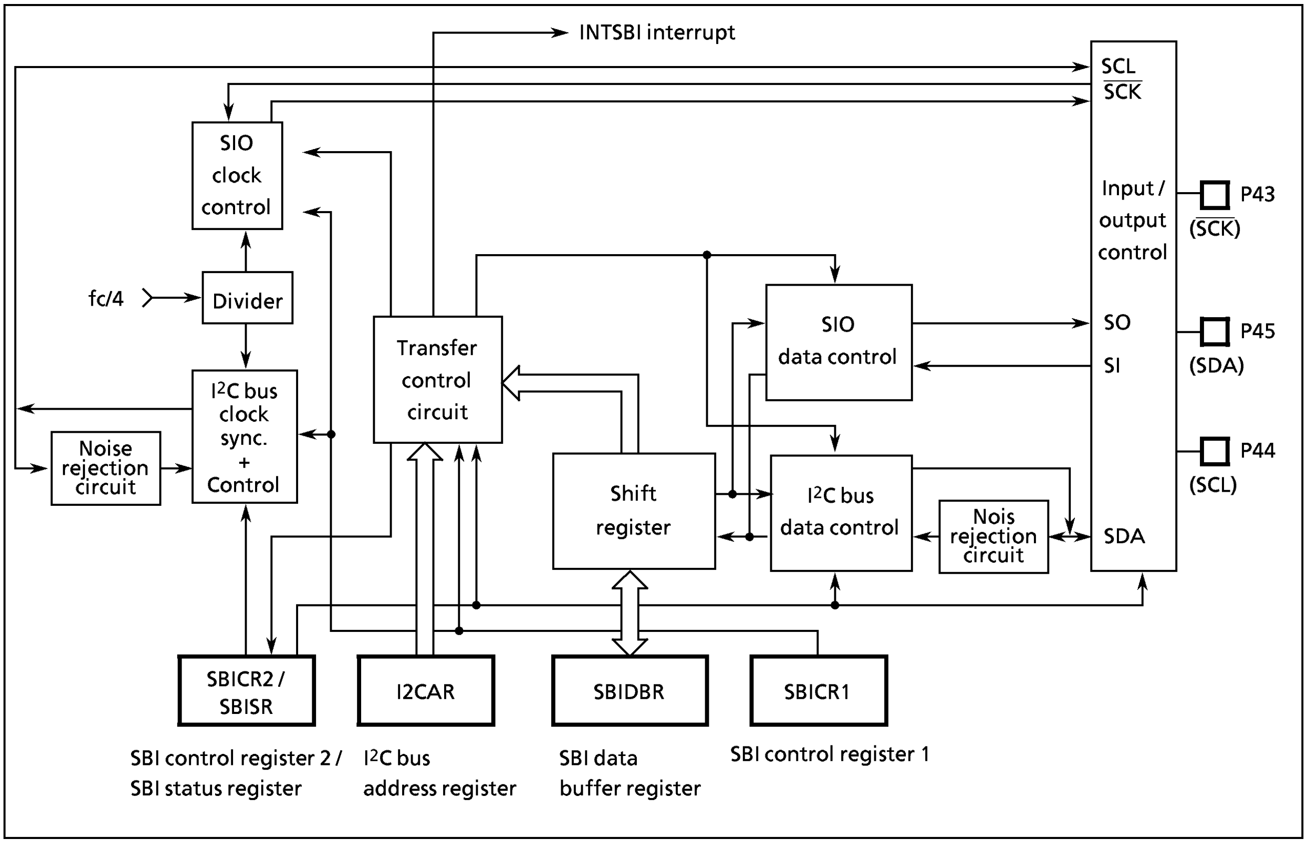


Figure 2-53. Serial Bus Interface (SBI)

2.13.2 Control

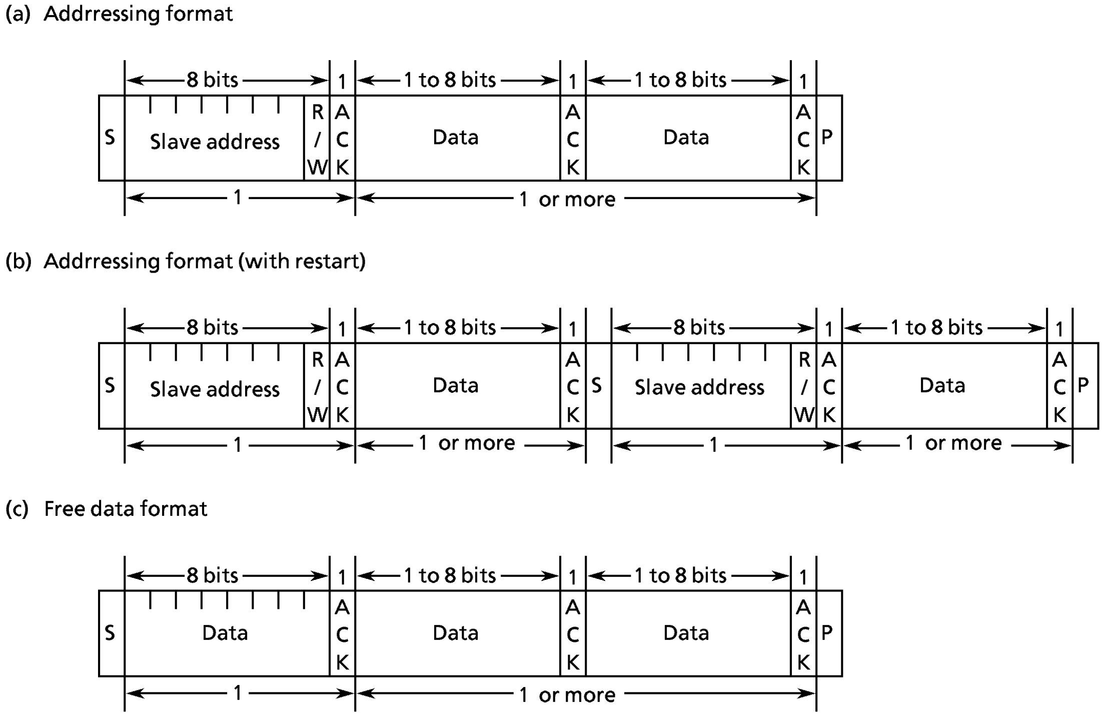
The following registers are used for control and operation status monitoring when using the serial bus interface (SBI).

- Serial bus interface control register 1 (SBICR1)
- Serial bus interface control register 2 (SBICR2)
- Serial bus interface data buffer register (SBIDBR)
- I^{2}C bus address register (I2CAR)
- Serial bus interface status register (SBISR)

The above registers differ depending on an mode to be used. Refer to Section "2.13.4 I^{2}C bus mode control" and "2.13.6 Clocked-synchronous 8-bit SIO mode control".

2.13.3 The Data Format in The I^{2}C Bus Mode

The data format when using the 88C060 in the I^{2}C bus mode are shown in figure 2-36.


(a) Addrressing format

(b) Addrressing format (with restart)

(c) Free data format

Notes : *S* : *Start condition*
R/$\overline{W}$: *Direction bit*
ACK : *Acknowledge bit*
P : *Stop condition*

Figure 2-54. Data Format in I^{2}C Bus Mode

2.13.4 I^2C Bus Mode Control

The following registers are used for control and operation status monitoring when using the serial bus interface (SBI) in the I^2C bus mode.

Serial Bus Interface Control Register 1

SBICR1
(00020$_H$)

7	6	5	4	3	2	1	0
BC			ACK	"0"	SCK		

(Initial value : 0000 *000)

		BC	ACK = 0		ACK = 1		
			Number of Clock	Bits	Number of Clock	Bits	
BC	Number of transferred bits	000 001 010 011 100 101 110 111	8 1 2 3 4 5 6 7	8 1 2 3 4 5 6 7	9 2 3 4 5 6 7 8	8 1 2 3 4 5 6 7	Write only
ACK	Acknowledge mode specification	0 : Acknowledge not returned to transmitter. 1 : Acknowledge returned to transmitter.					Read/ Write
SCK	Serial clock selection	000 : 181.8 kHz 001 : 105.3 kHz 010 : 57.1 kHz 011 : 29.9 kHz 100 : 15.3 kHz 101 : 7.72 kHz 110 : 3.88 kHz 111 : reserved	at fc = 8 MHz (Output on SCL pin)				Write only

*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 : SBICR1 include write-only resister and must not be used with any of read-modify-write instructions.

Serial Bus Interface Data Buffer Register

SBIDBR
(00021$_H$)

7	6	5	4	3	2	1	0

Read / Write

Note 1 : For writing transmitted data, start from the MSB (bit 7).

Note 2 : Cannot read the data which was written into SBIDBR, since a write data buffer and a read buffer are independent in SBIDBR. Therefore, cannot access it any of in read-modify-write instruction such as bit operate, etc.

I^2C bus Address Register

I2CAR
(00022$_H$)

7	6	5	4	3	2	1	0
Slave address							ALS
SA6	SA5	SA4	SA3	SA2	SA1	SA0	

(Initial value : 0000 0000)

SA	88C060 slave address selection		Write only
ALS	Address recognition more specification	0 : Slave address recognition 1 : Non slave address recognition	

Note : I2CAR include write-only-resister and must not be used with any of read-modify-write instructions.

Figure 2-55. Serial Bus Interface Control Register 1, Serial Bus Interface Data Buffer Register and I^2C Bus Address Register In The I^2C Bus Mode

Serial Bus Interface Control Register 2

SBICR2
(00023_H)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	SBIM		"0"	"0"

(Initial value : 0001 00**)

MST	Master / Slave selection (Write), Status monitor (Read)	0 : Slave 1 : Master	Read / Write
TRX	Transmitter / receiver selection (Write), Status monitor (Read)	0 : Receiver 1 : Transmitter	
BB	Start / stop generation (Write), I^2C bus status monitor (Read)	0 : Stop condition (Write), Bus free (Read) 1 : Start Condition (Write), Bus busy (Read)	
PIN	Cancel interrupt service request (Write), Status Monitor (Read)	0 : – (Write), Interrupt service requested (Read) 1 : Cancel interrupt service request (Write), canceled (Read)	
SBIM	Serial bus interface operating mode selection	00 : Port mode (serial bus interface output disable) 01 : SIO mode 10 : I^2C bus mode 11 : Reserved	Write only

*Note 1 : * ; don't care*

Note 2 : Switch a mode to port after confirming that the bus is free.

Note 3 : SBICR2 include write-only resister and must not be used with any of read-modify-write instructions.

SBISR
(00023_H)

7	6	5	4	3	2	1	0
MST	TRX	BB	PIN	AL	AAS	AD0	LRB

AL	Arbitration loss detection monitor	0 : – 1 : Arbitration loss detected	Read only
AAS	Slave address match detection monitor	0 : – 1 : Slave address match or "GENERAL CALL" detected	
AD0	"GENERAL CALL" detection monitor	0 : – 1 : "GENERAL CALL" detected	
LRB	Last received bit monitor	0 : Last received bit "0" 1 : Last received bit "1"	

Figure 2-56. Serial Bus Interface Control Register 2 and Serial Bus Interface Status Register In The I^2C Bus Mode

(1) Acknowledge mode specification

Set the ACK (bit 4 in SBICR1) to "1" for operation in the acknowledge mode. The 88C060 generates an additional clock pulse for an acknowledge signal when operating in the master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low level in order to generate the acknowledge signal.

Reset the ACK for operation in the non-acknowledge mode. The 88C060 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

(2) Number of transfer bits

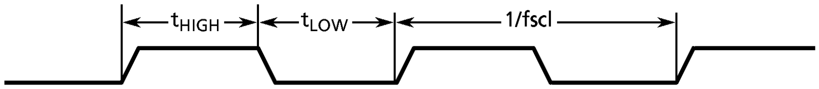
The BC (bits 7 to 5 in SBICR1) is used to select a number of bits for transmitting and receiving data.

Since the BC is cleared to "000" as 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.

(3) Serial clock

a. Clock source

The SCK (bits 2 to 0 in SBICR1) is used to select a maximum transfer frequency output from the SCL pin in the master mode.



$t_{LOW} = 2^n/fc$

$t_{HIGH} = 2^n/fc + 12/fc$

$fscl = 1/(t_{Low} + t_{HIGH})$

Note : fc ; High-frequency clock

SCK (bits 2 to 0 in the SBICR1)	n
000	4
001	5
010	6
011	7
100	8
101	9
110	10

Figure 2-57. Clock Source

b. Clock synchronization

In the I²C bus mode, 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 master device with a high-level clock pulse needs to detect the situation and implement the following procedure.

