CMOS 4-bit Microcontroller

TMP47C101P, TMP47C201P,

TMP47C101M, TMP47C201M

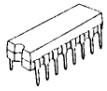
The 47C101/201 are high speed and high performance 4-bit single chip microcomputers, integrating ROM, RAM, input/output ports and timer/counters on a chip. The 47C101/201 are the standard type devices in the TLCS-47E series.

Part No.	ROM	RAM	Package	OTP	Piggyback + Adapter
TMP47C101P	1024 x 8-bit	64 x 4-bit	DIP16	TMP47P201VP	TMP47C990E + BM1160 (for DIP)
TMP47C101M			SOP16	T.B.D.	
TMP47C201P	2048 x 8-bit	128 x 4-bit	DIP16	TMP47P201VP	
TMP47C201M			SOP16	T.B.D.	

Features

- 4-bit single chip microcomputer
- Instruction execution time: 1.3μs (at 6MHz)
- Low voltage operation: 2.2V (at 2MHz RC)
- 89 basic instructions
  - Instruction set is the same as TLCS-47 series
- ROM table look-up instructions
- Subroutine nesting: 15 levels max.
- 5 interrupt sources (External: 2, Internal: 3)
  - All sources have independent latches each, and multiple interrupt control is available
- I/O port (11 pins)
- 12-bit Timer/Counters (TC2)
  - Timer, event counter, and pulse width measurement mode
- 12-bit programmable Timer (TC1)
- Interval Timer
- High current outputs
  - LED direct drive capability: typ. 20mA x 4 bits (Port R4)
- Hold function
  - Battery/Capacitor back-up
- Real Time Emulator: BM4721A + BM1160 (for DIP)

DIP16-P-300A

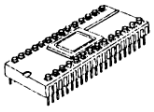


TMP47C101P
TMP47C201P
TMP47P201VP

SOP16-P-300

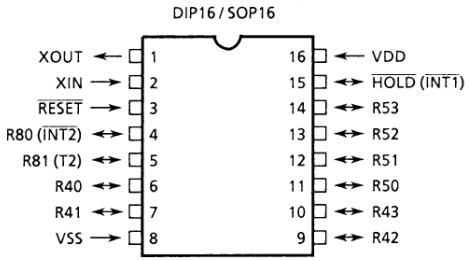
TMP47C101M
TMP47C201M

SDIC42



TMP47C990E

Pin Assignment (Top View)



Pin Function

Pin Name	Input/Output	Functions	
R43 to R40	I/O	4-bit I/O port with latch. When used as input port, the latch must be set to "1". Every bit data is possible to be set, cleared and tested by the bit manipulation instruction of the L-register indirect addressing.	
R53 to R50			
R81 (T2)	I/O (Input)	2-bit I/O port with latch. When used as input port, external interrupt pin, or timer/counter external input pin, the latch must be set to "1".	Timer/Counter 2 external input
R80 ($\overline{INT2}$)			External interrupt 2 input
XIN	Input	Resonator connecting pins, For inputting external clock, XIN is used and XOUT is opened.	
XOUT	Output		
$\overline{RESET}$	Input	Reset signal input	
$\overline{HOLD}$ ($\overline{INT1}$)	I/O (Input)	Hold request/release signal input	External interrupt 1 input and R82 I/O
VDD	Power Supply	+5 V	
VSS		0V (GND)	

Operational Description

Concerning the above component parts, the configuration and functions of hardwares are described.

1. System Configuration

Internal CPU Function

- 2.1 Program Counter (PC)
- 2.2 Program Memory (ROM)
- 2.3 H Register, L Register
- 2.4 Data Memory (RAM)
  - Stack
  - Stack Pointer Word (SPW)
  - Data Counter (DC)
- 2.5 ALU, Accumulator
- 2.6 Flags
- 2.7 System Controller
- 2.8 Interrupt Controller
- 2.9 Reset Circuit

Peripheral Hardware Function

- 3.1 I/O Ports
- 3.2 Interval Timer
- 3.3 Timer/Counters (TC1, TC2)

2. Internal CPU Function

2.1 Program Counter (PC)

The program counter is a 11-bit binary counter which indicates the address of the program memory storing the next instruction to be executed.Normally, the PC is incremented by the number of bytes of the instruction every time it is fetched. When a branch instruction or a subroutine instruction has been executed or an interrupt has been accepted, the specified values listed in Table 2-1 are set to the PC. The PC is initialized to "0" during reset.

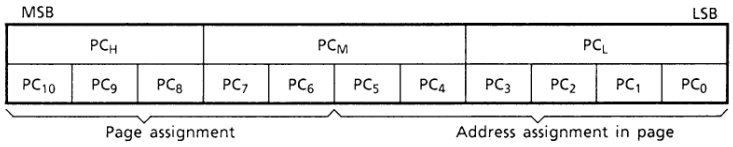


Figure 2-1. Configuration of Program Counter

The PC can directly address a 2048-byte address space. However, with the short branch, the following points must be considered:

- Short branch instruction [BSS a]

In [BSS a] instruction execution, when the branch condition is satisfied, the value specified in the instruction is set to the lower 6 bits of the PC.That is, [BSS a] becomes the in-page branch instruction. When [BSS a] is stored at the last address of the page, the upper 5 bits of the PC point the next page, so that branch is made to the next page.

