CMOS 4-BIT MICROCONTROLLER

TMP47E885AF

Based on the TLCS-470 series, TMP47E885AF is a high-speed, advanced, single-chip, 4-bit microcomputer with a built-in 64 × 8-bit E²PROM, enhanced timer/counter functions, an 8-bit A/D converter, 12-bit pulse width modulation (PWM) output, and an 8-bit asynchronous serial interface (UART).

PART No.	ROM	E²PROM	RAM	PACKAGE	OTP	PIGGYBACK
TMP47E885AF	8192 × 8-bit	64 × 8-bit	512 × 4-bit	QFP44-P-1414-0.80D	TMP47P885F	–

FEATURES

◆4-bit single chip microcomputer

◆Minimum instruction execution time
: 1.3 μs (at 6 MHz), 244 μs (32.8 kHz)

◆Basic machine instructions : 92
- Table look-up instructions

◆Subroutine nesting: 15 levels max.

◆6 interrupt sources (External : 2, Internal : 4)
All sources have independent latches each, and multiple interrupt control is available.

◆I/O port (36 pins)
- Input: 2 ports 5 pins
- I/O: 8 ports 31 pins

◆ Interval timer

◆16-bit timer/counter: one channel
Free running timer, event counter, input capture, output compare

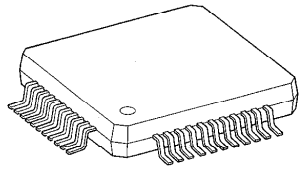
◆Watchdog timer

◆8-bit successive approximate type A/D converter
- Analog input : 8 channels
- Conversion time : 34.6 μs (6 MHz)

◆12-bit pulse width modulation (PWM) output: two channels

◆8-bit buffered synchronous serial interface (SIO)
- Simultaneous transmit/receive
- External/internal clock, leading/trailing edge shift, 4-bit/8-bit mode

QFP44-P-1414-0.80D



TMP47E885AF

Caution:
- Because timer/counter functions have been enhanced, the description of the operation of TC1 and TC2 in the assembler, C-like compiler and software simulator manuals cannot be used.

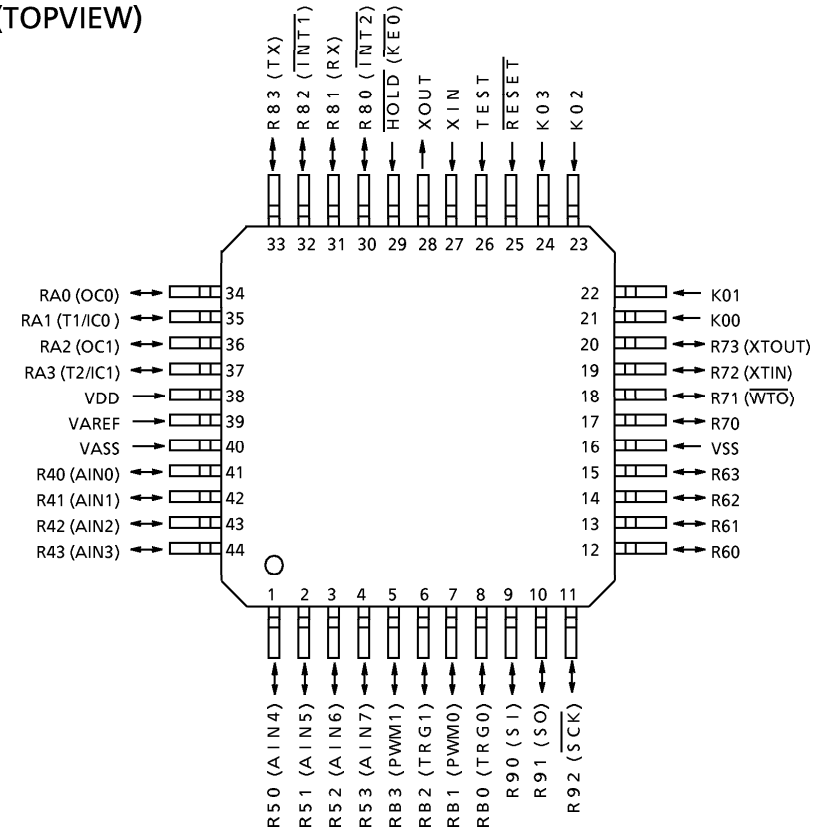
980901EBP1

◆8-bit buffered asynchronous serial interface (UART)
◆64 × 8-bit E^2PROM
- Word-by-word rewrite
- Automatic rewrite time setting (timer built in)
- Ready/Busy status monitor

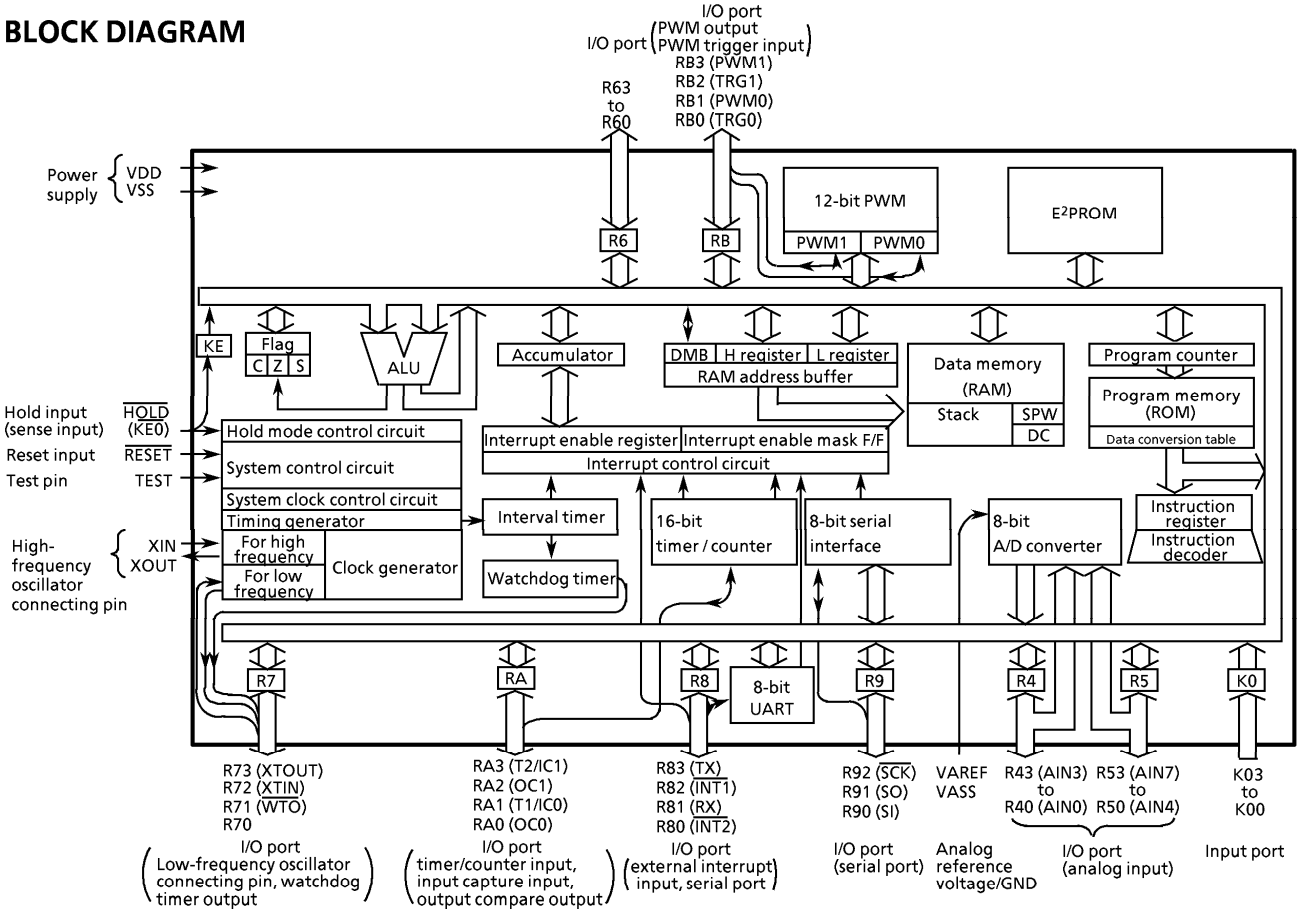
◆Dual-clock mode
High-speed mode /low power consumption operating mode
◆Hold function
battery/capacitor backup
◆Emulator : BM47E885AF0A

PIN ASSIGNMENT (TOPVIEW)

QFP44-P-1414-0.80D



BLOCK DIAGRAM



PIN FUNCTION

PIN NAME	Input/Output	Function	(Function)
K03 to K00	Input	4-bit latched I/O port.	
R53 (AIN7) to R40 (AIN0)	I/O (Input)	When using as an input port or for analog input, set the latch to 1. Can be set, cleared, or tested in units of bits by bit manipulation instructions using L register indirect addressing.	A/D converter analog input
R63 to R60	I/O	4-bit latched I/O port. When using as input port, set latch to 1. Can be set, cleared, or tested in units of bits by bit manipulation instructions using	
R73 (XTOUT)	I/O (Output)	4-bit latched I/O port. When using R73 and R72 as oscillators, set to dual-clock mode. When using as input port or for watchdog timer output, set latch to 1. Can be set, cleared, or tested in units of bits by bit manipulation instructions using L register indirect addressing.	Oscillator connecting pin (low frequency).
R72 (XTIN)	I/O (Input)		Input an external clock to XIN. XOUT is open circuit.
R71 ($\overline{\text{WTO}}$)	I/O (Output)		Watchdog timer output
R70	I/O		
R83 (TX)	I/O (Output)	4-bit latched I/O port. When using as input port, as external interrupt input pin, or as asynchronous serial port (UART) pin, set latch to 1.	Asynchronous serial data output
R82 ($\overline{\text{INT1}}$)	I/O (Input)		External interrupt 1 input
R81 (RX)			Asynchronous serial data input
R80 ($\overline{\text{INT2}}$)			External interrupt 2 input
R92 ($\overline{\text{SCK}}$)	I/O (I/O)	3-bit latched input/output port. When using as input port or synchronous serial port, set latch to 1.	Synchronous serial clock input/output
R91 (SO)	I/O (Output)		Synchronous serial data output
R90 (SI)	I/O (Input)		Synchronous serial data input
RA3 (T2/IC1)	I/O (Input)	4-bit latched input/output port. When using as input port or for timer/input capture input, set latch to 1.	Timer/input capture 1 input
RA2 (OC1)	I/O (Output)		Output compare 1 output
RA1 (T1/IC0)	I/O (Input)		Timer/input capture 0 input
RA0 (OC0)	I/O (Output)		Output compare 0 output
RB3 (PWM1)	I/O (Output)	4-bit latched input/output port. When using as input port or for PWM trigger input, set latch to 1.	PWM 1 output
RB2 (TRG1)	I/O (Input)		PWM trigger 1 input
RB1 (PWM0)	I/O (Output)		PWM 0 output
RB0 (TRG0)	I/O (Input)		PWM trigger 0 input
XIN	Input	Oscillator connecting pin (high frequency).	
XOUT	Output	Input an external clock to XIN. XOUT is open circuit.	
$\overline{\text{RESET}}$	Input	Reset signal input	
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input (Input)	Hold request/release signal input	Sense input
TEST	Input	Delivery test pin. Fix at low level.	

Pin Name	Input/Output	Function	(Function)
VDD	Power supply	+ 5V	
VSS		0V (GND)	
VAREF	Power supply	Analog reference voltage for A/D conversion	
VASS		Analog reference GND for A/D conversion	

OPERATIONAL DESCRIPTION

1. SYSTEM CONFIGURATION

- CPU Core Functions
  - 2.1 Program Counter (PC)
  - 2.2 Program Memory (ROM)
  - 2.3 H register, L register, DMB selector
  - 2.4 Data Memory (RAM)
    - Stack
    - Stack pointer word (SPW)
    - Data counter (DC)
  - 2.5 ALU and Accumulator
  - 2.6 Flags
  - 2.7 System Control Circuit
  - 2.8 Interrupt Control Circuit
  - 2.9 Reset Circuit
- Hardware Functions
  - 3.1 I/O Ports
  - 3.2 Interval Timer
  - 3.3 Timer/Counter
  - 3.4 Watchdog Timer
  - 3.5 A/D Converter
  - 3.6 PWM
  - 3.7 Serial Interface
  - 3.8 E^2PROM

The following sections describe the hardware configuration and operation for the above components.

2. CPU Core Functions

2.1 Program Counter (PC)

The program counter is a 13-bit register that indicates the address of the next program instruction to be executed. At each instruction fetch, the register is incremented by the number of bytes fetched.
Table 2.1 shows the operation of the program counter when a branch or subroutine instruction is executed, or an interrupt is received. At reset, the PC is initialized to 0.

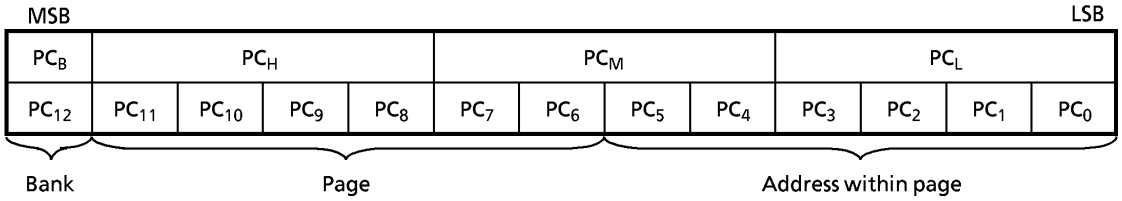


Figure 2-1. Configuration of Program Counter

The program counter directly addresses 8192 bytes of memory. Note the following points concerning short and middle branch instructions and subroutine instructions.

(1) Short branch instruction [BSS a]

If the branch condition is satisfied when the [BSS a] instruction is executed, the branch value specified by the instruction is set in the lower six bits of the PC. That is, the [BSS a] instruction is a in-page branch instruction.

If the [BSS a] instruction occurs at the final address in a page, the upper seven bits of the PC point to the next instruction, which is in the next page. Therefore, the instruction will branch to an address within the next page.

(2) Middle branch instruction [BS a]

If the branch condition is satisfied when the [BS a] instruction is executed, the branch value specified by the instruction is set in the lower twelve bits of the PC. That is, the [BS a] instruction is a in-bank branch instruction.

If the first byte or the second byte of the [BS a] instruction is stored at the final bank address, the MSB of the PC points to the next bank. Therefore, the instruction will branch to an address within the next bank.

(3) Subroutine call instruction [CALL a]

After the contents of the program counter are saved to the stack when the [CALL a] instruction is executed, the call value specified by the instruction is set in the PC. Addresses specifiable with this instruction are 11-bit addresses. The upper two bits of the PC are always set to 0. Therefore, the start address of the subroutine must be 0000 - 07FF$_H$.

Instruction or Operation		Condition		Program Counter (PC)												
				PC_{12}	PC_{11}	PC_{10}	PC_9	PC_8	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0
Execution of Instruction	BSL a	SF = 1 (branch condition satisfied)		Immediate data specified by instruction												
		SF = 0 (branch condition not satisfied)		+ 3												
	BS a	SF = 1	Lower 12 bits of address ≠ FFE, FFF$_H$	No change	Immediate data specified by instruction											
			Lower 12 bits of address = FFE, FFF$_H$ (final address in bank)	+ 1	Immediate data specified by instruction											
		SF = 0		+ 2												
	BSS a	SF = 1	Lower six bits of address ≠ 3F$_H$	No change						Immediate data specified by instruction						
			Lower six bits of address = 3F$_H$ (final address in page)	+ 1						Immediate data specified by instruction						
		SF = 0		+ 1												
	CALL a			0	0	Immediate data specified by instruction										
	CALLS a			0	0	0	0	0	Value generated from immediate data specified by instruction					1	1	0
	RET			Return address restored from the stack												
	RETI			Return address restored from the stack												
	Other instruction			Incremented by the number of bytes of instruction												
At interrupt reception				0	0	0	0	0	0	0	0	0	Interrupt vector			0
At reset				0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-1. Status Change of Program Counter

2.2 Program Memory (ROM)

Programs and fixed data are stored in program memory. The program counter indicates the address of the next instruction to be executed.

(1) Table look-up instructions [LDL A, @DC], [LDH A, @DC+]

The table look-up instructions read the upper or lower four bits of data stored at the ROM address specified by the data counter (DC) and load the bits to the accumulator. (The instruction [LDL A, @DC] reads the lower four bits. The instruction [LDH A, @DC+] reads the upper four bits.) The DC is 12 bits in length and can specify addresses 1000 - $1FFF_H$ in program memory.

(2) 5 - to 8-bit data conversion instruction [OUTB @HL]

The 5 - to 8-bit data conversion instruction reads the fixed (8-bit) data stored in the data conversion table in program memory (1FE0 - $1FFF_H$). The instruction specifies the upper four bits to be output to port 2 and the lower four bits to be output to port 1. However, do not use this instruction because the operation is invalid as there is no port 1 or port 2.

2.2.1 Program Memory Map

Figure 2-3 shows the program memory map. Program memory addresses 0000 - 0086_H and the final 32-byte space can also be used for special applications.

2.2.2 Program Memory Capacity

The processor incorporates 8192 x 8 bits (0000 - $1FFF_H$) of program memory (mask ROM).

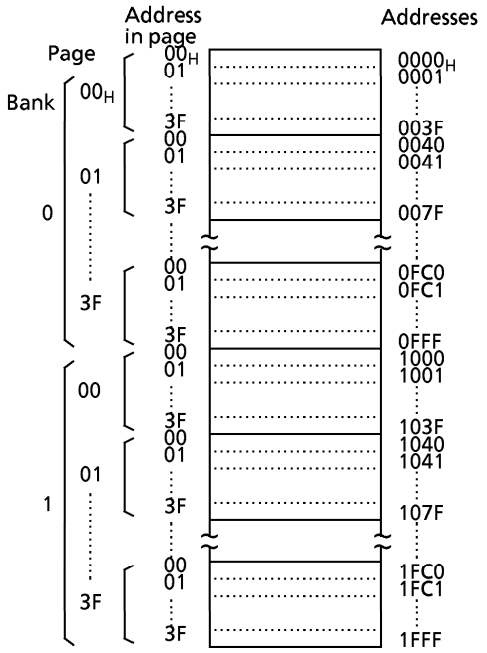


Figure 2-2. Configuration of Program Memory

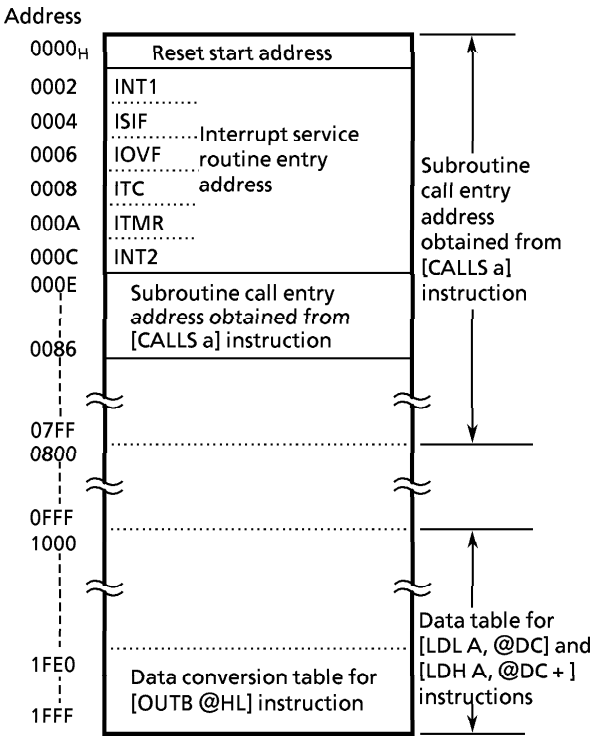


Figure 2-3. Program Memory Map

2.3 H Register, L Register, and Data Memory Bank Selector (DMB)

The H and L registers are 4-bit general-purpose registers. The two registers can be used as a pair, called the HL register, which can be used as a pointer to indicate a RAM data memory address. The data memory has a page structure with 16 words per page. The H register indicates the page and the L register indicates the address within the page. The data memory consists of two banks (bank 0 and bank 1). The data memory bank selector (DMB) is a 1-bit register. It specifies the data memory bank. Set the DMB using the [CLR DMB] and [SET DMB] instructions. The [TEST DMB] and [TESTP DMB] instructions can be used to ascertain which bank is currently selected. At reset, DMB is initialized to 0.

The L register has an auto post-increment or decrement feature, which permits compound instructions. For example, the [ST A,@HL+] instruction automatically increments the contents of the L register after data transfer.

Similarly, the L register is used to specify the bit corresponding to the I/O port (pins R73 - R40) when the [SET @L], [CLR @L], or [TEST @L] instructions are executed (indirect addressing of the port bits using the L register).

Example 1 : Write immediate values 5 and F_H to data memory addresses 10_H and 11_H respectively.

```
CLR     DMB
LD      HL, #10H
ST      #5, @HL+
ST      #0FH, @HL+
```

Example 2 : Use a bit manipulation instruction to set the pin R71 output latch to 1 with L register indirect addressing.

```
LD      L, #1101B
SET     @L
```

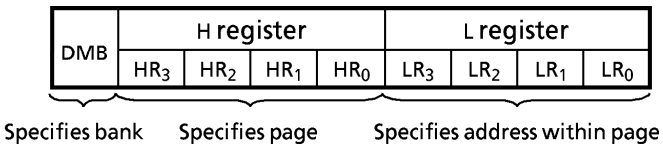


Figure 2-4. configuration of H Register, L Register, and DMB

2.4 Data Memory (RAM)

Data memory is used to save user data. Data memory consists of eight pages each consisting of 16 words (4-bit words). Data memory addresses can be specified by the following methods:

(1) Register indirect addressing mode

The DMB specifies the bank. The H register specifies the page. The L register specifies the RAM address within the page.

Example: LD A, @HL ; Acc ← RAM [HL]

(2) Direct addressing mode

The second byte of the instruction field (operand) specifies an address in the bank. The DMB specifies the bank.

Example: LD A, 2CH ; Acc ← RAM [$2C_H$]

(3) Page zero addressing mode

The lower four bits in the second byte of the instruction field (operand) specify an address in page zero (00 - $0F_H$) of bank 0.

Example: ST #3, 05H ; RAM [05_H] ← 3

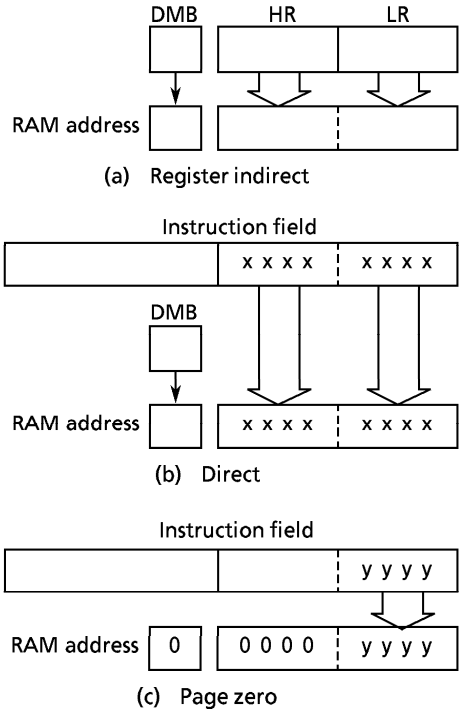
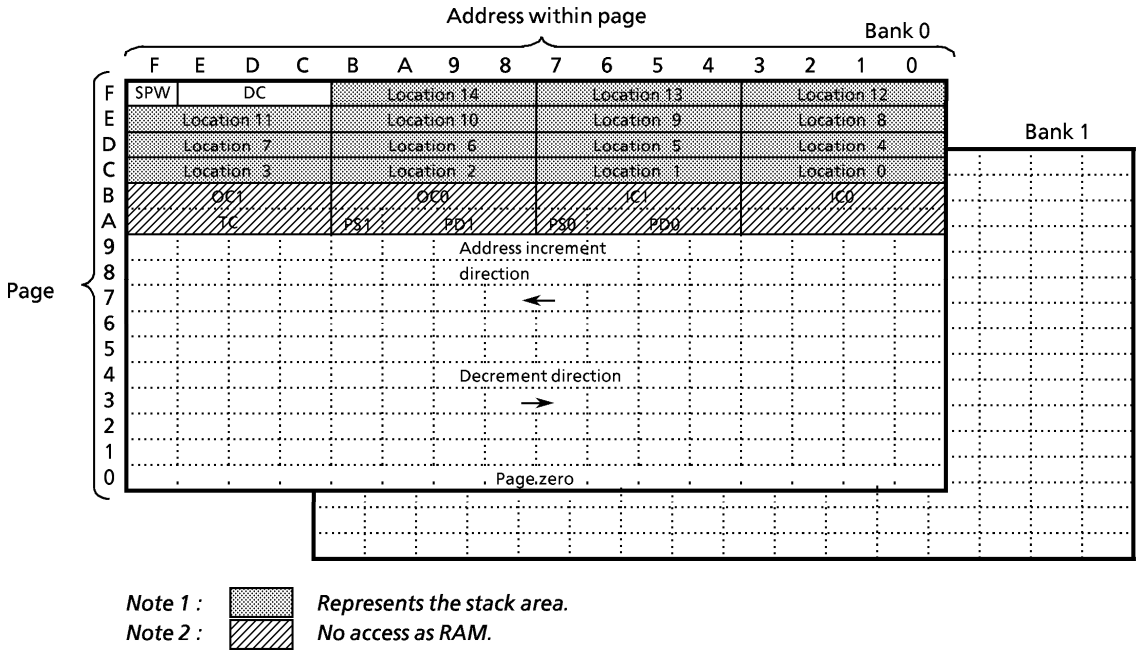


Figure 2-5. Addressing Modes

2.4.1 Data Memory Map

Figure 2-6 shows the data memory map. Data memory is also used for the following purposes, but only in bank 0.

① Stack
② Stack pointer word (SPW)
③ Data counter (DC)
④ Timer/counter register (TC, IC1, IC0, OC1, OC0)
⑤ PWM output register (PD1, PD0, PS1, PS0)
⑥ Page zero



Note 1 : *Represents the stack area.*
Note 2 : *No access as RAM.*

TC, OC1, OC	IC1, IC0	PS1, PD1, PS0, PD0	RAM Address (bank 0) $A3_H$ to $A0_H$
write/read	read only	write only	Access prohibited

Figure 2-6. Data Memory Map

(1) Stack

The stack area is used to save the contents of the program counter (return address) and flags when a subroutine call instruction is executed or an interrupt received. The processor saves the data to the stack before jumping to the processing routine.

When returning from the processing routine, the subroutine return instruction [RET] restores the contents of the program counter from the stack. The interrupt return instruction [RETI] restores the contents of the program counter and flags from the stack.

The stack is allocated to data memory addresses C0 - FB_H in bank 0, providing a maximum of 15 levels (locations 14 to 0). Each stack level consists of four words of data memory.

The save/restore stack location is determined by the stack pointer word. The stack pointer word is automatically decremented after saving, and automatically incremented prior to restoring. That is, the stack pointer word holds the next free location on the stack.

(2) Stack pointer word (SPW)

Data memory address FF_H (in bank 0) is known as the stack pointer word. The SPW determines the location when the stack is accessed (saving or restoring).

The stack pointer word is normally in the range 0 to 14, allowing up to 15 levels of nesting. Because the stack pointer word is allocated to address FF_H, the contents of the stack pointer word may never be 15.

The stack pointer word is automatically rewritten at subroutine calls and on receipt of interrupts. Note that if the stack exceeds the area allocated to the stack in data memory, user data are corrupted. (For example, if the user data area is allocated to address range 00 - CF_H, the stack area is available up to location 4. If an interrupt is received while location 4 is in use, the user data at addresses CC - CF_H corresponding to location 3 are corrupted.)

The stack pointer word is not initialized by hardware. Use a software initialization routine to write initial value 14 (the number of the first stack location to use).

Example : Stack pointer word initial setting

```
LD    A, #14
CLR   DMB
ST    A, 0FFH
```

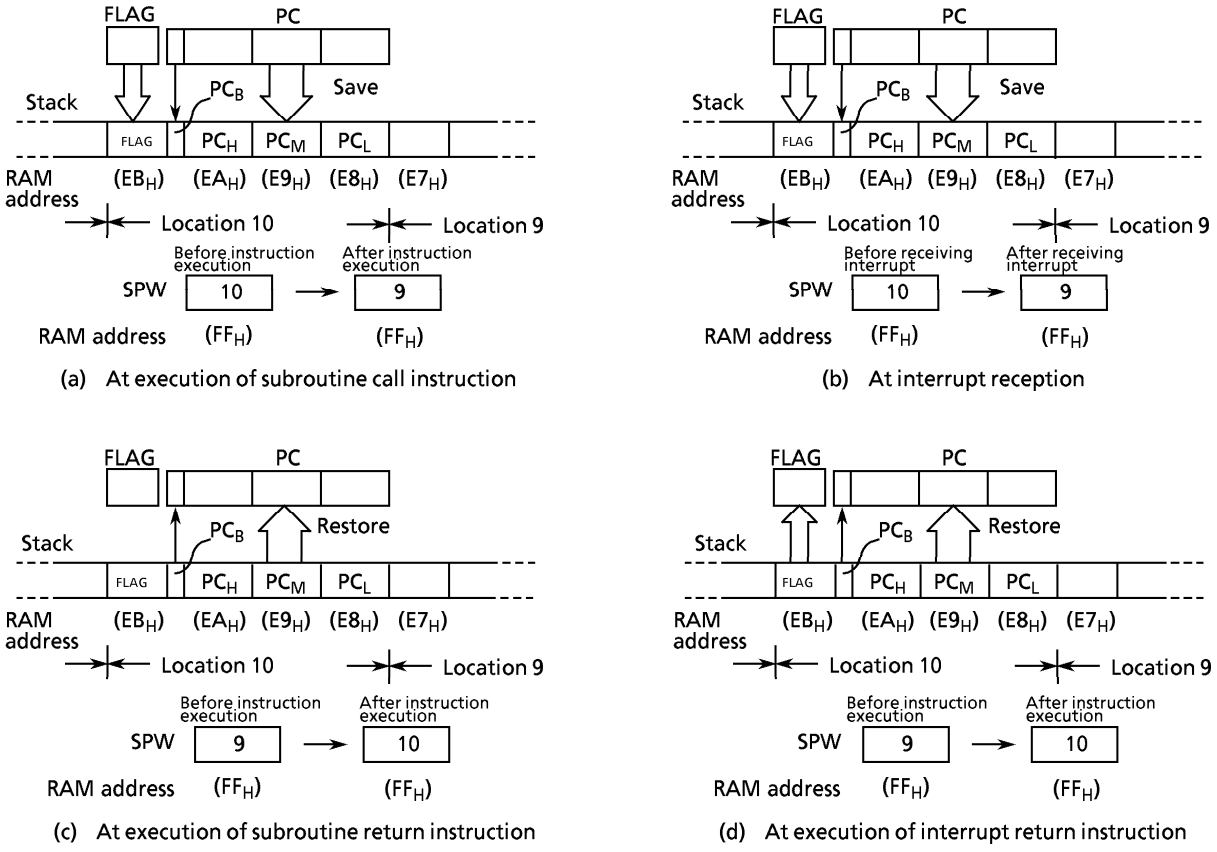


Figure 2-7. Stack Access (Save/Restore)

(3) Data Counter (DC)

The data counter is a 12-bit counter used to specify the address of the data table in program memory (ROM). The data table is referenced by the table look-up instructions [LDL A, @DC] and [LDH A, @DC+]. Data tables can be allocated from address 1000_H in the program memory area (the MSB of the PC is fixed to 1).

The data counter is allocated to RAM addresses in units of 4 bits. Use the RAM manipulation instructions to set initial values or read the contents of the data counter.

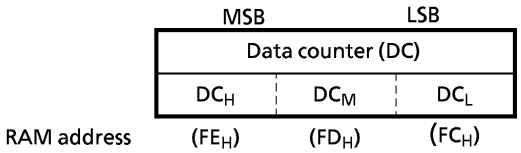


Figure 2-8. Data Counter

Example : Set the data counter to 780_H

```
CLR     DMB
LD      HL, #0FCH
ST      #0H, @HL+
ST      #8H, @HL+
ST      #7H, @HL+
```

(4) Timer/Counter Registers (TC, IC1, IC0, OC1, OC0)

The timer/count registers are allocated to RAM in units of 4 bits. Use the RAM manipulation instructions to set initial values or read the contents of the count registers.

However, because the data memory at addresses corresponding to the registers cannot be accessed as RAM, the data memory cannot be used to store user data. Registers IC1 and IC0 are read only. Registers TC, OC1, and OC0 can be read or written.