The 88C060 has a clock synchronization function for normal data transfer even when more than one master exists on a bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

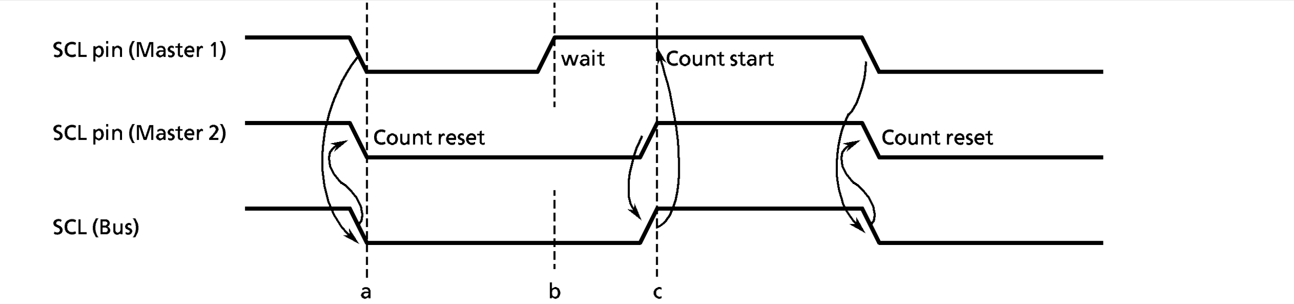


Figure 2-58. Clock Synchronization

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.
The clock pulse on the bus is deteminded 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.

(4) Slave address and address recognition mode specification
When the 88C060 is used as a slave device, set the slave address and ALS to the I2CAR. Set "0" to the ALS for the address recognition mode.

(5) Master / slave selection
Set the MST (bit 7 in SBICR2) to "1" for operating the 88C060 as a master device.
Reset the MST for operation as a slave device. The MST is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(6) Transmitter / receiver selection
Set the TRX (bit 6 in SBICR2) to "1" for operating the 88C060 as a transmitter. Reset the TRX for operation as a receiver. When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2CAR or a GENERAL CALL is received (all 8-bit data are "0" after a start condition), the TRX is set to "1" if the direction bit (R/$\overline{\text{W}}$) sent from the master devices is "1", and is cleared to "0" if the bit is "0". In the master mode, after an acknowledge signal is returned from the slave device with the hardware, the TRX is set to "0" if a transmitted direction bit is "1", and set to "1" if it is "0". When an acknowledge signal is not returned, the current condition is maintained.
The TRX is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(7) Start / stop condition generation
A start condition and the slave address and the direction bit written to the data buffer register are output on the bus by writing "1" to the MST, TRX and BB when the BB (bit 5 in SBICR2) is "0". It is necessary to set "1" to ACK beforehand.

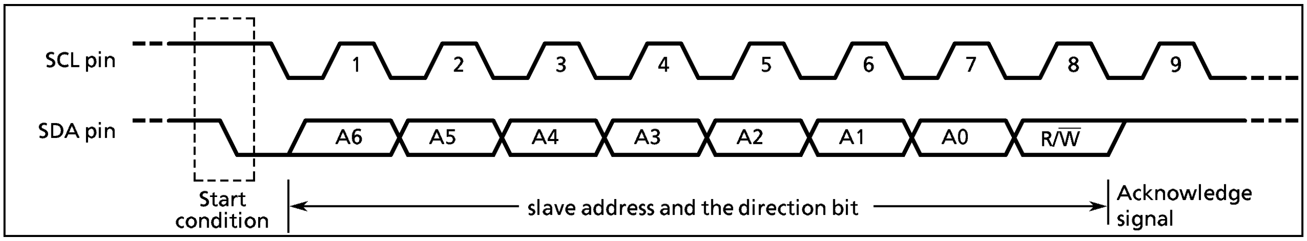


Figure 2-59. Start Condition Generation and Slave Address Generation

A stop condition is output on a bus by writing "1" to the MST and TRX when the BB is "1".

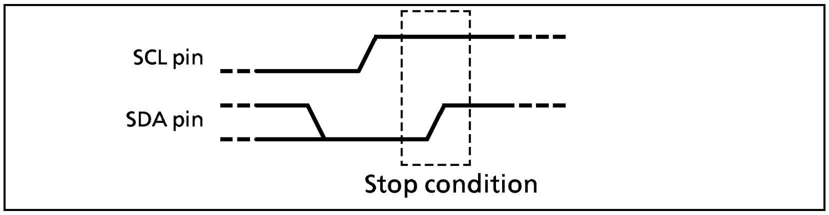


Figure 2-60. Stop Condition Generation

The bus condition can be indicated by reading the contents of the BB (bit 5 in SBISR). 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.

(8) Interrupt service request cancel
When a serial bus interface interrupt request (INTSBI) occurs, the PIN (bit 4 in SBISR) is cleared to "0". During the timer that the PIN is "0", the SCL pin is pulled down to the low level.
The PIN is cleared to "0" when 1-word of data is transmitted or received. Either writing / reading data to / 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.
In the address recognition mode (ALS = 0), the PIN is cleared to "0" when the received slave address is the same as the value set at the I2CAR or when a GENERAL CALL is received (all 8-bit data are "0" after a start condition). Although the PIN (bit 4 in SBICR2) can be set to "1" by the program, the PIN is not set to "0" when "0" is written.

(9) Serial bus interface operating mode
The SBIM (bits 3, 2 in SBICR2) is used to specify the serial bus interface operation mode. Set the SBIM to "10" when used in the I^{2}C bus mode.
Switch a mode to port after making sure that a bus is free.

(10) Arbitration lost detection monitor
Since more than one master device can exist simultaneously on a bus in the I^{2}C bus mode, 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 the bus. Master 1 and Master 2 output the same data until point "a". After Master 1 outputs "1" and Master 2, "0", the SDA line of the bus is wired AND and the SDA line is pulled down to the low level by Master 2. When the SCL line of the 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 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.

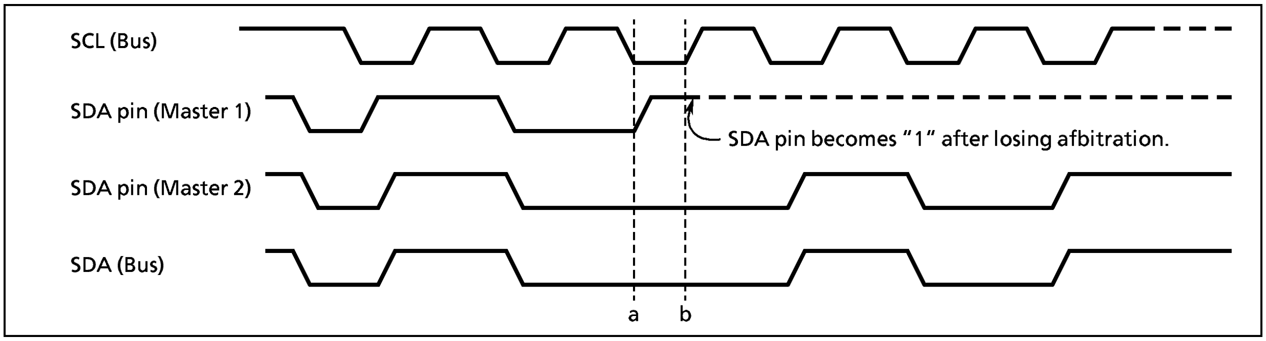


Figure 2-61. Arbitration Lost

The 88C060 compares levels of the SDA line of the bus with those of the 88C060 SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and the AL (bit 3 in SBISR) is set to "1".

When the AL is set to "1", the MST and TRX are reset to "0" and the mode is switched to a slave receiver mode. The 88C060 generates the clock pulse until data, when the AL is "1", is transmitted.

The AL is reset to "0" by writing / reading data to / from the SBIDBR or writing data to the SBICR2.

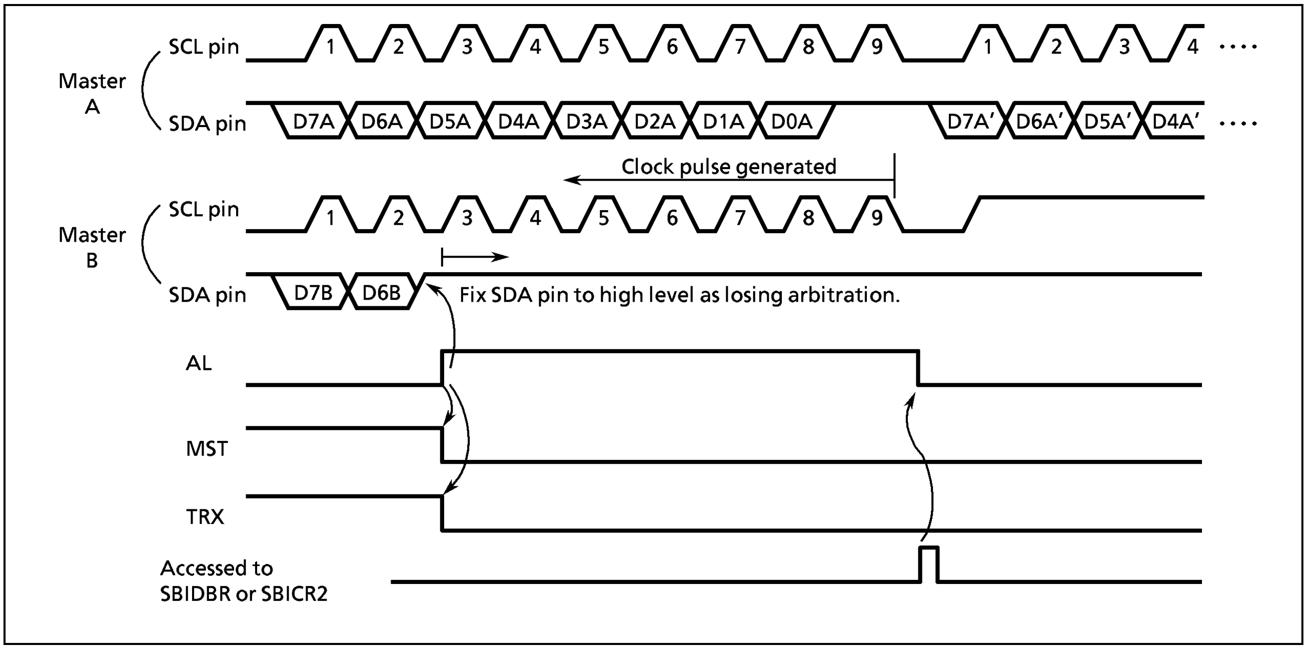

Master A
SCL pin
1 2 3 4 5 6 7 8 9 1 2 3 4 ····
SDA pin
D7A D6A D5A D4A D3A D2A D1A D0A D7A' D6A' D5A' D4A' ····
Clock pulse generated
Master B
SCL pin
1 2 3 4 5 6 7 8 9
SDA pin
D7B D6B
Fix SDA pin to high level as losing arbitration.
AL
MST
TRX
Accessed to SBIDBR or SBICR2

Figure 2-62. Example of when 88C060 is a Master B

(11) Slave address match detection monitor

The AAS (bit 2 in SBISR) is set to "1" in the slave mode, in the address recognition mode (ALS = 0), or when receiving a slave address with same value that sets a GENERAL CALL or I2CAR. When the ALS is "1", the AAS is set to "1" after receiving the first 1-word of data. The AAS is cleared to "0" by after wring / reading data to / from a data buffer register.

(12) GENERAL CALL detection monitor

The AD0 (bit 1 in SBISR) is set to "1" in the slave mode, when all 8-bit data received immediately after a start condition are "0". The AD0 is cleared to "0"when a start or stop condition is detected on the bus.

(13) Last received bit monitor

The SDA value stored at the rising edge of the SCL line is set to the LRB (bit 0 in SBISR). When the contents of the LRB are read immediately after an INTSBI interrupt request is generated in the acknowledge mode, and ACK signal is read.

2.13.5 Data Transfer in I^2C Bus Mode

(1) Device initialization
Set the ACK and SCK in the SBICR1, specify "0" to bus 7 to 5 and 3.
Set a slave address and the ALS (ALS = "0" when an addressing format) to the I2CAR.
For specifying the setting to a slave receiver mode, assign "0" to the MST, TRX, and BB in the SBICR2 ; "1" to the PIN, "10" to the SBIM, and "0" to bits 1 and 0.

(2) Start condition, slave address generation
Observe a bus free status (when BB = 0).
Set the ACK to "1" and specify a slave address and a direction bit to be transmitted to the SBIDBR.
When writing "1" to the MST, TRX, and BB, the slave address and the direction bit which are set to the SBIDBR and the start condition are output on the bus. A slave device receives these data and pulls down the SDA line of the bus to the low level at the acknowledge signal timing. An INTSBI interrupt request occurs at the 9th falling edge of the 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.

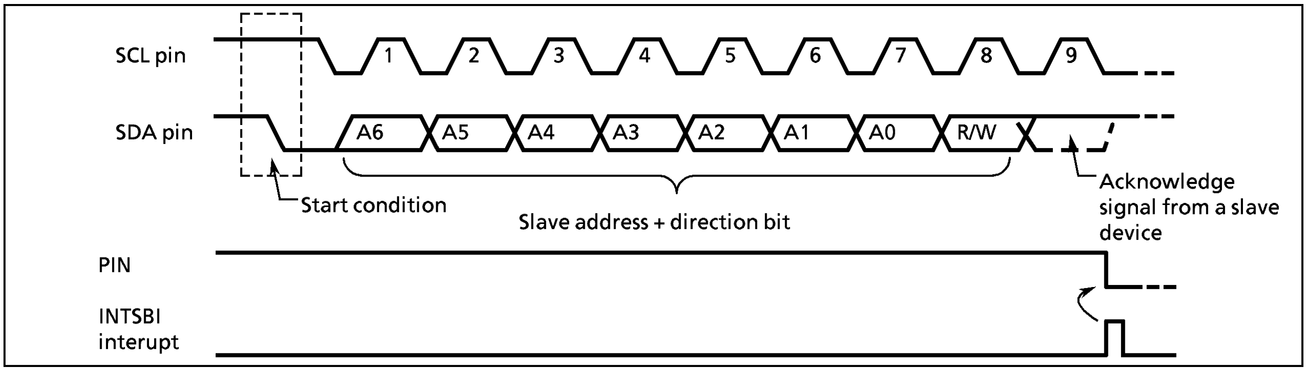


Figure 2-63. Start Condition Generation and Slave Address Transfer

(3) 1-word data transfer
Test the MST by the INTSBI interrupt process after a 1-word data transfer is concluded, and determine whether the mode is a master or slave.

a. When the MST is "1" (Master mode)
Test 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 request new data. When the next transmitted data is other than 8 bits, set the BC 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 new 1-word of data from the SCL pin, and then the 1-word data is transmitted from the SDA pin. After the data is transmitted, an INTSBI interrupt request occurs. The PIN becomes "0" and the SCL pin is pulled down to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the LRB test above.