Table 2-1 Status Change of Program Counter

Instruction or Operation		Condition		Program Counter (PC)										
				PC_{10}	PC_9	PC_8	PC_7	PC_6	PC_5	PC_4	PC3	PC_2	PC_1	PC_0
Execution of Instruction	BS a	SF = 1 (Branch condition is satisfied)		Immediate data specified by the instruction										
		SF = 0 (Branch condition is not satisfied)		+ 2										
	BSS a	SF = 1	Lower 6-bit address ≠ 111111	Hold							Immediate data specified by the instruction			
			Lower 6-bit address = 111111 (last address in page)	+ 1							Immediate data specified by the instruction			
		SF = 0		+ 1										
	CALL a			Immediate data specified by the instruction										
	CALLS a			0	0	0	The data generated by the immediate data specified by the instruction					1	1	0
	RET			The return address restored from stack										
	RETI			The return address restored from stack										
	Others			Incremented by the number of bytes in the instruction										
Interrupt acceptance				0	0	0	0	0	0	0	Interrupt vector			0
Reset				0	0	0	0	0	0	0	0	0	0	0

2.2 Program Memory (ROM)

Programs and fixed data are stored in the program memory. The instruction to be executed next is read from the address indicated by the contents of the PC.

The fixed data can be read by using the table look-up instructions.

- Table look-up instructions

  [LDL A, @DC], [LDH A, @DC+]

  The table look-up instructions read the lower and upper 4 bits of the fixed data stored at the address specified in the data counter (DC) to place them into the accumulator. [LDL A, @DC] instruction reads the lower 4 bits of fixed data, and [LDH A, @DC+] instruction reads the upper 4 bits.

  The DC is a 12-bit register, allowing it to address the entire program memory space.

  Example: When [LDL A, @DC] instruction is executed with the DC value being $7AO_H$ being 58_H, "8" is stored in the accumulator; when [LDH A, @DC+] instruction is executed, "5" is stored in the accumulator and the DC value is incremented to $7A1_H$.

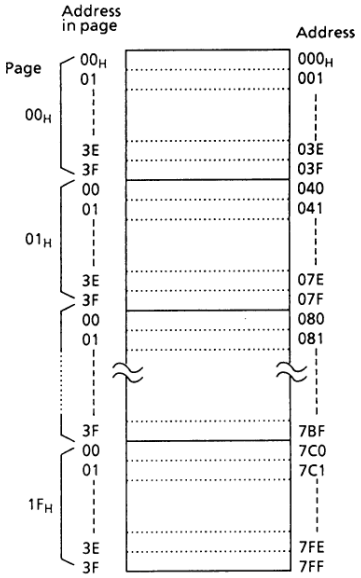


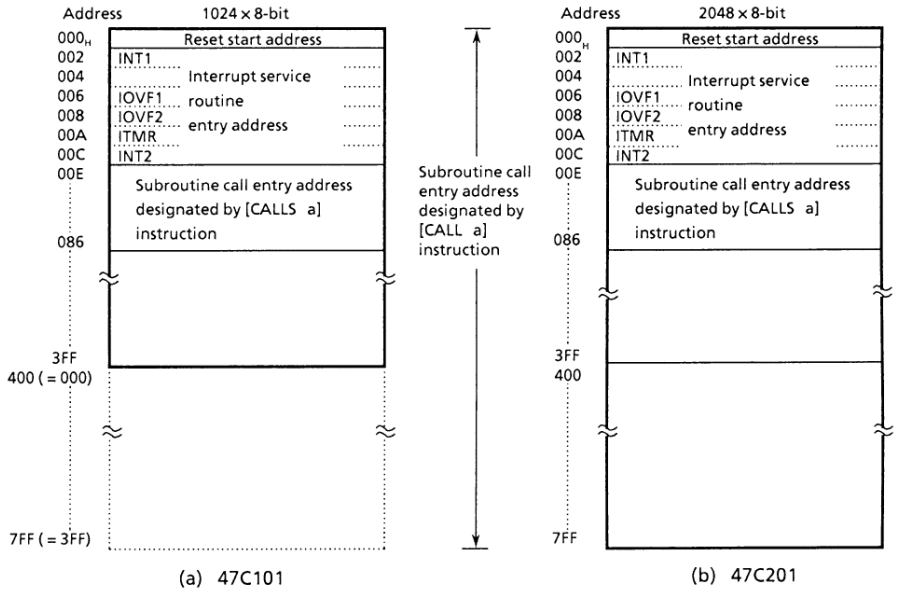
Figure 2-2. Configuration of Program Memory

2.2.1 Program Memory Capacity

The 47C101 has 1024 x 8 bits (addresses 000_H through $3FF_H$) of program memory (mask ROM), the 47C201 has 2048 x 8 bits (addresses 000_H through $7FF_H$).

2.2.2 Program Memory Map

Figure 2-3 shows the program memory map. Address 000_H - 086_H of the program memory are also used for special purposes. On the 47C101, no physical program memory exists in the address range 400_H through $7FF_H$. However, if this space is accessed by program, the most significant bit of each address is always regarded as "0" and the contents of the program memory corresponding to the address is always regarded as "0" and the contents of the program memory corresponding to the address 000_H through $3FF_H$ are read.



Note: Address 004_H and 005_H can be used to store ordinary user's processing data.

Figure 2-3. Program Memory Map

2.3 H Register and L Register

The H register and L register are 4-bit general registers. They are also used as a register pair (HL) for the data memory (RAM) addressing pointer. The RAM consists of pages, each page being 16 words long (1 word = 4 bits). The H register specifies a page and the L register specifies an address in the page.

The L register has the auto-post-increment/decrement capability, implementing the execution of composite instructions. For example, [ST A, @HL+] instruction automatically increments the contents of the L register after data transfer.

During the execution of [SET @L], [CLR @L], or [TEST @L] instructions, the L register is also used to specify the bits corresponding to I/O port pins R73 through R40 (the indirect addressing of port bits by the L register).