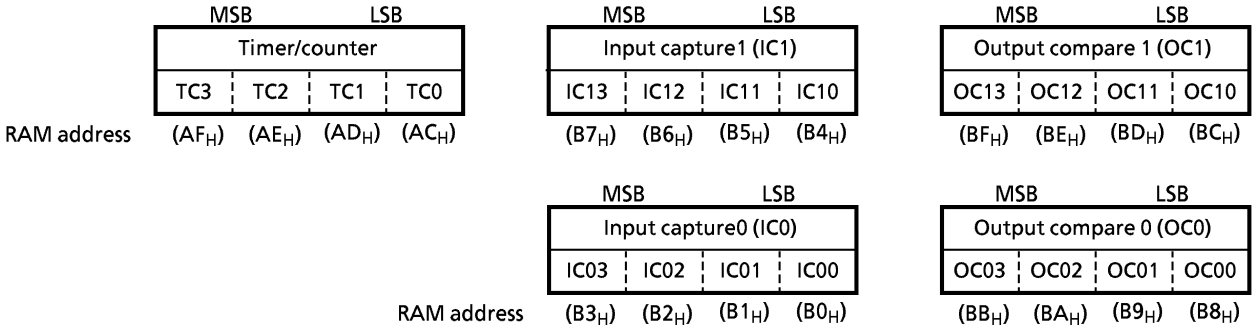


Figure 2-9. Timer/Counter Registers

(5) PWM output registers (PD1, PD0, PS1, PS0)

The PWM output registers are allocated to RAM in units of 4 bits. Use the RAM manipulation instructions to set initial values.

However, because the data memory at addresses corresponding to the registers cannot be accessed as RAM, the data memory cannot be used to store user data. Registers PD1, PD0, PS1, and PS0 are write only.

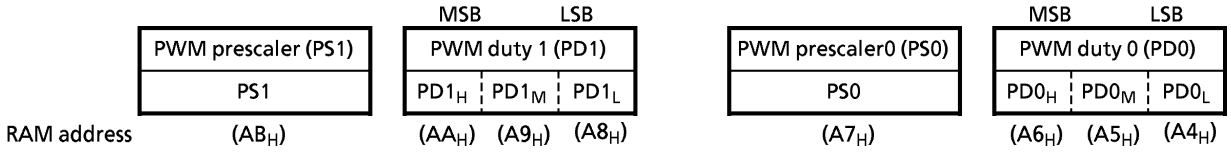


Figure 2-10. PWM Output Register

(6) Page Zero

The sixteen words of (bank 0) page zero (addresses 00 - $0F_H$) data memory can be used for user flags and pointers by utilizing the page zero addressing mode instructions (compare, add, transfer, and bit manipulation). This enables a highly efficient program.

Example : If bit 2 of data memory address 04_H is 1, write 8 to address 09_H.

```
    TEST    04H, 2
    B       SKIP
    ST      #8,09H
SKIP :
```

2.4.2 Data Memory Capacity

47E885AF provides 256 x 4 bits of two internal data memory (RAM) banks (bank 0 and bank 1). Because the contents of the data memory are undefined when the power supply is switched on, set initial values using the initializing routine.

```
Example:  Clear RAM.
          LD      HL, #00H
SCLR1 :   CLR     DMB
SCLR2 :   ST      #0, @HL+
          B       SCLR2
          SET     DMB
SCLR3 :   ST      #0, @HL+
          B       SCLR3
          ADD     H, #1
          B       SCLR1
```

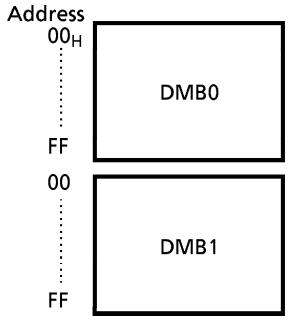


Figure 2-11. Data Memory (RAM)

2.5 ALU and Accumulator

2.5.1 ALU

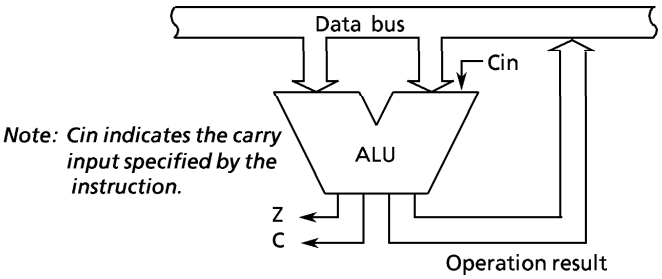
The ALU (arithmetic logic unit) performs operations on 4-bit binary data. The ALU executes an instruction and outputs the (4-bit) operation result, carry data (C), and zero detect data (Z).

(1) Carry data (C)

In additions, the carry data represent the carry from the uppermost bit. As subtraction is processed as a two's complement addition, carry data indicate no borrow from the lowest bit. For rotation instructions, carry data indicate that data shifted out of the accumulator.

(2) Zero detect data (Z)

The processor sets zero detect to 1 when an operation result or data transfer to the accumulator or data memory is 0000_B.



Note: Cin indicates the carry input specified by the instruction.

Figure 2-12. ALU

Example : Carry data (C) and zero detect data (Z) resulting from 4-bit addition and subtraction.

Operation		Operation result	C	Z
4 + 2	=	6	0	0
7 + 9	=	0	1	1
8 – 1	=	7	1	0
2 – 2	=	0	1	1
5 – 8	=	– 3 (1101_B)	0	0

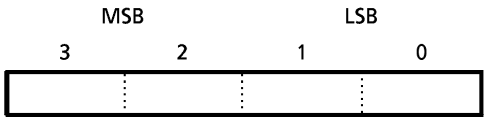


Figure 2-13. Accumulator

2.5.2 Accumulator (Acc)

The accumulator is a 4-bit register used to store the source and result data for operations.

2.6 Flags (FLAG)

47E885AF has a carry flag (CF), zero flag (ZF), and status flag (SF). The flags are set or cleared depending on conditions specified by the instruction.

When an interrupt is received, the flags are saved to the stack together with the program counter. Executing the interrupt return instruction [RETI] restores the flags from the stack. This restores the state immediately prior to receiving the interrupt.

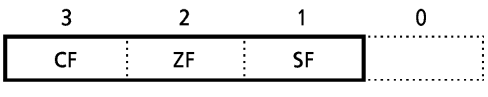


Figure 2-14. Flag

(1) Carry flag (CF)

The carry flag stores the carry data from the ALU at execution of an addition or subtraction instruction with carry, execution of a compare instruction, or execution of a rotate instruction. When a carry flag test instruction is executed, the value specified by instruction is set in the carry flag.

1. Addition or subtraction instruction with carry [ADDC A,@HL] [SUBRC A,@HL]
   The carry flag is used as an ALU input (Cin) and to store the carry data (carry/non-borrow).
2. Compare instructions [CMPR A,@HL] [CMPR A,#k]
   The carry flag is shifted into the accumulator and stores the carry data (data shifted out of the accumulator).
3. Rotate instructions [ROLC A], [RORC A]
   The carry flag is shifted into the accumulator and stores the carry data (data shifted out of the accumulator).
4. Carry flag test instructions [TESTP CF], [TEST CF]
   The [TESTP CF] instruction transfers the contents of the carry flag to the status flag. After the transfer, the carry flag is set to 1.
   The [TEST CF] instruction transfers an inverted carry to the status flag. After the transfer, the carry flag is cleared to 0.

(2) Zero flag (ZF)

The zero flag stores the zero detect data (Z) from the ALU at the execution of an arithmetic instruction, a rotate instruction, an input instruction, or an instruction to transfer to the accumulator. The [TESTP ZF] instruction transfers the contents of the zero flag to the status flag.

(3) Status flag (SF)

The status flag contains the branch conditions for branch instructions. A branch occurs if the status flag is 1. The status flag is normally set to 1; branch with the status flag set to 1 is regarded as an unconditional branch instruction. A branch made immediately after the status flag is set/cleared according to the condition satisfied by the instruction is regarded as a conditional branch instruction. The status flag is initialized to 1 at reset. (Other flags are not initialized.)

Example: The table below lists the values of the accumulator and the flags after execution of the instructions in the table. Prior to the execution, the values are as follows: accumulator: CH H register: 0 L register: 7 Address 07_H in data memory: 5 carry flag: 1

Instruction	Acc after execution	Flag after execution		
		CF	ZF	SF
ADDC A, @HL	2_H	1	0	0
SUBRC A, @HL	9_H	0	0	0
CMPR A, @HL	$C0_H$	0	0	1
AND A, @HL	4_H	1	0	1
LD A, @HL	5_H	1	0	1

Instruction	Acc after execution	Flag after execution		
		CF	ZF	SF
LD A, #0	0_H	1	1	1
ADD A, #4	0_H	1	1	0
DEC A	B_H	1	0	1
ROLC A	9_H	1	0	0
RORC A	E_H	0	0	1

2.7 System Control Circuit

Figure 2-15 shows the configuration of the system control circuit.

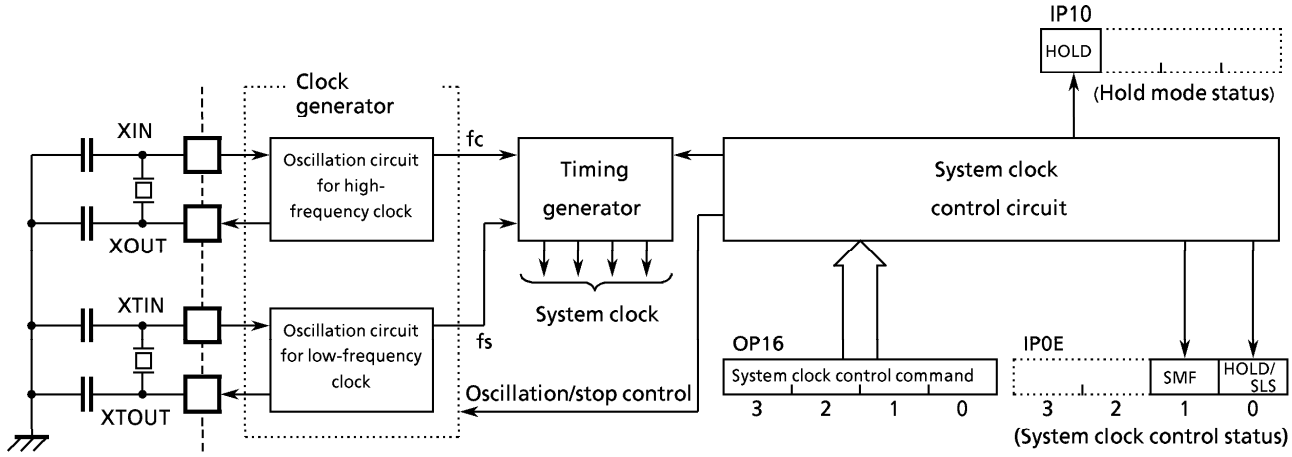


Figure 2-15. System Control Circuit Configuration

2.7.1 Clock Generator (CG)

The clock generator is a circuit to generate the fundamental clock, which is the reference for the system clock supplied to the CPU and peripheral hardware. The processor incorporates two types of oscillator circuits: one for the high-frequency clock and one for the low-frequency clock. To reduce power dissipation, switch to low-speed operation using the low-frequency clock in the system clock control circuit or completely stop the CPU.

A high-frequency clock and low-frequency clock can be easily obtained by connecting an oscillator respectively to the XIN and XOUT pins, or the XTIN and XTOUT pins. A clock can also be input from an external oscillator.

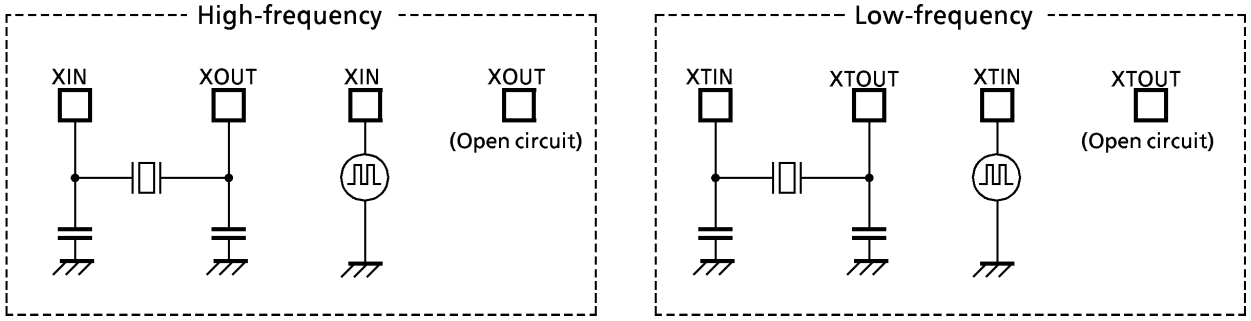


Figure 2-16. Oscillator Connection Examples

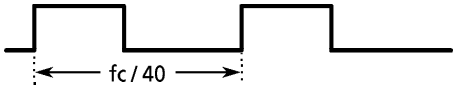
* Precise adjustment of oscillation frequency

The hardware does not provide the capability to directly monitor the fundamental clock externally. However, if interrupts are inhibited and the watchdog timer is halted, output pulses can be programmed to a port with a fixed frequency. The pulse can be monitored and the frequency adjusted. Create such a program for use in adjusting the oscillator frequency in systems that require such adjustment.

Example: Output a pulse (fc: 40Hz) to R70 for monitoring and adjustment of high-frequency oscillation.

```
SFCCHK :    SET     %OP07, 0
            CLR     %OP07, 0
            BSS     SFCCHK
```

Output waveform



2.7.2 Timing Generator (TG)

(1) Timing generator configuration

The timing generator consists of a 3-step prescaler, a 19-step binary counter, and an instruction cycle generator circuit. The input step and the clock input to the timing generator are set as follows according to the operating mode selected by the system clock control command register (OP16). The binary counter is cleared to 0 at reset. However, the prescaler is not cleared.

1. In single-clock mode
   The high-frequency clock (fc) is input to the timing generator first step. In this mode, SLCK (bit 2 of command register OP16) must be cleared to 0.
2. In dual-clock mode
   In this mode, the fc is input at the first step. The seventh and eighth steps are separated. The clock input to the eighth step can be selected by SLCK (bit 2 of command register OP16). Setting SLCK inputs fs, while clearing SLCK inputs fc/2^7. In slow mode, the oscillation of the high-frequency clock stops and fs is automatically input to the eighth step. The divider output from step 1 to step 7 is halted. Figure 2-17 shows the timing generator configuration.

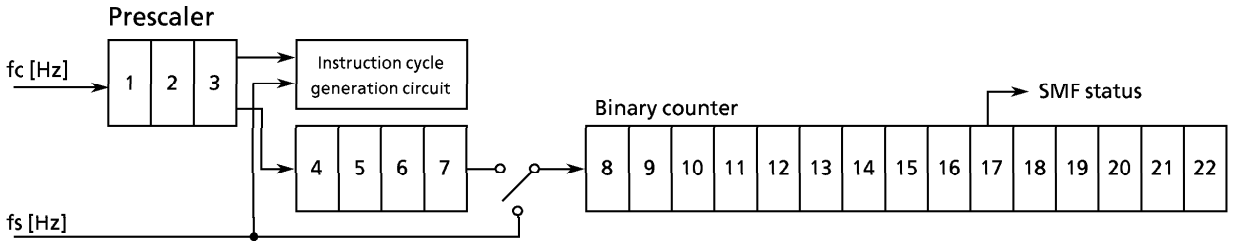


Figure 2-17. Timing Generator Configuration

(2) Timing generator functions

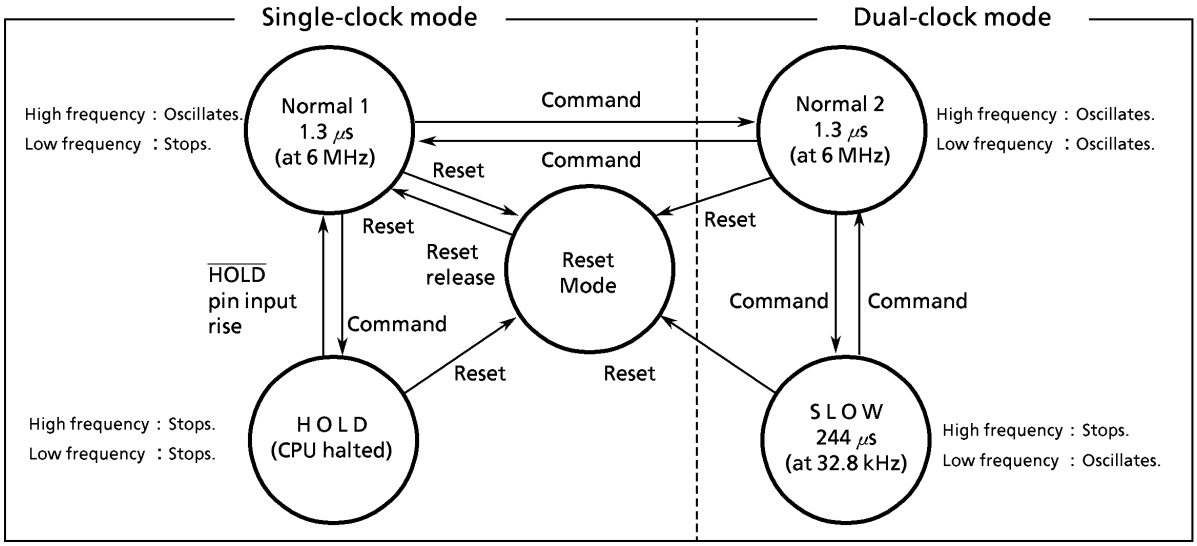
The timing generator functions are:

1. Generate instruction cycles
2. Generate internal pulses for the interval time
3. Generate internal serial clocks for serial interface
4. Warming-up counter when a hold is released
5. Generate clock source for watchdog timer

2.7.3 System Clock Control Circuit

The system clock control circuit controls the timing generators. It also operates or halts the oscillator circuits for the high-frequency clock and low-frequency clock.

The CPU can operate in single-clock or dual-clock mode depending on the command.



Note: Normal 1 and 2 operating modes are known collectively as normal operating mode.

Figure 2-18. Operating Mode Transition Dialgram

(1) System clock control

The command register (OP16) controls system clock. At reset, the register is set to 0, selecting single-clock mode. When operating modes are switched, all states can be read from the status register (IP0E).

System clock control command register
(port address : OP16 initial value : 0000_B)

3	2	1	0
DCLK3	SCLK	DCLK1	DWUT

DCLK3/DCLK1	Selects operating mode. (Note 2)
00	: Single-clock mode (Normal 1 mode)
01	: Do not use.
10	: Dual-clock mode (Normal 2 mode)
11	: Dual-clock mode (SLOW mode)

SLCK	Selects input clock for TG step 8. (Note 3)
	Example: fc = 4.19 MHz
0	: fc/2^7 [Hz] ····· 32.8 [kHz]
1	: fs

DWUT	Warming-up time setting (Note 4)
	Example: fc = 4.19 MHz, fs = 32.8 kHz
0	: 2^8/ fs + 2^9 / fc [s] ··· 7.9 [ms]
1	: 2^{11}/ fs + 2^9 / fc ··· 62.6

System clock control status register
(port address : IP0E initial value : $00**_B$)

3	2	1	0
(SIOF)	(SEF)	SMF	HOLD/SLS

SMF	Low-frequency clock oscillation status
TG step 17 output (fc/2^{17} or fs/2^{10} Hz)	

HOLD / SLS	Monitors $\overline{HOLD}$ pin status/operating status.
	Single clock / Dual clock
0	: $\overline{HOLD}$ pin is high. (in normal 2 mode)
1	: $\overline{HOLD}$ pin is low. (in slow mode)

Note 1 : fc ; High-frequency clock [Hz]
fs ; Low-frequency clock [Hz]
Note 2 : Note the bit configuration.
Note 3 : Valid only in normal 2 mode.
Note 4 : Valid only when switching to normal 2 mode from slow mode.
Note 5 : Accessing OP16 may advance the divider output after timing generator step 8 by 2^7fc [s] (max.) or 1/fs [s].

Figure 2-19. System Clock Control Command Register/Status Register

(2) Instruction Cycle

Instruction execution and internal hardware operation are synchronized with the fundamental clock.

The minimum unit of instruction execution is called an instruction cycle. Single-cycle instructions require one instruction cycle to execute; two-cycle instructions require two instruction cycles to execute. While three-cycle instructions require three instruction cycles to execute.

Instruction cycles consist of four states (S1 to S4). Each state consists of two fundamental clocks.

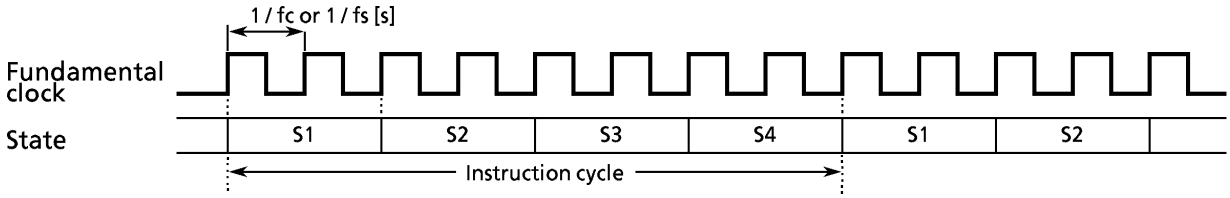


Figure 2-20. Instuction Cycle

2.7.4 Operation Mode

Operating mode consists of single-clock mode, which uses only the oscillator circuit for the high-frequency clock, and dual-clock mode, which uses the two types of oscillator circuits.

(1) Dual-clock mode

Dual-clock mode usually performs normal 2 operation, which generates instruction cycles from the high-frequency clock (fc). If necessary, slow operation can be performed by generating instruction cycles from the low-frequency clock (fs). The following describes switching between normal 2 and slow operation in dual-clock mode. At reset, the command register is initialized to single-clock mode and the low-frequency clock is stopped, so set to normal 2 operation in dual-clock mode. Switching to normal 2 mode starts the low-frequency clock.

a. Switching from normal 2 to slow operation

Setting DCLK1 (OP16 bit 1) to 1 switches to slow operation. However, several seconds are required for the low-frequency clock oscillation to stabilize. Therefore, if switching to slow operation immediately after switching to normal 2 operation, wait until the low-frequency clock oscillation has stabilized or check the oscillation by program. Use the SMF status register (IP0E bit 1) for this check.

If the high-frequency clock ($fc/2^7$) is input to TG step 8, first set SLCK (OP16 bit 2) to 1 and input the low-frequency clock (fs). Then, monitor SMF by program. After confirming the SMF has switched from 1 to 0 to 1, or from 0 to 1 to 0, set DCLK to 1. The high-frequency clock oscillation then stops.

b. Returning to normal 2 operation from slow operation

When clearing DCLK1 to 0, simultaneously set the warming-up time in DWUT (bit 0 of OP16) to return to normal mode. After the set warming-up time has elapsed, normal 2 operation starts.

The current operating mode can be confirmed by monitoring SLS (bit 0 of IP0E).

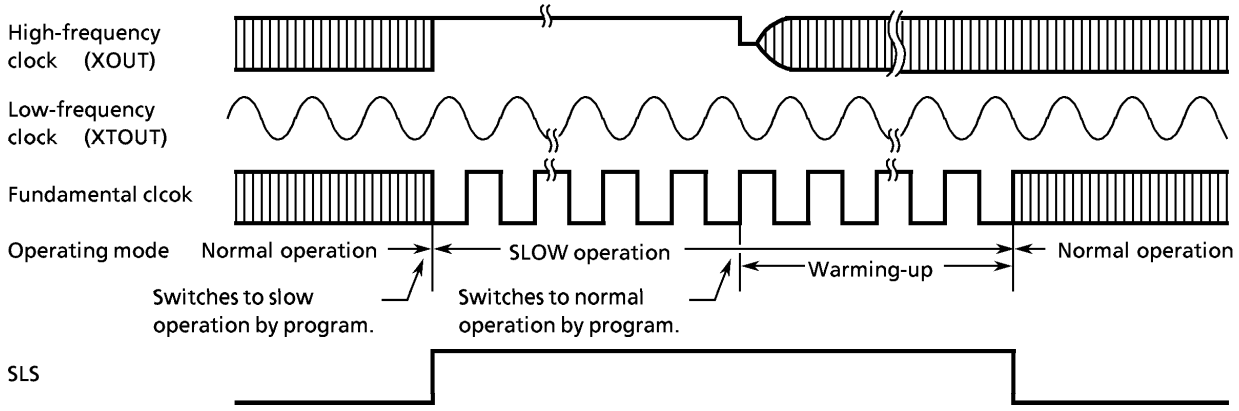


Figure 2-21. System Clock Switching Timing

Note : Slow mode automatically halts the high-frequency clock oscillation, enabling low-power dissipation and low-voltage operation. Power dissipation is reduced in the oscillator circuit and internal hardware; however, pin interface-related power dissipation (dependent on external circuits and the program) is not directly related to low-power dissipation operation. Therefore, consider power dissipation at the system and interface circuit design stage.

Although instruction execution is not cancelled by switching to slow operation, some peripheral hardware functions may be affected. See operational descriptions on this point.

(2) Single-clock mode

This mode uses only the oscillator circuit for the high-frequency clock. R72 (XTIN) and R73 (XTOUT) are set to an I/O port. After reset is released, enters single-clock mode (normal 1 mode). The system clock control command register (OP16) is not required in single-clock mode. Hold mode provides low-power dissipation.

2.7.5 HOLD Operation

The HOLD function halts the operation of the system and preserves the internal status immediately prior to the halt. Processor power dissipation is low during a HOLD. The HOLD function is controlled through the command register (OP10) and the $\overline{\text{HOLD}}$ pin input. The input status of the $\overline{\text{HOLD}}$ pin can be monitored by the status register. In single-clock mode, bit 0 of IP0E can be used. In dual-clock mode, this bit is used for monitoring operating status (SLS) and is therefore not available. Instead, use bit 3 of IP10. This bit can be used in single-clock mode.

(1) Starting HOLD mode

Setting a command in the command register in single-clock mode starts HOLD mode. The states during a hold are as follows:

1. Oscillation is disabled and all internal operation halted.
2. The timing generator binary counter is cleared to 0.
3. Data memory, registers, and port latches maintain their states immediately prior to the hold.
4. The program counter stores the address of the instruction following the instruction used to trigger the hold ([OUT A, %OP10] or [OUT @HL, %OP10]).

HOLD Mode Command Register

(Port address : OP10 initial value : *0**)

3	2	1	0
HLDMS		HWUT	

HLDMS	Sets mode/starts hold.
01	Starts HOLD in edge release mode
11	Starts HOLD in level release mode
*0	(do not use)

HWUT	Warming-up time at HOLD release	Example: fc = 4 MHz
00	2^{18} / fc [s]	65.5 [ms]
01	2^{14} / fc	4.1
10	(do not use)	
11	2^{6} / fc	0.016

HOLD mode status register

(Port address : IP0E initial value : $00**_B$)

3	2	1	0
(SIOF)	(SEF)	(SMF)	HOLD (KE0)

(Port address : IP10 initial value : $****_B$)

3	2	1	0
HOLD (K103)			

HOLD	$\overline{\text{HOLD}}$ pin input status
0	$\overline{\text{HOLD}}$ pin at high level
1	$\overline{\text{HOLD}}$ pin at low level

*Note 1: * ; don't care*
*Note 2: * ; When not using HOLD mode, do not access command register OP10.*

Figure 2-22. HOLD Mode Command Register and Status Register

HOLD mode includes level release mode and edge release mode.

a. Level release mode

Setting the $\overline{\text{HOLD}}$ pin to high releases HOLD mode. This mode is used for capacitor backup or long-duration battery backup when an outage occurs in the main power supply.

The processor does not enter HOLD mode if the $\overline{\text{HOLD}}$ pin input is high when the instruction for starting HOLD mode is executed. Instead, the CPU immediately switches to the release sequence (warming up). Accordingly, when starting a HOLD in level release mode, the program should first check that the $\overline{\text{HOLD}}$ pin input is low (HOLD requested). There are the following two methods to check the $\overline{\text{HOLD}}$ pin input state:

1. Test $\overline{\text{HOLD}}$ (bit 0 of IP0E or bit 3 of IP10).
2. Input $\overline{\text{HOLD}}$ pin input to $\overline{\text{INT1}}$ pin as well and generate an external interrupt 1 request.

Example : Test $\overline{\text{HOLD}}$, then start a HOLD in level release mode (warming-up time is 2^{14} / fc [s]).

```
SHOLDH :    TEST    %IP0E, 0
            B       SHOLDH
            LD      A, #1101B
            OUT     A, %OP10
```

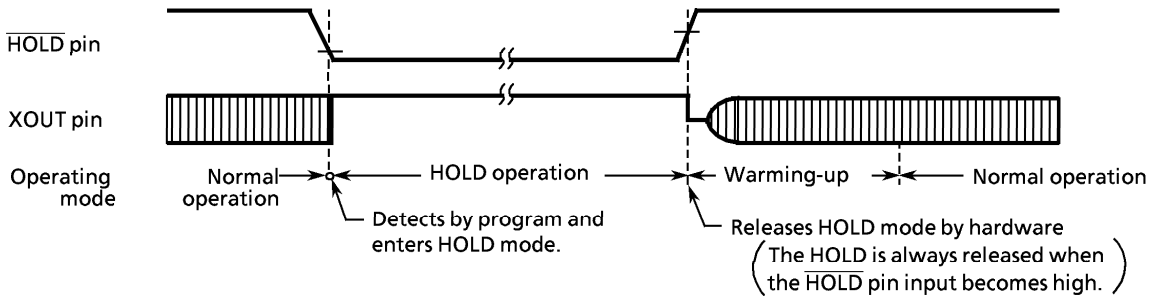


Figure 2-23 (a). Level Release Mode

b. Edge release mode

This mode releases the HOLD at a rising edge of input to the $\overline{HOLD}$ pin. Edge release mode is used for repeating a relatively short programs at a fixed interval. Input a signal at a fixed interval (for example, a clock from a low power dissipation oscillator) to the $\overline{HOLD}$ pin.
In edge release mode, the processor enters HOLD mode even if the $\overline{HOLD}$ pin input is high.

Example: Start HOLD in edge release mode (warming-up time set to 2^{14}/fc [s]).

```
LD      A, #0101B  ;  OP10←0101B
OUT     A, %OP10
```

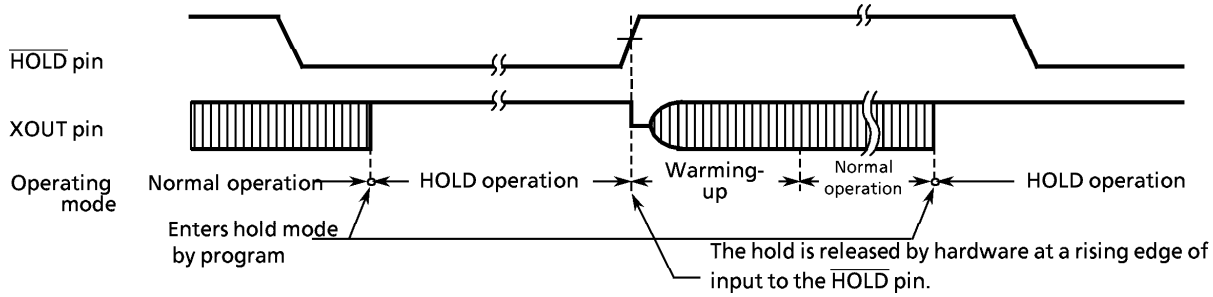


Figure 2-23 (b). Edge Release Mode

Note: HOLD mode reduces power dissipation in the oscillator circuit and internal hardware. However, pin interface-related power dissipation (dependent on external circuits and the program) is not directly related to the hardware operation of the HOLD function. Therefore, consider power dissipation at the system and interface circuit design stage.
Current is minimal when the input level of the CMOS circuit is stable at the power supply voltage level (V_{DD}/V_{SS}). However, current flows if the input level fluctuates beyond the power supply voltage level (about 0.3 to 0.5V). Therefore, when a pin signal is at high impedance because the output transistor at an input/output port (pins with open-drain outputs and input transistors) is disconnected, current may flow through the input transistor. Fix the input level using pull-up resistors.

(2) HOLD mode release

HOLD mode is released in the following sequence:

1. Oscillation begins.
2. Warms up for a sufficient time to stabilize the oscillation. During the warming-up time, internal operations remain halted. The program can select one of three warming-up times to suit the characteristics of the oscillator.
3. When the required warming-up time has elapsed, operation resumes from the instruction following the instruction that started HOLD mode.

* The warming-up time is obtained by dividing the frequency of the high-frequency clock using the timing generator. Therefore, if fluctuations occur in the oscillation at the release of HOLD mode, the warming-up time will vary from that shown in Figure 2-23 Treat the warming-up time as an approximate value only.

HOLD mode can also be released by setting the $\overline{RESET}$ pin to low. In this case a normal reset occurs immediately.