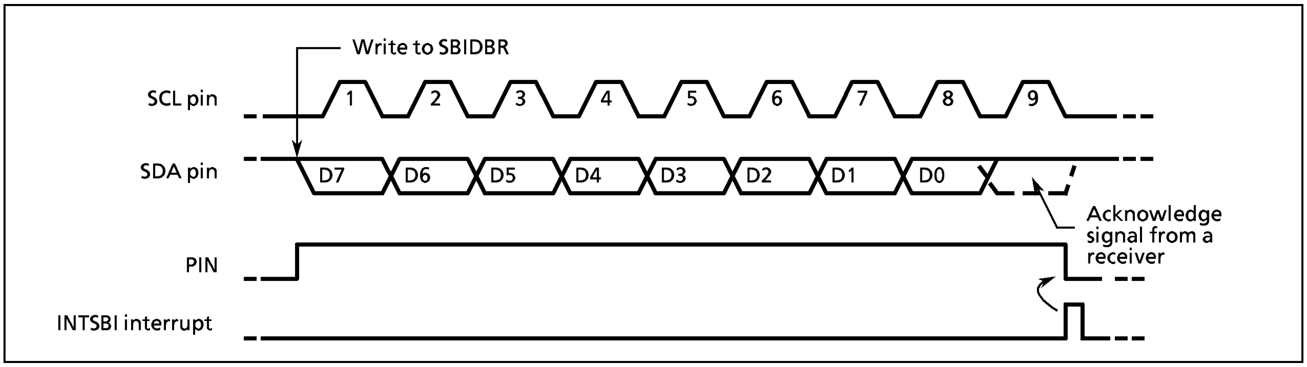


Figure 2-64. Example of when BC = "000", ACK = "1"

② When the TRX is "0" (receiver mode)
When the next transmitted data is other than of 8 bits, set the BC is set again. Set the ACK to "1" and read the received data from the SBIDBR (data which is read immediately after a slave address is sent is undefined). After the data is read, the PIN becomes "1". The 88C060 outputs a serial clock pulse to the SCL to transfer new 1-word of data and sets the SDA pin to "0" at the acknowledge signal timing.
An INTSBI interrupt request then occurs and the Pin becomes "0". The SCL pin is pulled down to the low level. The 88C060 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBIDBR.

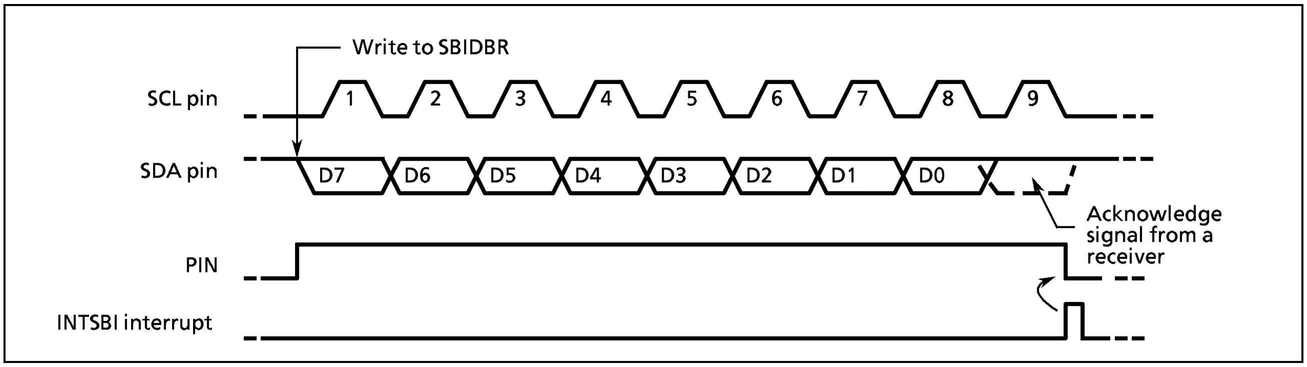


Figure 2-65. Example of when BC = "000", ACK = "1"

In order to terminate transmitting data to a transmitter, reset the ACK before reading data which is 1 word before the last data to be received. The last data does not generate a clock pulse for the acknowledge signal. After the data transmitted and an interrupt request has occurred, set the BC to "001" and read the data. The 88C060 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line of the 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 an and interrupt request has occurred, the 88C060 generates a stop condition an terminates data transfer.

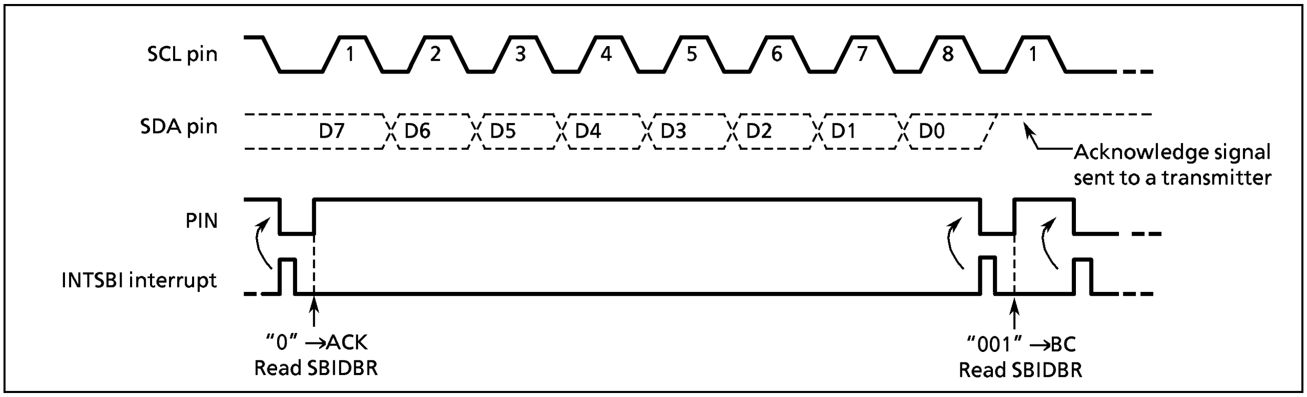


Figure 2-66. Termination of Data Transfer in Master Receiver Mode

b. When the MST is "0" (Slave mode)

In the slave mode, an INTSBI interrupt request occurs when the 88C060 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete after matching a received slave address. In the master mode, the 88C060 operates in a slave mode if it is losing arbitration. An INTSBI interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBI interrupt request occurs the PIN (bit 4 in SBICR2) is reset, and the SCL pin is pulled down to the low level. Either reading / writing from / to the SBIDBR or setting the PIN to "1" releases the SCL pin after taking tLOW time.

In the slave mode, the 88C060 operates either in normal slave mode or in slave mode after losing arbitration.

The 88C060 tests the AL (bit 3 in SBISR), the TRX (bit 6 in SBISR), the AAS (bit 2 in SBISR), and the AD0 (bit 1 in SBISR) and implements processes according to conditions listed in table 2-19.

Table 2-9. Operation in The Slave Mode

TRX	AL	AAS	AD0	Conditions	Process
1	1	1	0	The 88C060 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	1	In the slave receiver mode, the 88C060 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" sine the receiverdoes not request further data. Then, reset the TRX to release the bus. If the LRB is set to "0", set the number of bits in a word to the BC and write transmitted data to the SBIDBR since the receiver requests further data.
0	1	1	1/0	The 88C060 loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL of which the value of the direction bit sent from another master is "0".	Read the SBIDBR for setting the PIN to "1" (reading dummy data) or write "1" to the PIIN.
		0	0	The 88C060 loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, the 88C060 receives a slave address or general CALL of which the value of the direction bit sent from the master is "0".	
		0	1/0	In the slave receiver mode, the 88C060 terminates receiving of 1-word data.	Set the number of bits in a word to the BC and read received data from the SBIDBR.

(4) Stop condition generation

Writing "1" to the MST, TRX, and PIN, and "0" to the BB generates a stop condition on the bus.

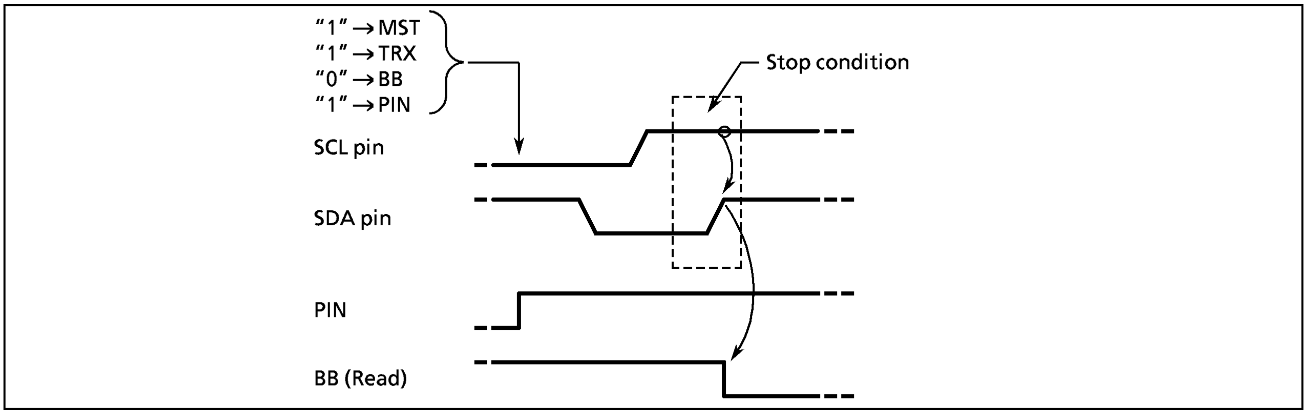


Figure 2-67. Stop Condition Generation

(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 when the 88C060 is in the master mode.

Specify "0" to the MST, TRX, and BB and "1" to the PIN and release the bus. The SDA pin retains the high level and the SCL pin is released. Since a stop condition is not generated on the bus, the 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 the 88C060 is released. Test the LRB until it becomes "1" to check that the SCL line of the bus is not pulled down to the low level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure (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 the bus is free until the time to generate the start condition.

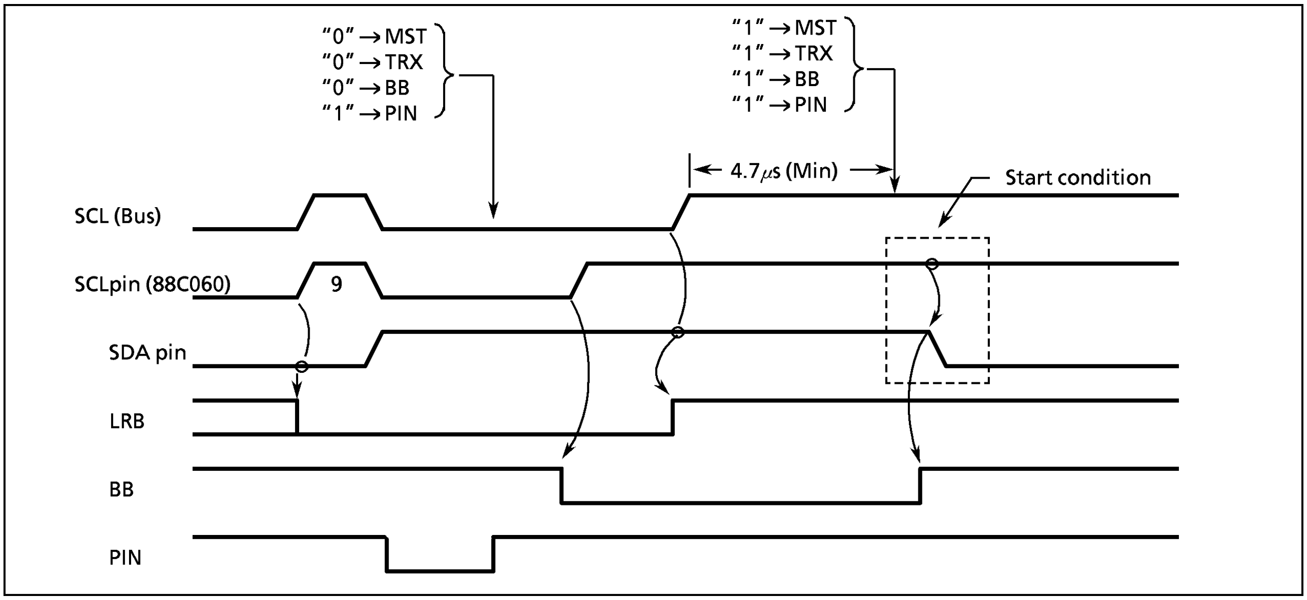


Figure 2-68. Timing Diagram when Restarting The 88C060

2.13.6 Clocked-synchronous 8-bit SIO Mode Control

The following registers are used for control and operation status monitoring when using the serial bus interface (SBI) in the clocked-synchronous 8-bit SIO mode.