Example: To Write immediate values "5" and FH" to data memory addresses 10H and 11H.

```
LD    HL, #10H     ;    HL 10H
ST    #5,@HL+      ;    RAM [10H] 5H, LR←LR + 1
ST    #0FH,@HL+    ;    RAM [11H] FH, LR←LR + 1
```

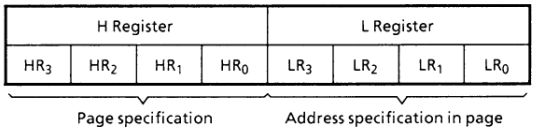


Figure 2-4. Configuration of H and L Registers

2.4 Data Memory (RAM)

The 47C101 has 64 x 4 bits (addresses 00_H through $3F_{H)}$ of the data memory (RAM), and the 47C201 has 128 x 4 bits (addresses 00_H through $7F_H$).

The RAM is addressed in one of the three ways (addressing modes):

(1) Register-indirect addressing mode

In this mode, a page is specified by the H register and an address in the page by the L register.

Example: LD A, @HL ; Acc←RAM [HL]

(2) Direct addressing mode

In this mode, an address is directly specified by the 8 bits of the second byte (operand) in the instruction field.

Example: LD A, 2CH ; Acc←RAM [$2C_H$]

(3) Zero-page addressing mode

In this mode, an address in zero-page (addresses 00_H through $0F_H$) is specified by the lower 4 bits of the second byte (operand) in the instruction field.

Example: ST #3, 05H ; RAM [05_H]←3

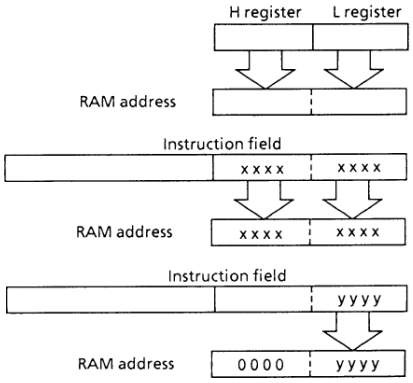


Figure 2-5. Addressing Mode

2.4.1 Data Memory Map

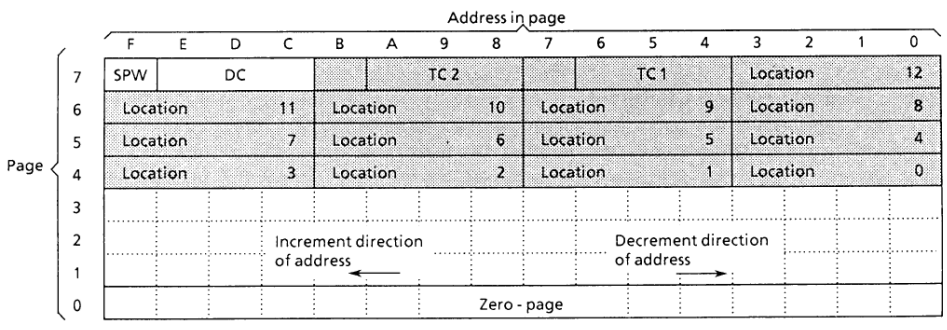
Figure 2-6 shows the data memory map. The data memory is also used for the following special purpose.

① Stack and Stack Pointer Word (SPW)

② Data Counter (DC)

③ Count registers of the timer/counters (TC1, TC2)

④ Zero-page



Note1. ▭ *denotes the stack area.*
Note2. *The TC1 and TC2 areas are shared by the locations 13 and 14.*

Figure 2-6. Data Memory Map (47C201)

(1) Stack

The stack provides the area in which the return address is saved before a jump is performed to the processing routine at the execution of a subroutine call instruction or the acceptance of an interrupt. When a subroutine call instruction is executed, the contents (the return address) of the program counter are saved; when an interrupt is accepted, the contents of the program counter and flags are saved.

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

The stack consists of up to 15 levels (locations 0 through 14) which are provided in the data memory (addresses 40_H through $7B_H$). Each location consists of 4-word data memory. Locations 13 and 14 are shared with the count registers of the timer/counters (TC1, TC2) to be described later.

The save/restore locations in the stack are determined by the stack pointer word (SPW). The SPW is automatically decremented after save, and incremented before restore.That is, the value of the SPW indicates the stack location number for the next save.

(2) Stack Pointer Word (SPW)

Address $7F_H$ ($3F_H$ for the 47C101) in the data memory is called the stack pointer word, which identifies the location in the stack to be accessed (save or restore).

Generally, location number 0 to 12 can be set to the SPW, providing up to 13 levels of stack nesting. Locations 13 and 14 are shared with the timer/counters to be described later; therefore, when the timer/counters are not used, the stack area of up to 15 levels is available. Address $7F_H$ is assigned to the SPW, so that the contents of the SPW cannot be set “15” in any case.

The SPW is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost.(For example, when the user-processed data area is in an address range 00_H through $4F_H$, up to location 4 of the stacks are usable.If an interrupt is accepted with location 4 already used, the user-processed data stored in addresses $4C_H$ through $4F_H$ corresponding to the location 3 area is lost.)

The SPW is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of “12” is used.

Example: To initialize the SPW (when the stack is used from location 12)

```
LD    A,#12   ;    SPW←12
ST    A,0FFH
```

(3) Data Counter (DC)

The data counter is a 12-bit register to specify the address of the data table to be referenced in the program memory (ROM). Data table reference is performed by the table look-up instructions [LDL A, @DC] and [LDH A, @DC +]. The data table may be located anywhere within the program memory address space.

The DC is assigned with a RAM address in unit of 4 bits. Therefore, the RAM manipulation instruction is used to set the initial value or read the contents of the DC.