Note: Note the following when releasing a HOLD at low voltage.
Before releasing the hold, raise the power supply voltage to the operation. At this time, the $\overline{RESET}$ pin input becomes high together with the power supply voltage. If a time constant circuit, for example, is connected to the $\overline{RESET}$ pin, the rise in voltage at the $\overline{RESET}$ pin is slower than the rise in power supply voltage. If at this time, the $\overline{RESET}$ pin input voltage exceeds the non-inverted high level, an unintended reset may be automatically generated.

2.8. Interrupt Functions

(1) Interrupt control circuit

Six interrupt sources, two external and four internal, are supported. Prioritized multiple interrupts are also possible.

Interrupt latches (IL_5 to IL_0) are provided to store the requests that accompany the interrupts. The generation of an interrupt request sets the interrupt latch to 1. The interrupt latch requests the CPU to receive the interrupt. The program can selectively enable or disable interrupts using the interrupt enable master flip-flop (EIF) and the interrupt enable register (EIR). If multiple interrupt requests are generated simultaneously, the CPU receives the requests according to the priority defined in the hardware (the highest priority interrupt is received first).

Interrupt source			Hardware priority		Interrupt latch	Interrupt enable/disable condition set by program	Entry Address
External	External interrupt 1	(INT1)	(high)	1	IL_5	EIF = 1	0002_H
Internal	Serial interface interrupt	(ISIF)		2	IL_4	EIF = 1, EIR_3 = 1	0004_H
	Timer/counter overflow interrupt	(IOVF)		3	IL_3	EIF = 1, EIR_2 = 1	0006_H
	Timer/counter interrupt	(ITC)		4	IL_2	EIF = 1, EIR_1 = 1	0008_H
	Interval timer interrupt	(ITMR)		5	IL_1		$000A_H$
External	External interrupt 2	(INT2)	(low)	6	IL_0	EIF = 1, EIR_0 = 1	$000C_H$

Table 2-2. Interrupt Sources

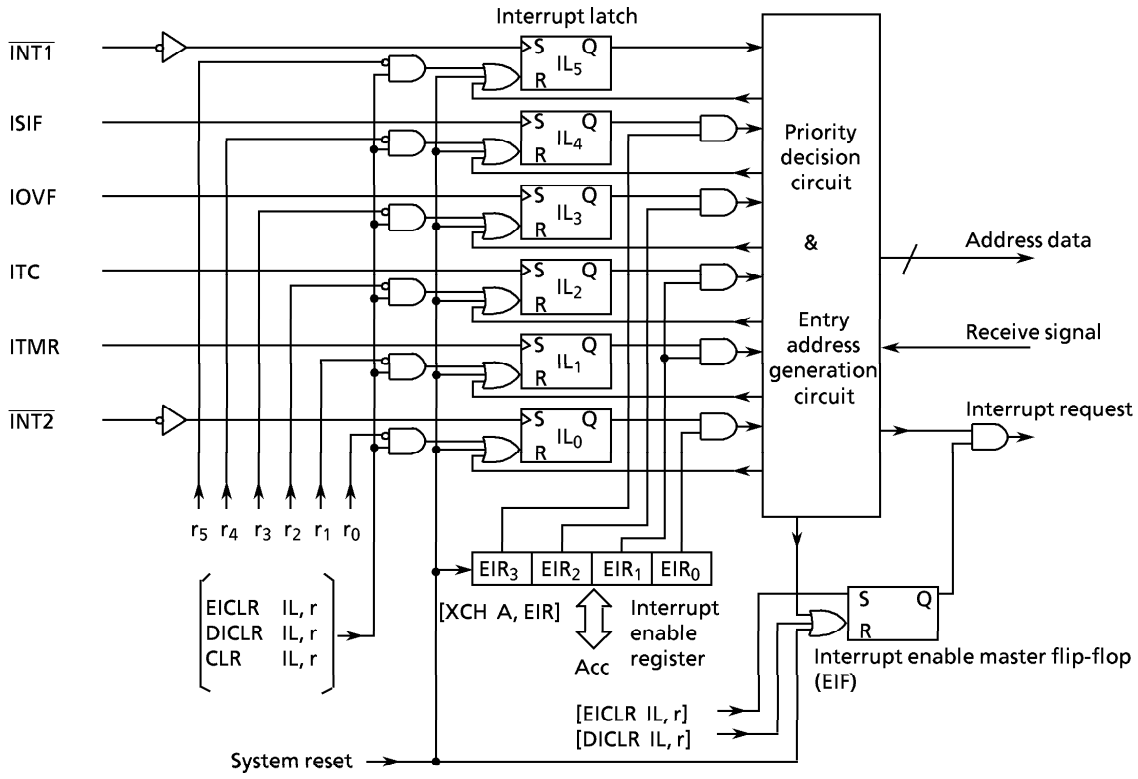


Figure 2-24. Interrupt Control Circuit Block Diagram

a. Interrupt enable master flip-flop (EIF)

This flip-flop enables or disables all interrupts. Clearing the flip-flop to 0 disables reception of all interrupts. Setting the flip-flop to 1 enables reception of all interrupts.

When an interrupt is received, the flip-flop is cleared to 0, temporarily disabling reception of subsequent interrupts. After the interrupt service program is executed, the flip-flop is reset to 1 by the execution of the interrupt return instruction, again enabling interrupts.

Use the [EICLR IL, r] or [DICLR IL, r] instruction to set or clear the EIT. At reset, the flip-flop is initialized to 0.

b. Interrupt enable register (EIR)

The EIR is a 4-bit register that enables or disables reception of the ISIF, IOVF, ITC, ITMR, and INT2 interrupts. If the applicable bit of this register is 1, the interrupt is enabled. If the applicable bit is 0, the interrupt is disabled. The ITC and ITMR interrupt sources share bit 1 of the EIR (EIR_1).

Use the [XCH A,EIR] instruction to read or write the EIR. At reset, the EIR is initialized to 0.

c. Interrupt latch (IL)

An interrupt latch is provided for each interrupt. The IL is set to 1 by the generation of an interrupt request, whereupon the IL requests that the CPU receive the interrupt. Immediately the interrupt is accepted, the latch is cleared to 0. At reset, all interrupt latches are initialized to 0.

The interrupt latches can be individually cleared by the interrupt latch instructions [EICLR IL,r], [DICLR IL, r], and [CLR IL, r]. Interrupt requests can be cancelled or initialized within the program. When the value of the instruction field, r, is 0, the interrupt latch is cleared. When the value is 1, the interrupt latch value is maintained. The interrupt latch cannot be set by an instruction.

Example 1: IOVF, INT1, INT2 interrupts.

```
LD      A, #0101B
XCH     A, EIR
EICLR   IL, 111111B
```

Example 2: Clear interrupt latches other than ISIF and set EIF to 1.

```
EICLR   IL, 010000B
```

(2) Interrupt processing

Interrupt requests are saved until the interrupt is received, or until the interrupt latch is cleared by a reset or an interrupt latch manipulation instruction. Interrupt receive processing is executed in two instruction cycles after the instruction currently under execution is completed. The interrupt processing program terminates on execution of the interrupt return instruction [RETI].

The processor performs the following operations to receive an interrupt:

1. Saves the contents of the program counter and flags to the stack.
2. Sets the interrupt entry address for the interrupt in the program counter.
3. Sets the status flag to 1.
4. Clears the interrupt enable master flip-flop (EIF) to 0 to temporarily disable further interrupts.
5. Clears the interrupt latch for the received interrupt to 0.
6. Branches to the instruction stored at the interrupt entry address. (The program memory at the interrupt entry address usually contains a branch instruction to the service program for the interrupt. As the interrupt entry addresses are allocated every 2 bytes, the addresses typically cannot contain a long branch instruction. Accordingly, the interrupt service program is allocated to program memory addresses 0000 - $0FFF_H$.)

To handle multiple interrupts, set the EIF to 1 in the interrupt service program. The EIR determines which interrupts can be received at this time. However, as the INT1 interrupt cannot be disabled by the EIR, disable processing of INT1 by software.

Example : Disable the interrupt processing for INT1 by software (use bit 0 of RAM [05H], the interrupt processing disable switch).
If RAM $[05H]_0 = 1$, processing returns without processing the interrupt.

```
PINT1 :   TEST   05H, 0
          B      SINT1
          RETI
SINT1 :   ⋮
```

The interrupt return instruction performs the following operations:

1. Restores the contents of the program counter and flags from the stack.
2. Sets the interrupt enable master flip-flop (EIF) to 1 to re-enable interrupts. This means that the next interrupt can be received immediately after execution of the interrupt return instruction.

Note : If the interrupt processing time is longer than the interval between interrupt requests, only the interrupt service program is executed (the main program is not executed).

In interrupt processing, the program counter and flags are automatically saved or restored. However, the accumulator and other registers (for example, the HL register pair, DMB, and DC) are not automatically saved or restored. If the accumulator and other registers are used in the interrupt handler, their contents should be saved on entry and restored on exit. When multiple interrupts are generated, be careful that the save areas in RAM do not overlap.

Example 1: Save and restore the accumulator and HL register pair.

```
        XCH   HL, GSAV1       ; RAM [GSAV1] ↔ HL
        XCH   A, GSAV1 + 2    ; RAM [GSAV1 + 2] ↔ Acc
```

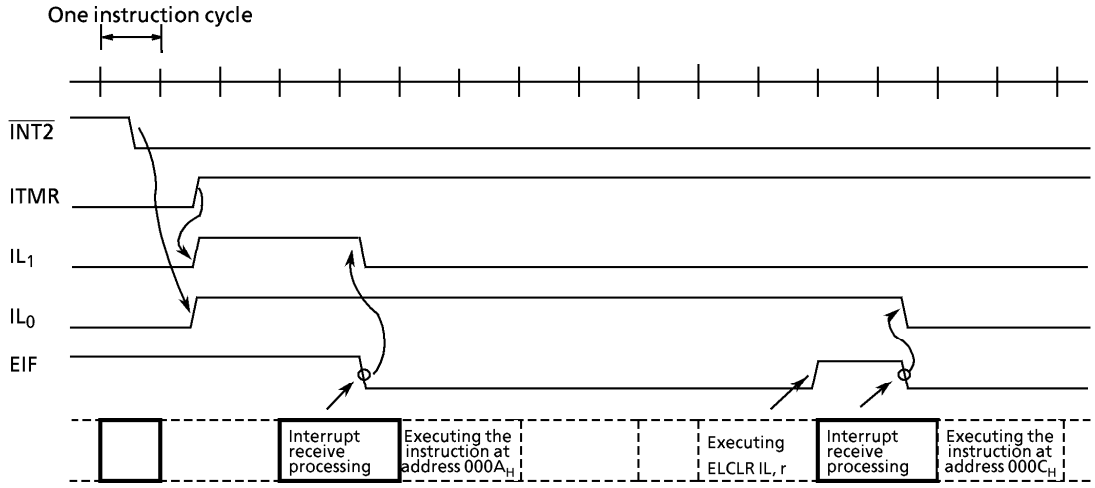
Note : The lower two bits of GSAV1 must be set to 0.

Example 2: Save the DMB to bit 0 at address GSAV3 in data memory (page zero).

```
        CLR   GSAV3, 0        ; RAM [GSAV3] 0 ← 0
        TEST  DMB             ; If DMB = 0, skips.
        B     SKIPS
        SET   GSAV3, 0        ; RAM [GSAV3] 0 ← 1
SKIPS :
```

Example 3: Restore the DMB from bit 0 at address GSAV3 in data memory (page zero).

```
        CLR   DMB             ; DMB ← 0
        TEST  GSAV3, 0        ; f RAM [GSAV3]0 = 0, skips.
        B     SKIPR
        SET   DMB             ; DMB ← 1
SKIPR :
```



Note 1: Assumes no other interrupt requests and that EIR = 0011_B.
Note 2: The "r" of instruction [EICLR IL,r] is assumed to be 111111_B.
Note 3: [ ] indicates instruction execution.

Figure 2-25. Interrupt Timing Chart (Example)

(3) External interrupt

The interrupt latches for external interrupts INT1 and INT2 are set at falling edges of the corresponding pin inputs ($\overline{INT1}$ and $\overline{INT2}$).

As the INT1 interrupt cannot be disabled by the EIR, this interrupt is always received as long as interrupts are enabled (EIF = 1). Consequently, use this interrupt for high priority interrupts, for example, emergency interrupts. Interrupt INT1 is generated at the falling edge of R82 ($\overline{INT1}$) pin input. Thus, when using the pin as an I/O port, write an interrupt return instruction [RETI] at the interrupt entry address to perform dummy interrupt processing.

Interrupt INT2 can be enabled or disabled by the EIR. When using the R80 ($\overline{INT2}$) pin as an I/O port, interrupt INT2 is generated at the falling edge of the pin input. However, simply clearing EIR bit 0 to 0 disables interrupt INT2.

The external interrupt inputs are hysteresis inputs. Both high and low levels must be maintained for at least two instruction cycles for the interrupt to be detected correctly.

2.9 Reset Function

If the $\overline{\text{RESET}}$ pin is maintained at a low level for a minimum of three instruction cycles with stable oscillation and the power supply voltage within the operating voltage range, a reset occurs and the internal state is initialized.
The reset is released when the $\overline{\text{RESET}}$ pin returns to high level. This restarts execution from address 0000H.
The $\overline{\text{RESET}}$ pin is a hysteresis input with a pull-up resistor (typ. 220kΩ). Connecting a capacitor and a diode externally provides a simple power-on reset.

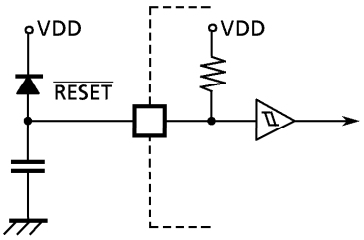


Figure 2-26. Simple Power-On Reset Circuit

Internal hardware		Initial value	Internal hardware	Initial value
Program counter	(PC)	0000_H	TG binary counter	"0"
Status flag	(SF)	1	Output latch	See the I/O
Data memory bank selector	(DMB)	0	(Input/output or output port)	circuit section.
Interrupt enable master flip-flop	(EIF)	0		See the
Interrupt enable register	(EIR)	0_H	Command registers	description of
Interrupt latch	(IL)	"0"		each command

Table 2-3. Internal Hardware Initialization by Reset

2.9.1 Warm Start

Warm start - reset while saving contents of data memory after HOLD is released - is not supported by hardware. However, warm start can be performed in the following way.

1. Backs up the voltage supplied to the VDD pin.
2. Also applies to the $\overline{\text{HOLD}}$ pin a waveform in sync with the change in power supply voltage.
3. Detects a power drop and sets to HOLD mode during power cut.
4. After release from HOLD, resets by program using sink open-drain (initial: Hi-Z) output port.
5. Applies the POWER ON detect signal to the input port and skips the RAM initialize routine.

Figure 2-27 is an example of a warm start circuit.

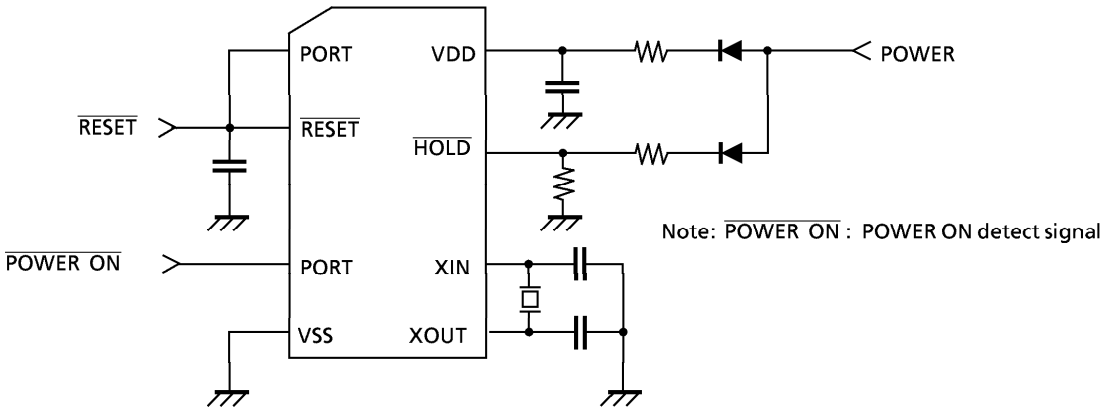


Figure 2-27. Warm Start Circuit Example

3. Peripheral Hardware Functions

3.1 Ports

The 13 input/output instructions transfer data with external circuits. The instructions also transfer commands, read states, and data with internal circuits.

The four types of ports are:
1. Input/output ports ; Transfer data with external circuits.
2. Command registers ; Control internal circuits.
3. Status registers ; Read status signals from internal circuits.
4. Data registers ; Transfer data with internal circuits.

Ports are assigned a port address (00 - $1F_H$). Individual ports are selected by specifying the port address in the input/output instruction. Table 3-1 shows the port address allocation and the input/output instructions that access the ports.

3.1.1 Input/Output Timing

(1) Input timing

The processor reads external data from input ports or input/output ports at the S3 state of the second cycle of the input instruction (a 2-cycle instruction). As this timing cannot be identified externally, the program must deal with transient input data such as chatter.

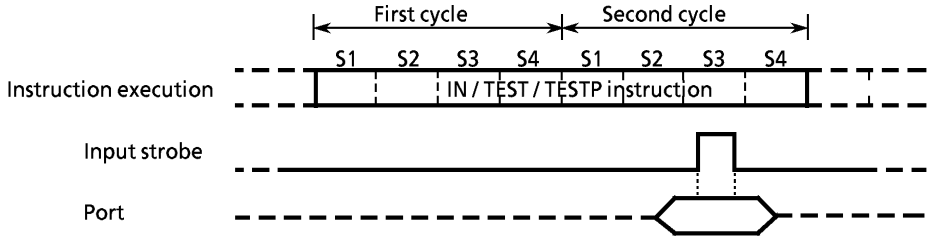


Figure 3-1. Input Timing

(2) Output timing

The processor outputs data to output ports or input/output ports at the S4 state of the second cycle of the output instruction (a 2-cycle instruction).

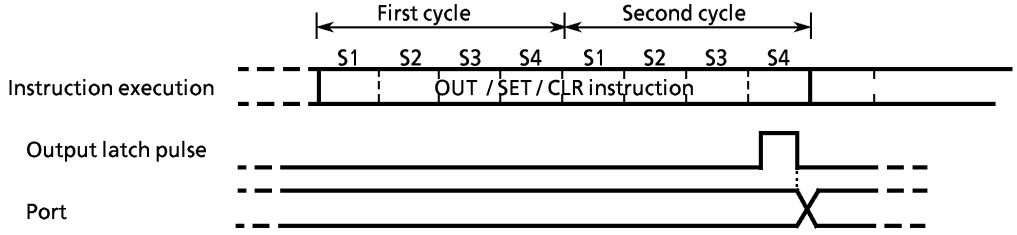


Figure 3-2. Output Timing

PORT ADDRESS (**)	PORT		INPUT/OUTPUT INSTRUCTION						
	INPUT (IP**)	OUTPUT (OP**)	IN %p, A IN %p, @HL	OUT A, %p OUT @HL, %p	OUT #k, %p	OUTB @HL	SET %p, b CLR %p, b	TEST %p, b TESTP %p, b	SET @L CLR @L TEST @L
00_H	K0 input port	E^2PROM command	○	○	○	–	–	○	–
01	E^2PROM read data	E^2PROM write data	○	○	○	–	–	○	–
02	E^2PROM read data	E^2PROM write data	○	○	○	–	–	○	–
03	——	——	–	–	–	–	–	–	–
04	R4 input port (analog input)	R4 output port	○	○	○	–	○	○	○
05	R5 input port (analog input)	R5 output port	○	○	○	–	○	○	○
06	R6 input port	R6 output port	○	○	○	–	○	○	○
07	R7 input port (XTIN)	R7 output port ($\overline{WTO}$ output, XTOUT)	○	○	○	–	○	○	○
08	R8 input port (UART, external interrupt input)	R8 output port (UART output)	○	○	○	–	○	○	–
09	R9 input port (SIO input)	R9 output port (SIO output)	○	○	○	–	○	○	–
0A	RA input port (capture, external input)	RA output port (compare output)	○	○	○	–	○	○	–
0B	RB input port (PWM trigger input)	RB output port (PWM output)	○	○	○	–	○	○	–
0C	UART receive data buffer	UART transmit data buffer	○	○	○	–	–	–	–
0D	UART status	Output compare 0 command	○	○	○	–	–	–	–
0E	Status input (SIOF, SEF, SMF, KE0)	Output compare 1 command	○	○	○	–	–	○	–
0F	SIO receive data buffer	SIO transmit data buffer	○	○	○	–	–	–	–
10_H	$\overline{HOLD}$ pin status	HOLD mode command	○	○	–	–	–	–	–
11	——	——	–	–	–	–	–	–	–
12	A/D conversion value	A/D conversion start flag/andlog input channel select command	○	○	–	–	–	–	–
13	A/D, E^2PROM status	E^2PROM address	○	○	–	–	–	–	–
14	——	E^2PROM address	–	○	–	–	–	–	–
15	——	Watchdog timer command	–	○	–	–	–	–	–
16	——	System clock command	–	○	–	–	–	–	–
17	——	PWM command 1	–	○	–	–	–	–	–
18	——	PWM command 2	–	○	–	–	–	–	–
19	——	Interval timer interrupt command	–	○	–	–	–	–	–
1A	——	UART command 1	–	○	–	–	–	–	–
1B	Serial I/F interrupt status	UART command 2	○	○	–	–	–	–	–
1C	——	Timer/counter command	–	○	–	–	–	–	–
1D	IC/OC interrupt status	Input capture command	○	○	–	–	–	–	–
1E	——	SIO command 1	–	○	–	–	–	–	–
1F	——	SIO command 2	–	○	–	–	–	–	–

Note 1 : "—— " indicates a reserved address. Do not access reserved addresses, as they cannot be used by user programs.

Note 2 : Do not use the 5- to 8-bit data conversion instruction [OUTB @HL] because ports 1 and 2 are not supplied.

Note 3 : Port address 00 is assigned port K0 for input commands; the E^2PROM command register, for output commands.

Table 3-1. Port Address Allocation and Corresponding Input/Output Instructions

3.1.2 Input/Output Ports

TMP47E885AF has ten input/output ports and 36 pins.

1. Port K0 ; 4-bit input
2. Port R4, R5 ; 4-bit input/output (also used for A/D converter analog input)
3. Port R6 ; 4-bit output/input
4. Port R7 ; 4-bit input/output (used as a low-frequency oscillator connecting pin and for watchdog timer output)
5. Port R8 ; 4-bit input/output (used for external interrupt input and as asynchronous serial port)
6. Port R9 ; 3-bit input/output (also used as a synchronous serial port)
7. Port RA ; 4-bit input/output (used as a capture input pin, a counter input pin, and a compare output pin)
8. Port RB ; 4-bit input/output (also used as the PWM output pin)
9. Port KE ; 1-bit sense input (also used as the HOLD mode request/release signal input)

All output ports incorporate latches, which store the output data. The input ports do not have latches. Therefore, data input from an external source should be either stored externally or processed after several reads.

(1) Port K0 (K03 to K00)

K0 is a 4-bit input-only port.

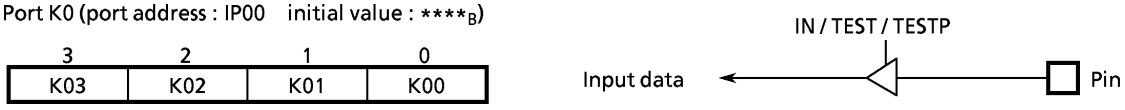


Figure 3-3. Port K0

(2) R4 (R43 to R40), R5 (R53 to R50), R6 (R63 to R60), R7 (R73 to R70) Port

These are 4-bit latched input/output ports. When using as input ports, set the latch to 1. At reset, the latch is initialized to 1. The 16 pins of these four ports use the L register indirect addressing bit manipulation instructions [SET @L], [CLR @L], and [TEST @L] to set, clear, or test the bit specified by the L register. Table 3-2 shows the correspondence between the contents of the L register and pins (input/output ports).

Example : Clear pin R43 using a bit manipulation instruction in L register indirect addressing mode.

```
LD      L, #0011B       ; Sets the R43 address in the L register.
CLR     @L              ; R43 ← 0
```

L register				Pin
3	2	1	0	
0	0	0	0	R40
0	0	0	1	R41
0	0	1	0	R42
0	0	1	1	R43

L register				Pin
3	2	1	0	
0	1	0	0	R50
0	1	0	1	R51
0	1	1	0	R52
0	1	1	1	R53

L register				Pin
3	2	1	0	
1	0	0	0	R60
1	0	0	1	R61
1	0	1	0	R62
1	0	1	1	R63

L register				Pin
3	2	1	0	
1	1	0	0	R70
1	1	0	1	R71
1	1	1	0	R72
1	1	1	1	R73

Table 3-2. Correspondence Between L Register Contents and I/O Port Bits

(a) Ports R4 and R5

Ports R4 and R5 are also used for A/D converter analog input. When using as input ports or for analog inputs, set the latch to 1. To maintain accuracy during A/D conversion, do not execute output instructions.

At reset, the latch is set to 1. Pin R40 (AIN0) is selected for analog input.

Port R5
(Port address : OP05 / IP05 initial : 1111_B)

3	2	1	0
R53 (AIN7)	R52 (AIN6)	R51 (AIN5)	R50 (AIN4)

Port R4
(Port address : OP04 / IP04 initial : 1111_B)

3	2	1	0
R43 (AIN3)	R42 (AIN2)	R41 (AIN1)	R40 (AIN0)

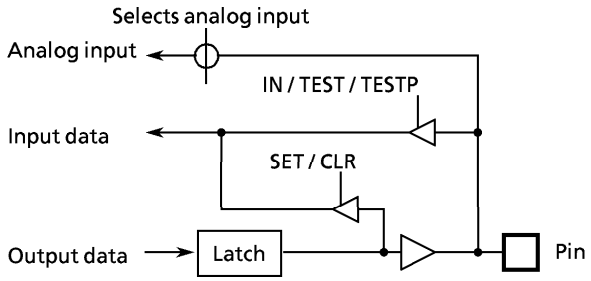


Figure 3-4 (a). Ports R4 and R5

(b) Port R6

Port R6 is a standard input/output port.

Port R6 (port address : OP06/IP06 initial value : 1111_B)

3	2	1	0
R63	R62	R61	R60

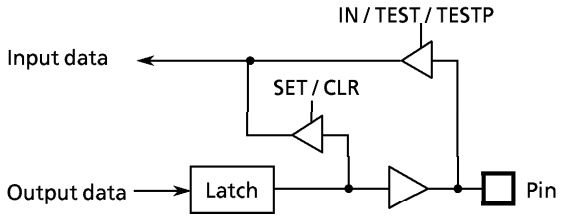


Figure 3-4 (b). Port R6

(c) R7ポート

Port R7 is used as both a watchdog timer output pin ($\overline{WTO}$) and low-frequency oscillator connecting pins (XTIN and XTOUT).

In dual-clock mode, connect the low-frequency oscillator (typ. 32.768kHz) to pins R72 (XTIN) and R73 (XTOUT). In single-clock mode, pins R72 and R73 can be used as standard I/O ports.

When using the watchdog timer, R71 ($\overline{WTO}$) is used as the watchdog timer output pin. At that time, set the R71 output latch to 1. When using R71 as a standard I/O port, disable the watchdog timer. R70 is a standard input/output pin.

Port R7 (port address : OP07 / IP07 initial value : 1111_B)

3	2	1	0
R73 (XTOUT)	R72 (XTIN)	R71 ($\overline{WTO}$)	R70

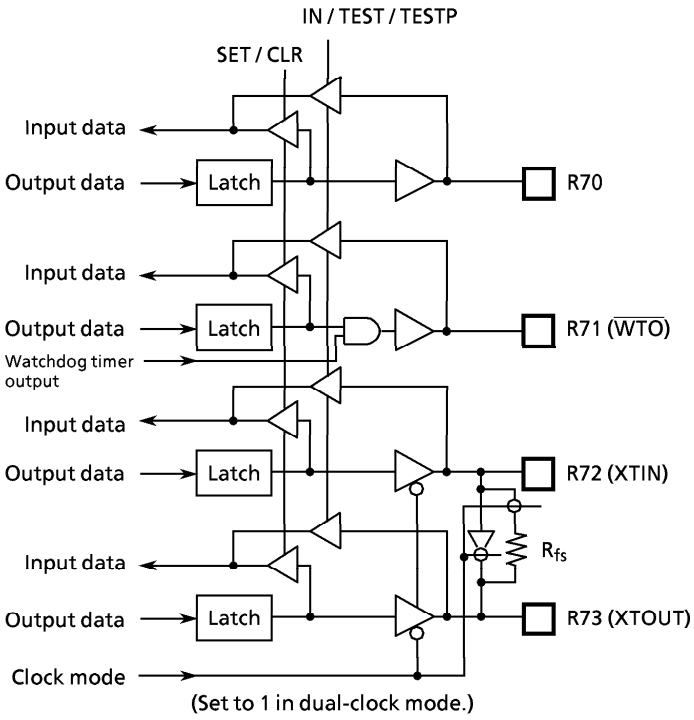


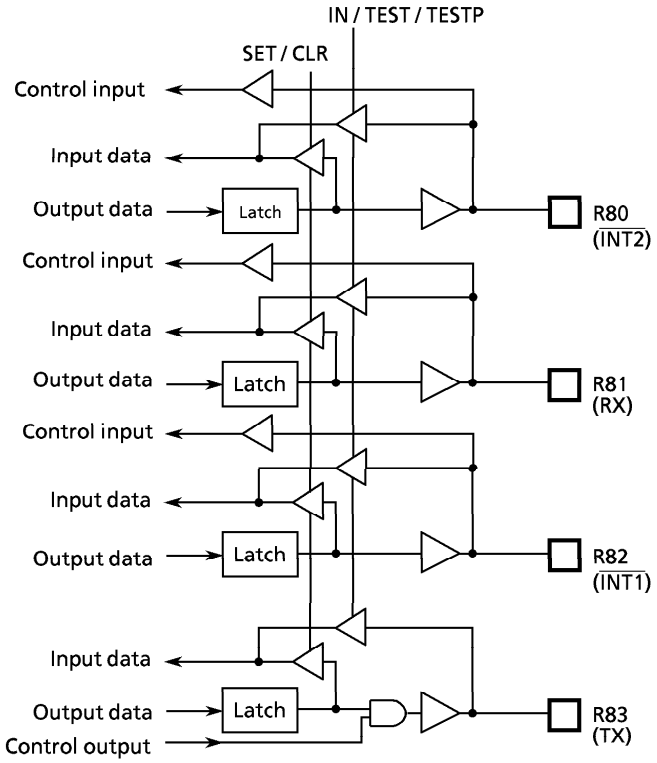
Figure 3-4 (c). Port R7

(3) Port R8 (R83 to R80)

R8 is a 4-bit latched I/O port. When using as an input port, set the latch to 1. At reset, the latch is initialized to 1.

Port R8 is used as both an external interrupt input pin and asynchronous serial port. When using as an external interrupt input pin and asynchronous serial port RX and TX pins, set the latch to 1. When using as a standard I/O port, disable either external interrupts or the asynchronous serial ports.

Note: When using pin R82 ($\overline{INT1}$) as an I/O port, external interrupt 1 is generated when a falling edge is detected at the pin input. If the interrupt enable mask flip-flop is enabled, an interrupt request is always received. Therefore, execute dummy interrupt processing (execute only the interrupt return instruction [RETI]). With pin R80 ($\overline{INT2}$), external interrupt 2 is generated in the same circumstances. However, simply clear in advance bit 0 (EIR_0) in the interrupt enable register to 0 to disable the interrupt request.



Port R8 (port address : OP08/IP08 initial value : 1111_B)

3	2	1	0
R83 (TX)	R82 ($\overline{INT1}$)	R81 (RX)	R80 ($\overline{INT2}$)

Figure 3-5. Port R8

(4) Port R9 (R92 to R90)

R9 is a 3-bit latched I/O port. When using as an input port, set the latch to 1. At reset, the latch is initialized to 1. Port R9 is also used as a synchronous serial port. When using as a synchronous serial port, set the latch to 1. When using as a standard I/O port, disable the synchronous serial port.

While pin R93 is not provided, executing instructions to set or clear pin R93 ([SET %OP09,3] or [CLR %OP09,3]) affects the internal CPU. Therefore, do not use R93 set/clear instructions. However, input/output instructions other than set/clear instructions can be used. If an input instruction is executed, undefined values are read at input execution.

Port R9 (port address : OP09/IP09 initial value : 1111_B)

3	2	1	0
	R92 ($\overline{SCK}$)	R91 (SO)	R90 (SI)

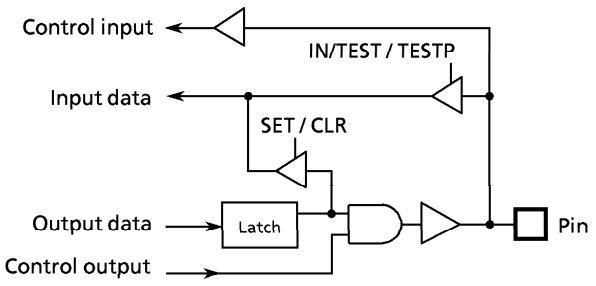


Figure 3-6. Port R9

(5) Port RA (RA3 to RA0)

RA is a 4-bit latched I/O port. When using as an input port, set the latch to 1. At reset, the latch is initialized to 1.