Serial Bus Interface Control Register 1

SBICR1
(00020$_H$)

7	6	5	4	3	2	1	0
SIOS	SIO INH	SIOM		"0"	SCK		

(Initial value : 0000 *000)

SIOS	Indicate transfer start / stop	0 : Stop 1 : Start	Write only
SIOINH	Continue / abort transfer	0 : Continue transfer 1 : Abort transfer (automatically cleared after abort)	
SIOM	Transfer mode select	00 : 8-bit transmit mode 01 : reserved 10 : 8-bit transmit / receive mode 11 : 8-bit receive mode	
SCK	Serial clock select	000 : $fc/2^5$ (250 kHz) 001 : $fc/2^6$ (125 kHz) 010 : $fc/2^7$ (62.5 kHz) 011 : $fc/2^8$ (31.25 kHz) at fc = 8 MHz (Output on $\overline{\text{SCK}}$ pin) 100 : $fc/2^9$ (15.62 kHz) 101 : $fc/2^{10}$ (7.81 kHz) 110 : $fc/2^{11}$ (3.90 kHz) 111 : External clock (input from $\overline{\text{SCK}}$ pin)	

*Note 1 : fc ; High-frequency clock [Hz], * ; don't care*
Note 2 : Set the SIOS to "0" when setting the transfer mode or serial clock.
Note 3 : SBICR1 is a write-only register and must not be used with any of read-modify-write instructions.

Serial Bus Interface Data Register 1

SBIDBR
(00021$_H$)

7	6	5	4	3	2	1	0

Read / Write

Note : Cannot read the data which was written into SBIDBR, since a write data buffer and read buffer are independent in SBIDBR. Therefore, cannot access it any of in read-modify-write instruction such as bit operate, etc.

Serial Bus Interface Control Register 2

SBICR2
(00023$_H$)

7	6	5	4	3	2	1	0
"0"	"0"	"0"	"1"	SBIM		"0"	"0"

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

SBIM	Serial bus interface operation mode selection	00 : Port mode (serial bus interface output disable) 01 : SIO mode 10 : I^2C bus mode 11 : reserved	Write only

*Note 1 : * ; don't care*
Note 2 : Switch a mode to port after data transfer is complete.
Note 3 : SBICR2 is a write-only register and must not be used with any of read-modify-write instructions.

Serial Bus Interface Status Register

SBISR
(00023$_H$)

7	6	5	4	3	2	1	0
"1"	"1"	"1"	"1"	SIOF	SEF	"1"	"1"

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

Figure 2-69. Control Register / Data Buffer Register / Status Register in SIO Mode

(1) Serial clock

a. Clock source

The SCK (bits 2 to 0 in SBICR1) is used to select the following functions.

① Internal clock

In an internal clock mode, and of seven frequencies can be selected. The serial clock is output to the outside on the $\overline{SCK}$ pin. The $\overline{SCK}$ pin becomes a high level when data transfer starts. When writing (in the transmit mode) or reading (in the receive mode) data cannot 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 complete.

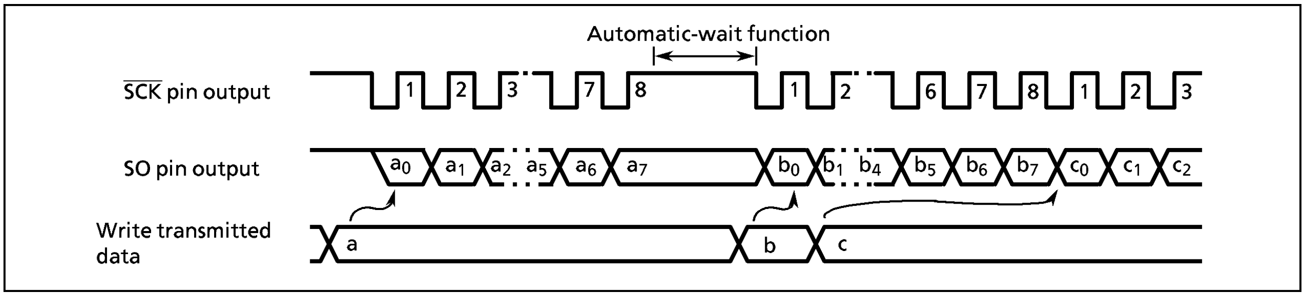


Figure 2-70. Automatic Wait Function

② External (SCK = "111")

An external clock supplied to the $\overline{SCK}$ pin is used as the serial clock. In order to ensure shift operation, a pulse width of at least 4 machine cycles is required for both high and low levels in the serial clock. The maximum data transfer frequency is 390.6kHz (fc = 12.5 MHz).

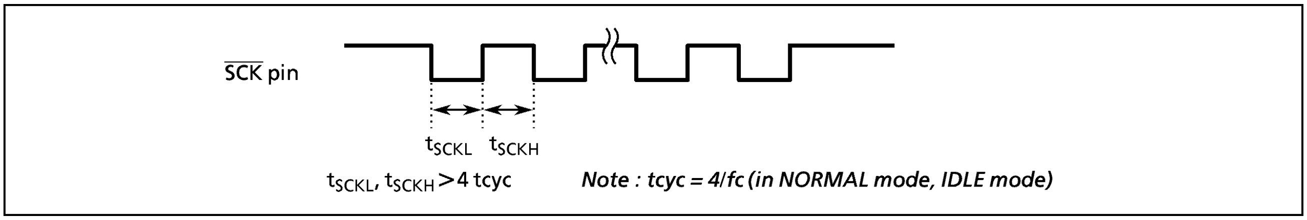

$\overline{SCK}$ pin
t_{SCKL}
t_{SCKH}
$t_{SCKL}, t_{SCKH} > 4$ tcyc
Note : tcyc = 4/fc (in NORMAL mode, IDLE mode)

Figure 2-71. The Maximum Data Transfer Frequency in The External Clock Input

b. Shift edge

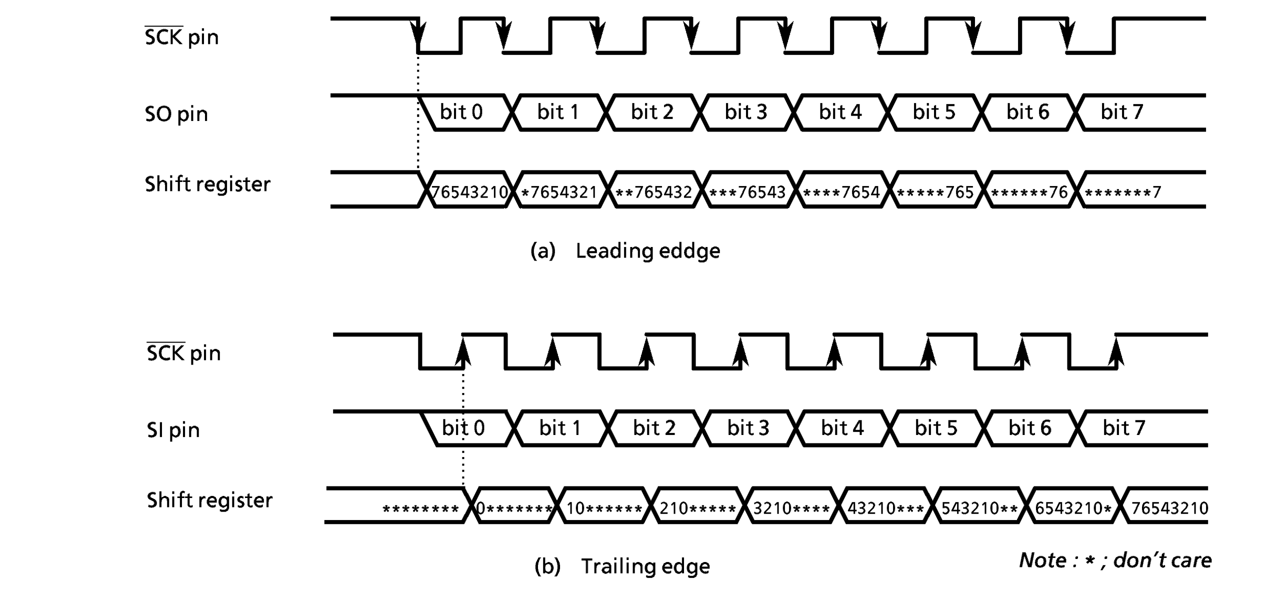
The leading edge is used to transmit data, and the trailing edge is used to receive data.

① Leading edge

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

② Trailing edge

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



(a) Leading eddge

(b) Trailing edge

*Note : * ; don't care*

Figure 2-72. Shift Edge

(2) Transfer mode

The SIOM (bits 5 and 4 in SIO1CR) is used to select a transmit, receive, or transmit / receive mode.

a. <u>8-bit transmit mode</u>

Set a control register to a transmit mode and write data to the SBIDBR.

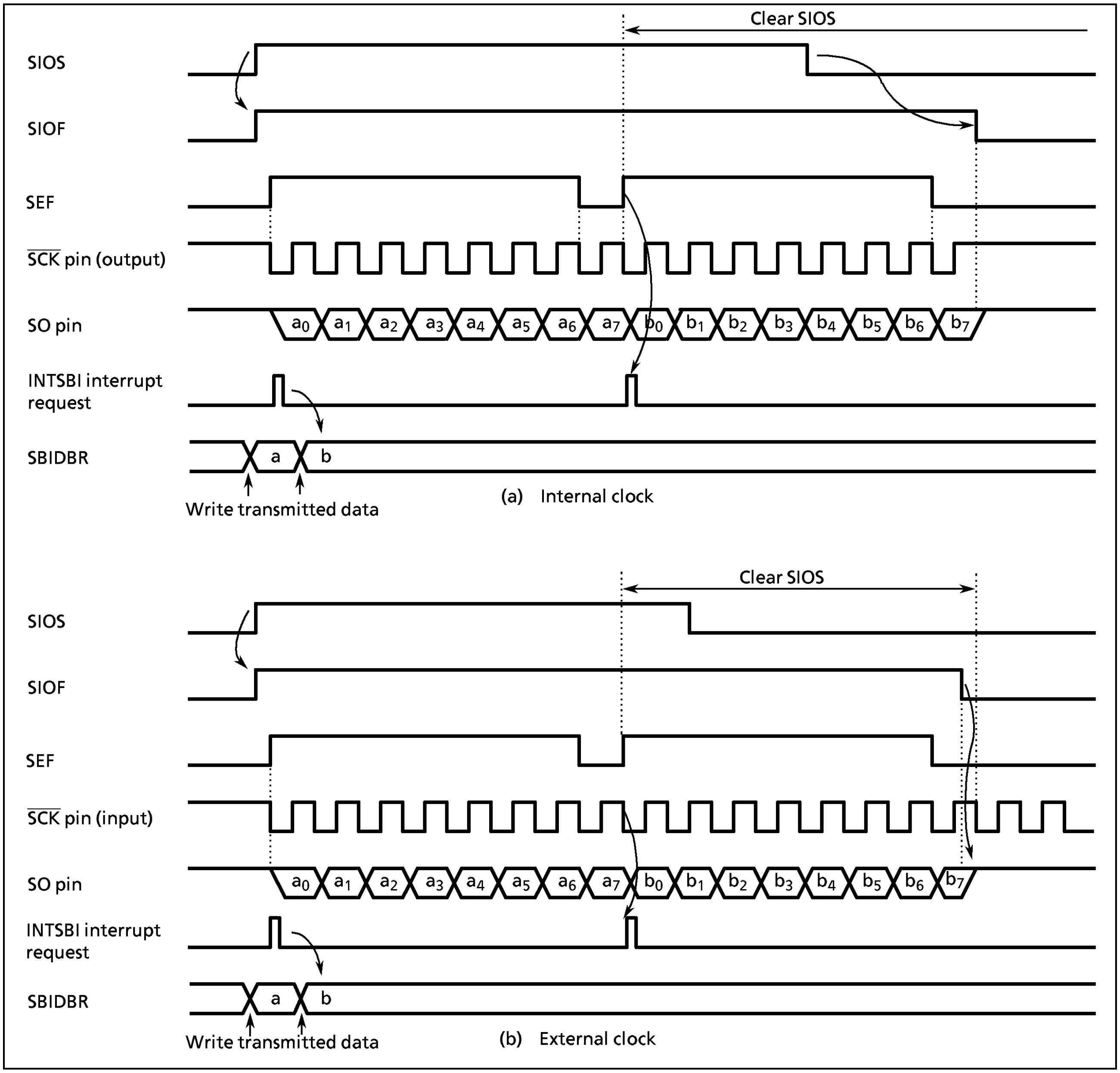
After the data is written, set the SIOS to "1" to start data transfer. The transmitted data is transferred from the SBIDBR to the shift register and output to the SO pin in synchronous with the serial clock, starting from the least significant bit (LSB). When the data is transferred to the shift register, the SBIDBR becomes empty. The INTSBI (buffer empty) interrupt request is generated to request new data.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new data is written, automatic-wait function is canceled.