Example: To set the DC to 380_H.

```
LD    HL,#07CH   ;  Sets RAM address of
                    DCL to HL register pair.
ST    #0H,@HL+   ;  DC←380H
ST    #8H,@HL+
ST    #3H,@HL+
```

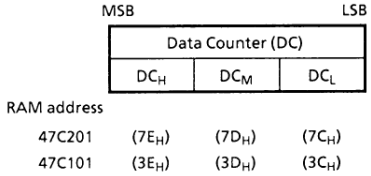


Figure 2-7. Data Counter

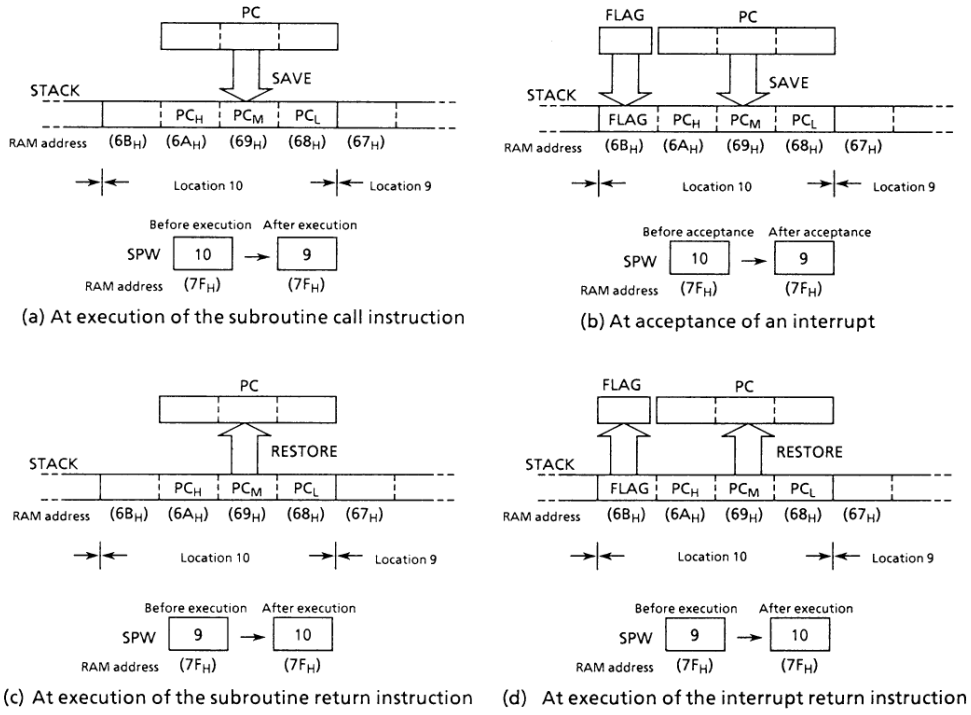


Figure 2-8. Accessing Stack (Save/Restore) at the 47C201

(4) Count registers of the timer/counters (TC1, TC2)

The 47C101/201 has two channels of 12-bit timer/counters. The count register of the timer/counter is assigned with a RAM addresses in unit of 4 bits, so that the initial value is set and the contents are read by using the RAM manipulation instruction.

The count registers are shared with the stack area (locations 13 and 14) described earlier, so that the stack is usable from location 13 when the timer/counter 1 is not used. When none of timer/counter 1 and timer/counter 2 are used, the stack is usable from location 14.

When both timer/counter 1 and timer/counter 2 are used, the data memory locations at addresses 77_H and $7B_H$ (37_H and $3B_H$ for the 47C101) can be used to store the user-processed data.

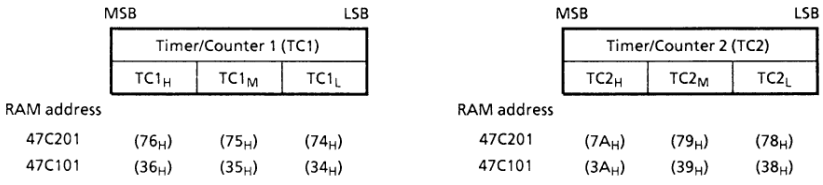


Figure 2-9. Count Registers of the Timer/Counters (TC1, TC2)

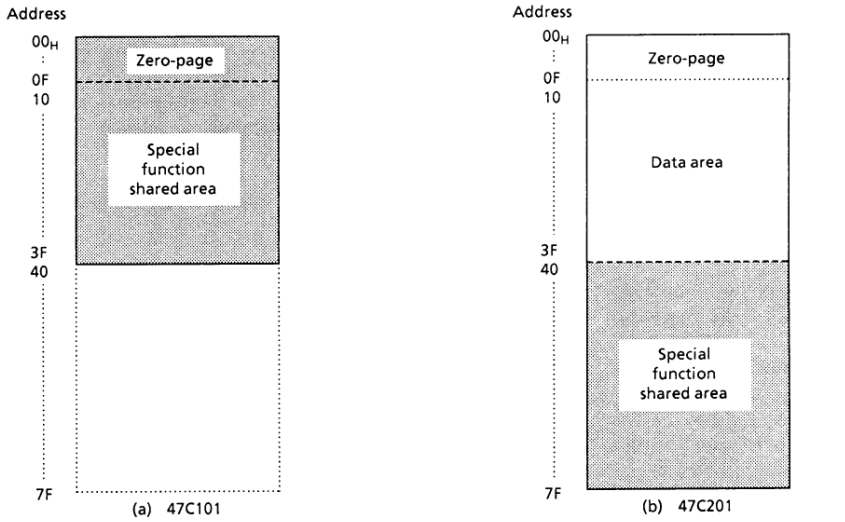
(5) Zero-page

The 16 words (at addresses 00_H through $0F_H$) of the zero page of the data memory can be used as the user flags or pointers by using zero-page addressing mode instructions (comparison, addition, transfer, and bit manipulation), providing enhanced efficiency in programming.