The RA port is also used for the timer/counter. When using as a timer/counter input pin, set the latch to 1. When using as a standard I/O port, disable the timer/counter.

Port RA (port address : OP0A/IP0A initial value : 1111_B)

3	2	1	0
RA3 (T2/IC1)	RA2 (OC1)	RA1 (T1/IC0)	RA0 (OC0)

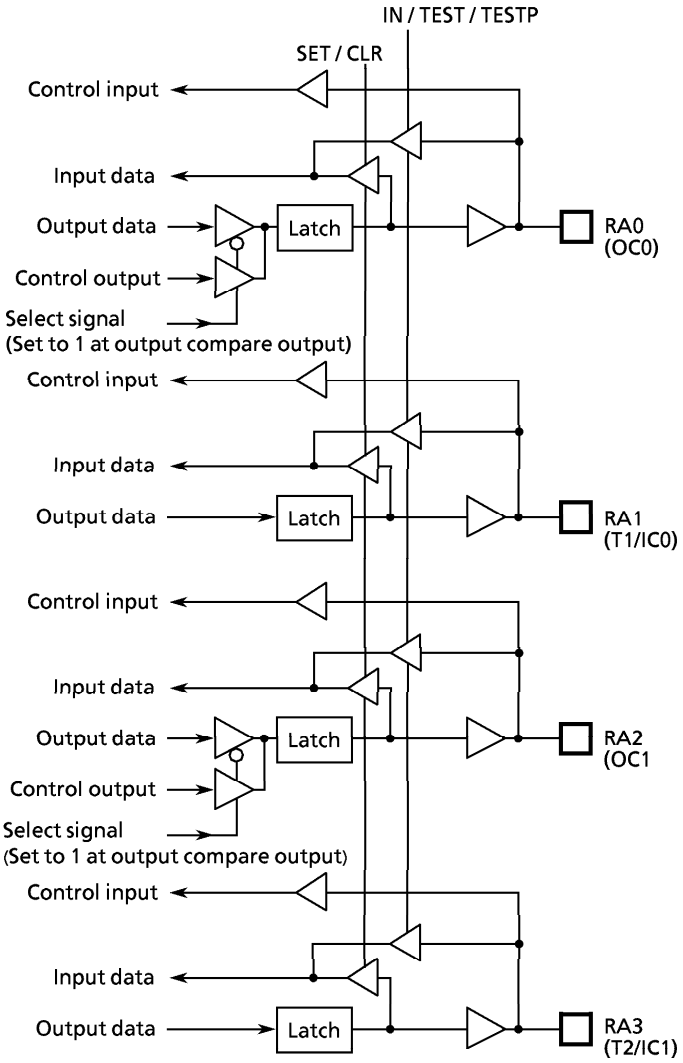


Figure 3-7. Port RA

(6) Port RB (RB3 to RB0)

RB is a 4-bit latched I/O port. When using as an input port, set the latch to 1. At reset, the latch is initialized to 1.

The RB port is also used for the PWM. When using as a PWM trigger input pin, set the latch to 1. When using as a standard I/O port, disable the PWM.

Port RB (port address : OP0B/IP0B initial value : 1111_B)

3	2	1	0
RB3 (PWM1)	RB2 (TRG1)	RB1 (PWM0)	RB0 (TRG0)

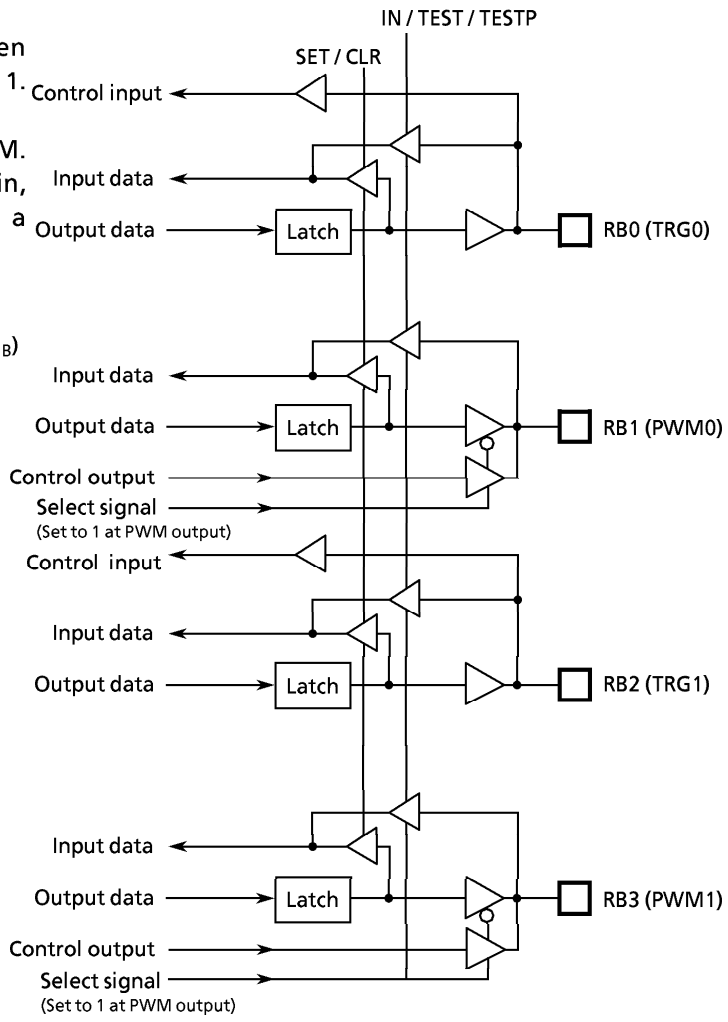


Figure 3-8. Port RB

(7) Port KE ($\overline{KE0}$)

Port KE is a 1-bit sense input port. The port is also used as a HOLD request/release signal input pin ($\overline{HOLD}$ pin). Port KE is allocated to the LSB of port address IP0E, and the MSB of port address IP10. Port KE data are processed as inverted data. For example, if an input instruction is executed when the pin is at high level, 0 is read.

In dual-clock mode, KE0 input is used to monitor the SLOW operation and is therefore not available. When monitoring the $\overline{HOLD}$ pin status, use the K103 input instead.

Example: Wait until the $\overline{KE0}$ pin is at low level.

```
SWAIT :   TEST    %IP0E, 0        ; Wait if KE0 pin = High
          B       SWAIT
```

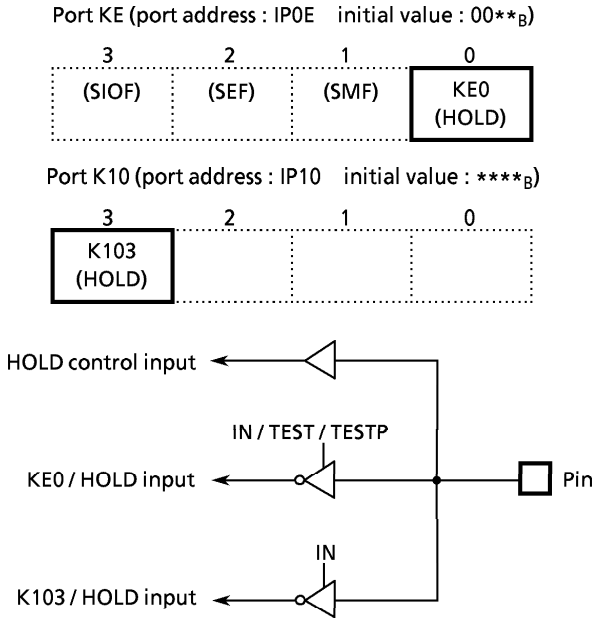


Figure 3-9. Port KE

3.2 Interval Timer

An interval timer interrupt function to generate fixed frequency interrupts is built into the system. A command register (OP19) controls the interval timer interrupts (ITMR).

Interval timer interrupts are generated after the first rising edge of output from the timing generator after a command is set in the command register. As the interval timer is not cleared by the command, the first interrupt may be generated earlier than the specified interrupt cycle.

Interval timer interrupt command register (port address : OP19 initial value : 0000_B)

3	2	1	0
TMRE		TMRF	

TMRE	Interrupt enable/disable
00	Disable
01	Enable
1*	Do not use

TMRF	Interrupt frequency
00	$fc/2^{11}$ or $fs/2^4$ [Hz]
01	$fc/2^{13}$ or $fs/2^6$
10	$fc/2^{15}$ or $fs/2^8$
11	$fc/2^{17}$ or $fs/2^{10}$

*Note 1 : * ; don't care*
Note 2 : fc ; Low-frequency clock [Hz]
fs ; Low-frequency clock [Hz]

Figure 3-10. Interval Timer Interrupt Command Register

Example : Set interval timer interrupt frequency to $fc/2^{15}$ [Hz].

```
LD      A, #0110B
OUT     A, %OP19
```

TMRF	Normal 1 operation	Normal 2 operation (TG Step 8 input)		SLOW operation	At fc = 4.194304 MHz, fs = 32.768 kHz
		SLCK = 0 ($fc/2^7$)	SLCK = 1 (fs)		
00	$fc/2^{11}$ [Hz]		$fs/2^4$ [Hz]	Do not use	2048 [Hz]
01	$fc/2^{13}$		$fs/2^6$	Do not use	512
10	$fc/2^{15}$		$fs/2^8$	Do not use	128
11	$fc/2^{17}$		$fs/2^{10}$	$fs/2^{10}$	32

Table 3-3. Interrupt Frequency Example

3.3 Timer/Counter

A channel of a 16-bit timer/counter is built into the system. The data registers used by the timer/counter (TC, IC1, IC0, OC1, OC0) are allocated to a RAM address in units of 4 bits. Use RAM manipulation instructions to set the initial value or read the timer/counter contents. The data memory at the addresses corresponding to these registers cannot be accessed as RAM and is therefore not available for saving user data.

3.3.1 Timer/Counter Configuration

Figure 3-11 shows the timer/counter configuration.

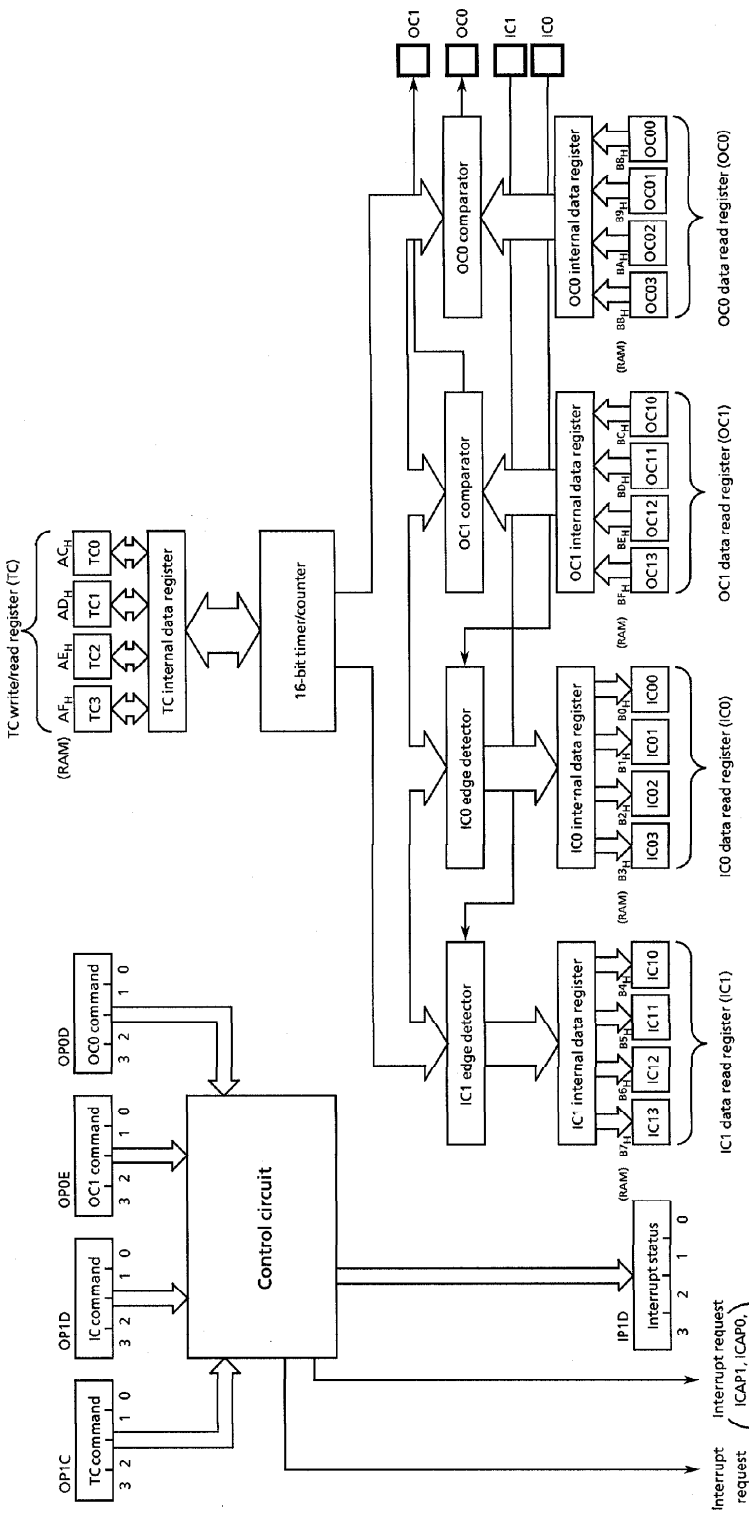


Figure 3-11. Timer/Counter Configuration

3.3.2 Timer/Counter Functions

The functions of the timer/counters are as follows.

1. Free-running timer
2. Event counter
3. Input capture
4. Output compare

3.3.3 Timer/Counter Control

The timer/counters are controlled by a timer/counter command register (OP1C), input capture command register (OP1D), output compare 1 command register (OP0E), output compare 0 command register (OP0D), and input capture/output compare interrupt status register (IP1D).

Timer/counter command register (port address : OP1C initial value : 0000_B)

3	2	1	0
TCEN	CPSEL		

TCEN	Selects timer/counter operation.
1	Enables timer/counter.
0	Disables timer/counter.

CPSEL	Selects count pulse/mode.
Free-running timer mode	
000	2/fc or 2/fs [s]
001	4/fc or 4/fs
010	8/fc or 8/fs
011	Counts at T1 input rising edge.
Event counter mode	
100	Counts at T1 input falling edge.
101	Counts at T1 input rising edge.
110	When T2 input is low, counts at T1 input rising edge.
111	When T2 input is high, counts at T1 input rising edge.

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

Figure 3-12. Timer/Counter Command Register

Input capture command register (port address : OP1D initial value : 0000_B)

3	2	1	0
ICCNT1		ICCNT0	

ICCNT1	Input capture 1 command
00	Disables capture.
01	Rising-edge capture
10	Falling-edge capture
11	Both-edge capture

ICCNT0	Input capture 0 command
00	Disables capture.
01	Rising-edge capture
10	Falling-edge capture
11	Both-edge capture

Figure 3-13. Input Capture Command Registers

Output compare 1 command register
(port address : OP0E initial value : 0000_B)

3	2	1	0
CMP1E	CMP1I	OMD1	

CMP1E	Output compare 1
1 :	Enables compare.
0 :	Disables compare.

CMP1I	Output compare 1 interrupt
1 :	Enables interrupt.
0 :	Disables interrupt.

OMD1	Selects output mode
00 :	No change in output
01 :	Outputs 1.
10 :	Outputs 0.
11 :	Inverts output.

Output compare 0 command register
(port address : OP0D initial value : 0000_B)

3	2	1	0
CMP0E	CMP0I	OMD0	

CMP0E	Output compare 0
1 :	Enables compare.
0 :	Disables compare.

CMP0I	Output compare 0 interrupt
1 :	Enables interrupt.
0 :	Disables interrupt.

OMD0	Selects output mode
00 :	No change in output
01 :	Outputs 1.
10 :	Outputs 0.
11 :	Inverts output.

Figure 3-14. Output Compare Command Registers

Input capture/output compare interrupt status register
(port address : IP1D initial value : 0000_B)

3	2	1	0
ICAP1	ICAP0	ICMP1	ICMP0

ICAP1	Input capture 1 interrupt
1 :	Interrupt
0 :	No interrupt

ICAP0	Input capture 0 interrupt
1 :	Interrupt
0 :	No interrupt

ICMP1	Output compare 1 interrupt
1 :	Interrupt
0 :	No interrupt

ICMP0	Output compare 0 interrupt
1 :	Interrupt
0 :	No interrupt

Figure 3-15. Input Capture/Output Compare Interrupt Status Register

3.3.4 Timer/Counter Write/Read Register (TC)

The timer/counter write/read registers are allocated to RAM addresses in bank 0. Read from and write to the timer/counter registers are performed using RAM manipulation instructions. The data memory at the addresses corresponding to these registers cannot be accessed as RAM and is therefore not available for saving user data.

(1) Write

Writing data to the most significant nibble in the TC (address AF_H) transfers 16-bit data to the timer/counter. Therefore, when writing a value to the TC, always write the most significant nibble last. Also, before writing, set the TCEN bit of the timer/counter control register to 0 to halt timer/counter operation.

(2) Read

Reading the least significant nibble of the TC (address AC_H) latches the current 16-bit timer/counter value in the internal data register. Therefore, when reading a value from TC, always read the least significant nibble first.

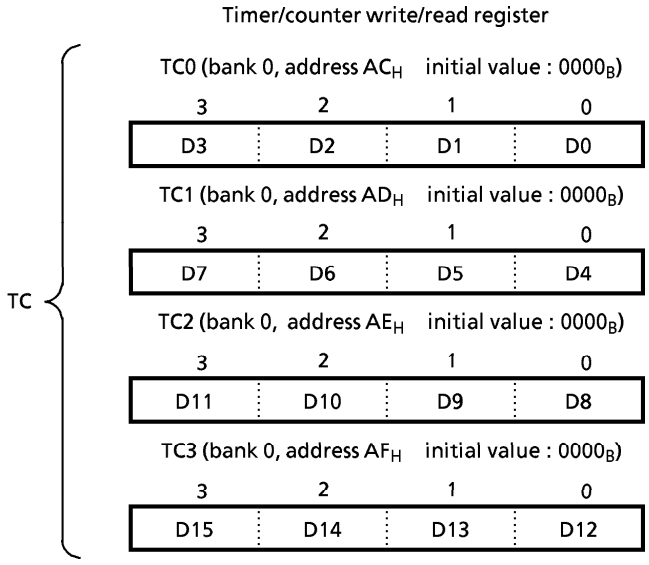


Figure 3-16. Timer/Counter Write/Read Registers

3.3.5 Input Capture Data Read Registers (IC1, IC0)

The input capture data read registers are allocated to RAM addresses in bank 0. Use data memory (RAM) manipulation instructions to operate the registers. Data can only be read from the registers, not written to them. The data memory at addresses corresponding to the registers cannot be accessed as RAM and is therefore not available to store user data.

Reading the most significant nibble of IC1 or IC0 (IC1: $B7_H$ or IC0: $B3_H$) clears the interrupt request of the relevant channel. Disable the next capture until the most significant nibble of the captured data is read.

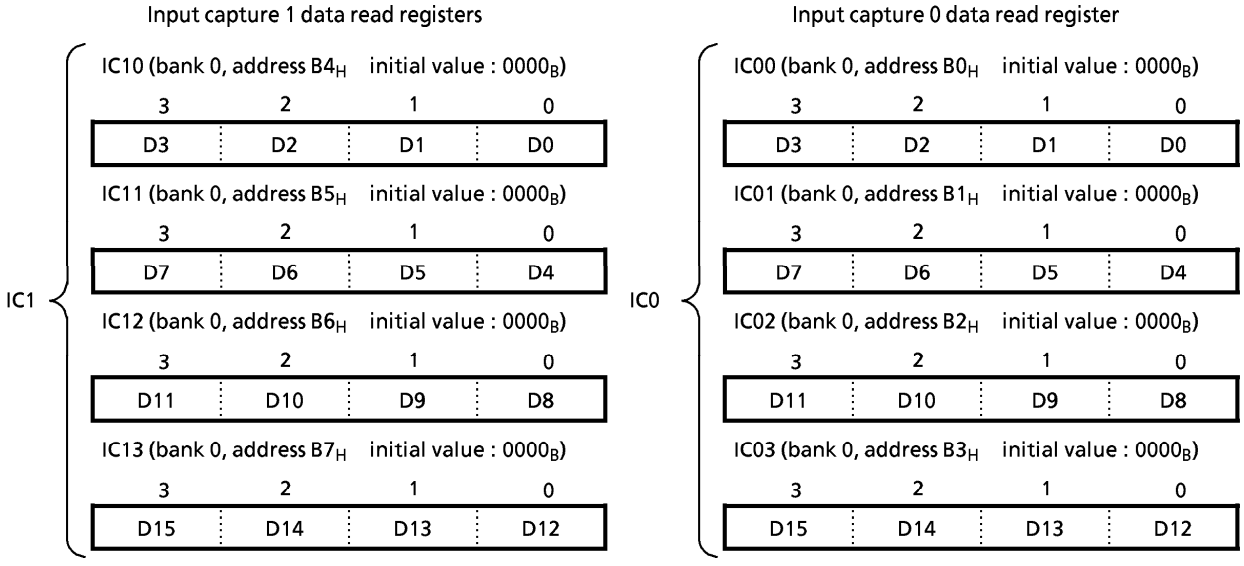


Figure 3-17. Input Capture Data Read Registers

3.3.6 Output Compare Data Write Registers (OC1, OC0)

The output compare data write registers are allocated to RAM addresses in bank 0. Use data memory (RAM) manipulation instructions to operate the registers. Data can be read from or written to the registers. The data memory at the addresses corresponding to the registers cannot be accessed as RAM and is therefore not available to store user data. Writing data to the least significant nibble of OC1 or OC0 (OC1: BC_H, OC0: $B8_H$) clears the interrupt request of the relevant channel. Writing data to the most significant nibble of the registers (OC1: BF_H, OC0: BB_H) transfers 16-bit data to the internal data register.

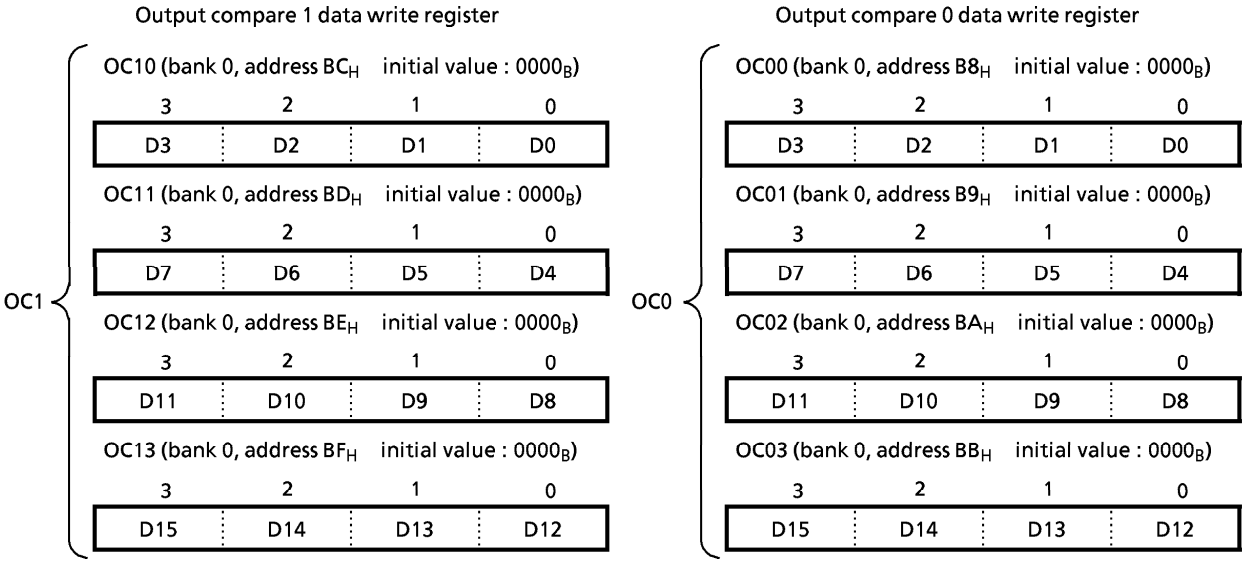


Figure 3-18. Output Compare Data Write Registers

3.3.7 Timer/Counter Operation

The timer/counter counts up at the rising edge of the count pulse. The count operation starts from the first rising edge of a count pulse after the command is set in the command register. Timer/counter operations have two modes: free-running timer mode when the internal pulse generated by the timing generator is used as the count pulse, and event counter mode when the external pin (T1 or T2) input is used as the count pulse.

3.3.7.1 Free-Running Timer Function

The free-running timer function uses the internal pulse, generated by the timing generator, as a count pulse. Four clock cycle rates can be selected for the count pulse using the lower two bits of timer/counter command register CPSEL. To enter free-running timer mode, set CPSEL to 0XXB. When CPSEL is set to 011B, the timer/counter is counted up at the rising edge of the external input pin T1. In this case, the maximum frequency that may be applied to T1 is fc/16Hz. Since T1 is sampled at 4/fc[s], the pulse width of the input signal must be longer than 4/fc[s].

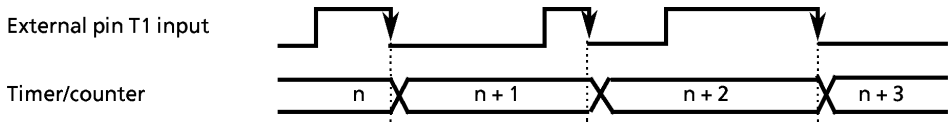
If an overflow is generated in free-running timer mode while the timer is operating (the timer/counter value shifts from $FFFF_H$ to 0000_H), an overflow interrupt is generated.

Before writing a value in the timer, set the TCEN bit to 0 to halt the timer/counter. Always write from the least significant nibble.

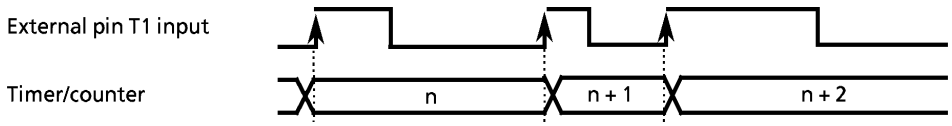
3.3.7.2 Event Counter Function

The event counter function uses the input from external pin T1 or T2 as a count pulse. Four count-up conditions can be selected using the lower two bits of timer/counter command register CPSEL. Figure 3.3.7.2 show the timing charts for these conditions.

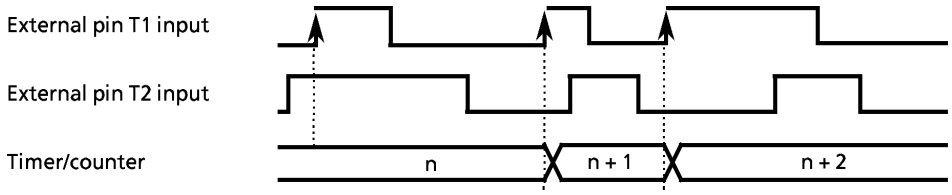
(1) CPSEL = 100_B : Counts up at T1 falling edge.



(2) CPSEL = 101_B : Counts up at T1 rising edge.



(3) CPSEL = 110_B : Counts up at T1 rising edge when T2 = L.



(4) CPSEL = 111_B : Counts up at T1 rising edge when T2 = H.

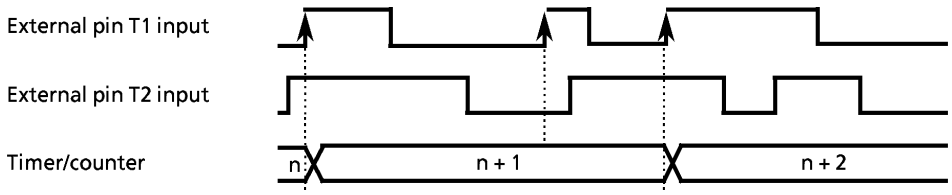


Figure 3-19. Event Counter Timing Charts

Pins T1 and T2 are also used as pins RA3 and RA1. When using pins T1 and T2 for timer/counter input, set the RA3 and RA1 output latches to 1. At reset, the output latch is initialized to 1.

To enter event counter mode, set CPSEL to 1XXB. In this mode, the maximum frequency that may be applied to the external input pin is fc/16Hz. While the counter is operating, the external input signal (T1, T2) is sampled at 4/fc[s]. Accordingly, the pulse width of the input signal must be longer than 4/fc[s].

If an overflow is generated in event counter mode (the timer/counter value shifts from $FFFF_H$ to 0000_H), an overflow interrupt is generated.

3.3.7.3 Input Capture Function

When the external pin (IC1, IC0) input changes, the input capture function automatically saves the timer/counter value to the input capture register corresponding to the external pin and generates an interrupt request. The input capture function also sets to 1, bit ICAP1 or ICAP0 of the input capture/output compare interrupt status register corresponding to the pin that caused generation of the interrupt request.

IC1 and IC0 of the input capture command registers ICCNT1 and ICCNT0 independently enable or disable data capture, or specify the capture on either edge of the external pin input signal.

Pins IC1 and IC0 are also used as pins RA3 and RA1. When used for input capture input, set RA3 and RA1 output latches to 1. At reset, the RA3 and RA1 output latches are initialized to 1.

Since the IC1, IC0 input signal is sampled at 4/fc[s] and the edge of the signal is detected, input a pulse width longer than 4/fc[s].

Figure 3-20 shows an input capture timing chart.

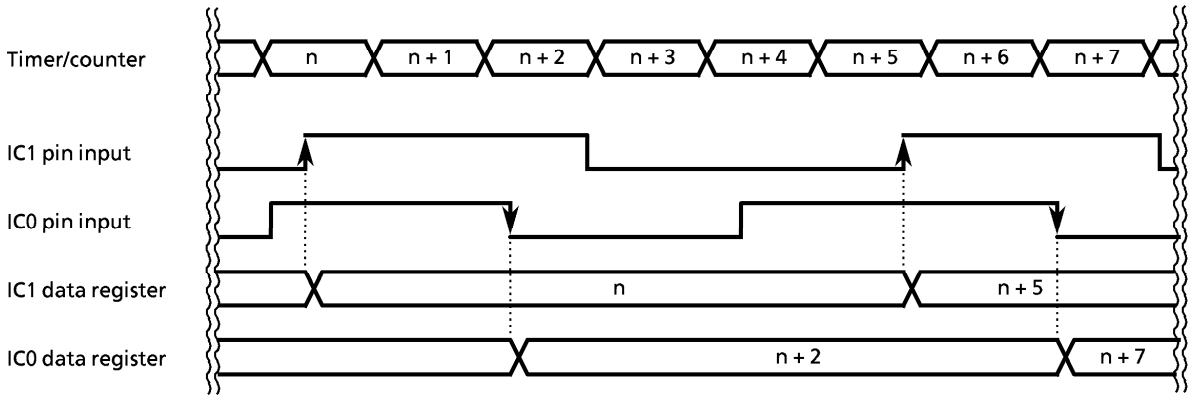


Figure 3-20. Input Capture Timing Chart
(When ICCNT1 = 01B, ICCNT0 = 10B)

[Input capture function usage example]

(a) Measurement of external input waveform cycle

First, write 10XXB to the timer/counter command register, set the timer/counter to free-running timer mode, and enable the timer/counter. Next, write 01_B or 10_B to the ICCNT register for the channel whose input capture command register cycle is measured, set the input capture timing to the rising or falling edge, and perform an input capture. At the first generation of an input capture interrupt, read the contents of the input capture data register using the interrupt service program and save the contents to RAM. At the second generation of an input capture interrupt (for the same channel as the first interrupt), read the contents of the input capture data register using the interrupt service program. The difference between the previously saved input capture data and the input capture data read at the generation of the second interrupt is multiplied by the clock rate to become the external waveform cycle input to the applicable channel. Figure 3.3.7.3.a shows the timing chart.