When the external clock is used, data should be written to the SBIDBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to the SBIDBR by the interrupt service program.

Transmitting data is ended by cleaning the SIOS to "0" by the buffer empty interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the SIOF (bit 3 in the SBISR) to be sensed. The SIOF is cleared to "0"when transmitting is complete. When the SIOINH is set, transmitting data stops. The SIOF turns "0".

When the external clock is used, it is also necessary to clear the SIOS to "0" before new data is shifted ; otherwise, dummy data is transmitted and operation ends.


Clear SIOS
SIOS
SIOF
SEF
$\overline{SCK}$ pin (output)
SO pin
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
INTSBI interrupt request
SBIDBR
a
b
Write transmitted data
Clear SIOS
SIOS
SIOF
SEF
$\overline{SCK}$ pin (input)
SO pin
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7 b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
INTSBI interrupt request
SBIDBR
a
b
Write transmitted data

(a) Internal clock

(b) External clock

Figure 2-73. Transfer Mode

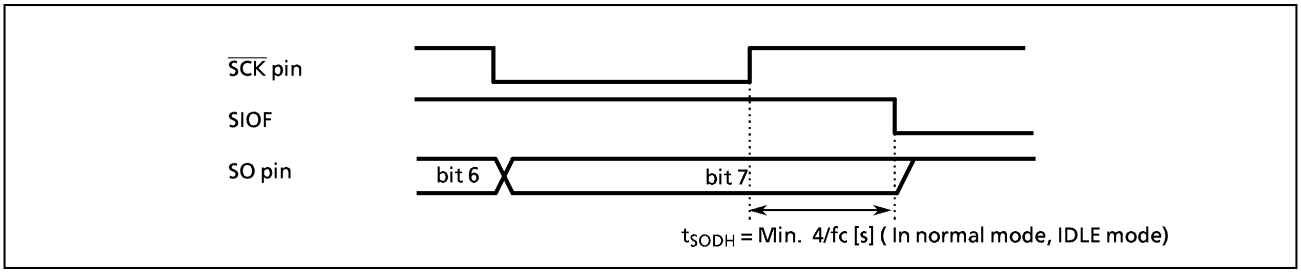


Figure 2-74. Transmitted Data Hold Time at End of Transmit

b. 8-bit receive mode

Set a control register to a receive mode and the SIOS to "1" for switching to a receive mode. Data is received from the SI pin to the shift register in synchronous with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBIDBR. The INTSBI (buffer full) interrupt request is generated to request of reading the received data. The data is then read from the SBIDBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and automatic-wait function will be initiated until the received data is read from the SBIDBR.

When the external clock is used, since shift operation is synchronized with the clock pulse provided externally, the received data should be read before new data is transferred to the SBIDBR. If the received data is not read, further data to be received is canceled. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read.

Receiving data is ended by clearing the SIOS to "0" by the buffer full interrupt service program or setting the SIOINH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm if data is surely received by the program, set the SIOF (bit 3 in SBIDBR) to be sensed. The SIOF is cleared to "0" when receiving is complete. After confirming that receiving has ended, the last data is read. When the SIOINH is set, receiving data stops. The SIOF turns "0" (the received data becomes invalid, therefore no need to read it).

*Note : When the transfer mode is switched, the **SBIDBR** contents are lost. In case that the mode needs to be switched, conclude receiving data by clearing the **SIOS** to "0", read the last data, and then switch the mode.*

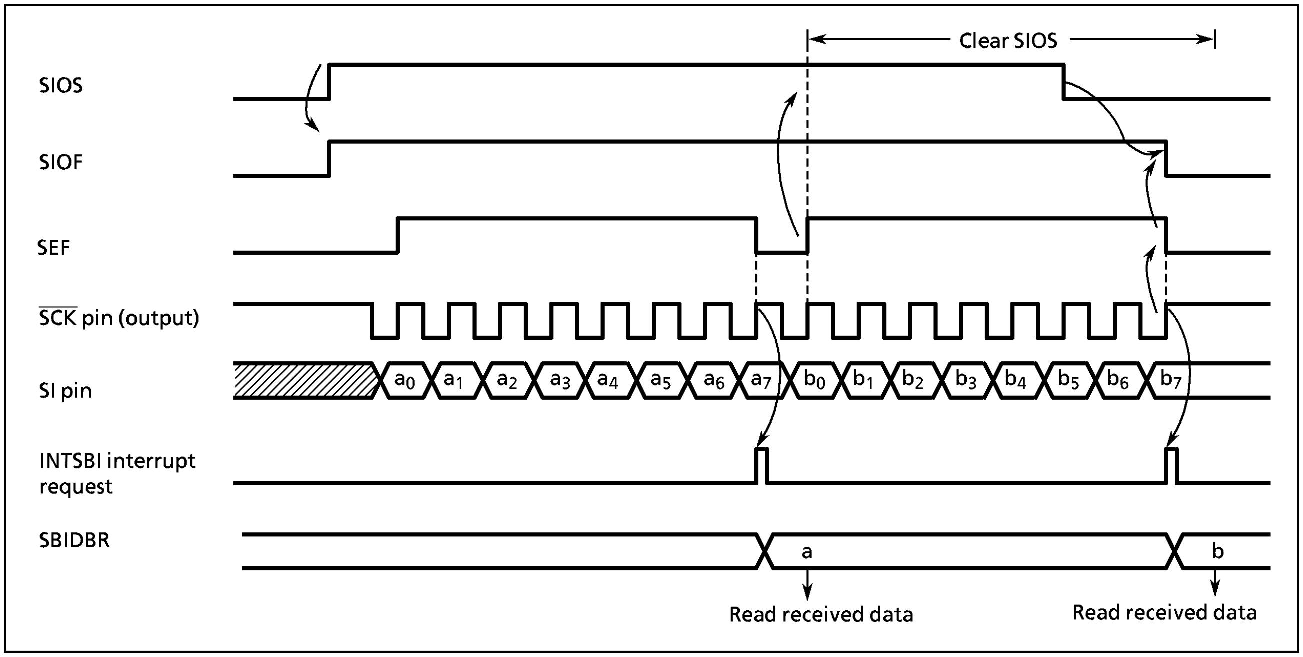


Figure 2-75. Receive Mode (Example: Internal clock)

c. 8-bit transmit / receive mode

Set a control register to a transmit / receive mode and write data to the SBIDBR. After the data is written, set the SIOS to "1" to start transmitting / receiving. When transmitting, the data is output from the SO pin on the leading edges in synchronous with the serial clock, starting from the least significant bit (LSB). When receiving, the data is input to the SI pin on the trailing edges of the serial clock. 8-bit data is transferred from the shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the shift register to the SBIDBR, and the INTSBI interrupt request occurs. The interrupt service program reads the received data from the data buffer register and writes data to be transmitted. The SBIDBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, automatic-wait function is initiated until received data is read and next data is written.

When the external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before new shift operation is executed. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when received data is read and transmitted data is written.

Transmitting / receiving data is ended by cleaning the SIOS to "0" by the INTSBI interrupt service program or setting the SIONH to "1". When the SIOS is cleared, received data is transferred to the SBIDBR in complete blocks. The transmit / receive mode ends when the transfer is complete. In order to confirm if data is surely transmitted / received by the program, set the SIOF (bit 3 in SBISR) to be sensed. The SIOF becomes "0" after transmitting / receiving is complete. When the SIONH is set, transmitting / receiving data stops. The SIOF turns "0".

Note : When the transfer mode is switched, the SBIDBR contents are lost. In case that the mode needs to be switched, conclude transmitting / receiving data by clearing the SIOS to "0", read the last data, and then switch the transfer mode.

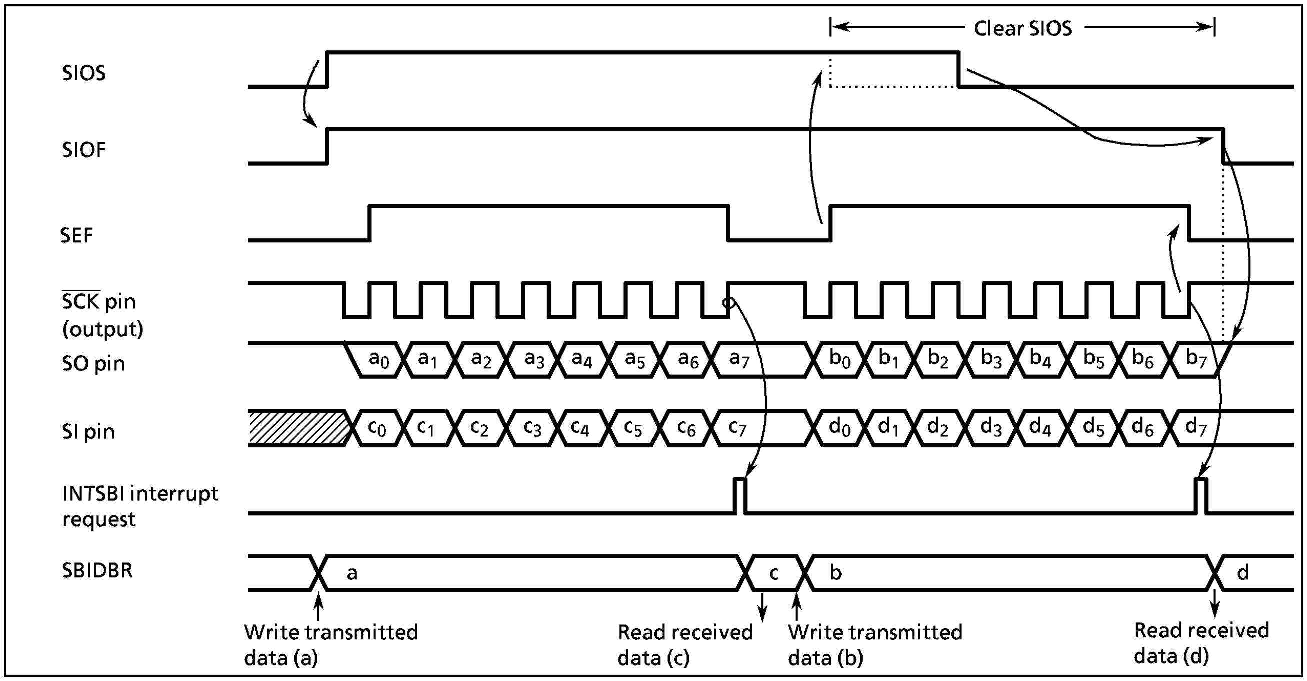

Clear SIOS
SIOS
SIOF
SEF
SCK pin (output)
SO pin
a0 a1 a2 a3 a4 a5 a6 a7 b0 b1 b2 b3 b4 b5 b6 b7
SI pin
c0 c1 c2 c3 c4 c5 c6 c7 d0 d1 d2 d3 d4 d5 d6 d7
INTSBI interrupt request
SBIDBR
a
c
b
d
Write transmitted data (a)
Read received data (c)
Write transmitted data (b)
Read received data (d)

Figure 2-76. Transmit / Receive Mode (Example : Internal clock)

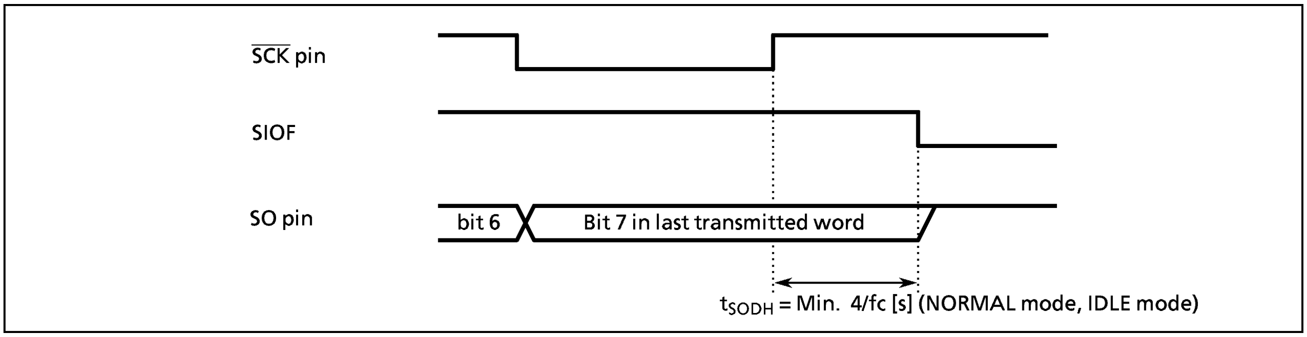

$\overline{SCK}$ pin
SIOF
SO pin
bit 6
Bit 7 in last transmitted word
t_{SODH} = Min. 4/fc [s] (NORMAL mode, IDLE mode)

Figure 2-77. Transmitted Data Hold Time at End of Transmit / Receive

2.14 10-bit A/D Converter (ADC)

The 88C060 has a 10-bit successive approximation type A/D converter.