Example: To write immediate data "8" to address 09_H if bit 2 at address 04_H in the RAM is "1".

```
TEST  04H,2   ;  Skips if bit 2 at address 04H in the RAM is "0".
B     SKIP
ST    #8, 09H ;  Writes "8" to address 09H in the RAM
SKIP:
```

2.4.2 Data Memory Capacity



Note: In the 47C101, Zero-page and Special function shared area are overlapped. Programing should be performed assuming that the RAM is assigned to addresses 00 to $0F_H$ and D0 to FF_H. The technical data sheets for the 47C990 shall also be referred to.

Figure 2-10. Data Memory Capacity and Address Assignment

When power-on is performed, the contents of the RAM become unpredictable, so that they must be initialized by the initialization routine.

Example: To clear RAM (use common to the 47C101 and 47C201)

```
        LD    HL, #00H    ;  HL←00H
SCLRRAM: ST   #0, @HL+    ;  RAM [HL]←0, LR
                             LR + 1
        B     SCLRRAM
        ADD   H, #1       ;  HR←HR + 1
        B     SCLRRAM
```

2.5 ALU and Accumulator

2.5.1 Arithmetic/Logic Unit (ALU)

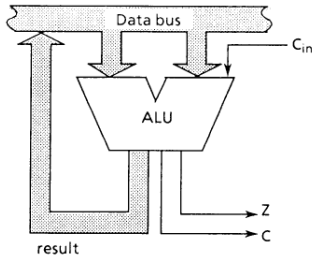
The ALU performs the arithmetic and logic operations specified by instructions on 4-bit binary data and outputs the result of the operation, the carry information(C), and the zero detect information (z).

(1) Carry information (C)

The carry information indicates a carry-out from the most significant bit in an addition. A subtraction is performed as addition of two's complement, so that, with a subtraction, the carry information indicates that there is no borrow to the most significant bit. With a rotate instruction, the information indicates the data to be shifted out from the accumulator.

(2) Zero detect information (Z)

This information is "1" when the operation result or the data to be transferred to the accumulator/data memory is "0000_B".



Note. C_{in} indicates the carry input specified by instruction

Figure 2-11. ALU

Example: The carry information and zero detect information for 4-bit additions and subtractions.

Operation	Result	C	Z
4 + 2 =	6	0	0
7 + 9 =	0	1	1
9 + 9 =	2	1	0

Operation	Result	C	Z
8 - 1 =	7	1	0
2 - 2 =	0	1	1
5 - 8 =	3 (1101_B)	0	0

2.5.2 Accumulator (Acc)

The accumulator is a 4-bit register used to hold source data or results of the operations and data manipulations.

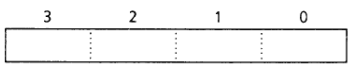


Figure 2-12. Accumulator

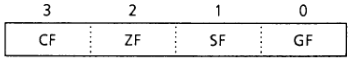


Figure 2-13. Flags

2.6 Flags

There is a carry flag (CF), a zero flag (ZF), a status flag (SF), and a general flag (GF), each consisting of 1 bit. These flags are set or cleared according to the condition specified by an instruction. When an interrupt is accepted, the flags are saved on the stack along with the program counter. When the [RETI] instruction is executed, the flags are restored from the stack to the states set before interrupt acceptance.

(1) Carry flag (CF)

The carry flag holds the carry information received from the ALU at the execution of an addition/subtraction with carry instruction, a compare instruction, or a rotate instruction. With a carry flag test instruction, the CF holds the value specified by it.

① Addition/subtraction with carry instructions [ADDC A, @HL], [SUBRC A, @HL]

The CF becomes the input (C_{in}) to the ALU to hold the carry information.

② Compare instructions [CMPR A, @HL], [CMPR A, #k]
The CF holds the carry information (non-borrow).

③ Rotate instructions [ROLC A], [RORC A]
The CF is shifted into the accumulator to hold the carry information (the data shifted out from the accumulator).

④ Carry flag test instructions [TESTP CF], [TEST CF]
With [TESTP CF] instruction, the content of the CF is transferred to the SF then the CF is set to "1".
With [TEST CF] instruction, the value obtained by inverting the content of the CF is transferred to the SF then the CF is cleared to "0".

(2) Zero flag (ZF)

The zero flag holds the zero detect information (Z) received from the ALU at the execution of an operational instruction, a rotate instruction, an input instruction, or a transfer-to-accumulator instruction.

(3) Status flag (SF)

The status flag provides the branch condition for a branch instruction. Branch is performed when this flag is set to "1". Normally the SF is set to "1", so that any branch instruction can be regarded as an unconditional branch instruction. When a branch instruction is executed upon set or clear of the SF according to the condition specified by an instruction, this instruction becomes a conditional branch instruction. During reset, the SF is initialized to "1", other flags are not affected.

(4) General flag (GF)

This is a 1-bit general-purpose flag which can be set, cleared, or tested by program.

Example: When the following instruction are executed with the accumulator, H register, L register, data memory (address 07H), and carry flag being set to "C_H", "0", "7", "5", and "1" respectively, the contents of the accumulator and flags become as follows:

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	C_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 Controller

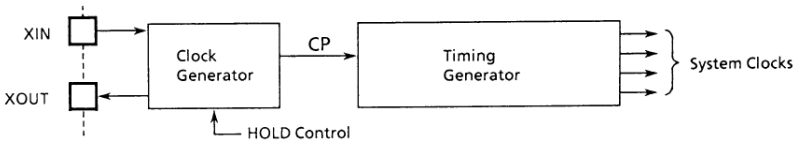


Figure 2-14. Clock Generator and Timing Generator

2.7.1 Clock Generator

The clock generator provides the basic clock pulse (CP) by which the system clock to be supplied to the CPU and the peripheral hardware is produced. The CP can be easily obtained by connecting the resonator to the XIN and XOUT pins. (RC oscillation is also possible, depending on the mask option) The clock from the external oscillator is also available. In the hold operating mode, the clock generator stops oscillating.