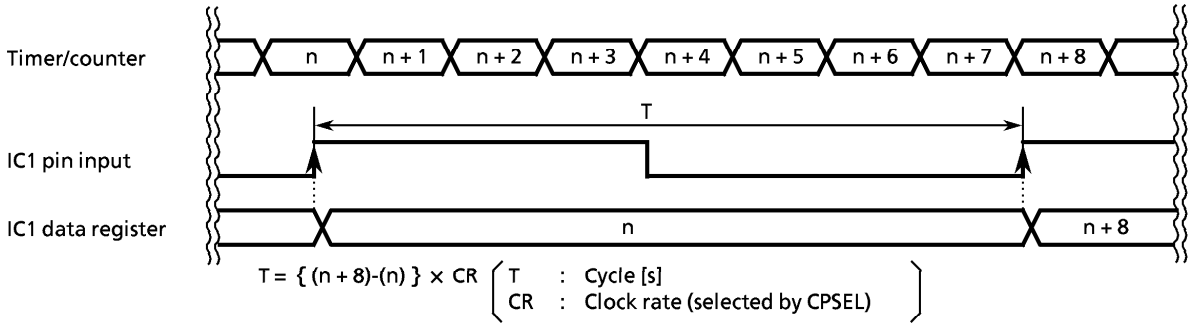


Figure 3-20 (a). External Input Waveform Cycle Measurement Example
(waveform input channel: IC1, ICCNT1 = 01B)

(b) External input waveform pulse width measurement

First, write 10XXB to the timer/counter command register, set the timer/counter to free-running timer mode, and enable the timer/counter. Next, write 11_B to the ICCNT register for the channel whose input capture command register cycle is measured, set the input capture timing to both edges, and perform an input capture. At the first generation of an input capture interrupt, read the contents of the input capture data register using the interrupt service program and save the contents to RAM. At the second generation of an input capture interrupt (for the same channel as the first interrupt), read the contents of the input capture data register using the interrupt service program. The difference between the previously saved input capture data and the input capture data read at the generation of the second interrupt is multiplied by the clock rate to become the pulse width of the external waveform input to the applicable channel. Figure 3-20 (b) shows the timing chart.

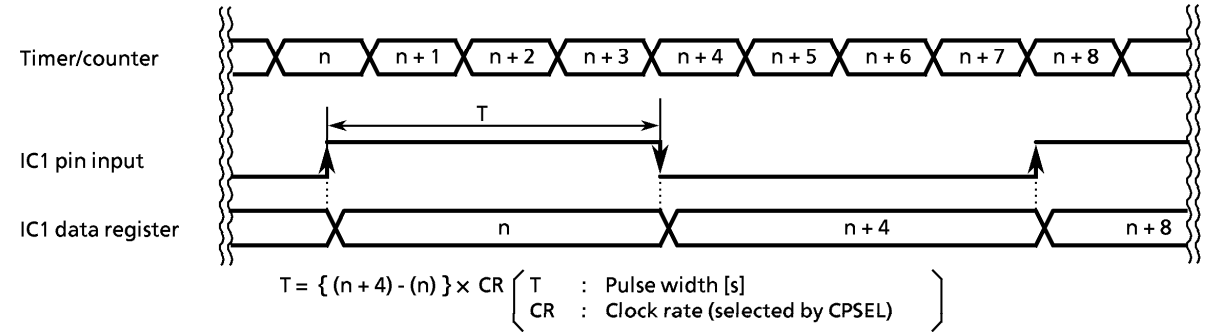


Figure 3-20 (b). External Input Waveform Pulse Width Measurement Example (waveform input channel: IC1)

3.3.7.4 Output Compare Function

The output compare function changes the output of pin OC1 or OC0 for the data register when the value of the output compare data register matches that of the timer/counter. If bit CMP1I of the output compare command 1 register or bit CMP0I of the output compare command 0 register is set to 1, an interrupt request is generated and the bit corresponding to the input capture/output compare interrupt status register interrupt source (ICMP1 and ICMP0) is set to 1. Two output compare registers are provided. Both can be set independently. Output mode of pins OC1 and OC0 are selected by command registers OMD1 and OMD0. Command registers CMP1E and CMP0E independently enable or disable the output compare.

Pins OC1 and OC0 are also used as RA2 and RA0.

Figure 3-21 shows the timing chart for the output compare.

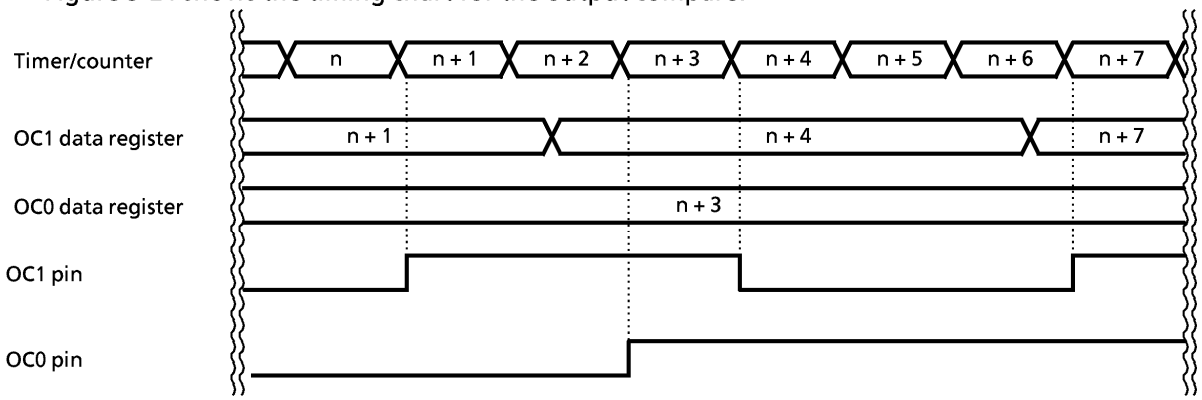


Figure 3-21. Output Compare Timing Chart (When OMD1 = 11_B, OMD0 = 01_B)

[Output compare function usage example]

(a) Timed output

First, write $10XX_B$ to the timer/counter command register, set the timer/counter to free-running timer mode, and enable the timer/counter. Next, write $0X01_B$ or $0X10_B$ or $0X11_B$ to the output compare command register for the channel used for timed output and set to output mode. If the value of X at this time is 1, an internal interrupt is generated at output compare output. Also, set the desired output time in the output compare data register of the applicable channel. When setting, write the data from the least significant nibble. When the data are written, set the CMPE bit of the output compare command register of the applicable channel to 1 and begin the output compare. If the output time written in the output compare data register matches the timer/counter value, the waveform is output in the set output mode from the output compare output pin for the applicable channel to the output compare command register. If bit CMP1 of the output compare command register for the applicable channel is 1, an internal interrupt is generated.

Figure 3-21 (a) shows the timing chart.

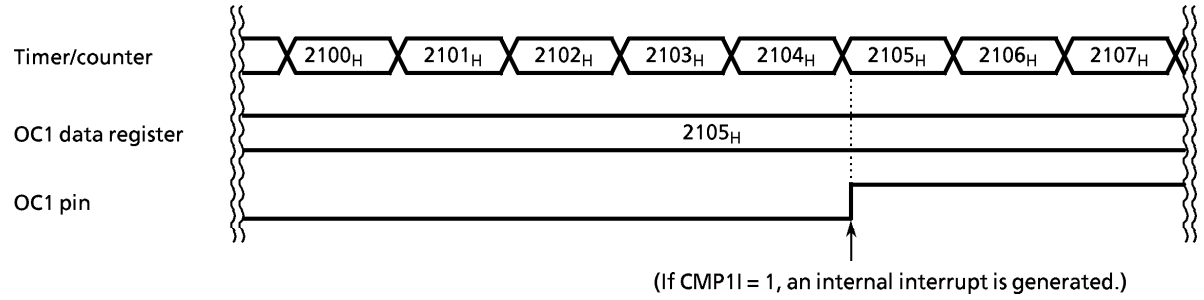


Figure 3-21 (a). Timed Output Usage Example (output compare channel: OC1, output mode: Outputs 1.)

(b) Time-specified interrupt

First, write $10XX_B$ to the timer/counter command register, set the timer/counter to free-running timer mode, and enable the timer/counter. Next, write 0100_B to the output compare command register for the channel (**performing the timed-specified interrupt, enable an output compare interrupt, and set the output mode to no output change. Also, set the desired interrupt time in the output compare data register of the applicable channel. When setting, write the data from the least significant nibble. When the data are written, set the CMPE bit of the output compare command register of the applicable channel to 1 and begin the output compare. If the interrupt time written in the output compare data register matches the timer/counter value, an internal interrupt is generated for the applicable channel. (The value output from the output compare output pin does not change.)

Figure 3-21 (b) shows the timing chart.

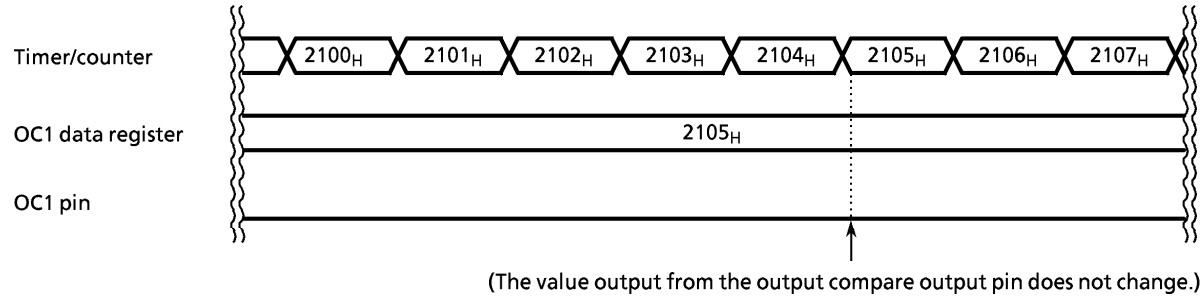
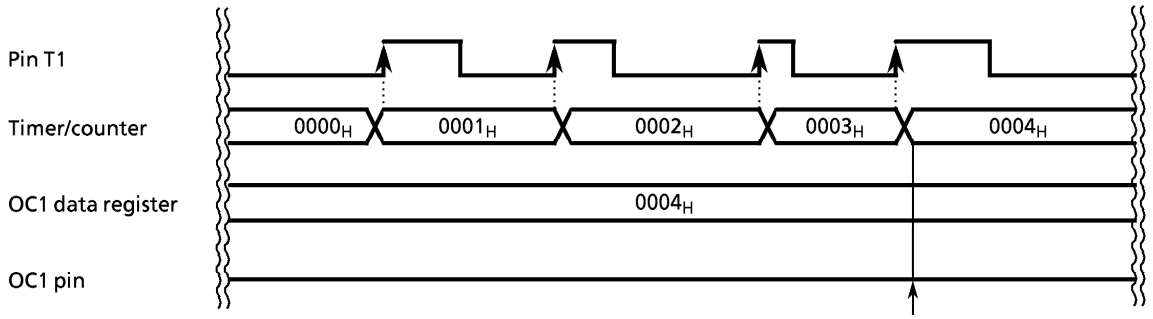


Figure 3-21 (b). Time-Specified Interrupt Usage Example (output compare channel: OC1)

(c) Event count interrupt

First, write $11XX_B$ to the timer/counter command register, set the timer/counter to event counter mode, and enable the timer/counter. Next, write 0100_B to the output compare command register for the channel performing the event count interrupt, enable an output compare interrupt, and set the output mode to no output change. Also, set the desired interrupt event count in the output compare data register of the applicable channel. When setting, write the data from the least significant nibble. When the data are written, set the CMPE bit of the output compare command register of the applicable channel to 1 and begin the output compare. If the interrupt event count written in the output compare data register matches the event count, an internal interrupt is generated for the applicable channel. (The value output from the output compare output pin does not change.)

Figure 3-21 (c) shows the timing chart.



(The value output from the output compare output pin does not change.)

Figure 3-21 (c). Event Count Interrupt Usage Example (output compare channel: OC1) (Event input pin: T1 (counts at the rising edge of input waveform (CPSEL = 101_B))

3.3.7.5 Timer/Counter Interrupts

Timer/counter interrupts have a total of five interrupt sources: a timer counter overflow interrupt source, two input capture interrupt sources (IC1 and IC0), and two output compare interrupt sources (OC1 and OC0). However, there are only two types of interrupt requests: timer/counter overflow (IOVF) requests and timer/counter interrupt (ITC) requests. The ITC request is ORed by any of four interrupt sources excluding the timer/counter overflow interrupt source. Therefore, when an ITC request is generated, the input capture/output compare interrupt status register must be read during the interrupt processing routine to determine the type of ITC interrupt source. Also, be sure to clear the corresponding interrupt request during the interrupt processing routine. The interrupt request is repeated until the interrupt request is cleared.

(1) Timer/counter overflow interrupt (IOVF)

When the timer/counter overflows (the count changes from $FFFF_H$ to 0000_H), an IOVF interrupt is generated. In interrupt enabled state, if the overflow interrupt is accepted immediately after generation, the IOVF is processed in the sequence shown in Figure 3-22 If count is requested, the count continues after generation of the overflow.

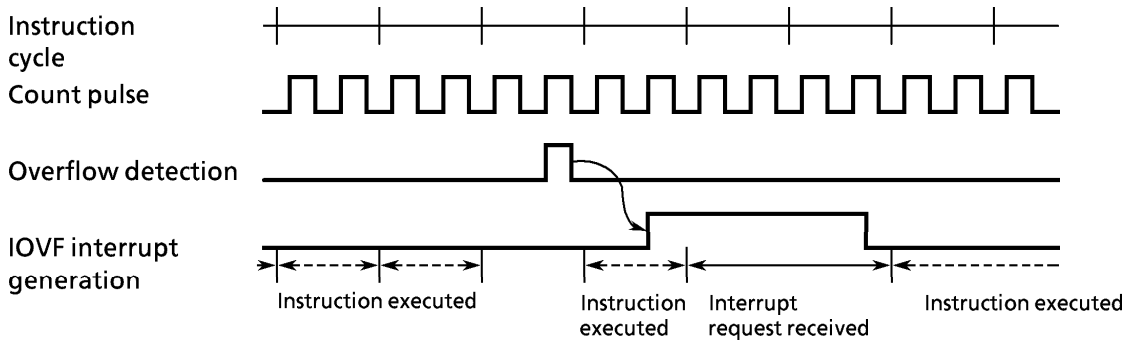


Figure 3-22. Timer/Counter Overflow Interrupt Timing Chart

(2) Input capture interrupt (ICAP1, ICAP0)

If the timer/counter value is saved to the input capture data register during the input capture, an input capture interrupt request (ICAP1, ICAP0) is generated and the corresponding bit of the interrupt status register is set to 1.

Reading the most significant nibble of the input capture data register (IC1: $B7_H$, IC0: $B3_H$) clears input capture interrupt requests. Figure 3-23 shows the timing chart for clearing input capture interrupt requests.

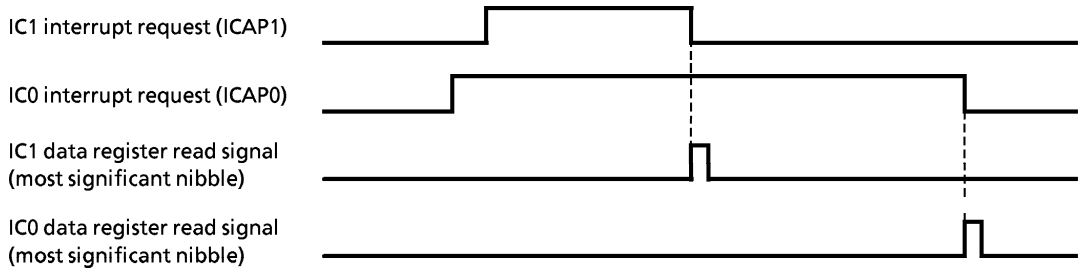


Figure 3-23. Input Capture Interrupt Request Clear Timing Chart

(3) Output compare interrupt (ICMP1, ICMP0)

If CMP1I = 1 or CMP0I = 1, and the timer/counter value and output compare data register value match during an output compare, an output compare interrupt request (ICMP1 or ICMP0) is generated and the corresponding bit in the interrupt status register is set to 1.

Writing to the least significant nibble of the output compare data register clears output compare interrupt requests. Figure 3-24 shows a timing chart for clearing output compare interrupt requests.

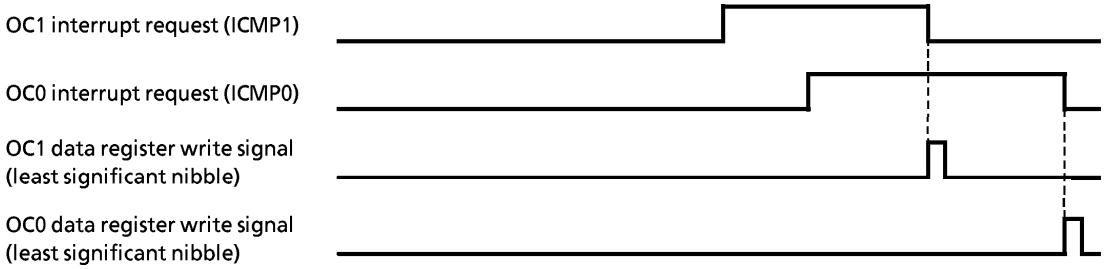


Figure 3-24. Output Compare Interrupt Request Clear Timing Chart

[Notes when processing timer/counter interrupts]

The timer/counter has multiple interrupt sources. However, only one interrupt request signal (ITC) is provided. Accordingly, when multiple interrupts are generated simultaneously or, before interrupt processing begins, a dummy interrupt may be generated after the termination of interrupt processing. Make sure that if a dummy interrupt is generated, the program is not affected. The following describes the case of no dummy interrupt generation and the case of dummy interrupt generation.

(a) No dummy interrupt

The interrupt processing routine checks the interrupt sources and begins processing one of the multiple interrupts. The routine completes the processing using the RETI instruction without processing the other interrupts. Where each interrupt is sequentially processed, dummy interrupts do not occur.

The interrupt request is maintained until all the interrupts are processed. In the case of multiple interrupts, after processing is completed for one interrupt, reception is again enabled and the next interrupt is processed. The number of interrupts generated equals the number of interrupt sources. Processing is repeated until all the interrupts are processed.

(b) Dummy interrupt

The interrupt processing routine checks for interrupt sources and processes all interrupts. When the routine completes interrupt processing and the RETI instruction is executed, a dummy interrupt is generated.

The dummy interrupt is acknowledged by the interrupt processing routine when it checks to see if there are any interrupt sources and detects none. Then the interrupt processing routine processes the dummy interrupt. Interrupt processing completes after executing the RETI instruction again.

3.4 Watchdog Timer

The internal watchdog timer function aims to swiftly detect CPU malfunction (runaway) due, for example, to noise, and aims to return operation to normal. The runaway detect signal is output to pin R71 ($\overline{WTO}$) as watchdog timer output. When using the watchdog timer, set the R71 output latch to 1. Resetting sets the R71 output latch to 1 and disables the watchdog timer.

Connecting the $\overline{WTO}$ pin to the $\overline{RESET}$ pin triggers a reset on detection of a runaway CPU.

3.4.1 Watchdog Timer Configuration

The internal watchdog timer function aims to swiftly detect CPU malfunction (runaway) due, for example, to noise, and aims to return operation to normal. The runaway detect signal is output to pin R71 ($\overline{WTO}$) as watchdog timer output. When using the watchdog timer, set the R71 output latch to 1. Resetting sets the R71 output latch to 1 and disables the watchdog timer.

Figure 3-25 shows the configuration of the watchdog timer

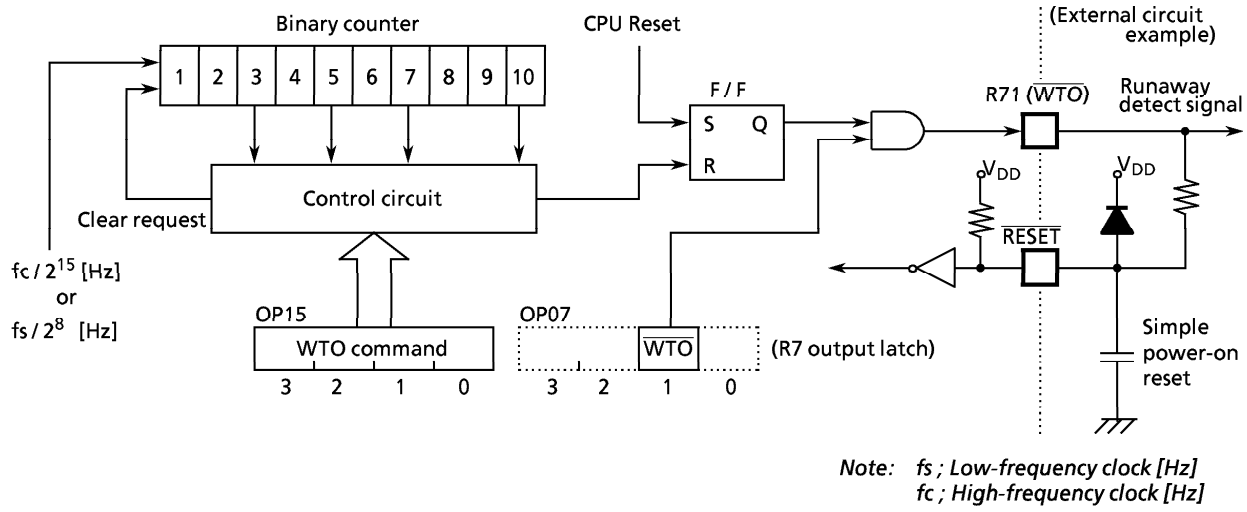


Figure 3-25. Watchdog Timer configuration

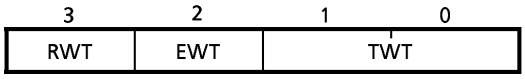
3.4.2 Watchdog Timer Control

The watchdog timer is controlled by command register OP15. At reset, the command register is initialized to 1000_B.

Use the watchdog timer for detecting CPU runaway as follows:

1. Set the watchdog timer detect time and clear the binary counter.
2. Enable the watchdog timer.
3. Then, clear the binary counter within the set watchdog timer detect time. If the CPU runs away for some reason but the binary counter is not cleared, the flip-flop is cleared to 0 at the rising edge of the binary counter. This activates the runaway detect signal. At this time, if the R71 output latch is 1, the $\overline{WTO}$ output is low level.

Watchdog timer command register (port address : OP15 initial value : 1000)



RWT : Clears binary counter.
0 : Clears binary counter.
(After clearing, automatically set to 1.)

EWT : Enables or disables watchdog timer.
0 : Disables watchdog timer.
1 : Enables watchdog timer.

TWT : Sets watchdog timer detect time.
00 : 3×2^{15} / fc or 3×2^{8} / fs [s]
01 : 15×2^{15} / fc or 15×2^{8} / fs
10 : 63×2^{15} / fc or 63×2^{8} / fs
11 : 511×2^{15} / fc or 511×2^{8} / fs

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

Figure 3-26. Watchdog Timer Command Register

Example : Enable the watchdog timer detect time for 63×2^{15} / fc [s] .

```
        LD      A, #0010B
        OUT     A, %OP15
        LD      A, #0110B
        OUT     A, %OP15
        :
Within WDT detect time
        :
        LD      A, #0110B
        OUT     A, %OP15
        :
Within WDT detect time
        :
        LD      A, #0110B
        OUT     A, %OP15
```

Note: Before enabling the watchdog timer, always clear the binary counter. Before switching the system clock, first disable the watchdog timer during the warming-up time, from slow mode to normal 2 mode.

TWT	Normal 1 operation	Normal 2 operation (TG step 8 input)		SLOW operation	At fc = 4.194304 MHz, fs = 32.768 kHz
		SLCK = 0 (fc/2^7)	SLCK = 1 (fs)		
00	3×2^{15} / fc [s]		3×2^{8} / fs [s]		23.4 [ms]
01	15×2^{15} / fc		15×2^{8} / fs		117
10	63×2^{15} / fc		63×2^{8} / fs		492
11	511×2^{15} / fc		511×2^{8} / fs		3992

Table 3-4. Watchdog Timer Detect Time

3.5 A/D Converter

TMP47E885AF incorporates an 8-bit successive approximation type A/D converter with an 8-channel analog input multiplexer and a sample and hold circuit.

3.5.1 Configuration of A/D Converter

Figure 3-27 shows the configuration of the A/D converter.

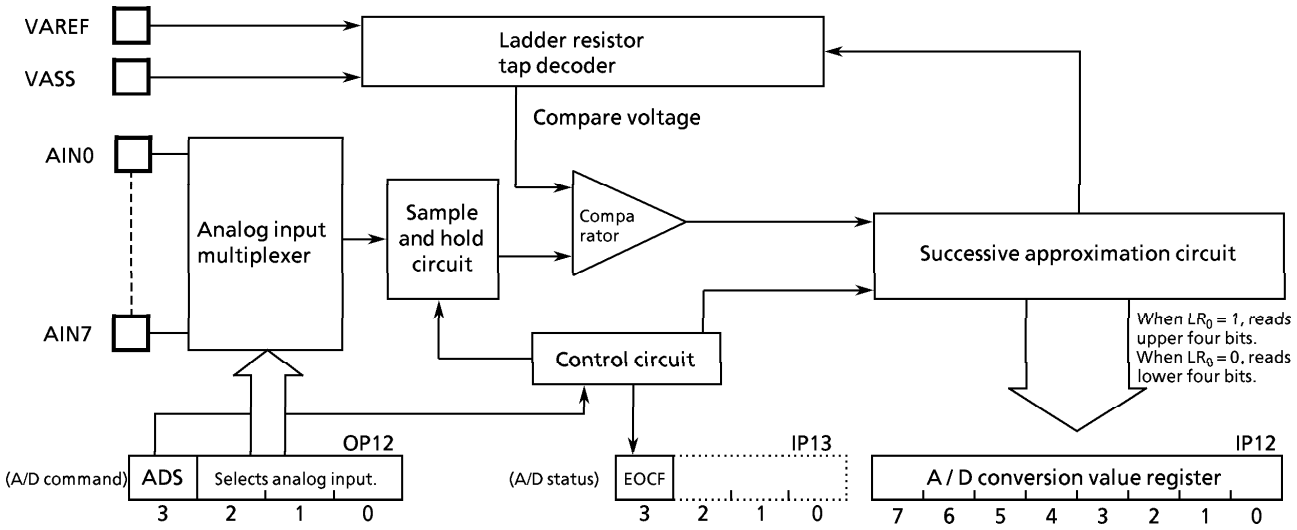


Figure 3-27. Configuration of A/D Converter

3.5.2 A/D Converter Control

The A/D converter is controlled by a command register (OP12), conversion value register (IP12), and status register (IP13).

(1) A/D conversion start register (ADS) and analog input selection register (SAIN)

The lower three bits are used to select the analog input channel (AIN0 - AIN7) for A/D conversion. Setting the most significant bit of ADS to 1 starts A/D conversion. When A/D conversion starts, ADS is automatically cleared to 0. Setting ADS to 1 during A/D conversion immediately starts a new A/D conversion. During A/D conversion, the sample and hold circuit is used to hold the analog input voltage.

A/D conversion start register and analog input selection register
(port address : OP12 initial value : 0000_B)

3	2	1	0
ADS	SAIN		

ADS	Starts A/D conversion.
1 :	Starts A/D conversion. (After start, automatically cleared to 0.)

SAIN	Selects analog input channel.
000 :	R40 (AIN0)
001 :	R41 (AIN1)
010 :	R42 (AIN2)
011 :	R43 (AIN3)
100 :	R50 (AIN4)
101 :	R51 (AIN5)
110 :	R52 (AIN6)
111 :	R53 (AIN7)

Figure 3-28. A/D Conversion Start Register and Analog Input Selection Register

(2) End of A/D conversion flag

The end of conversion flag (EOCF) is a 1-bit flag to indicate the completion of A/D conversion. At the end of conversion, the flag is set to 1. The EOCF flag is cleared to 0 when the upper four bits and lower four bits of conversion data is read, or when A/D conversion is started.

A/D converter status register (port address : IP13 initial value:0011_B)

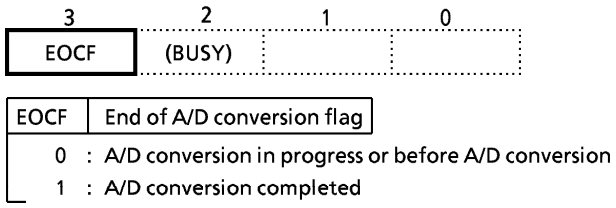


Figure 3-29. A/D Status Register

(3) A/D conversion value register

This register contains A/D-converted 8-bit values. The 8-bit A/D converted values are split into the upper four bits and the lower four bits, and read using LR0 (bit 0 of L register).

A/D conversion value register (port address: IP12 initial value : 0000_B)

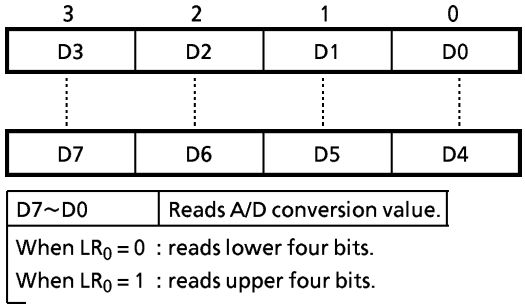


Figure 3-30. A/D Conversion Value Register

3.5.3 A/D Converter Operation

The high analog reference voltage is applied to the VAREF pin; the low analog reference voltage, to the VASS pin. The A/D converter uses resistors to configure a resistor ladder network which distributes the VAREF-VASS reference voltage to the corresponding bits, then compares the result with the analog input voltage.

(1) Starting A/D conversion

Before A/D conversion is started, select one of AIN0 to AIN7 using the A/D conversion start register and the lower three bits of the analog input selection register. Set the output latch of the analog input pin used for A/D conversion to 1 so that it becomes high impedance. Although the pins not used as analog input pins are normally available as I/O pins; however, during conversion, do not issue output instructions to any of these pins. Setting the A/D conversion start register and the analog input selection register (ADS) to 1 starts A/D conversion. When the conversion is complete in 26 instruction cycles, the EOCF indicating A/D conversion completion is set to 1.

The analog input voltage is sampled in two instruction cycles after A/D conversion starts.

Note: As the sample and hold circuit incorporates a 12pF capacitor via a 5kΩ resistor (typ.), the capacitor must be charged within two instruction cycles.

(2) Reading A/D converted value

The (8-bit) converted values stored in the A/D conversion value register are split by LR_0 (L register bit 0) into their upper four bits and lower four bits to be read. When $LR_0 = 0$, the lower four bits can be read. When $LR_0 = 1$, the upper four bits can be read. On conversion, the converted values are stored sequentially. The values are read after confirmation of conversion completion (EOCF = 1). The converted values can be saved to RAM by the [IN %p,@HL] instruction. If reading is attempted during A/D conversion, values are undefined.

(3) HOLD during A/D conversion

Upon entering a HOLD during A/D conversion, conversion is stopped and the converted value is undefined. After returning from a HOLD, the EOCF remains cleared to 0. However, upon entering a HOLD after A/D conversion (after the EOCF is set), the A/D converted value and the EOCF state are maintained.

Example : After selecting pin AIN4 as an analog input pin, perform A/D conversion, check the EOCF, and save the upper and lower four bits of converted data to RAM addresses 10_H and 11_H respectively.

```
        ; AIN   SELECT
        ; A/D   START
        LD      A, #0CH
        OUT     A, %OP12
        ; EOCF = 1?
SLOOP : IN      %IP13, A
        TEST    A, 3
        B       SLOOP
        ; DATA READ
        LD      HL, #10H
        IN      %IP12, @HL
        INC     L
        IN      %IP12, @HL
```

3.6 12-Bit Pulse Width Modulation (PWM) Output

The TMP47E885AF incorporates two channels with 12-bit pulse width modulation (PWM) output. A prescaler and duty write register are allocated to RAM addresses in units of 4 bits. Values are set using RAM manipulation instructions. However, memory at addresses corresponding to the prescaler and duty write register (PS1, PS0, PD1, PD0) cannot be accessed as RAM and is therefore not available to user variable data. The PWM is also used as the RB port. When used as the PWM trigger input pin, set the RB port output latch to 1.

3.6.1 Configuration of PWM Output

Figure 3-31 shows the configuration of PWM output.