2.14.1 Configuration

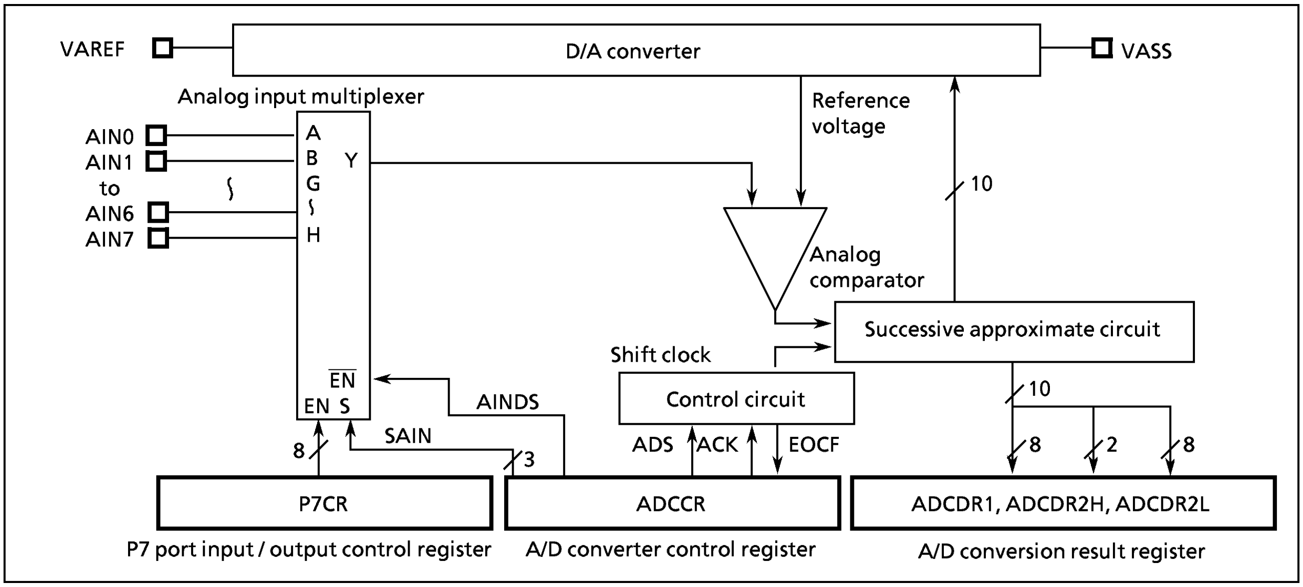


Figure 2-78. A/D Converter (ADC)

2.14.2 Control

The A/D converter is controlled by the A/D converter control register (ADCCR) and port P7 input / output control register (P7CR). The operating state of the A/D converter is confirmed by reading EOCF in ADCCR. The A/D conversion value is confirmed by reading the contents of the A/D conversion value registers such as ADCDR1, or ADCDR2H and ADCDR2L.

A/D Converter Control Register

	7	6	5	4	3	2	1	0
ADCCR (0000E_H)	EOCF	ADS	ACK	AINDS		SAIN		

(Initial value : 0000 *000)

SAIN	Analog input channel select	000 : Selects AIN0 001 : Selects AIN1 010 : Selects AIN2 011 : Selects AIN3 100 : Selects AIN4 101 : Selects AIN5 110 : Selects AIN6 111 : Selects AIN7	R/W
AINDS	Analog input control	0 : Analog input enable 1 : Analog input disable	
ACK	Conversion time select	0 : 184/fcgck 1 : 736/fcgck	
ADS	A/D Conversion start	0 : – 1 : A/D conversion start	
EOCF	A/D Conversion end flag	0 : Under conversion or before conversion 1 : End of conversion	Read only

Note 1 : The analog input channel must be selected when the A/D conversion is stopped.
Note 2 : The ADS is automatically cleared to "0" after starting the A/D conversion.
Note 3 : The EOCF is cleared to "0" by reading the A/D conversion registers such as ADCDR1, ADCDR2H, or ADCDR2L.
Note 4 : The EOCF is read-only, so the written data is invalid.
*Note 5 : fc ; High-frequency clock [Hz] , * ; don't care*

Figure 2-79. A/D Converter Control Register

A/D Conversion Result Register

	7	6	5	4	3	2	1	0	
ADCDR1 (0000F$_H$)	DATA9	DATA8	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	Read only

	7	6	5	4	3	2	1	0	
ADCDR2H (00025$_H$)	0	0	0	0	0	0	DATA9	DATA8	Read only

	7	6	5	4	3	2	1	0	
ADCDR2L (00024$_H$)	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	Read only

Port P7 Input / Output Control

	7	6	5	4	3	2	1	0	
P7CR (0000D$_H$)									(Initial value : 0000 0000)

		ADCCR	P7CR		
		AINDS	0	1	
P7CR	I/O control for port P7	0	Input port except for the analog input selected by SAIN	Output port except for the analog input selected by SAIN	Write only
		1	Input port	Output port	

Figure 2-80. A/D Conversion Register and Port P7 Input / Output Control Register

2.14.3 A/D converter operation

The high side of an analog reference voltage is applied to the VAREF pin, and the low side is applied to the VASS pin. Dividing a reference voltage between VAREF and VASS to the voltage corresponding to a bit by a rudder resistance and comparing it with the analog input voltage converts the A/D.

(1) Starting A/D conversion

First, select one of the analog input channels (AIN7 to AIN0) with the SAIN (bit 2 to 0 in ADCCR). Clear the AINDS (bit 4 in ADCCR) to "0" and set the corresponding the P7 input control (P7CR) to "1" for analog input. The pins to be not used as the analog input can be used as normal input / output ports. During conversion, the input / output instruction is performed to any pins to remain A/D accuracy.
The A/D conversion time is set by the ACK (bit 5 in ADCCR).
The A/D operation is started by setting the ADS (bit 6 in ADCCR) to "1".
The A/D conversion time needs 184/fcgck [s] (46 machine cycles) at ACK = 0 after starting A/D conversion until the conversion results are set to the ADCDR1 or the ADCDR2H, ADDR2L. For example, the A/D conversion time takes 43.7 μs at fcgck = fc = 4.2 MHz. Ending the A/D conversion sets the EOCF (bit 7 in ADCCR) to "1", which indicates the conversion end.
When the ADS is set to "1" during A/D conversion, the A/D conversion is initialized, restarted converting from the first.

Table 2-20. A/D Conversion Time (Example)

ACK	A/D conversion time [μs]			
	fc = 4.2 MHz		fc = 12.5 MHz	
	fcgck = fc	fcgck = fc/2	fcgck = fc	fcgck = fc/2
0	43.7	87.4	–	29.4
1	174.8	350	58.9	117.7

(2) Reading A/D converted value

The converted value stored in the A/D conversion registers such as ADCDR1, ADCDR2H, or ADCDR2L must be read out after the conversion is ended (EOCF = 1). The EOCF is automatically cleared to "0" by reading the converted value. When the converted value is read out during A/D conversion, the undefined value is read out.

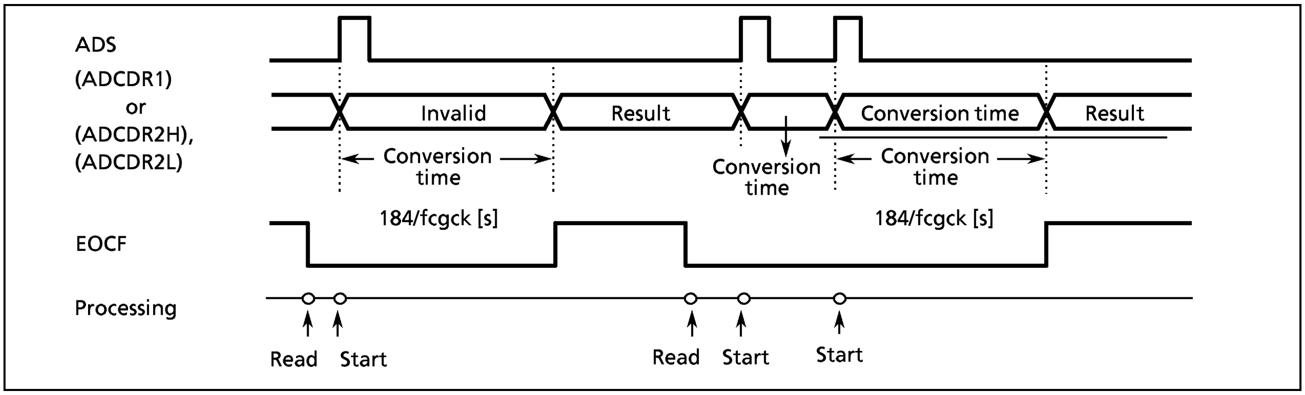


Figure 2-81. A/D Conversion Timing Chart

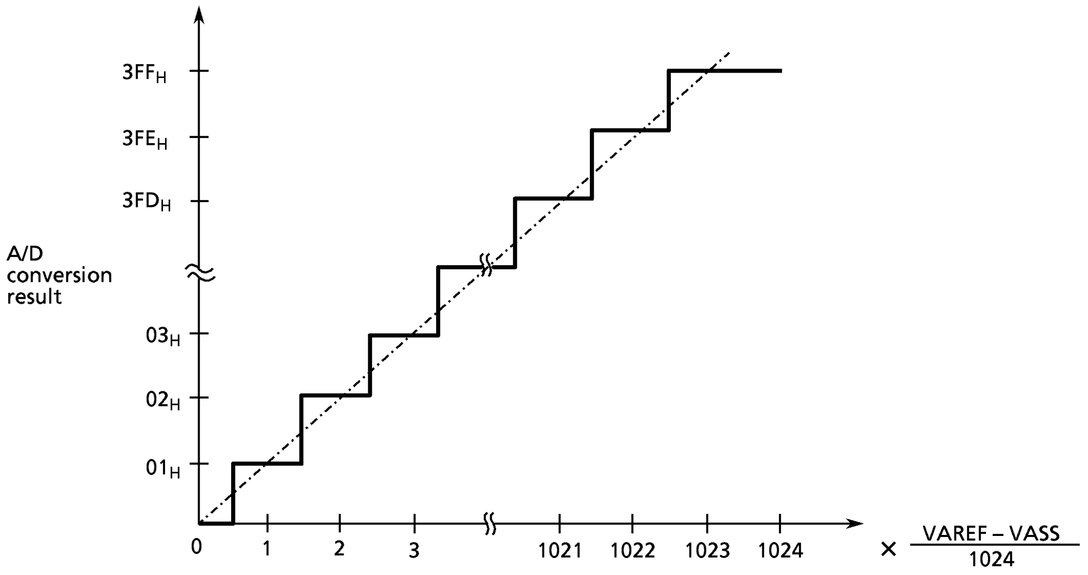
(3) A/D conversion in STOP mode

When the MCU places in the STOP mode during the A/D conversion, the conversion is stopped and the ADCDR contents become indefinite. The EOCF remains to be cleared to "0" after returning to the STOP mode. However, if the STOP mode is started after the end of conversion (EOCF = 1), the ADCDR contents are held.

Example 1 : The A/D conversion starts after the AIN4 pin is selected as the conversion time and the analog input channel. Confirming the EOCF, the converted value is read out, and the upper 2 bits data is stored to address 0009E$_H$ in RAM and the lower 8 bits data is stored to address 0009F.

```
         ; AIN SELECT
             LD (ADCCR), 00100100B      ; Selects the conversion time and AIN4
         ; A/D CONVERT START
             SET     (ADCCR). 6         ; ADS = 1
SLOOP :  TEST   (ADCCR). 7              ; EOCF = 1 ?
         JRS     T, SLOOP
         ; RESULT DATA READ
         LD (9EH), (ADCDR2H)
         LD (9DH), (ADCDR2L)
```

The analog input voltage is corresponded to the 10-bit digital value converted by the A/D as shown in figure 2-82.



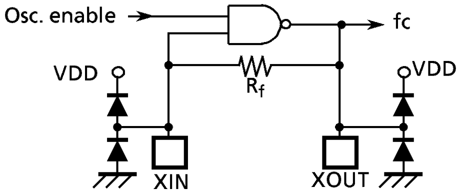
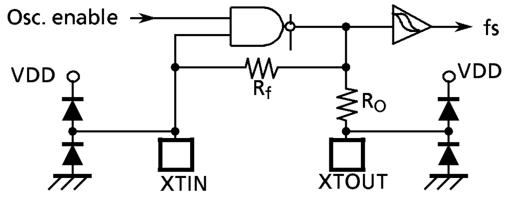
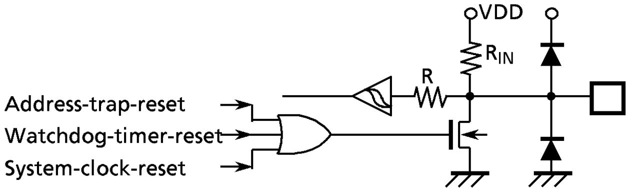
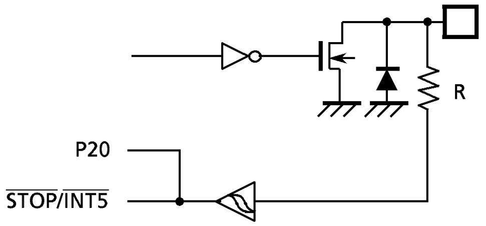
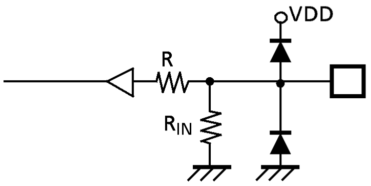
Analog Input voltage vs. A/D conversion result (typ.)