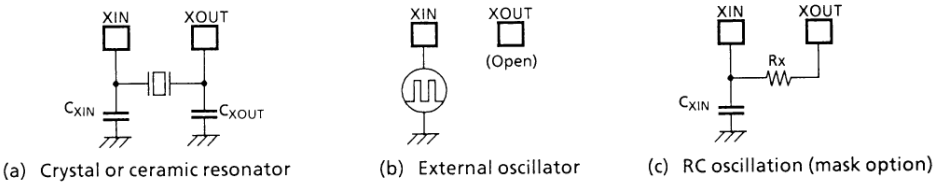
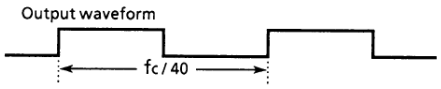


Figure 2-15. Examples of Oscillator Connection

Note: Accurate adjustment of the oscillation frequency.
Although the hardware to externally and directly monitor the CP is not provided, the oscillation frequency can be adjusted by making the program to output the pulse with a fixed frequency to the port with the all interrupts disabled and timer/counters stopped and monitoring this pulse. With a system requiring the oscillation frequency adjustment the adjusting program must be created beforehand.

Example: To output the oscillation frequency adjusting monitor pulse to port R40.

```
SFCCHK:   SET    %OP04,0
          CLR    %OP04,0
          BSS    SFCCHK
```



2.7.2 Timing Generator

The timing generator produces the system clocks from basic clock pulse (CP) which are supplied to the SPU and the peripheral hardware.

The timing generator consists of a 18-stage binary counter with a divided-by-16 prescaler. The basic clock (frequency: fc) provides the timing generator. Therefore, the output frequency at the last stage is $fc/2^{22}$[Hz]. During reset, the binary counter is cleared to "0", however, the prescaler is not cleared.

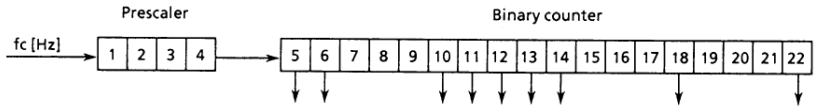


Figure 2-16. Configuration of Interval Timer

The timing generator provides the following functions:

① Generation of an internal source clock for interval timer

② Generation of an internal source clock for timer/counters

③ Generation of a warm-up time for releasing of the hold operating mode

2.7.3 Instruction Cycle

The instruction execution and the on-chip peripheral hardware operations are performed in synchronization with the basic clock pulse (CP: fx [Hx]). The smallest unit of instruction execution is called an instruction cycle. The instruction set of the TLCS-47 series consists of 1-cycle instructions and 2-cycle instructions. The former requires 1 cycle for their execution; the latter, 2 cycles. Each instruction cycle consists of 4 states (S1 through S4). Each state consists of 2 basic clock pulses.

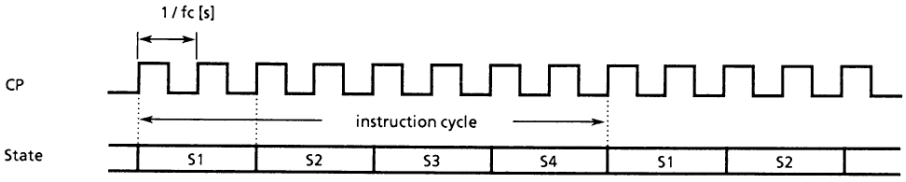


Figure 2-17. Instruction Cycle

2.7.4 Hold Operating Mode

The hold feature stops the system and holds the system's internal states active before stop with a low power. The hold operation is controlled by the command register (OP10) and the $\overline{\text{HOLD}}$ pin input. The $\overline{\text{HOLD}}$ pin input state can be known by the status register (IPOE). The $\overline{\text{HOLD}}$ pin is wired with the R82 output latch. To use this port for hold operating mode, the R82 output latch should be set to "1".

(1) Starts Hold Operating Mode

The hold operating mode consists of the level-sensitive release mode and the edge-sensitive release mode. The hold operation is started when the command is set to the command register and holds the following states during the hold operation:

① The oscillator stops and the system's internal operations are all held up.

② The timing generator is cleared to "0".

③ The states of the data memory, registers, and latches valid immediately before the system is put in the hold state are all held.

④ The program counter holds the address of the instruction to be executed after the instruction ([OUT A, %OP10] or [OUT @HL, %OP10]) which starts the hold operating mode.

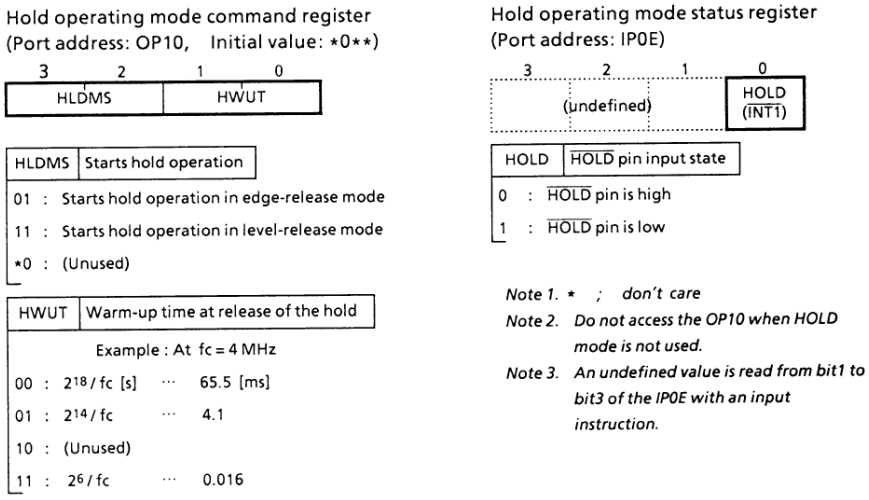


Figure 2-18. Hold Operating Mode Command Register/Status Register

a. Level-sensitive release (back-up) mode

In this mode, the hold operation is released by setting the $\overline{\text{HOLD}}$ pin to the high level. This mode is used for the capacitor backup with power off or for the battery backup for long hours.