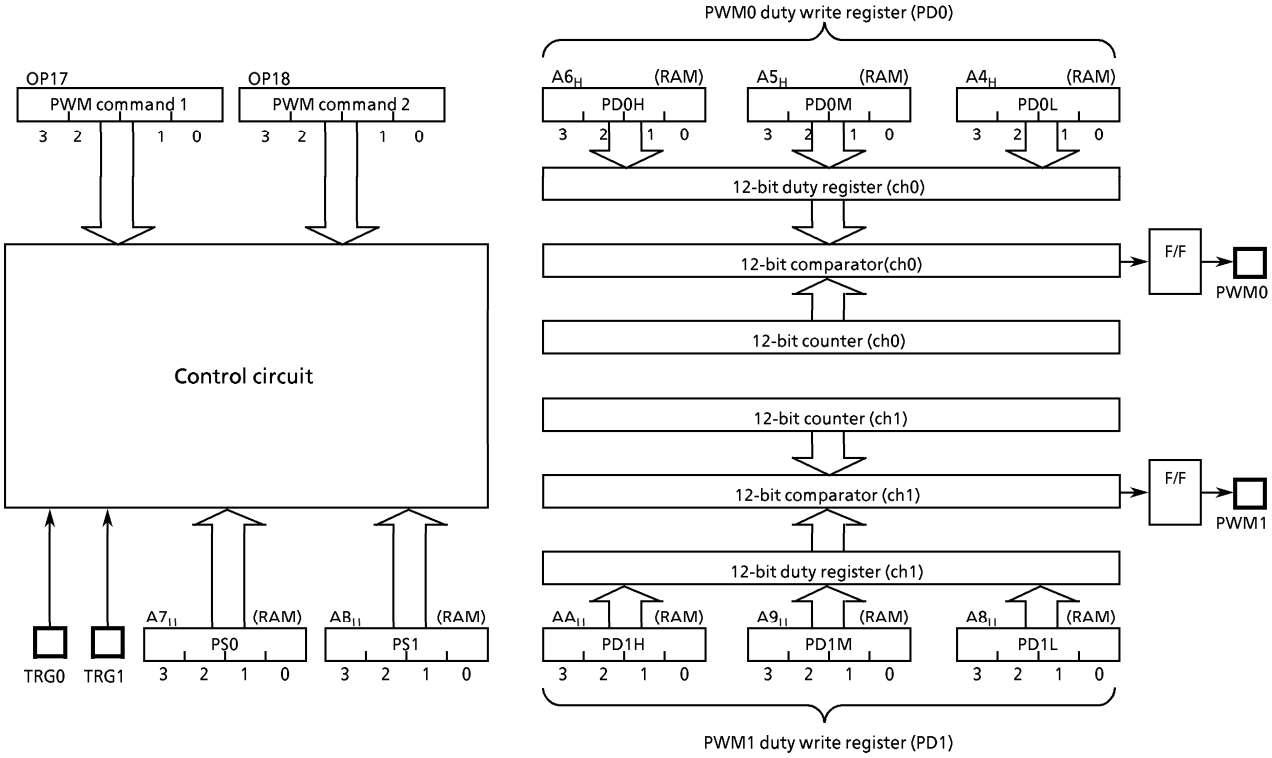


Figure 3-31. Configuration of PWM Output

3.6.2 PWM Control

The PWM output is controlled by PWM command register 1 (OP17), PWM command register 2 (OP18), the PWM1 prescaler register (at AB_H in RAM bank 0), the PWM0 prescaler register (at $A7_H$ in RAM bank 0), the PWM1 duty write register (at $A8_H$, $A9_H$, and AA_H in RAM bank 0), and the PWM0 duty write register (at $A4_H$, $A5_H$, and $A6_H$ in RAM bank 0).

PWM command register 1 (port address : OP17 initial value : 0011_B)

3	2	1	0
PWM1E	PWM0E		

PWM1E	Channel 1 enable
1	: Channel 1 output enable
0	: Channel 1 output disable

PWM0E	Channel 0 enable
1	: Channel 0 output enable
0	: Channel 0 output disable

Figure 3-32. PWM Command Register 1

PWM command register 2 (port address : OP18 initial value : 0000_B)

3	2	1	0
S/$\overline{C}$1	$\overline{POL1}$	S/$\overline{C}$0	$\overline{POL0}$

S/$\overline{C}$1	Channel 1 output mode (Note 1)
1	Single-shot output by trigger 1 input
0	Continuous output

S/$\overline{C}$0	Channel 0 output mode (Note 1)
1	Single-shot output by trigger 0 input
0	Continuous output

$\overline{POL1}$	Selects channel 1 output polarity.
1	Outputs negative pulse.
0	Outputs positive pulse.

$\overline{POL0}$	Selects channel 0 output polarity.
1	Outputs negative pulse.
0	Outputs positive pulse.

Note 1 : Before setting channel output mode, disable channel output.

Note 2 : When the most significant nibble in the PWM duty register of each channel is set, the bits used to select the channel output polarity are internally latched.

Figure 3-33. PWM Command Register 2

PWM1 prescaler register (PS1)
(address AB_H of bank 0, initial value: 0000_B)

3	2	1	0
PS1			

PS1	Selects channel 1 output cycle.
0000	2/fc × 4096 = 8192/fc [s]
0001	4/fc × 4096 = 16384/fc
0010	6/fc × 4096 = 24576/fc
⋮	⋮
1111	32/fc × 4096 = 131072/fc

PWM0 prescaler register (PS0)
(address $A7_H$ of bank 0, initial value: 0000_B)

3	2	1	0
PS0			

PS0	Selects channel 0 output cycle.
0000	2/fc × 4096 = 8192/fc [s]
0001	4/fc × 4096 = 16384/fc
0010	6/fc × 4096 = 24576/fc
⋮	⋮
1111	32/fc × 4096 = 131072/fc

Note : fc : High-frequency clock [Hz]

Figure 3-34. PWM Prescaler Register

The PWM prescaler register is a write-only register allocated to an address in bank 0 in data memory (RAM) and accessible using data memory (RAM) manipulation instructions. Data memory at addresses corresponding to the register cannot be accessed as RAM and is therefore not available to user variable data.

Because the PWM supports independent counters for channel 1 and channel 0, PWM output cycle can be set independently for each channel. The prescaler selects the output frequency. The prescaler can be set to 0000_B - 1111_B. Sixteen cycles are supported. The fundamental clock cycle of the internal 12-bit counter for PWM output is obtained by dividing 2/fc[s] by the prescaler register value. Multiplying the fundamental clock cycle 4096 times (8192/fc [sec] min.) produces the PWM output cycle. The following equation shows the derivation of the PWM output cycle based on the prescaler set value.

PWM output cycle (Tc) = (prescaler set value + 1) × (2/fc) × 4096 [s]

Note : fc ; High-frequency clock [Hz]

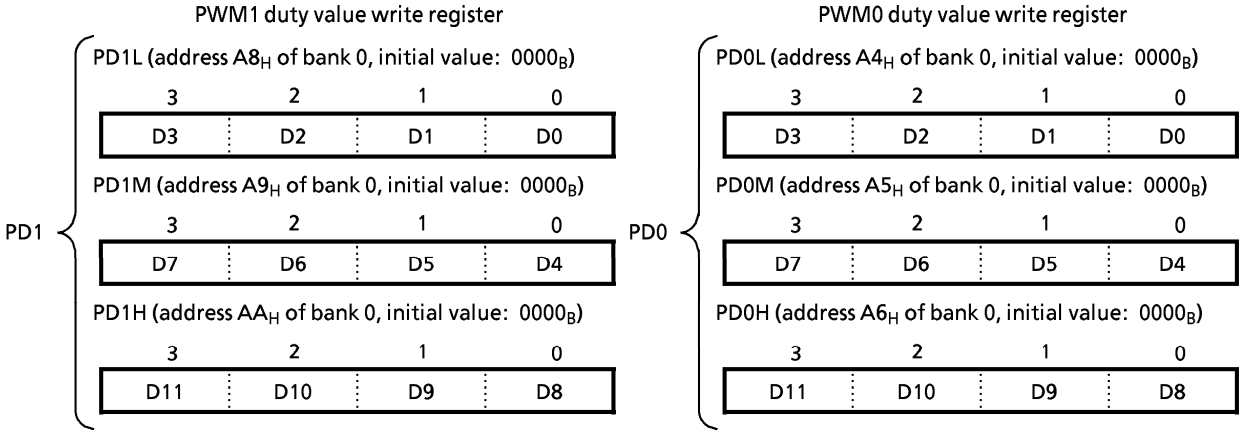


Figure 3-35. Duty Write Registers

The PWM duty write register is a write-only register allocated to an address in bank 0 in data memory (RAM) and accessible with data memory (RAM) manipulation instructions. Data memory at addresses corresponding to the register cannot be accessed as RAM and is therefore not available to user variable data. When data are written to the most significant nibble of the PWM duty write register (PD1H: AA_H, PD0H: $A6_H$), duty write register values are transferred to the internal 12-bit duty register.

3.6.3 PWM Output Waveform

In the following description, an asterisk (*) indicates the channel (ch0 or ch1).
The following three parameters determine the PWM output waveform.

(a) Cycle (Tc)
Set by the PWM* prescaler register (PS*).

(b) Polarity
Set by the $\overline{POL*}$ bit of PWM command register 2.

(c) Duty (Td)
Set by the PWM* duty write register (PD*).

Figure 3-36 shows the correspondence between parameters and PWM output waveform.

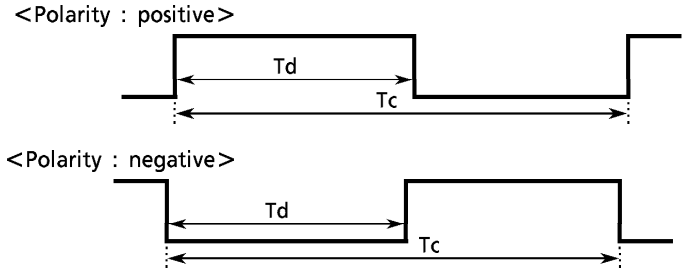


Figure 3-36. PWM Output Waveform

3.6.4 Output Mode

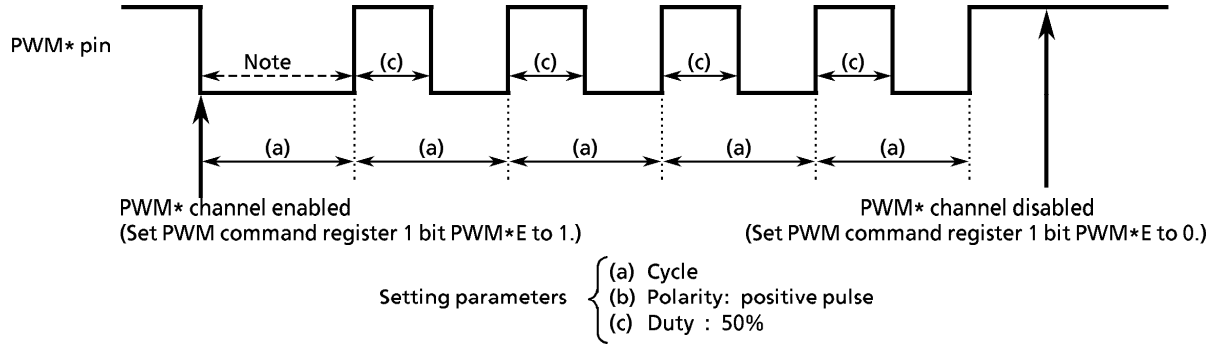
In the following description, an asterisk (*) indicates the channel (ch0 or ch1).
There are the following two output modes for 12-bit PWM output.

(1) Continuous output mode
(2) Trigger input single-shot output mode

Set the mode with the S/$\overline{C}$* bit of PWM command register 2. Before setting the mode, first disable the PWM output.

3.6.4.1 Continuous Output Mode

Continuous output mode is used to output the waveform continuously while the PWM output is enabled. Setting the S/$\overline{C}$* bit of PWM command register 2 to 0 enters this mode. In continuous output mode, the output waveform is determined by the three parameters in 3.6.3, PWM Output Waveform. In this mode, the TRG* pin can be used as a standard I/O port.
Figure 3.6.4.1 shows an example of an output waveform in continuous output mode.



Note : For one output cycle directly after continuous output mode is enabled, the set duty and polarity are not valid. The duty and polarity become effective from the next output cycle.

Figure 3-37. Output Waveform in Continuous Output Mode

3.6.4.2 Trigger Input Single-Shot Output Mode

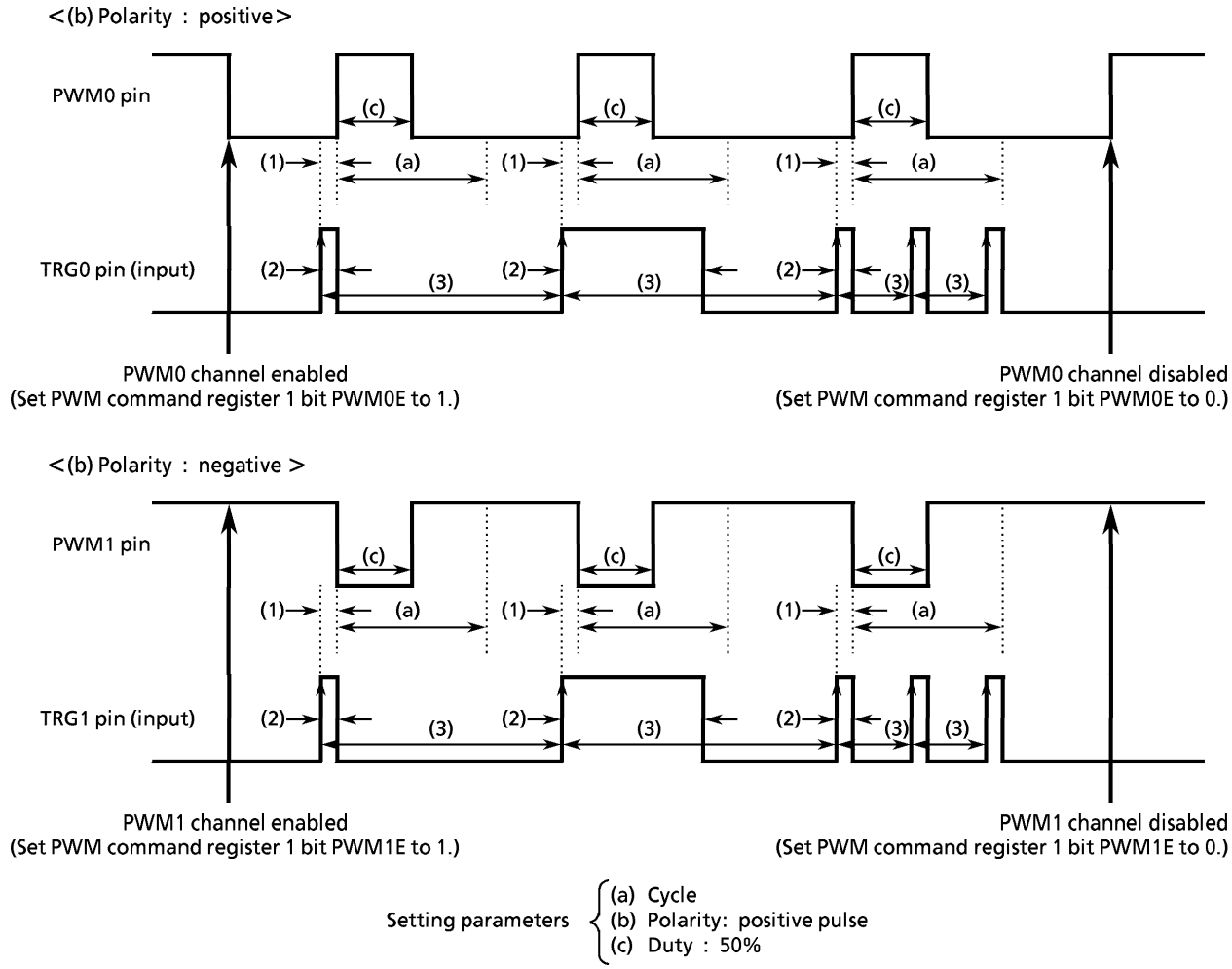
While PWM output is enabled, trigger input single-shot output mode is used to output from the PWM trigger input pin (TRG*) a single-shot waveform in sync with the input signal. Setting the PWM command register 2 S/$\overline{C}$* bit to 1 enters this mode. In trigger input single-shot output mode, the output waveform is determined by the three parameters in 3.6.3, PWM Output Waveform. A single-shot waveform is output for one cycle (prescaler register set value + 1) x 2/fc [s]) of the PWM 12-bit counter clock determined by the PWM* prescaler from the rising-edge timing of the PWM trigger input signal. The trigger input signal cannot be received for one PWM output cycle after the single-shot waveform output starts. Therefore, the cycle of the trigger input signal must be longer than the cycle of the PWM output. As the PWM trigger input signal is internally sampled at a rate of 4/fc, the trigger input signal pulse width must be greater than 4/fc.
The following shows the correspondence between the PWM trigger input pin (TRG*) and the PWM output pin (PWM*).

*PWM trigger input pins	PWM output pin
TRG0 (RB0)	PWM0 (RB1)
TRG1 (RB2)	PWM1 (RB3)

The pin name when PWM output is disabled is enclosed in parentheses ( ).

* PWM trigger input pins (TRG0, TRG1) are used for trigger inputs. Therefore, before trigger input, set the output latch to 1.

Figure 3-38 shows an example of the output waveform in trigger input single-shot output mode.



Setting parameters
- (a) Cycle
- (b) Polarity: positive pulse
- (c) Duty : 50%

(1) From the rising timing of the trigger input signal until the start of the single-shot waveform output
: 0- (prescaler register set value + 1) × (2/fc) [s]

(2) Trigger input signal pulse width : >> fc/4 [s]

(3) Trigger input signal cycle : >> setting parameter (a) cycle

Figure 3-38. Output Waveform in Trigger Input Single-Shot Output Mode

3.6.5 Changes in PWM Output Cycle

In the following description, an asterisk (*) indicates the channel (ch0 or ch1).

The PWM output cycle is determined by the PWM* prescaler register and is valid immediately after writing to the register. Accordingly, changing the PWM* prescaler register setting during the PWM output cycle alters the 12-bit counter count clock. The current cycle and duty become undefined. From the next cycle, the waveform is output in different cycle according to the altered PWM* prescaler register setting.

Figure 3-39 shows the waveform when the PWM* prescaler register is altered during the output cycle.

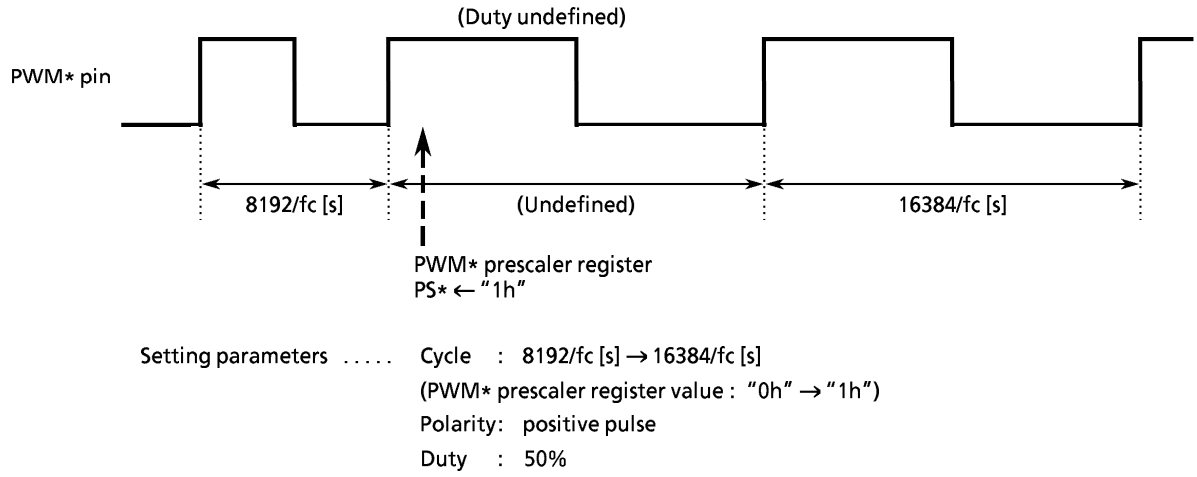


Figure 3-39. Changes in PWM Output Cycle

3.6.6 PWM Duty Change

In the following description, an asterisk (*) indicates the channel (ch0 or ch1).
The duty of the PWM output waveform is set using the PWM* duty write register.
The value set in the PWM* duty write register is transferred to the internal 12-bit counter when the most significant nibble (PD*H) of the PWM* duty write register is written. Accordingly, when changing the duty, write to the PWM* duty write register in the following order:

1. Least significant nibble (PD*L)
2. Middle nibble (PD*M)
3. Most significant nibble (PD*H)

Writing data to the PWM* duty write register in the above order transfers the set value to the internal 12-bit counter at the execution of step 3 above. The waveform in which the changed duty becomes valid is output in the output cycle following the cycle where the set value is transferred. Figure 3-40 shows the waveform when the PWM duty is changed.

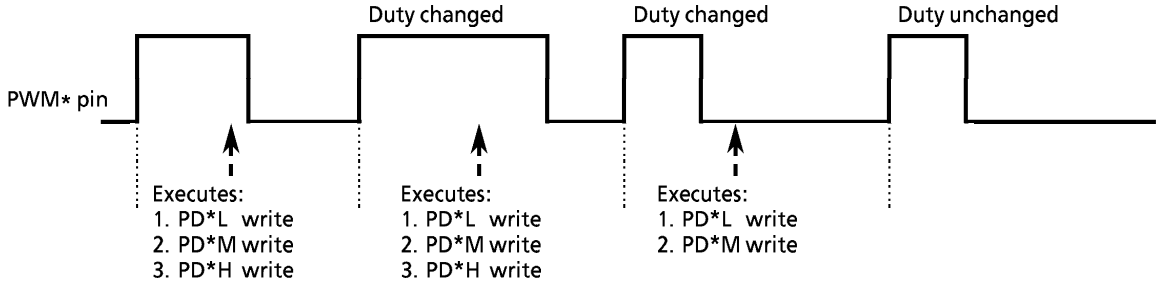


Figure 3-40. PWM Duty Change

3.6.7 PWM Output Polarity Change

In the following description, an asterisk (*) indicates the channel (ch0 or ch1).
The $\overline{POL*}$ bit setting is not internally latched at the time of writing to PWM command register 2. The $\overline{POL*}$ bit setting is internally latched when the most significant nibble (PD*H register) of the PWM* duty register is written. The internally latched $\overline{POL*}$ bit data are valid from the cycle following the PD*H register write cycle. That is, the $\overline{POL*}$ setting change becomes valid in the waveform output from the cycle following the PD*H register write cycle. Accordingly, change the PWM output polarity in the following order.

1. Change $\overline{POL*}$ bit.
2. Write to PD*H register.

The polarity change becomes valid from the cycle following that when the POL* bit was changed and the PD*H register was written. Figure 3-41 shows the waveform when the PWM output polarity is changed.

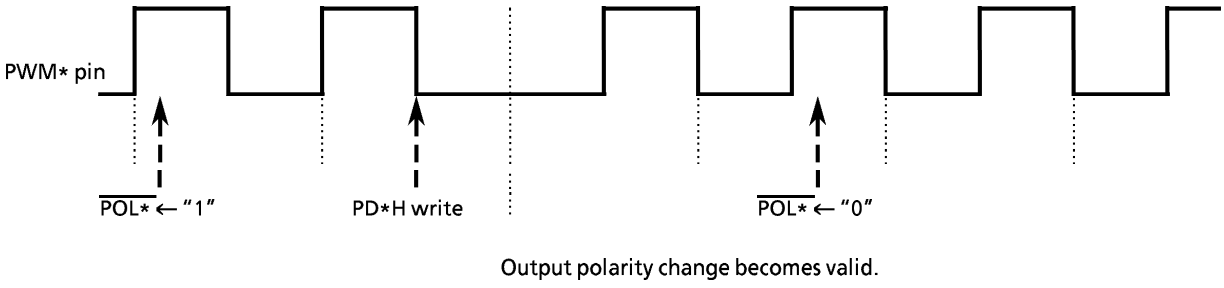


Figure 3-41. Output Polarity Change

3.6.8 Timing of PWM Output Waveform Duty and Polarity Changes

The PWM* duty register is used to set the duty of the PWM output waveform. The $\overline{POL*}$ bit of PWM command register 2 sets the polarity. After the register settings are written to PD*H, the settings are saved internally. The timing for the PWM output waveform becomes valid from the next PWM output cycle after data are written to the PD*H register.
If data are written to the PD*H register several times during a single PWM output cycle, the last duty settings in one duty cycle become valid at the next PWM output cycle. The polarity set before the last write to the PD*H register in one cycle become effective at the next PWM output cycle. Therefore, if data are not written to the PD*H register after setting $\overline{POL*}$, the polarity is not switched at the next output cycle.

★ Changing polarity but not duty

To change only the polarity without changing the duty during PWM output, set the polarity in the $\overline{POL*}$ bit of PWM command register 2, then write the previous setting to the most significant nibble (PD*H) of the PWM* duty register.

3.6.9 PWM Pin

In the following description, an asterisk (*) indicates the channel (ch0 or ch1).
In PWM output mode, the RB pin has the following functions.

PWM output pin	enable/disable		RB3 (PWM1)	RB2 (TRG1)	RB1 (PWM0)	RB0 (TRG0)
	CH1	CH2				
Continuous	×	×	RB3	RB2	RB1	RB0
	×	○	RB3	RB2	PWM0	RB0
	○	×	PWM1	RB2	RB1	RB0
	○	○	PWM1	RB2	PWM0	RB0
Single-shot	×	×	RB3	RB2	RB1	RB0
	×	○	RB3	RB2	PWM0	RB0
	○	×	PWM1	TRG1	RB1	TRG0
	○	○	PWM1	TRG1	PWM0	TRG0

○ ··· enable
× ··· disable

RB* : Can be used as a standard I/O pin.
PWM* : PWM output pin
TRG* : PWM trigger input pin

3.7 Serial Interface

TMP47E885AF incorporates two types of serial interfaces.

1. Synchronous serial interface (SIO)
2. Asynchronous serial interface (UART)

3.7.1 Synchronous Serial Interface (SIO)

This 8-bit buffered synchronous serial interface is built into the processor and is connected to external circuits through the R92 ($\overline{SCK}$), R91 (SO), and R90 (SI) pins. The interface shares port R9. When used as a serial port, set the port R9 output latch to 1. In transmit mode, pin R90 can be used as a standard I/O port. In 4-bit receive mode, R91 can be used as a standard I/O port.

3.7.1.1 Configuration of Serial Interface

Figure 3-42 shows the configuration of the synchronous serial interface.

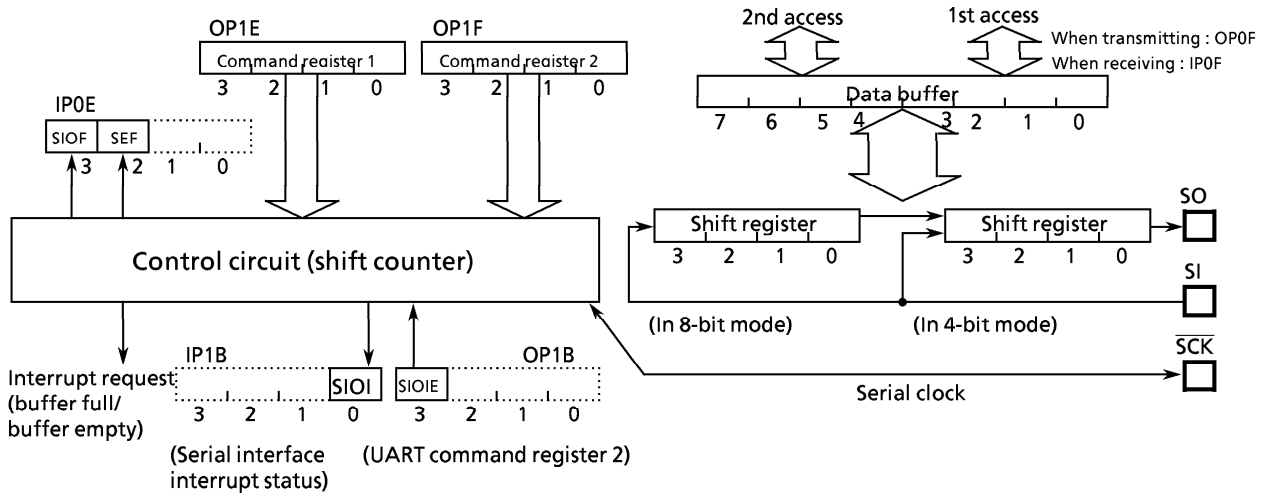


Figure 3-42. Configuration of Synchronous Serial Interface

3.7.1.2 Serial Interface Control

The serial interfaces are controlled by SIO command register 1 (OP1E), SIO command register 2 (OP1F), the status register (IP0E), the interrupt status register (IP1B), and UART command register 2 (OP1B).

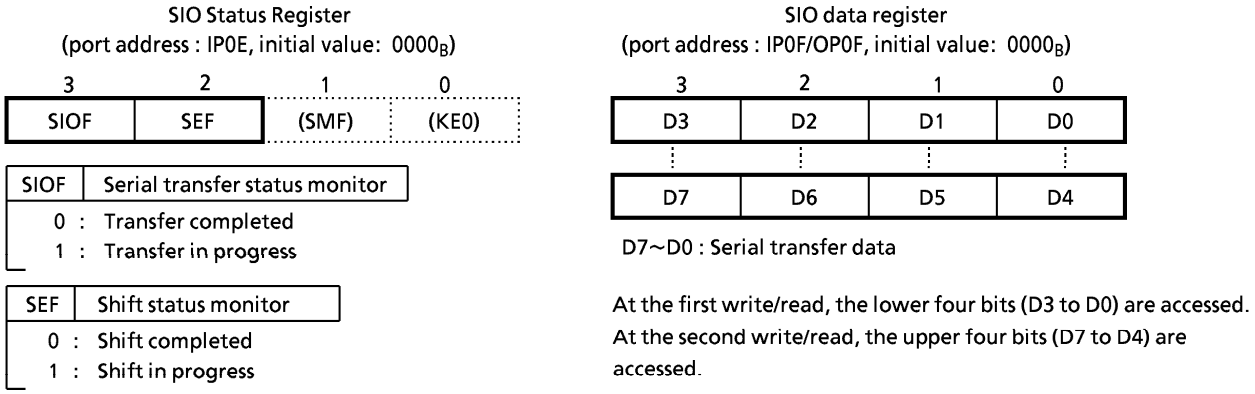


Figure 3-43. SIO Status Register and SIO Data Register

SIO command register 1
(port address : OP1E, initial value: 0001_B)

3	2	1	0
INH	SBIT	CKR	

INH	Serial transfer mandatory stop
0 :	Continues transfer.
1 :	Mandatory stop

SBIT	Number of transfer bits
0 :	4-bit transfer
1 :	8-bit transfer

CKR	Serial clock frequency
00 :	fc/64 [Hz]
01 :	fc/128
10 :	fc/512 or fs/4
11 :	fc/4096 or fs/32

SIO command register 2
(port address : OP1F, initial value: 0000_B)

3	2	1	0
ESIO	RM	LM	ECKM

ESIO	Starts/ends serial transfer.
0 :	Ends serial transfer.
1 :	Starts serial transfer.

RM	Selects transfer mode.	(Note 2)
	4-bit transfer	8-bit transfer
0 :	Transmit mode	Transmit mode
1 :	Receive mode	Receive mode

LM	Selects shift edge.	(Note 3)
0 :	Shifts at serial clock trailing edge.	
1 :	Shifts at serial clock leading edge.	

ECKM	Selects serial clock.
0 :	Internal clock (output to $\overline{SCK}$ pin)
1 :	External clock (input from $\overline{SCK}$ pin)

Notes
1. fc ; High-frequency clock [Hz]
   fs ; Low-frequency clock [Hz]
2. To enter transfer mode, set ESIO = 0 .
3. In transmit/receive mode, set "don't care". In transmit mode, set LM = 1.

Figure 3-44. SIO Command Registers

Serial interface interrupt status register
(port address : IP1B, initial value: 0010B)

3	2	1	0
(TDBEI)	(RDBFI)		SIOI

SIOI	SIO interrupt
1 :	Interrupt
0 :	No interrupt

UART command register 2
(port address : OP1B, initial value: 0000_B)

3	2	1	0
SIOIE	(TXIE)	(RXIE)	(STOP)

SIOIE	Controls SIO interrupt
1 :	Enables SIO interrupt.
0 :	Disables SIO interrupt.

Figure 3-45. Serial Interface Interrupt Status Register and UART Command Register 2

3.7.1.3 Serial Clock

The serial clock can be selected as follows.

(1) Clock source selection

The ECKM (bit 0 of SIO command register 2) is used to select the following:

(a) Internal clock (ECKM = "0")

The internal serial clock frequency is selected according to CKR (lower two bits of SIO command register 1). The internal serial clock is output externally from the $\overline{SCK}$ pin. At transmission start, the $\overline{SCK}$ pin output is high level. When data write (transmitting) or read (receiving) cannot keep pace with the serial clock rate, a wait function automatically stops the internal serial clock and holds up the next shift until processing is complete. The highest transfer speed for internal clock operation is 93750 bits/s (at fc = 6MHz).

(b) External clock (ECKM = "1")
The clock input externally to pin $\overline{SCK}$ is used as a serial clock. To use this clock, set the R92 ($\overline{SCK}$) output latch to 1. A reliable shift requires at least two instruction cycles whether the external serial clock is high or low level. When the transfer is complete, hold high level for at least two instruction cycles after the final shift clock.

(2) Shift edge selection
Select the following using the LM (SIO command register 2, bit 1):
(a) Leading edge shift (LM = "1")
Shifts data at the leading edge of the serial clock ($\overline{SCK}$ pin falling edge).
(b) Trailing shift (LM = "0")
Shifts data at the trailing edge of the serial clock ($\overline{SCK}$ pin rising edge). In transmit mode, the trailing edge shift is not supported.

3.7.1.4 Number of Transfer Bits

Select 4-bit serial transfer or 8-bit serial transfer using the SBIT (command register 1, bit 2).