Figure 2-82. Analog Input Voltage and A/D Conversion Result (typ.)

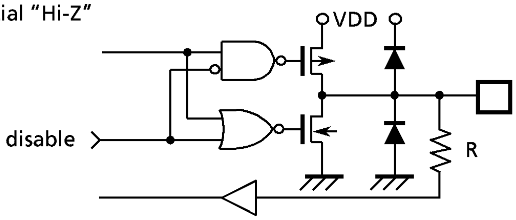
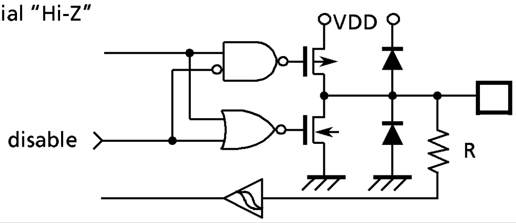
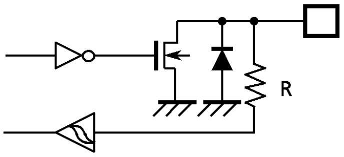
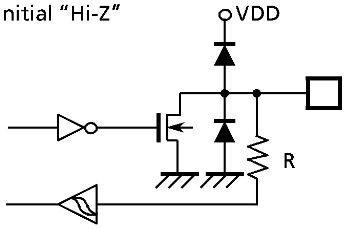
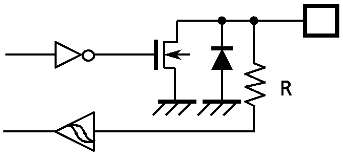
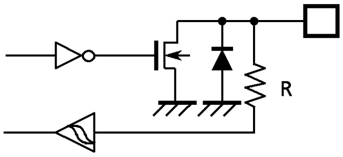
INPUT / OUTPUT CIRCUIT

(1) Control pins

The input / output circuits of the 88C060 are shown below.

Port	I/O	Input / Output circuitry and Code	Remarks
XIN XOUT	I/O		High-frequency resonator connect pin R_f = 1.2 MΩ (typ.)
XTIN XTOUT	I/O	NM1 Refer to P2 port NM2  Osc. enable fs VDD R_f R_O VDD XTIN XTOUT	Low-frequency resonator connect pin R_f = 6 MΩ (typ.) R_O = 220 kΩ (typ.)
$\overline{\text{RESET}}$	I/O	 VDD R_{IN} R Address-trap-reset Watchdog-timer-reset System-clock-reset	Sink open drain output Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{STOP}}$ / $\overline{\text{INT5}}$	Input	 R P20 $\overline{\text{STOP}}$/$\overline{\text{INT5}}$	Hysteresis input R = 1 kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

(2) Input / Output ports

<table>
<tr><th>Port</th><th>I/O</th><th colspan="2">Input / Output circuitry and Code</th><th>Remarks</th></tr>
<tr><td>P0
P7</td><td>I/O</td><td colspan="2">initial "Hi-Z"
 </td><td>Try state input / output
R = 1 kΩ (typ.)</td></tr>
<tr><td>P1</td><td>I/O</td><td colspan="2">initial "Hi-Z"
 </td><td>Try state input / output
Hysteresis input
R = 1 kΩ (typ.)</td></tr>
<tr><td rowspan="2">P2</td><td rowspan="2">I/O</td><td>P20, P23</td><td>P21, P22</td><td rowspan="2">Sink open drain output
Hysteresis input
R = 1 kΩ (typ.)</td></tr>
<tr><td>initial "Hi-Z"
 </td><td>initial "Hi-Z"
 </td></tr>
<tr><td>P3</td><td>I/O</td><td colspan="2">initial "Hi-Z"
 </td><td>Sink open drain output
Hysteresis input
Large current output
R = 1 kΩ</td></tr>
<tr><td>P4</td><td>I/O</td><td colspan="2">initial "Hi-Z"
 </td><td>Sink open drain output
Hysteresis input
R = 1 kΩ</td></tr>
</table>

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATING

(V_{SS} = 0 V)

PARAMETER	SYMBOL	CONDITIONS	RATING	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}	P21, P22, $\overline{\text{RESET}}$, Tri-st	– 0.3 to V_{DD} + 0.3	
	V_{OUT2}	P20, P23, Sink Open Drain Port	– 0.3 to 5.5	
Output Current (Per 1 pin)	I_{OUT1}	P0, P1, P2, P4, P7 port	3.2	mA
	I_{OUT2}	A19-0, D7-0, $\overline{\text{RD}}$, $\overline{\text{WR}}$	12	
	I_{OUT3}	P3	30	
Output Current (Total)	ΣI_{OUT1}		80	
	ΣI_{OUT2}		120	
Power Dissipation (Topr = 70 °C)	PD		330	mW
Soldering Temperature (time)	Tsld		260 (10 s)	°C
Storage Temperature	Tstg		– 55 to 125	
Operating Temperature	Topr		– 40 to 85	

RECOMMENDED OPERATING CONDITIONS

(V_{SS} = 0 V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	PINS	CONDITIONS		Min.	Max.	UNIT
Supply Voltage	V_{DD}		fc = 12.5 MHz	NORMAL1, 2 mode	4.5	5.5	V
				IDLE1, 2 mode	2.7		
			fc = 4.2 MHz	NORMAL1, 2 mode			
				IDLE1, 2 mode			
			fs = 32.768 kHz	SLOW mode			
				SLEEP mode			
				STOP mode	2.0		
Input High Voltage	V_{IH1}	Except hysteresis and TTL input	$V_{DD} \geqq 4.5$ V		$V_{DD} \times 0.70$	V_{DD}	V
	V_{IH2}	Hysteresis input			$V_{DD} \times 0.75$		
	V_{IH3}	Except TLL input	$V_{DD} < 4.5$ V		$V_{DD} \times 0.90$		
	V_{IH4}	TTL input (Data bus)	V_{DD} = 5 V		2.2		
			V_{DD} = 3 V		$V_{DD} - 0.2$		
Input Low Voltage	V_{IL1}	Except hysteresis and TTL input	$V_{DD} \geqq 4.5$ V		0	$V_{DD} \times 0.30$	V
	V_{IL2}	Hysteresis input				$V_{DD} \times 0.25$	
	V_{IL3}	Except TTL input	$V_{DD} < 4.5$ V			$V_{DD} \times 0.10$	
	V_{IL4}	TTL input (Data bus)	V_{DD} = 5 V			0.8	
			V_{DD} = 3 V			0.2	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 4.5 V to 5.5 V (Normal 1, 2 modes)		1.0	12.5	MHz
			V_{DD} = 2.7 V to 5.5 V			4.2	
	fs	XTIN, XTOUT			30.0	34.0	kHz

Note : fc (Min.) are calculated at using clock Gear as follow :
(Minimum value of fc) = (pre-scaled ration) × 1 [MHz]

DC CHARACTERISTICS

(V_{SS} = 0 V, Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Typ.	Max.	UNIT
Hysteresis Voltage	V_{HS}	Hysteresis input		–	0.9	–	V
Input Current	I_{IN1}	TEST, $\overline{EA}$	V_{DD} = 5.5 V V_{IN} = 5.5 V / 0 V	–	–	± 2	μA
	I_{IN2}	Sink Open Drain, Tir-state Port					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input Resistance	R_{IN2}	$\overline{RESET}$		100	220	450	kΩ
	R_{IN3}	TEST		–	70	–	
Oscillator Feed-back Resistance	R_{fx}	XIN-XTOUT		–	1.2	–	MΩ
	R_{fxt}	XTIN-XTOUT		–	6	–	
Output Leakage Current	I_{LO1}	Sink Open Drain Port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
	I_{LO2}	Tir-st port	V_{DD} = 5.5 V, V_{OUT} = 5.5 V / 0V	–	–	± 2	
Output High Voltage	V_{OH2}	Tir-st port	V_{DD} = 4.5 V, I_{OH} = – 0.7 mA	4.1	–	–	V
	V_{OH3}	A19-0, D7-0, $\overline{RD}$, $\overline{WR}$	V_{DD} = 4.5 V, I_{OH} = – 400 μs	2.4	–	–	
Output Low Voltage	V_{OL3}	A19-0, D7-0, $\overline{RD}$, $\overline{WR}$	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.45	V
Output Low Voltage	I_{OL1}	Except XOUT, P3, A19-0, D7-0, $\overline{RD}$, $\overline{WR}$	V_{DD} = 4.5 V, V_{OL} = 0.4 V	–	1.6	–	mA
	I_{OL3}	P3	V_{DD} = 4.5 V, V_{OL} = 1.0 V	–	20	–	
Supply Current in NORMAL1, 2 mode	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V fc = 12.5 MHz fs = 32.768 kHz	–	15	20	mA
Supply Current in IDLE1, 2 mode				–	6	8	
Supply Current in SLOW mode			V_{DD} = 3.0 V V_{IN} = 2.8 V / 0.2 V fs = 32.768 kHz	–	30	60	μA
Supply Current in SLEEP mode				–	15	30	
Supply Current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V / 0.2 V	–	0.5	10	μA

Note 1 : Typical values show those at T_{opr} = 25 °C, 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 : IDD : Except for IREF.

A.C. CHARACTERISTICS

(1) (V_{SS} = 0 V, V_{DD} = 4.5 to 5.5 V, Topr = – 40 to 85 °C)

(1) - ① CLOCK

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Machine Cycle Time	tcy	In NORMAL1, 2 mode	0.32	–	4	μs
		In IDLE1, 2 mode				
		In SLOW mode	117.6	–	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input) fc = 12.5 MHz	40	–	–	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XIN input) fs = 32.768 kHz	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

(1) - ② EXTERNAL MEMORY INTERFACE

PARAMETER	SYMBOL	Variable		12.5 MHz		UNIT
		Min.	Max.	Min.	Max.	
Address Setup to $\overline{RD}$	t_{ARD}	0.5t – 30	–	10	–	ns
Address Setup to $\overline{WR}$	t_{AWR}	1.5t – 30	–	90	–	ns
Address Hold Time After $\overline{RD}$ / $\overline{WR}$	t_{RDA} t_{WRA}	0.5t – 35	–	5	–	ns
Address to Valid Data In	t_{ADI}	–	3.5t – 95	–	185	ns
$\overline{RD}$ to Valid Data In	t_{RDDS}	–	3.0t – 100	–	140	ns
$\overline{RD}$ Low Pulse Width	t_{WRD}	0.3t – 40	–	200	–	ns
Input Data Hold After $\overline{RD}$	t_{RDDH}	0	–	0	–	ns
$\overline{WR}$ Low Pulse Width	t_{WWR}	2.0t – 40	–	120	–	ns
Data Setup to $\overline{WR}$	t_{DWR}	2.0t – 40	–	120	–	ns
Data Hold After $\overline{WR}$	t_{WRDH}	0.5t – 35	–	5	–	ns
XIN to Address Delay	t_{XINA}	–	140	–	140	ns
XTIN to Address Delay	t_{XTINA}	–	340	–	340	ns

Note : t = tcy / 4 (t = 80 ns @ 12.5 MHz)

(1) - ③ WAIT

PARAMETER	SYMBOL	Variable		12.5 MHz		UNIT
		Min.	Max.	Min.	Max.	
Address Setup to $\overline{\text{WAIT}}$	t_{AWTF}	–	1.5t – 100	–	85	ns
Address Setup to $\overline{\text{WAIT}}$	t_{AWTR}	1.5t + 20	–	140	–	ns
$\overline{\text{RD}}$ Setup to $\overline{\text{WAIT}}$	t_{RDWTF}	–	1.0t – 100	–	– 20	ns
$\overline{\text{RD}}$ Setup to $\overline{\text{WAIT}}$	t_{RDWTR}	1.0t + 20	–	100	–	ns
$\overline{\text{WR}}$ Setup to $\overline{\text{WAIT}}$	t_{WRWTR}	20	–	20	–	ns
Address Valid to $\overline{\text{CLK}}$	t_{ACLK}	–	4.0t + 35	–	355	ns
$\overline{\text{CLK}}$ Pulse Width	t_{WCLKL} t_{WCLKH}	2.0t – 50	–	110	–	ns
$\overline{\text{CLK}}$ Set up to WAIT	t_{CLKWT}	–	1.5t – 70	–	50	ns

(1) - ④ BUS ARBITRATION

PARAMETER	SYMBOL	Variable		12.5 MHz		UNIT
		Min.	Max.	Min.	Max.	
Bus Floating to $\overline{\text{BUSAK}}$	t_{BFAK}	0.5t – 30	–	50	–	ns

A.C. Meaurement Condition

Output Level : High 2.2 V / Low 0.8 V, CL = 100pF
Input level : High 2.4 V / Low 0.4 V (D7 to D0)
High 0.7 V_{DD} / Low 0.3 V_{DD} (WAIT)
High 0.8 V_{DD} / Low 0.2 V_{DD} (Except D7 to D0 and WAIT)

A.C. CHARACTIRISTICS (2) (V_{SS} = 0 V, V_{DD} = 2.7 to 5.5 V, Topr = – 40 to 85 °C)