If the instruction to start the hold operation is executed with the $\overline{\text{HOLD}}$ pin input being high, the hold operation does not start but the release sequence (warm-up) starts immediately. Therefore, to start the hold operation in the level-sensitive release mode, that the $\overline{\text{HOLD}}$ pin input being low (the hold operation request) must be recognized in program. This recognition is performed in one of the two ways below:

① Testing HOLD (bit 0 of the status register)

② Generating the external interrupt 1 request.

Example: To test HOLD to start the hold operation in the level-sensitive release mode (the warm-up time = 2^{14}/fc).

```
SHOLDH:   TEST  %IPOE, 0     ; Waits until HOLD pin input goes low.
          B     SHOLDH
          LD    A, #1101B    ; OP10←1101B
          OUT   A, %OP10
```

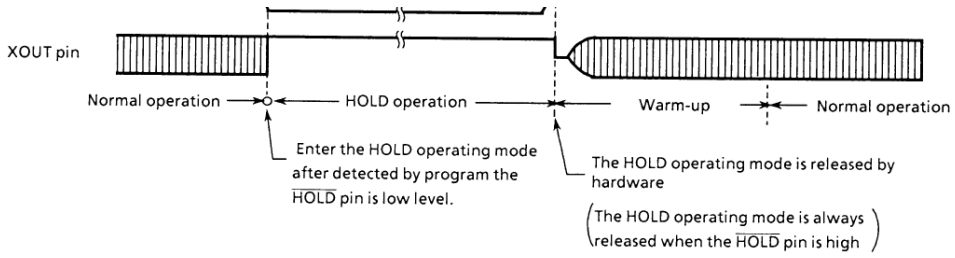


Figure 2-19. Level-Sensitive Release Mode

b. Edge-sensitive release (clock) mode

In this mode, the hold operation is released at the rising edge of the $\overline{HOLD}$ pin input. This mode is used for applications in which a relatively short-time program processing is repeated at a certain cycle. This cyclic signal (for example, the clock supplied from the low power dissipation oscillator). In the edge-sensitive mode, even if the $\overline{HOLD}$ pin input is high, the hold operation is performed.

Example: To start the hold operation in the edge-sensitive release mode (the warm-up time = 2^{14}/fc).

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

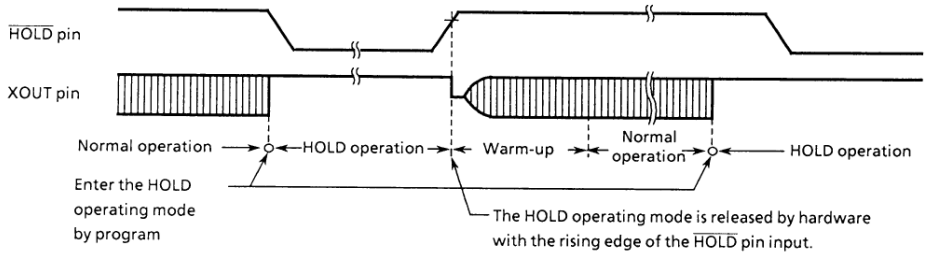


Figure 2-20. Edge-Sensitive Release Mode

Note 1: In the hold operation, the dissipation of the power associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the hold feature. This point should be considered in the system design and the interface circuit design.

Note 2: In the CMOS circuitry, a current does not flow when the input level is stable at the power voltage level (V_{DD}/V_{SS}): however, when the input level gets higher than the power voltage level (by approximately 0.3 to 0.5 V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port (the open drain output pin with an input transistor connected) puts the pin signal into the high-impedance state, a current flows across the ports input transistor, requiring to fix the level by pull-up or other means.

(2) Releases Hold Operating Mode

The hold operating mode is released in the following sequence:

① The oscillator starts

② Warm-up is performed to acquire the time for stabilizing oscillation. During the warm-up, the internal operations are all stopped. One of three warm-up times can be selected by program depending on the characteristics of the oscillator used.

③ When the warm-up time has passed, an ordinary operation restarts from the instruction next to the instruction which starts the hold operation. At this time, the interval timer starts from the reset state "0".

The warm-up time is obtained by dividing the basic clock by the interval timer, so that, if the frequency at releasing the hold operation is unstable, the warm-up time shown in Figure 2-18. includes an error. Therefore, the warm-up time must be handled as an approximate value. The hold operation is also released by setting the $\overline{\text{RESET}}$ pin to the low level. In this case, the normal reset operation follows immediately.

Note: To release the hold operation at a low hold voltage, the following points must be considered:
When the power voltage rises from the hold voltage to the operating voltage, the $\overline{\text{RESET}}$ pin input is also at the high level and its voltage rises with the power voltage.

In this case, if a time-constant circuit or the like is externally attached, the voltage rise of the $\overline{\text{RESET}}$ pin input occurs after the power voltage rise. If the voltage level of the $\overline{\text{RESET}}$ pin input gets under the non-inverted high input voltage of the $\overline{\text{RESET}}$ pin input (the hysteresis input), a reset operation may happen.

2.8 Interrupt Function

2.8.1 Interrupt Controller

There are 5 interrupt sources (2 external and 3 internal). The prioritized multiple interrupt capability is supported. The interrupt latches (IL_5 through IL_0) to hold interrupt requests are provided for the interrupt sources. Each interrupt latch is set to "1" when an interrupt request is made, asking the CPU to accept the interrupt. The acceptance of interrupt can be permitted or prohibited by program through the interrupt enable master flip-flop (EIF) and interrupt enable register (EIR). When two or more interrupts occur simultaneously, the one with the highest priority determined by hardware is serviced first.