(1) 4-bit serial transfer (SBIT = "0")
This mode transmits or receives in units of 4 bits. When the SIOIE bit of UART command register 2 is set to 1, an SIOI interrupt is generated after every four bits transferred.
The buffer register is accessed once to read receive data or write transmit data.

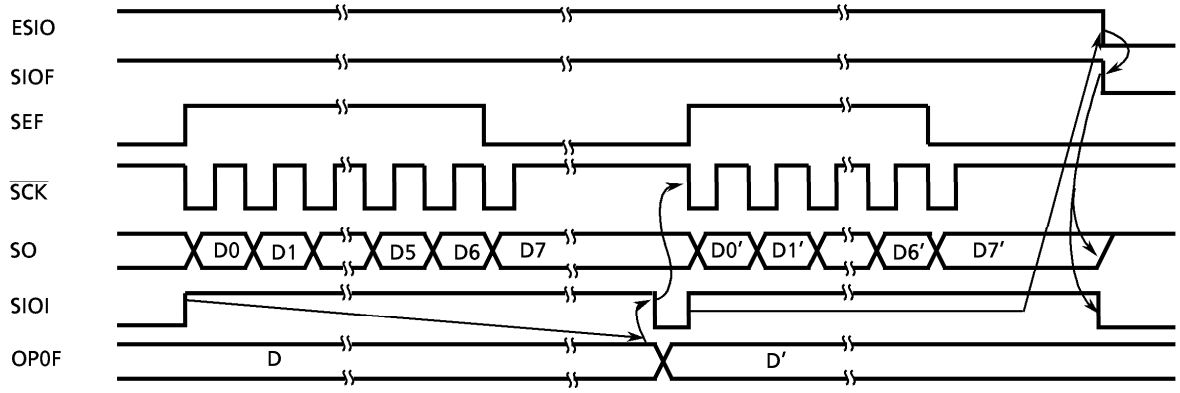
(2) 8-bit serial transfer (SBIT = "1")
This mode transmits or receives in units of 8 bits. When the SIOIE bit of UART command register 2 is set to 1, an SIOI interrupt is generated after every eight bits transferred. The buffer register is accessed twice to read receive data or write transmit data. After the transfer mode is set or an interrupt request is generated, at the first write/read the least significant four bits are accessed, and at the second write/read the most significant four bits are accessed.

3.7.1.5 Transfer Mode

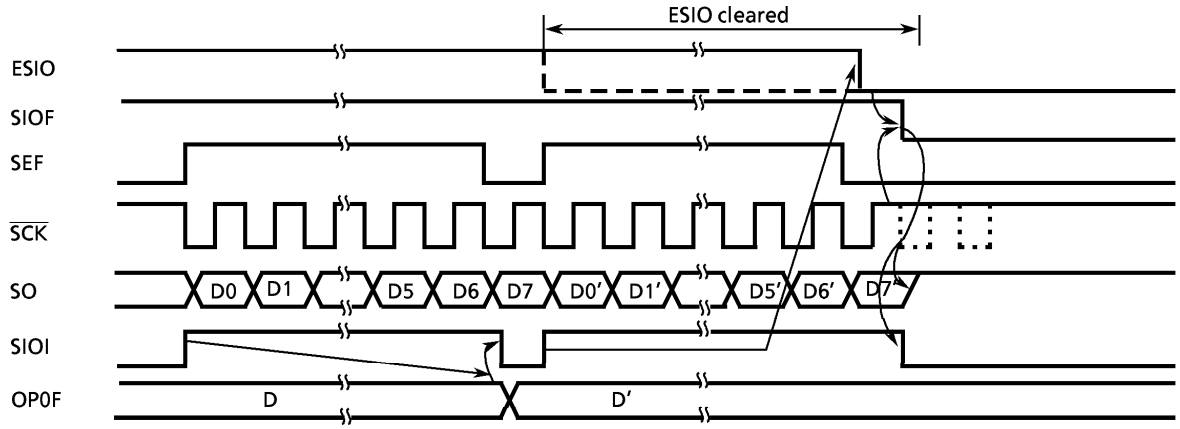
Select transmit mode, receive mode (at 4-bit transmission), or transmit/receive mode (at 8-bit transmission) using the RM (SIO command register 2, bit 2).

(1) Transmit mode (RM = 0 (4-Bit: SBIT = 0, 8-Bit: SBIT = 1))
After setting the transfer mode in the command register, write the first transmit data (4-bit or 8-bit) to the buffer register (OP0F) (if the transfer mode is not set to transmit mode, the data are not written in the buffer register). At 8-bit serial transfer, accessing the buffer register (OP0F) twice writes 8-bit data.
After transmit data are written, setting the ESIO to 1 starts transmission. Transmit data are synchronized with the serial clock and are output sequentially from the LSB to the SO pin. When the LSB data are output, the transmit data are shifted from the buffer register to the shift register. At that time, if the SIOIE bit of UART command register 2 is set to 1, an SIOI (SIO buffer empty) interrupt is generated to request the next transmit data. Writing the next transmit data to the buffer register by interrupt service program clears the interrupt request.
In internal clock mode, after transmitting all the 4-bit or 8-bit data, the serial clock stops and processing waits until the next transmit data are written to the buffer register.
In external clock mode, the data must be set in the buffer register prior to starting the next data shift. Therefore, the transfer rate is determined according to the maximum delay time from interrupt request generation until the interrupt service program writes data to the buffer register.

To terminate the transmission, instead of writing the next transmit data, clear the ESIO to 0 using the buffer empty interrupt service program. When ESIO is cleared, the transfer stops as soon as the transfer of the data currently shifted out ends. Transfer termination can be detected from the status of SIOF (status register, bit 3). (When transfer is complete, SIOF is cleared to 0).
Also, while the external clock is operating, clear ESIO to 0 before the next transmit data are shifted. If ESIO is not cleared before the data are shifted out, the transfer stops after transmitting the next 4-bit or 8-bit data (dummy data).



(a) Internal clock operation (with wait)



(b) External clock operation

Figure 3-46. Transfer Mode

Note : An SIOI interrupt is not generated unless the SIOIE bit of UART command register 2 is set to 1.

Example: Use the internal clock (fc/2^7) to transmit data (8-bit serial transfer) stored in data memory (specified by DMB and HL register pair).

```
LD      A, #0101B
OUT     A, %OP1E
LD      A, #0010B
OUT     A, %OP1F
OUT     @HL, %OP0F
INC     L
OUT     @HL, %OP0F
LD      A, #1010B
OUT     A, %OP1F
```

(2) 4-bit receive mode (RM = 1, SBIT = 0)

After receive mode is set in the command register, setting ESIO to 1 enables reception. The data are loaded in sync with the serial clock to the shift register via the SI pin. When the current 4-bit data are all loaded, they are transferred from the shift register to the buffer register (IP0F). If the SIOIE bit of UART command register 2 is set to 1 at that time, an SIOI (buffer full) interrupt is generated to request reading the receive data. The interrupt service program reads the data from the buffer register.

When the data are read, the interrupt request is cleared. After an interrupt is generated, the next data are read continuously. When the 4-bit data are loaded after the interrupt request is cleared, the data are transferred from the shift register to the buffer register.

In internal clock mode, if the previous transmit data are not read from the buffer register, serial clock stops and wait mode is entered until the data are read regardless of whether the loading of the next data is complete or not.

In external clock mode, because the shift is synchronized with an external clock, the data are read before the next receive data are transferred to the buffer register. If the previous data are not read, the receive data are not transferred to the buffer register and any receive data input subsequently are cancelled. The maximum transfer rate using an external clock is determined according to the maximum delay time from interrupt request generation until reading of the receive data. For receive data, a leading edge or a trailing edge shift can be selected. If a leading edge shift is selected, data are loaded at the leading edge of the serial clock. Therefore, before the first serial clock is applied at transfer start, the first shift data must be already input to the SI pin.

Note: If the SIOIE bit of UART command register 2 is not set to 1, no SIOI (buffer full) interrupt is generated even though the buffer is full at reception.

<1> Sufficiently slow transfer rate (external clock mode):
In external clock mode, clear ESIO to 0 before the next serial clock is applied after an SIOI (buffer full) interrupt. Then, read the final data.

Note: If the SIOIE bit of UART command register 2 is not set to 1, no SIOI (buffer full) interrupt is generated even though the buffer is full at reception.

Example: End reception at a sufficiently slow transfer rate. (Leading-edge shift)

```
LD      A, #0111B       ; ESIO ← 0 (Ends reception)
OUT     A, %OP1F
IN      %IP0F, A        ; Acc ← IP0F (Reads receive data)
```

<2> Fast transfer rate (internal/external clock mode)

When an interrupt request is received at a fast transfer rate, the loading of the next data may start before ESIO is zero-cleared. In such a case, before reading the next to last data, check that SEF (SIO status register, bit 2) is set to 1, clear ESIO to 0, then read the data. The interrupt service caused by the final data reception simply reads the receive data; reception terminates without further operation. This method is normally used when the internal clock operates. For external clock operations, the ESIO must be cleared and the receive data must be read before the final data are transferred to the buffer register.

Note: If the SIOIE bit of UART command register 2 is not set to 1, no SIOI (buffer full) interrupt is generated even though the buffer is full at reception.

Example: End reception at a fast transfer rate. (internal clock, leading-edge shift)

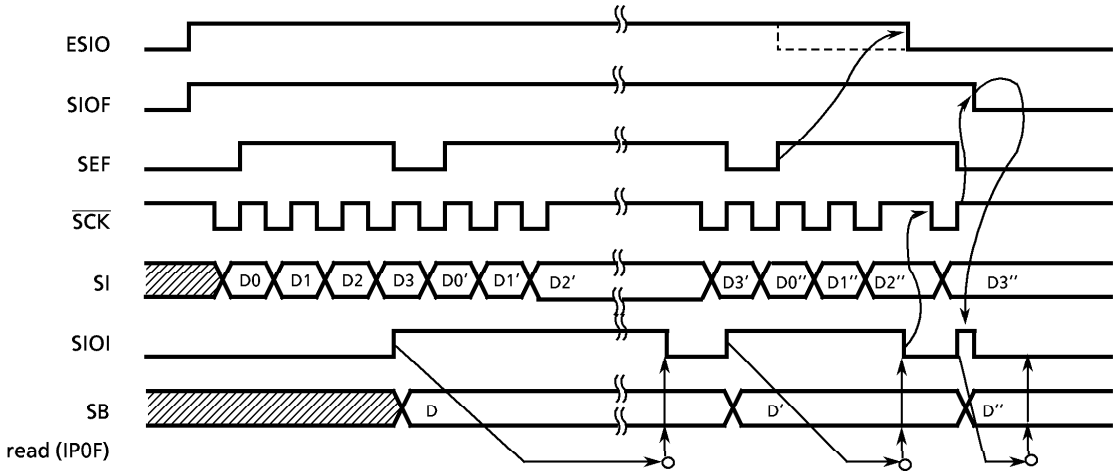
```
SSEF0 :   TEST   %IP0E, 2        ;  Waits until SEF = "1"
          B      SSEF0
          LD     A, #0110B       ;  ESIO ← 0
          OUT    A, %OP1F
          IN     %IP0F, A        ;  Acc ← IP0F (Reads receive data.)
```

<3> At one-word reception

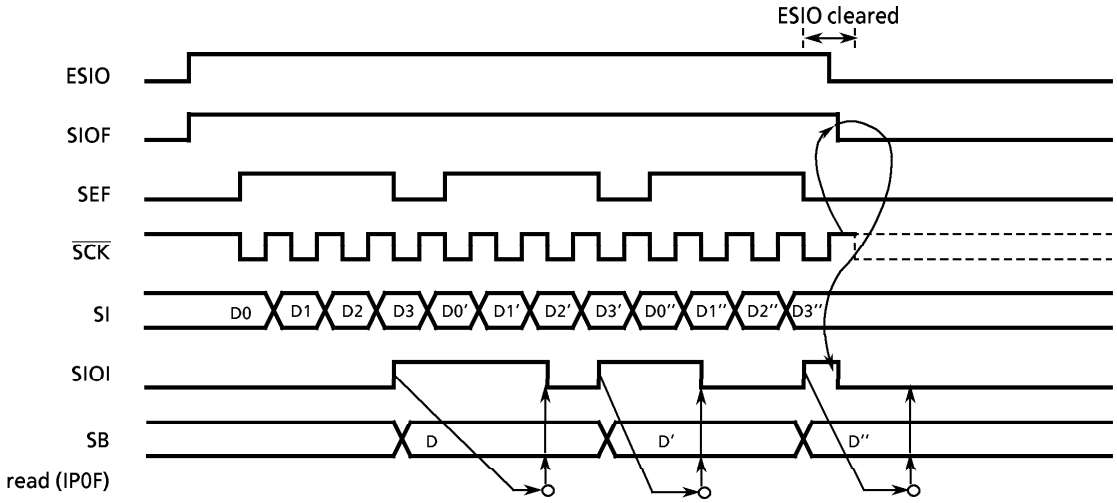
At one-word reception, set the SIOIE bit of UART command register 2 to 1, and set ESIO to 1 (indicates reception started). Check that SEF is set to 1 and returns ESIO to 0. As an SIO (buffer full) interrupt is generated once only, read the receive data using the interrupt service program.

Example: Start/end reception for one-word reception (internal clock, trailing edge)

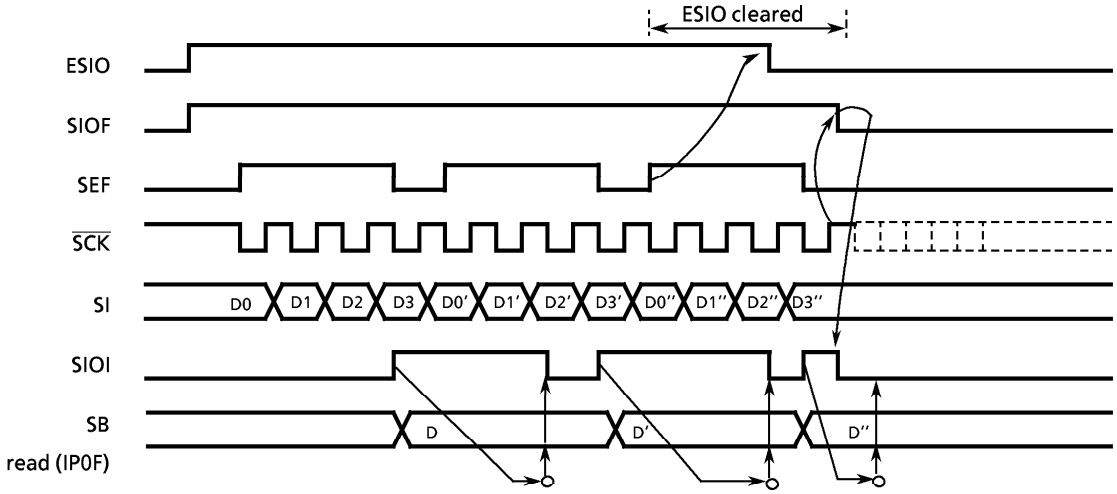
```
          LD     A, #0100B       ;  OP1F ← 0100B (Sets to receive mode.)
          OUT    A, %OP1F
          EI                     ;  EIF ← 1 (Enables interrupt.)
          LD     A, #1100B       ;  ESIO ← 1 (Starts reception.)
          OUT    A, %OP1F
SSEF0 :   TEST   %IP0E, 2        ;  Checks that SEF = 1.
          B      SSEF0
          LD     A, #0100B       ;  ESIO ← 0 (Ends reception.)
          OUT    A, %OP1F
```



(a) Internal clock operation, trailing-edge shift (with wait)



(b) External clock operation, leading-edge shift (slow transfer rate)



(c) Internal clock operation, leading-edge shift (fast transfer rate)

Note : If the SIOIE bit of UART command register 2 is not set to 1, no SIOI interrupt is generated.

Figure 3-47. 4-Bit Receive Mode

(3) 8-bit transmit/receive mode

After setting the RM to transmit/receive mode, write the first transmit data to the buffer register. Then, set ESIO to 1 to enable transmit/receive. Transmit data are output from the SO pin at the serial clock leading edge. Receive data are loaded from the SI pin at the trailing edge. When 8-bit data are loaded, the data are transferred from the shift register to the buffer register. At that time, if the SIOIE bit of UART command register 2 is set to 1, an SIOI (buffer full) interrupt is generated to request reading the receive data. The interrupt service program reads the receive data from the buffer register, then writes the transmit data.

In internal clock mode, processing waits until the receive data are read and the next transmit data are written.

In external clock mode, the shift is disabled until the receive data are read and the next transmit data are written. Therefore, the transmit data must be written before the next clock is supplied. The maximum transfer rate using an external clock is determined by the maximum delay time from interrupt request generation until the receive data are read and the transmit data are written.

The buffer register is used in both transmitting and receiving data. Therefore, when transmitting data, always write the transmit data after reading eight bits of receive data.

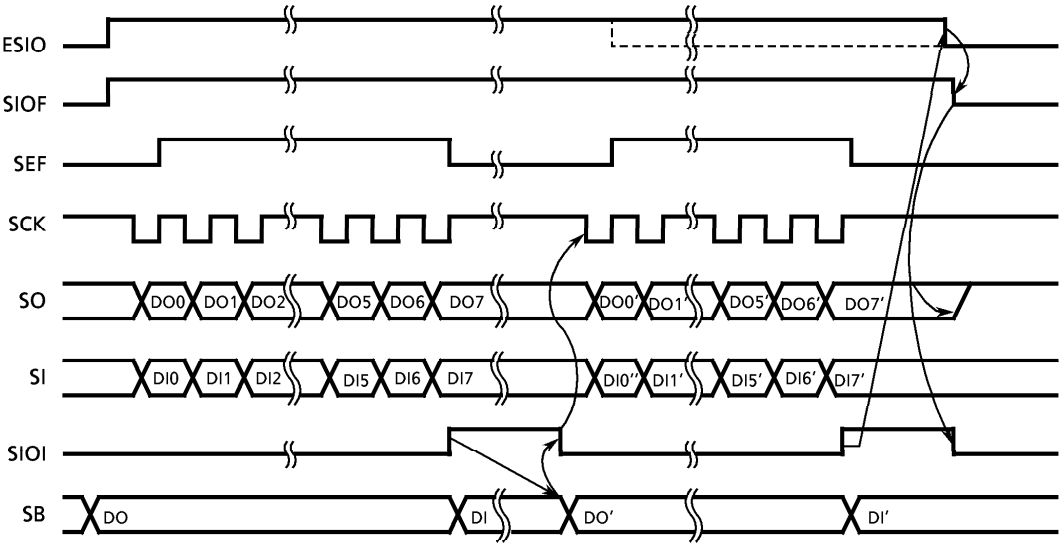
To terminate transmit or receive, clear ESIO to 0. When ESIO is zero-cleared, all 8 bits of the currently data are received and transferred to the buffer register, and transmit or receive terminates. To confirm by program the termination of transmit or receive, check the status of SIOF (SIO status register, bit 3), which is set to 0 at the termination of transmit or receive.

Example 1 : Write transmit data saved in RAM (8-bit transfer internal clock) and start transmit/receive

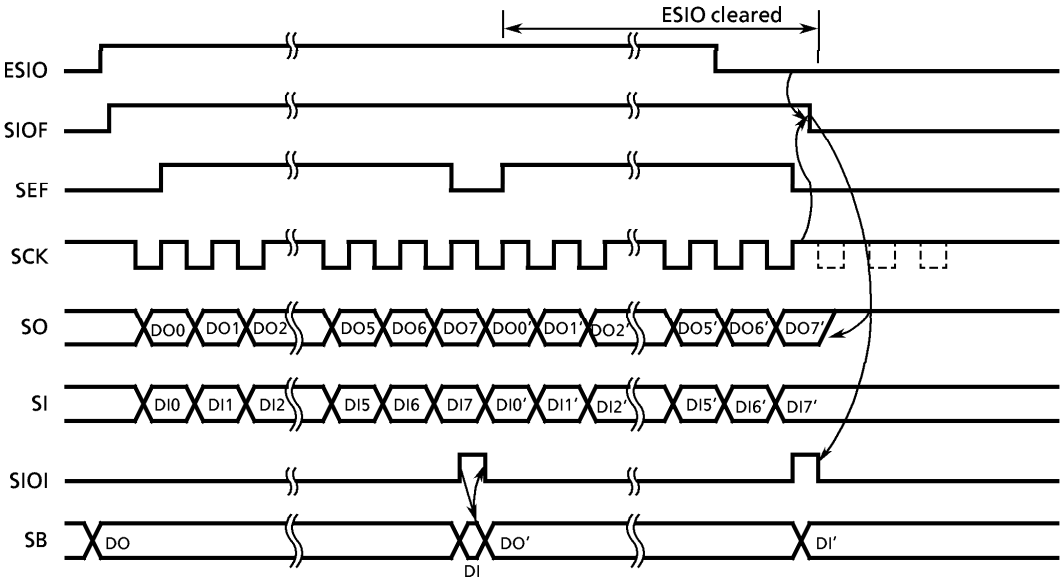
```
LD      A, #0100B
OUT     A, %OP1E
LD      A, #0110B
OUT     A, %OP1F
LD      HL, #20H
OUT     @HL, %OP0F
INC     L
OUT     @HL, %OP0F
LD      A, #1110B
OUT     A, %OP1F
 ⋮
```

Example 2 : Read receive data and write the next transmit data

```
LD      HL, #30H
IN      %IP0F, @HL
INC     L
IN      %IP0F, @HL
LD      HL, #22H
OUT     @HL, %OP0F
INC     L
OUT     @HL, %OP0F
```



(a) Internal clock operation



(b) External clock operation

Note : When the SIOIE bit of UART command register 2 is not set to 1, no SIOI interrupt is generated.

Figure 3-48. 8Bit Transmit/Receive Mode

3.7.1.6 Serial Transfer Mandatory Stop

During a serial transfer, the transfer can be forcibly halted.

Setting INH (command register 1, bit 3) to 1 clears ESIO to 0, stops the serial transfer (shift register operation), and initializes the shift counter. After the serial transfer is stopped, the $\overline{SCK}$ and SO pin outputs are set to high level and INH is automatically cleared to 0.

In transmit mode (including transmit/receive), the SO pin output is set to high level, but the contents of the shift register are not cleared. Therefore, the data immediately prior to the mandatory stop are output until the first shift data are output after the instruction to re-start transfer.

3.7.2 Asynchronous Serial Interface (UART)

TMP47E885AF incorporates an 8-bit buffered asynchronous serial interface. The asynchronous serial interface is connected to external circuits via pins R83 (TX) and R81 (RX). The serial port is also used as port R8. When using port R83 as a TX pin, set the port R83 output latch to 1. When using port R81 as an RX pin, set the port R81 output latch to 1.

3.7.2.1 UART Configuration

Figure 3-49 shows the UART configuration.

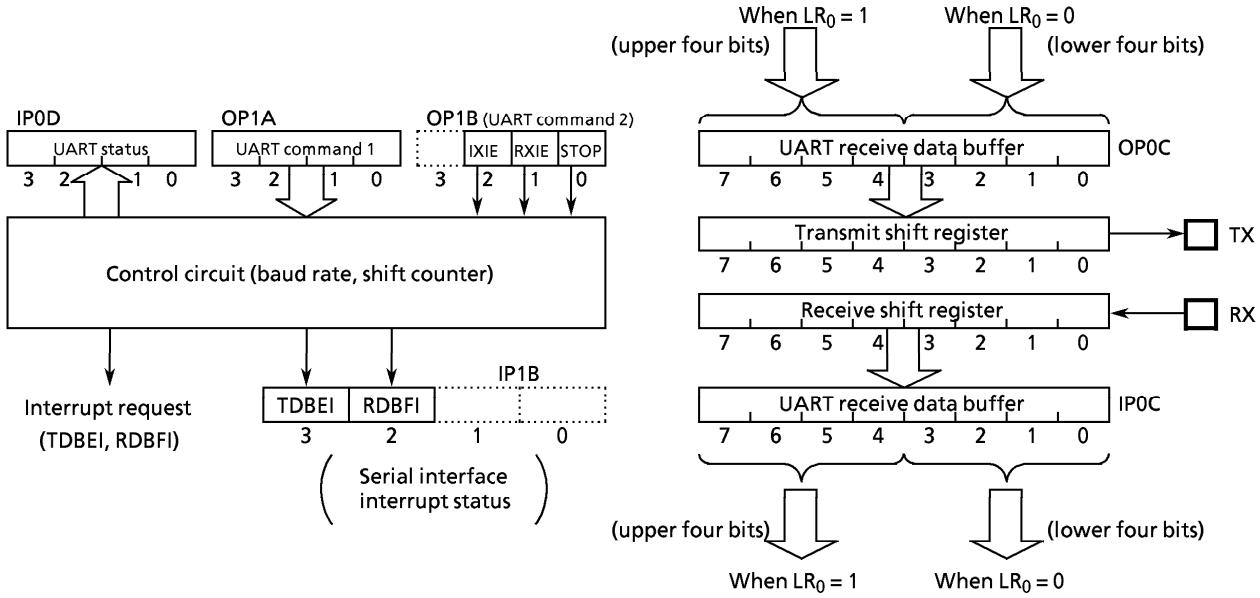


Figure 3-49. UART Configuration

3.7.2.2 UART Control

The UART is controlled by UART command register 1 (OP1A), UART command register 2 (OP1B), the UART status register (IP0D), and the serial interface interrupt status register (IP1B).

UART command register 1
(port address : OP1A, initial value : 0000_B)

3	2	1	0
TXE	RXE	BAUD RATE	

TXE	Enables transmitter.
1 :	Enables data transmit.
0 :	Disables data transmit.

RXE	Enables receiver.
1 :	Enables data receive.
0 :	Disables data receive.

BAUD RATE	Selects transfer clock.
00 :	10.4kbps (at 4MHz)
01 :	9600bps (at 4MHz)
10 :	4800bps (at 4MHz)
11 :	2400bps (at 4MHz)

UART command register 2
(port address : OP1B, initial value : 0000_B)

3	2	1	0
(SIOIE)	TXIE	RXIE	STOP

TXIE	Controls UART transmit interrupt.
1 :	Enables transmit data buffer empty interrupt.
0 :	Disables transmit data buffer empty interrupt.

RXIE	Controls UART receive interrupt.
1 :	Enables receive data buffer full interrupt.
0 :	Disables receive data buffer full interrupt.

STOP	Specifies number of transmit data stop bits.
1 :	One stop bit
0 :	Two stop bits

Note : When the TXE and RXE bits are set to 0 to disable interrupts, the interrupts are disabled after data are sent or received. When transmit data are saved in the transmit data buffer, the data are not transmitted until the next data are written, even when transmit is subsequently enabled.

Figure 3-50. UART Command Register

UART status register (port address : IP0D, initial value : 1000_B)

3	2	1	0
TDBE	RDBF	FE	ORE

TDBE Transmit data buffer empty
- 1 : Transmit data buffer empty. Enables transmit data write.
- 0 : Transmit data buffer full. Disables transmit data write.

RDBF Receive data buffer full
- 1 : Receive data. Enables receive data read.
- 0 : No receive data

FE Framing error flag
- 1 : Framing error
- 0 : No framing error

ORE Overrun error flag
- 1 : Overrun error
- 0 : No overrun error

Figure 3-51. UART Status Register

Serial interface interrupt status register (port address : IP1B, initial value : 0010_B)

3	2	1	0
TDBEI	RDBFI		(SIOI)

TDBEI UART transmit data buffer empty interrupt
- 1 : Interrupt
- 0 : No interrupt

RDBFI UART receive data buffer full interrupt
- 1 : Interrupt
- 0 : No interrupt

Figure 3-52. Serial Interface Interrupt Status Register

UART transmit data buffer register (port address : OP0C, initial value : 0000_B)

3	2	1	0	
D3	D2	D1	D0	(Writes when $LR_0 = 0$: clears TDBEI.)
D7	D6	D5	D4	(Writes when $LR_0 = 1$: clears TDBE and transmits data to transmit register.)

(D7 - D0 : Serial transmit data)

Writing data when $LR_0 = 0$ accesses the lower four bits (D3 to D0). Writing data when $LR_0 = 1$ accesses the upper four bits (D7 to D4).

Figure 3-53. UART Transmit Data Buffer Register

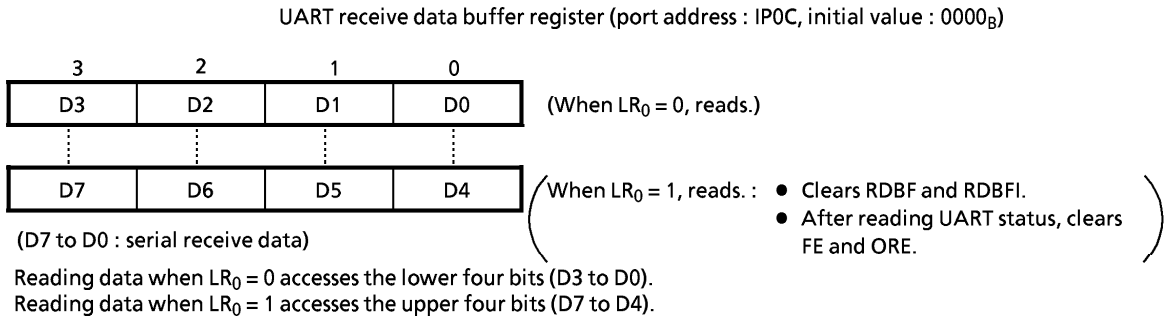


Figure 3-54. UART Receive Data Buffer Register

3.7.2.3 Data Transfer Baud Rate Selection

In UART mode, any of the four data transfer baud rates can be selected using BAUD RATE (UART command register 1, lower two bits).

3.7.2.4 Data Transfer Format

For data transferred in UART mode, one start bit (low level) and the number of stop bits (high level) specified by STOP (UART command register 2, bit 0) are added. Figure 3-55 shows the UART data transfer format.

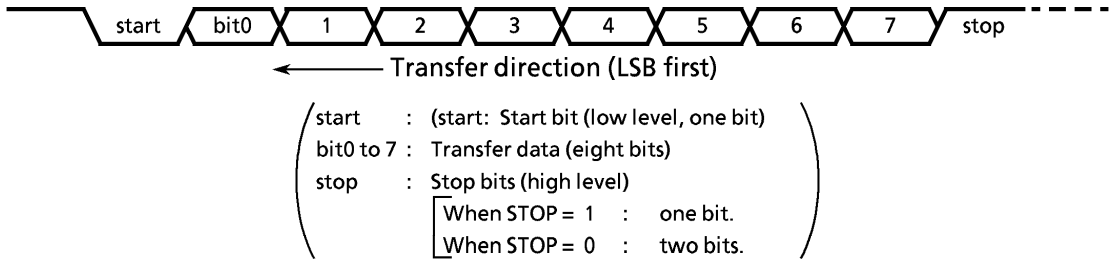


Figure 3-55. UART Data Transfer Format

3.7.2.5 UART Operation

(1) Data transfer

First, set port R83 to 1. Then, set bit TXE of UART command register 1 to 1 to enable the transmitter. Next, write the transmit data first to the lower four bits, then to the upper four bits of the transmit data buffer register. When the transmit data are written to the upper four bits of the transmit data buffer register, the eight bits of the transmit buffer register are transferred to the transmit shift register and sequentially transmitted from the TX pin. At that time, one start bit and the number of stop bits specified by STOP (UART command register 2, bit 0) are added to the data transmitted. The data transfer baud rate is selected by bits 1 and 0 of UART command register 1.

When the data written to the transmit data buffer are transferred to the transmit shift register, bit TDBE of the UART status register is set to 1 to show that the next data can be written. Before writing transmit data, read the UART status register. After confirming that the TDBE bit is set to 1, write the next transmit data. As data cannot be written when the TDBE bit is set to 0, wait until this bit is set to 1 before writing. When the next data are written in the UART transmit data buffer register, the TDBE bit is cleared to 0.

If TXIE (UART command register 2, bit 2) is set to 1, 8-bit data are transferred from the transmit data buffer register to the transmit shift register. When the transmit data buffer register is empty, an interrupt request (TDBEI) is issued. The interrupt request is cleared to 0 when data are written to the lower four bits of the transmit buffer register. When TXE is set to 1 and data transfer is enabled, TXIE can be set to 1.

When the TDBE bit is set to 0, do not write data to the transmit data buffer register.

(2) Data reception

First, set port R81 to 1. Set RXE (UART command register 1, bit 2) to 1 and enable reception. Then, when data are received from the RX pin, the receive data are transferred to the receive data buffer register. (The data are sampled at reception as described in 3.7.2.6.) At that time, one start bit and the number of stop bits specified by UART command register 2 are already included in the data received. However, only the receive data are read and transferred to the receive data buffer register. The data transfer baud rate is selected by BAUD RATE (the lower two bits of UART command register 1).

When the data are transferred to the receive data buffer register, bit RDBF of the UART status register is set to 1 indicating that data have been received. After the UART status register is read, reading the data from the receive data buffer register clears the RDBF bit to 0.