(2) - ① CLOCK

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Machine Cycle Time	tcy	In NORMAL1, 2 mode	0.95	–	4	μs
		In IDLE1, 2 mode				
		In SLOW mode	117.6	–	133.3	
		In SLEEP mode				
High Level Clock Pulse Width	t_{WCH}	For external clock operation (XIN input) fc = 4.2 MHz	119	–	–	ns
Low Level Clock Pulse Width	t_{WCL}					
High Level Clock Pulse Width	t_{WSH}	For external clock operation (XTIN input) fs = 32.768 kHz	14.7	–	–	μs
Low Level Clock Pulse Width	t_{WSL}					

(2) - ② EXTERNAL MEMORY INTERFACE

PARAMETER	SYMBOL	Variable		4.2 MHz		UNIT
		Min.	Max.	Min.	Max.	
Address Setup to $\overline{RD}$	t_{ARD}	0.5t – 110	–	9	–	ns
Address Setup to $\overline{WD}$	t_{AWR}	1.5t – 120	–	237	–	ns
Address Hold Time After $\overline{RD}$ / $\overline{WR}$	t_{RDA} t_{WRA}	0.5t – 110	–	9	–	ns
Address to Valid Data In	t_{ADI}	–	3.5t – 270	–	563	ns
$\overline{RD}$ to Valid Data In	t_{RDDS}	–	3.0t – 205	–	509	ns
$\overline{RD}$ Low Pulse Width	t_{WRD}	0.3t – 40	–	436	–	ns
Input Data Hold After $\overline{RD}$	t_{RDDH}	0	–	0	–	ns
$\overline{WR}$ Low Pulse Width	t_{WWR}	2.0t – 85	–	391	–	ns
Data Setup to $\overline{WR}$	t_{DWR}	2.0t – 50	–	426	–	ns
Data Hold After $\overline{WR}$	t_{WRDH}	0.5t – 110	–	9	–	ns

Note : t = tcy / 4 (t = 238 ns @ 4.2 MHz)

(2) - ③ WAIT

PARAMETER	SYMBOL	Variable		4.2 MHz		UNIT
		Min.	Max.	Min.	Max.	
Address Setup to $\overline{\text{WAIT}}$	t_{AWTF}	–	1.5t – 257	–	100	ns
Address Setup to $\overline{\text{WAIT}}$	t_{AWTR}	1.5t + 125	–	482	–	ns
$\overline{\text{RD}}$ Setup to $\overline{\text{WAIT}}$	t_{RDWTF}	–	1.0t – 165	–	73	ns
$\overline{\text{RD}}$ Setup to $\overline{\text{WAIT}}$	t_{RDWTR}	1.0t + 125	–	363	–	ns
$\overline{\text{WR}}$ Setup to $\overline{\text{WAIT}}$	t_{WRWTR}	50	–	50	–	ns
Address Valid to $\overline{\text{CLK}}$	t_{ACLK}	–	4.0t + 70	–	1022	ns
$\overline{\text{CLK}}$ Pulse Width	t_{WCLKL}	2.0t – 118	–	358	–	ns
	t_{WCLKH}					
$\overline{\text{CLK}}$ Set up to WAIT	t_{CLKWT}	–	1.5t – 170	–	187	ns

(2) - ④ BUS ARBITRATION

PARAMETER	SYMBOL	Variable		4.2 MHz		UNIT
		Min.	Max.	Min.	Max.	
Bus Floating to $\overline{\text{BUSAK}}$	t_{BFAK}	0.5t – 109	–	10	–	ns

A.C. Meaurement Condition

Output Level	:	High 0.7 V_{DD} / Low 0.3 V_{DD}, CL = 100 pF
Input level	:	High 0.9 V_{DD} / Low 0.1 V_{DD}

A / D CONVERSION CHARACTERISTICS

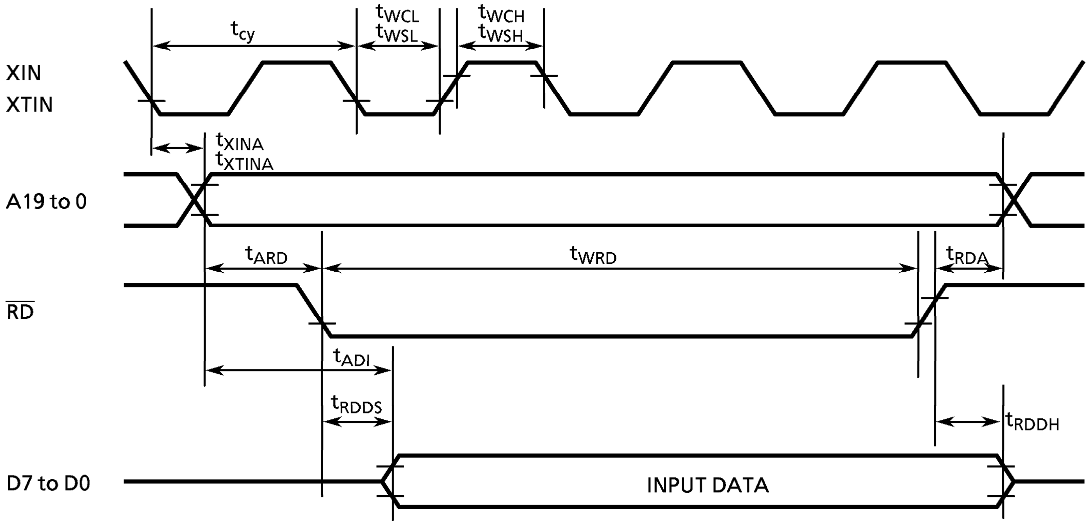
(Topr = – 40 to 85 °C)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Analog Reference Voltage	V_{AREF}		V_{DD} – 1.5	–	V_{DD}	V
	V_{ASS}		V_{SS}	–	V_{SS}	
Analog Reference Voltage Range	$\triangle V_{AREF}$		2.5	–	–	V
Analog Input Voltage	V_{AIN}		V_{ASS}	–	V_{AREF}	V
Analog Supply Current	I_{REF}	V_{DD} = AVDD = VAREF = 5.5 V VSS = AVSS = VASS = 0.0 V	–	0.5	1.0	mA
Non-Linearity Error		V_{DD} = 5.0 to 5.5 V, V_{SS} = 0.0V AVDD = VAREF = 5.000 V AVSS = VASS = 0.000 V low Speed Conversion (58.9 μs, @ 12.5 MHz)	–	–	± 2	LSB
Zero Point Error			–	–	± 2	
Full Scale Error			–	–	± 2	
Total Error			–	–	± 4	
Non-Linearity Error		V_{DD} = 2.7 to 5.5 V, V_{SS} = 0.0V AVDD = VAREF = 2.700 V AVSS = VASS = 0.000 V Hihg speed conversion (43.7 μs, @ 4.2 MHz)	–	–	± 2	LSB
Zero Point Error			–	–	± 2	
Full Scale Error			–	–	± 2	
Total Error			–	–	± 4	

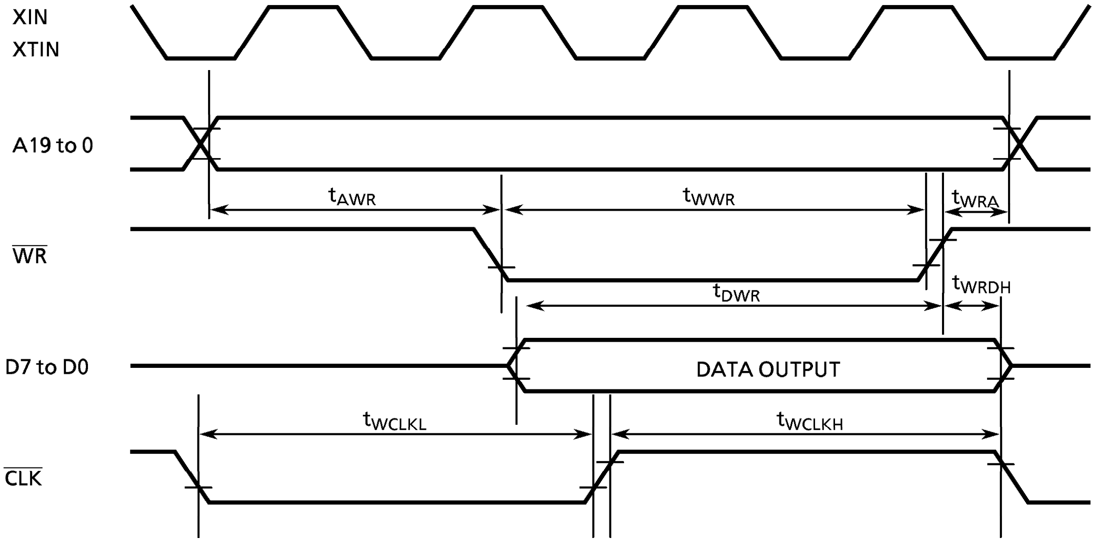
Note : $\triangle V_{AREF} = V_{AREF} - V_{ASS}$

Timing Chart

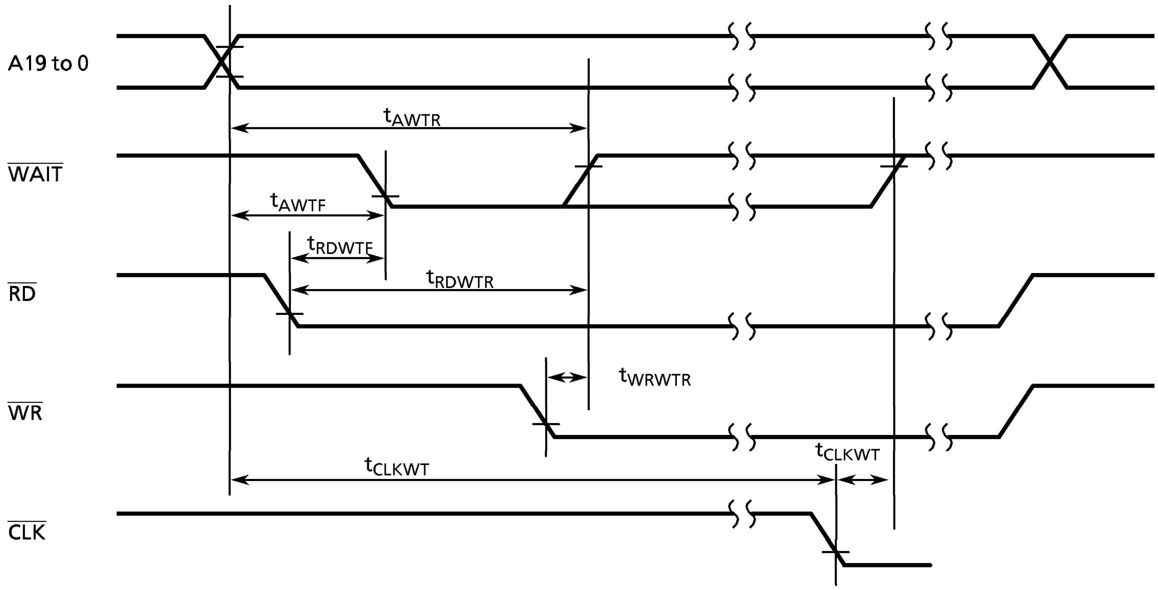
(1) Read Cycle



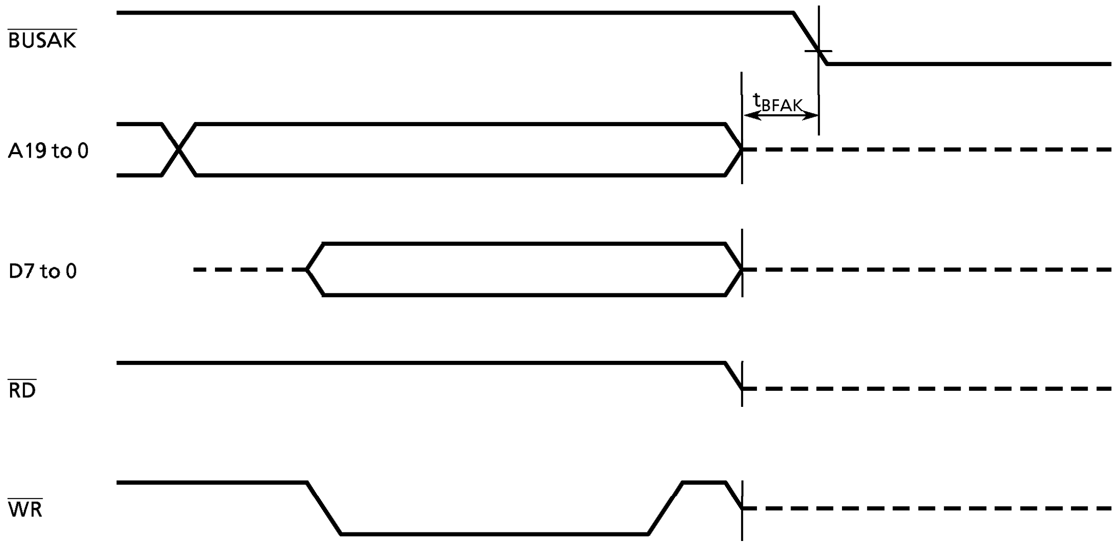
(2) Write Cycle



(3) Wait Timing



(4) Bus Arbitation


$\overline{BUSAK}$
t_{BFAK}
A19 to 0
D7 to 0
$\overline{RD}$
$\overline{WR}$