If RXIE (UART command register 2, bit 1) is set to 1, an interrupt request (RDBFI) is issued when bit RDBF is set to 1. Reading the upper four bits of the receive buffer register clears the interrupt request to 0. When RXE is set to 1 and data reception is enabled, RXIE can be set to 1.

If a framing or overrun error is detected at data reception, the FE or ORE bit of the UART status register is set to 1 to inform the processor of an error. When an error occurs, the data become invalid and are not transferred to the receive data buffer register. After the UART status register is read, reading the upper four bits of the receive data buffer register clears the FE or ORE bit to 0.

3.7.2.6 Data Sampling

The UART receiver samples the RX pin input at 16 times the baud rate selected by BAUD RATE (the lower two bits of UART command register 1) until the start bit is detected in the input. This rate, which is 16 times the baud rate, is called the RT (receive timing) rate or receiver clock. The start bit, all data bits, and the stop bit are sampled three times at intervals of 1RT (1RT is the position where the bits are expected to start) at the 7RT, 8RT, and 9RT positions as shown in Figure 3-56 Bit values are determined by a majority of the samplings (two or three of the three samplings).

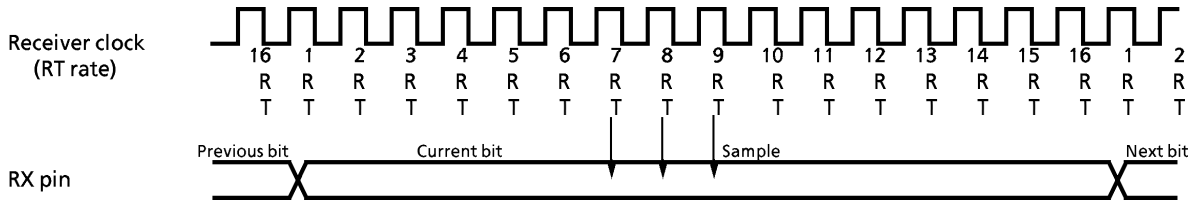
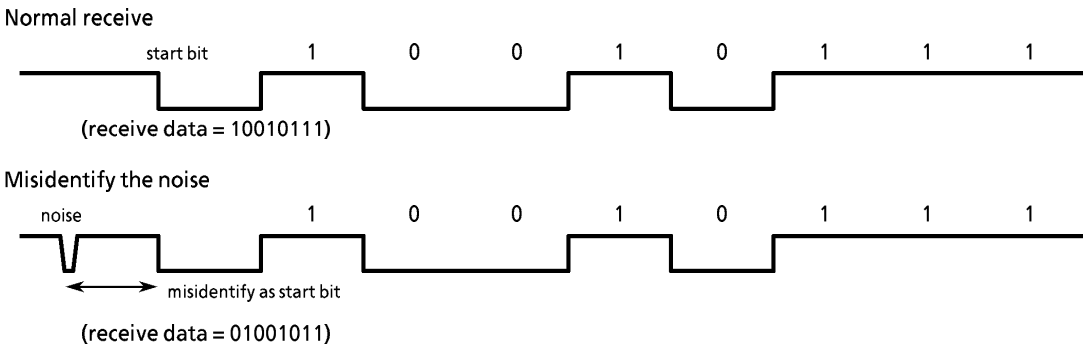


Figure 3-56. Sampling Used for All Bits

Note : Start bit is detected at the rising edge of RX. If noise occurs on RX, TMP47E885AF may start receive by misidentifying the noise as start bit.



3.7.3 Serial Interface Interrupt

There are three serial interface interrupt sources: a synchronous serial interface (SIO) interrupt source and two asynchronous serial interface (UART) interrupt sources. However, there is only one serial interface interrupt request (ISIF). An interrupt is requested by OR-ing three interrupt sources.
To determine the source of the interrupt, the interrupt handler routine reads the serial interface interrupt status register.

(1) SIO interrupt (SIOI)

Writing data to the buffer register at transmission or reading the buffer register at reception clears the status bit of SIO interrupt source to 0. Following that, if there is no transmit data, the interrupt request can be released by clearing the SIO interrupt control bit to 0 and terminating interrupt processing.

(2) UART interrupt (TDBEI, RDBFI)

Writing data to the lower four bits of the transmit buffer at transmission, or reading the upper four bits of the receive buffer register at reception clears the status bit of the UART interrupt source to 0. Following that, if there is no data for transmission, release the interrupt request by writing dummy data to the lower four bits of the transmit buffer register.

[Notes when processing timer/counter interrupts]
The timer/counter has multiple interrupt sources. However, only one interrupt request signal (ISIF) is provided. Accordingly, when multiple interrupts are generated simultaneously or, before interrupt processing begins, a dummy interrupt may be generated after the termination of interrupt processing. Make sure that if a dummy interrupt is generated, the program is not affected. The following describes the case of no dummy interrupt generation and the case of dummy interrupt generation.

(a) No dummy interrupt

The interrupt processing routine checks the interrupt sources and begins processing one of the multiple interrupts. The routine completes the processing using the RETI instruction without processing the other interrupts. Where each interrupt is sequentially processed, dummy interrupts do not occur.
The interrupt request is maintained until all the interrupts are processed. In the case of multiple interrupts, after processing is completed for one interrupt, reception is again enabled and the next interrupt is processed. The number of interrupts generated equals the number of interrupt sources. Processing is repeated until all the interrupts are processed.

(b) Dummy interrupt

The interrupt processing routine checks for interrupt sources and processes all interrupts. When the routine completes interrupt processing and the RETI instruction is executed, a dummy interrupt is generated.
The dummy interrupt is acknowledged by the interrupt processing routine when it checks to see if there are any interrupt sources and detects none. Then the interrupt processing routine processes the dummy interrupt. Interrupt processing completes after executing the RETI instruction again.

3.8 E²PROM

TMP47E885AF has 64 x 8 bits of internal E2PROM.

3.8.1 E²PROM Configuration

Figure 3-57 shows the E²PROM configuration.

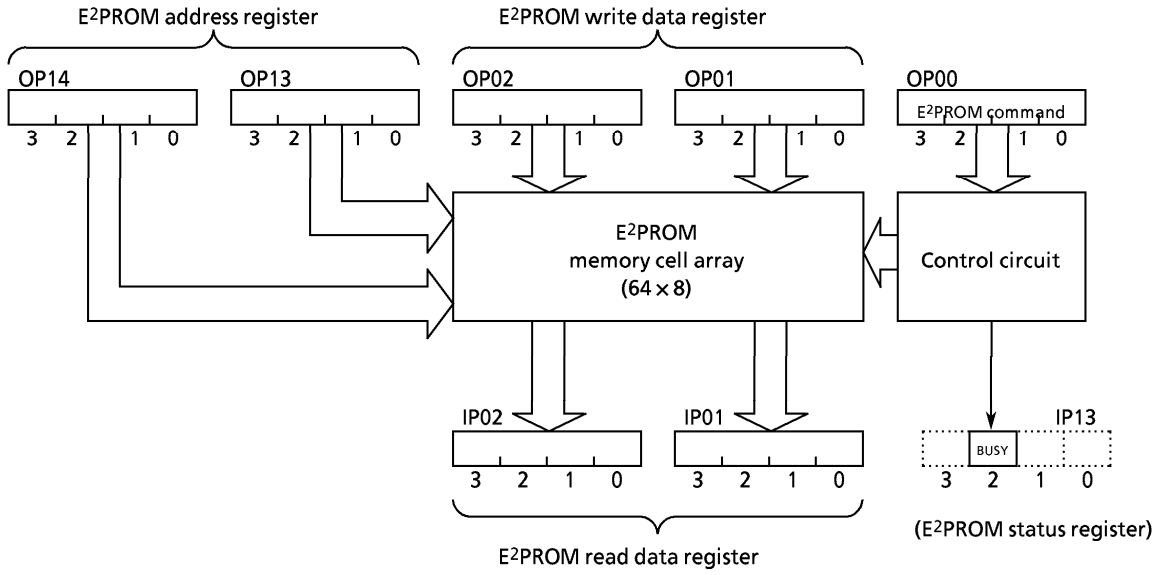


Figure 3-57. E²PROM Configuration

3.8.2 E²PROM Control

The E²PROM is controlled by the E²PROM command register (OP00), E²PROM address registers (OP14, OP13), E²PROM write data registers (OP02 and OP01), E²PROM read data registers (IP02 and IP01), and E²PROM status register.

(1) E²PROM command register

The E²PROM command register is used to set the E²PROM operating mode. Setting the operating mode in the command register executes the operation. When the operation terminates, the command register is automatically cleared to $X000_B$.

E²PROM command register (port address: OP00, initial value : 0000_B)

3	2	1	0
EEEN	MODE		

(Automatically cleared to X000 when the operation terminates.)

EEEN	Enables E²PROM.
0 :	Disables E²PROM
1 :	Enables E²PROM

MODE	Selects and executes operating mode.
000 :	No operation
001 :	Read
010 :	Write
011 :	Byte erase
100 :	Rewrite
Others :	Don't use.

Figure 3-58. E²PROM Command Register

(2) E²PROM address register

An 8-bit register that indicates the E²PROM address. For the lower four bits, access port address OP13. For the upper four bits, access port address OP14. At reset, the address register is initialized to 0.

The E²PROM addresses are 00_H - $3F_H$. Do not access the address range 40_H - FF_H.

E²PROM address register (port address : OP14, OP13, initial value : 00000000_B)

(OP14)				(OP13)			
3	2	1	0	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0

Figure 3-59. E²PROM Address Register

(3) E²PROM write data register

An 8-bit register that holds data to be written to E²PROM. For the lower four bits, access port address OP01. For the upper four bits, access port address OP02. At reset, the write data register is initialized to 0.

E²PROM write data register (port address : OP02, OP01, initial value : 00000000_B)

(OP02)				(OP01)			
3	2	1	0	3	2	1	0
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

Figure 3-60. E²PROM Write Data Register

(4) E²PROM read data register

An 8-bit register that holds data read from E²PROM. For the lower four bits, access port address IP01. For the upper four bits, access port address IP02. At reset, the read data register is undefined.

E²PROM read data register (port address : IP02, IP01, initial value : $********_B$)

(IP02)				(IP01)			
3	2	1	0	3	2	1	0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

Figure 3-61. E²PROM Read Data Register

(5) E²PROM status register

The status register is used to monitor the operating status of the E²PROM.

E²PROM status register (port address : IP13, initial value : 0011_B)

3	2	1	0
(EOCF)	BUSY		

BUSY : Monitors E²PROM operating status.
0 : Operation terminated or off
1 : Operating

Figure 3-62. E²PROM Status Register

3.8.3 E²PROM Operating Mode

When using E²PROM, set the EEEN bit of the E²PROM command register to 1. E²PROM is read in real time, but erase and write operations require an internal timer to be set. While the E²PROM is operating, the contents of the E²PROM command register, the address registers, and the write data registers cannot be altered. Before altering the contents of these registers, check that E²PROM operations have terminated, using the E²PROM status register. E²PROM erase, write, and read are supported only in normal operating mode.

(1) Read

Setting read mode in the E²PROM command register loads the data at the memory address specified by the address register to the read data register.

Before performing the read, first set the address in the E²PROM address register.

Example : Read data at E²PROM address $2C_H$ and load the data at data memory (RAM) addresses 30_H and 31_H.

```
LD      A, #0CH ; Sets address.
OUT     A, %OP13
LD      A, #2H
OUT     A, %OP14
OUT     #1001B, %OP00 ; Reads data.
LD      HL, #30H ; Loads data to RAM.
IN      %IP01, @HL
INC     L
IN      %IP02, @HL
```

(2) Byte erase

Setting byte erase mode in the E²PROM command register clears the data at the address specified in the address register to 00H.

Before performing a byte erase, first set the address in the E²PROM address register.

Example : Erase the byte at E²PROM address $1E_H$.

```
LD      A, #0E_H ; Sets address.
OUT     A, %OP13
LD      A, #1H
OUT     A, %OP14
OUT     #1011B, %OP00 ; Starts byte erase.
```

(3) Write

Setting write mode in the E²PROM command register writes the contents of the write data register to the address specified in the address register. (Data are written to the bits at the specified address whose data are erased.)

Before performing a write, first set the address in the E²PROM address register and set the write data in the write data register.

Example : Write $5A_H$ at E²PROM address 32_H.

```
      LD      A, #2H ; Sets address.
      OUT     A, %OP13
      LD      A, #3H
      OUT     A, %OP14
      OUT     #0AH, %OP01 ; Sets the write data.
      OUT     #5H, %OP02
      OUT     #1011B, %OP00 ; Starts byte erase.
LOOP  : TESTP %IP13, 2 ; Monitors status.
      B       LOOP
      OUT     #1010B, %OP00 ; Starts write.
```

(4) Overwrite

When overwrite mode is set in the E²PROM command register, the data at the address specified in the address register are erased and the contents of the write data register are written.

Before performing the overwrite, first set the address in the E²PROM address register, and set the data to write in the write data register.

Example : Overwrite data at E²PROM address 32_H with value $5A_H$.

```
LD      A, #2H ; Sets address.
OUT     A, %OP13
LD      A, #3H
OUT     A, %OP14
OUT     #0AH, %OP01 ; Sets the write data.
OUT     #5H, %OP02
OUT     #1100B, %OP00   ; Starts
                          rewriting
```

3.8.4 E²PROM Data Protection

E²PROM has no data protection. To access the E²PROM, set the registers of E²PROM by the instruction. If TMP47E885AF is operated out of the guaranteed range, data in the E²PROM may be changed by the runaways of the CPU. Under the condition out of the guaranteed range, such as power on or power off, please use the power-on-reset circuit and reset IC to reset the MCU certainly.

1. After power on, keep active Reset until Vcc stabilized.
2. Do not power off during E²PROM access.

ELECTRICAL CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (V_{SS} = 0V)

PARAMETER		SYMBOL	PIN	SPECIFICATION		UNIT
Power supply voltage		V_{DD}		– 0.3 to 6.5		V
Input Voltage		V_{IN}		– 0.3 to V_{DD} + 0.3		V
Output Voltage		V_{OUT1}	Ports R4, R5, R6, R7, R8, RA	– 0.3 to V_{DD} + 0.3		V
		V_{OUT2}	Ports R9, RB	– 0.3 to V_{DD} + 0.3		
Output Current (per 1 pin)		I_{OUT1}	Ports R4, R5, R6, R7, R8, RA	3.2		mA
		I_{OUT2}	Ports R9, RB (sink current)	3.2		
		I_{OUT3}	Ports R9, RB (source current)	1		
Output current (total for all pins)		ΣI_{OUT1}	Ports R4, R5, R6, R7, R8, RA	40		mA
		ΣI_{OUT2}	Ports R9, RB	20		
Power dissipation [Topr = + 85°C/ + 110°C]		PD		300		mW
Soldering Temperature (time)		Tsld		260 (10 s)		°C
Storage Temperature		Tstg		– 55 to 150		°C
Operating temperature	Product version	Topr		T_L	T_H	°C
	I version			– 40	+ 85	
	W version			– 40	+ 110	

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V, Topr = T_L to T_H)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Max.	UNIT
Power supply voltage	V_{DD}		At normal operation	4.5	5.5	V
			At slow operation	2.7		
			At hold operation	2.0		
High-level input voltage	V_{IH1}	Excluding hysteresis input	$V_{DD} \geqq 4.5V$	$V_{DD} \times 0.7$	V_{DD}	V
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
	V_{IH3}	At slow or hold operation	$V_{DD} < 4.5V$	$V_{DD} \times 0.9$		
Low-level input voltage	V_{IL1}	Excluding hysteresis input	$V_{DD} \geqq 4.5V$	0	$V_{DD} \times 0.3$	V
	V_{IL2}	Hysteresis input			$V_{DD} \times 0.25$	
	V_{IL3}	At slow or hold operation	$V_{DD} < 4.5V$		$V_{DD} \times 0.1$	
Clock frequency	fc	XIN, XOUT		0.4	6.0	MHz
	fs	XTIN, XTOUT		30	34	kHz

D.C. CHARACTERISTICS

(V_{SS} = 0V, Topr = T_L to T_H)

PARAMETER	SYMBOL	PINS	CONDITIONS		Min.	Typ.	Max.	UNIT
Hysteresis voltage	V_{HS}	Hysteresis Input			–	0.7	–	V
Input current	I_{IN1}	Ports $\overline{RESET}$, $\overline{HOLD}$, TEST, K0	V_{DD} = 5.5V, V_{IN} = 5.5V/0V		–	–	± 10	μA
	I_{IN2}	Ports R4 to RB						
resistance	R_{IN1}	$\overline{RESET}$			80	220	450	kΩ
	R_P	Ports R9, RB			3	10	35	
Output leak current	I_{LO}	Ports R4 to R8, RA	V_{DD} = 5.5V, V_{OUT} = 5.5V		–	–	+ 10	μA
High-level output voltage	V_{OH}	Ports R9, RB	V_{DD} = 4.5V, I_{OH} = – 60 μA		2.4	–	–	V
Low-level output voltage	V_{OL}	Ports R4 to RB	V_{DD} = 4.5V, I_{OL} = 1.6 mA		–	–	0.4	V
Low-level output current	I_{OL1}	Ports R4 to R8, RA	V_{DD} = 4.5V, V_{OL} = 1.0V		2.4	–	–	mA
	I_{OL2}	Ports R9, RB			2.4	–	–	
Power supply current at normal operation	I_{DD}	Except for E²PROM Erase / write	V_{DD} = 5.5V, fc = 4 MHz		–	3	6	mA
		During E²PROM Erase / write	V_{DD} = 5.5V, fc = 4 MHz		–	6	10	
Power supply current at slow operation	I_{DDS}		V_{DD} = 3.0V, fs = 32.768 kHz	version I	–	30	60	μA
				version W			120	
Power supply current at hold operation	I_{DDH}		V_{DD} = 5.5V	version I	–	0.5	20	μA
				version W			40	

Note 1 : Typ. values are based on Topr = 25 °C, V_{DD} = 5V

Note 2 : Input current : I_{IN1}, I_{IN2} : Excludes current due to built-in input (pull-up or pull-down) resistors.

Note 3 : Input current: I_{DD}, I_{DDH} : V_{IN} = 5.3V / 0.2V

Port R voltage level is assumed to be valid.

I_{DDS} : V_{IN} = 2.8V / 0.2V, low-frequency clock (XTIN, XTOUT connected) only oscillates.

A.C. CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 4.5V to 5.5V, Topr = T_L to T_H)

PARAMETER	SYMBOL	CONDITIONS	Min.	Max.	UNIT
Instruction cycle time	t_{cy}	At normal operation	1.3	20	μs
		At slow operation	235	267	
High-level clock pulse width	t_{WCH}	External clock operates.	80	–	ns
Low-level clock pulse width	t_{WCL}				
Reset pulse width	PW_{RSTL}	With stable oscillation	3	–	t_{cy}
External interrupt pulse width	PW_{EINT}		2	–	t_{cy}

A / D CONVERSION CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 4.5V to 5.5V, Topr = T_L to T_H)

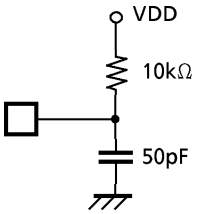
PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Analog reference power supply voltage	V_{AREF}		V_{DD} – 1.5	–	V_{DD}	V
	V_{ASS}		V_{SS}	–	1.5	
Analog reference power supply voltage range	ΔV_{AREF}	V_{AREF} – V_{ASS}	2.5	–	–	V
Analog input voltage range	V_{AIN}		V_{ASS}	–	V_{AREF}	V
Analog reference voltage power supply current	I_{REF}		–	0.5	1.0	mA
Non-linear error		V_{DD} = 5.0V, V_{SS} = 0.0V V_{AREF} = 5.000V V_{ASS} = 0.000V	–	–	± 1.5	LSB
Zero error			–	–	± 1.5	
Full-scale error			–	–	± 1.5	
Total error			–	–	± 2	
A/D conversion time	t_{ADC}		–	26	–	t_{cy}
A/D conversion sampling time	t_{AIN}	At fc = 4MHz	–	4	–	μs

SIO CHARACTERISTICS

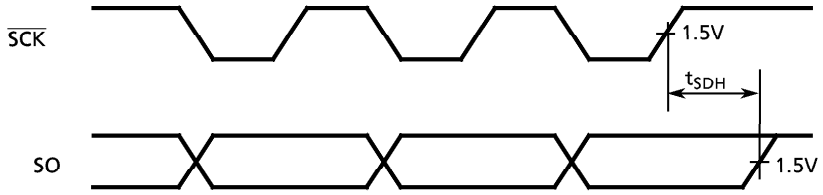
(V_{SS} = 0V, V_{DD} = 4.5 to 5.5V, Topr = T_L to T_H)

PARAMETER	SYMBOL	CONDITION	Min.	Max.	UNIT
Data transfer rate		When fc = 6MHz, internal clock operates.	–	93750	bps
Shift data hold time	t_{SDH}		$0.5t_{cy}$ – 300	–	ns
External clock pulse width	PW_{SCKH}	External clock operates.	2	–	t_{cy}
	PW_{SCKL}				

Note : Shift data hold time :
$\overline{SCK}$, SO pin External circuit



Serial port (end of transmission)



TIMER/COUNTER CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 4.5 to 5.5V, Topr = T_L to T_H)

PARAMETER	SYMBOL	CONDITION	Min.	Max.	UNIT
External count clock frequency	f_{CNT}	At normal operation	–	fc/16	Hz
		At slow operation	–	fs/16	
External input signal pulse width	PW_{TCIN}	At normal operation	4/fc	–	s
		At slow operation	4/fs	–	

PWM OUTPUT CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 4.5 to 5.5V, Topr = T_L to T_H)

PARAMETER	SYMBOL	CONDITION	Min.	Typ.	Max.	UNIT
PWM signal output frequency	f_{PWM}		–	–	fc/8192	Hz
Trigger signal input pulse width	PW_{TRG}		4/fc	–	–	s

E²PROM CHARACTERISTICS

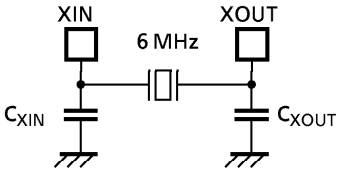
(V_{SS} = 0V, V_{DD} = 4.5 to 5.5V, Topr = T_L to T_H)

PARAMETER	SYMBOL	CONDITION		UNIT
Write time	t_{PW}		4.1 (Typ.)	ms
Erase time	t_{EW}		4.1 (Typ.)	ms
Number of rewrites		Topr = T_H	10^4 (Min.)	Times
Data retention characteristics		After executing 10^4 rewrites Ta = 55°C (average temperature)	10 (Min.)	Year

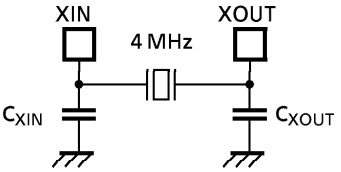
Note : Number of rewrites and data retention characteristics are intended as a guide to product capability.

RECOMMENDED OSCILLATION CONDITIONS (V_{SS} = 0V, V_{DD} = 4.5 to 5.5V, Topr = T_L to T_H)

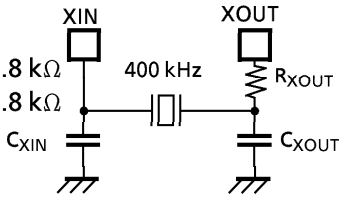
(1) 6 MHz
Ceramic oscillator
CSA6.00 MGU (MURATA) $C_{XIN} = C_{XOUT}$ = 30 pF
KBR-6.00 MS (KYOCERA) $C_{XIN} = C_{XOUT}$ = 30 pF



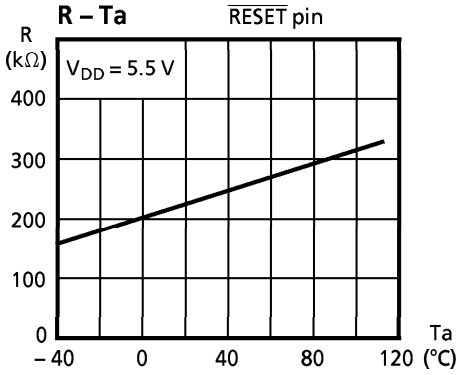
(2) 4 MHz
Ceramic oscillator
CSA4.00 MG (MURATA) $C_{XIN} = C_{XOUT}$ = 30 pF
KBR-4.00 MS (KYOCERA) $C_{XIN} = C_{XOUT}$ = 30 pF
Crystal Oscillator
204B-6F 4.0000 (TOYOCOM) CXIN = CXOUT = 20 pF

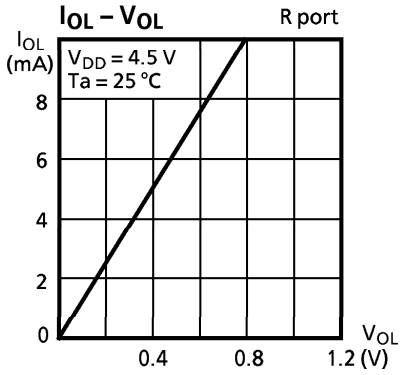


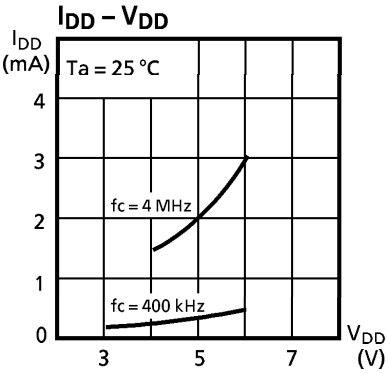
(3) 400 kHz
Ceramic oscillator
CSB400P (MURATA) $C_{XIN} = C_{XOUT}$ = 220 pF, R_{XOUT} = 6.8 kΩ
KBR-400B (KYOCERA) $C_{XIN} = C_{XOUT}$ = 220 pF, R_{XOUT} = 6.8 kΩ

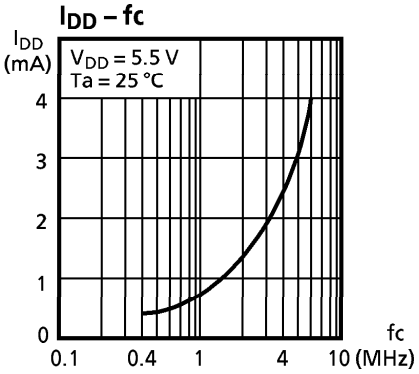


Electrical characteristics diagram









Port Condition by RESET Operation

The transition of Port condition by RESET operation is shown as below.

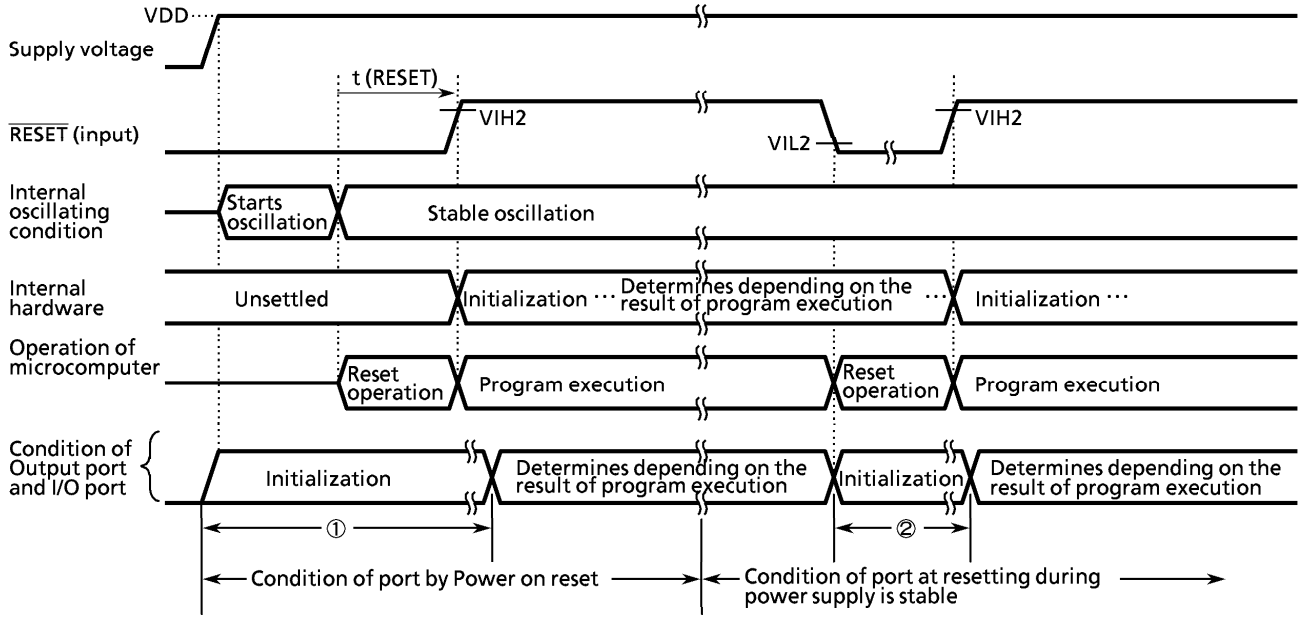


Figure 3-63. Port condition by Reset operation

Note 1: t (RESET) > 24/fc

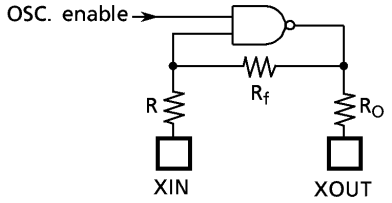
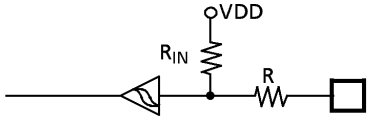
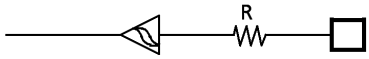
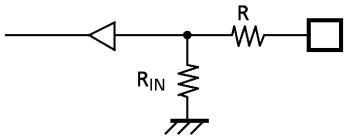
Note 2: VIL2 : Stands for low level input voltage of $\overline{\text{RESET}}$ pin.
VIH2 : Stands for high level input voltage of $\overline{\text{RESET}}$ pin.

Note 3: The term from power on reset to the time program is executed (above ①) and also the term starting from reset operation during power supply is stable to the program is executed (above ②), the port is on the initial condition. The initial condition of Port differs from I/O circuit by each port, refer to the section of "INPUT/OUTPUT CIRCUITRY". Thus, when using Port as an output pin, in the term of the above ① and ②, the voltage level on the signal that connects with the output pin of Port to the input pin of external application circuit should be determined by the external circuitry such as pull-up resistor and / or pull-down resistor.

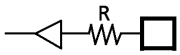
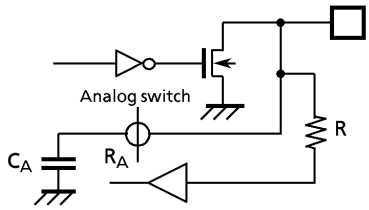
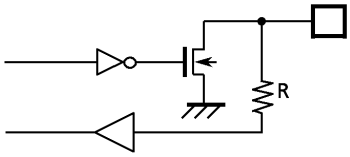
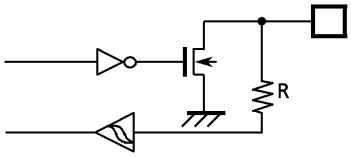
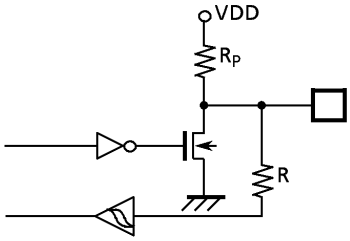
PIN I/O CIRCUITS

(1) Control pins

The following shows the control pin input/output circuits for TMP47E885AF.

Control Pin	I/O	CIRCUITRY	REMARKS
XIN XOUT	Input Output		Oscillator connecting pins R = 1 kΩ (typ.) R_f = 1.5 MΩ (typ.) R_O = 2 kΩ (typ.)
$\overline{RESET}$	Input		Hysteresis input Internal pull-up resistors R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{HOLD}$ ($\overline{KE0}$)	Input (Input)		Hysteresis input (Sense input) R = 1 kΩ (typ.)
TEST	Input		pull-down resistors R_{IN} = 70 kΩ (typ.) R = 1 kΩ (typ.)

(2) I/O ports

Port	I/O	CIRCUIT	REMARKS
K0	Input		R = 1 kΩ (typ.)
R4 R5	Input Output		Sink open drain output Initial "Hi-Z" R = 1 kΩ (typ.) Analog input R_A = 5 kΩ (typ.) C_A = 12 pF (typ.)
R6 R7	Input Output		Sink open drain output Initial "Hi-Z" R = 1 kΩ (typ.)
R8 RA	Input Output		Sink open drain output Initial "Hi-Z" Hysteresis input R = 1 kΩ (typ.)
R9 RB	Input Output		Sink open drain output (Internal pull-up resistor) Initial "Hi-Z" R_P = 10 kΩ (typ.) (In hold mode, turn pull-up resistor off.) Hysteresis input R = 1 kΩ (typ.)