Table 2-2. Interrupt Sources

Interrupt Source			Priority	Interrupt Latch	Enable conditions	Entry address
External	Extenal Interrupt 1	(INT1)	(highest) 1	IL_5	EIF = 1	002_H
Internal	TC1 overflow Interrupt	(IOVF1)	2	IL_3	EIF = 1, EIR_2 = 1	006_H
	TC2 overflow Interrupt	(IOVF2)	3	IL_2	EIF = 1, EIR_1 = 1	008_H
	Interval Timer Interrupt	(ITMR)	4	IL_1		$00A_H$
External	External Interrupt 2	(INT2)	(lowest) 5	IL_0	EIF = 1, EIR_0 = 1	$00C_H$

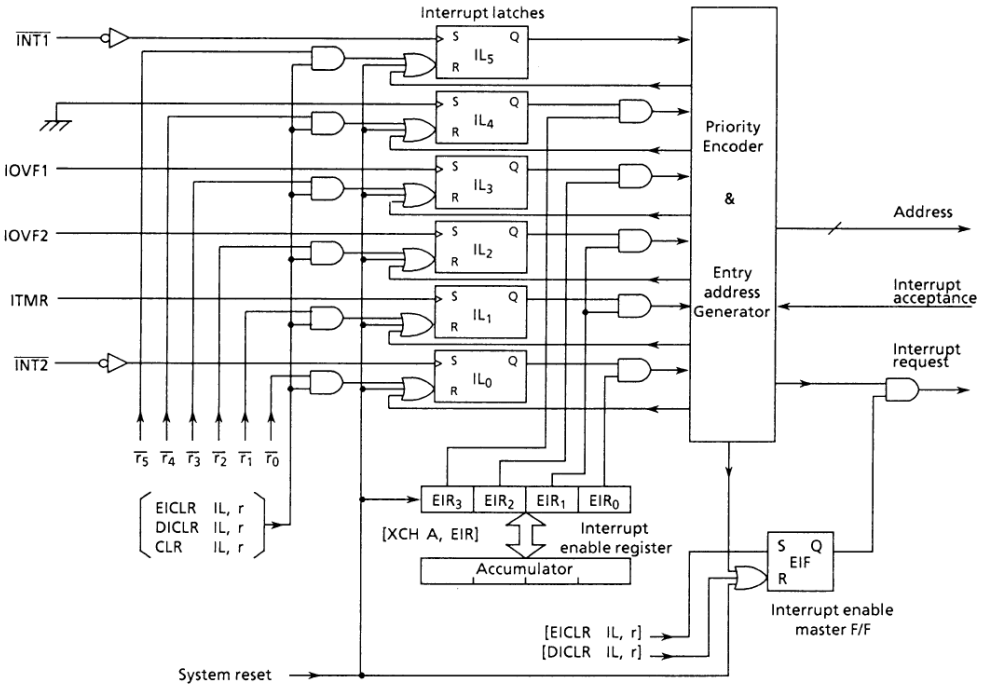


Figure 2-21. Interrupt Controller Block Diagram

(1) Interrupt enable master flip-flop (EIF)

The EIF controls the enable/disable of all interrupts. When this flip-flop is cleared to "0", all interrupts are disabled; when it is set to "1", the interrupts are enabled.

When an interrupt is accepted, the EIF is cleared to "0", temporarily disabling the acceptance of subsequent interrupts.

When the interrupt service program has been executed, the EIF is set to "1" by the execution of the interrupt return instruction [RETI], being put in the enabled state again.

Set or clear of the EIF in program is performed by instruction [EICLR IL, r] and [DICLR IL, R], respectively. The EIF is initialized to "0" during reset.

(2) Interrupt enable register (EIR)

The EIR is a 4-bit register specifies the enable or disable of each interrupt except INT1. An interrupt is enabled when the corresponding bit of the EIR is "1", and an interrupt is disabled when the corresponding bit of the EIR is "0". Bit 1 of the EIR (EIR_1) is shared by both IOVF2 and ITMR interrupts.

Read/write on the EIR is performed by executing [SCH a, EIR] instruction. The EIR initialized to "0" during reset.

(3) Interrupt latch (IL5 through IL0)

An interrupt latch is provided for each interrupt source. The IL is set to "1" when an interrupt request is made to ask the CPU for accepting the interrupt. Each IL is cleared to "0" upon acceptance of the interrupt. It is initialized to "0" during reset.

The ILS can be cleared independently by interrupt latch operation instructions ([EICLR IL, r], [DICLR IL, r], and [CLR IL, r]) to make them cancel interrupt requests or initialize by program. When the value of instruction field (r) is "0", the interrupt latch is cleared; when the value is "1", the IL is held. Note that the ILs cannot be set by instruction.

Example 1: To enable IOVF1, INT1, and INT2 interrupts.

```
LD      A,#0101B      ;  EIR←0101B
XCH     A,EIR
EICLR   IL,111111B    ;  EIF←1
```

Example 2: To set the EIF to "1", and to clear the interrupt latches except ITMR to "0".

```
EICLR   IL,000010B    ;  EIF←1, IL0←0, IL2-IL5←0
```

2.8.2 Interrupt Processing

An interrupt request is held until the interrupt is accepted or the IL is cleared by the reset of the interrupt latch operation instruction.The interrupt acknowledge processing is performed in 2 instruction cycles after the end of the current instruction execution (or after the timer/counter processing if any). The interrupt service program terminates upon execution of the interrupt return instruction [RETI].

The interrupt acknowledge processing consists of the following sequence:

① The contents of the program counter and the flags are saved on the stack.

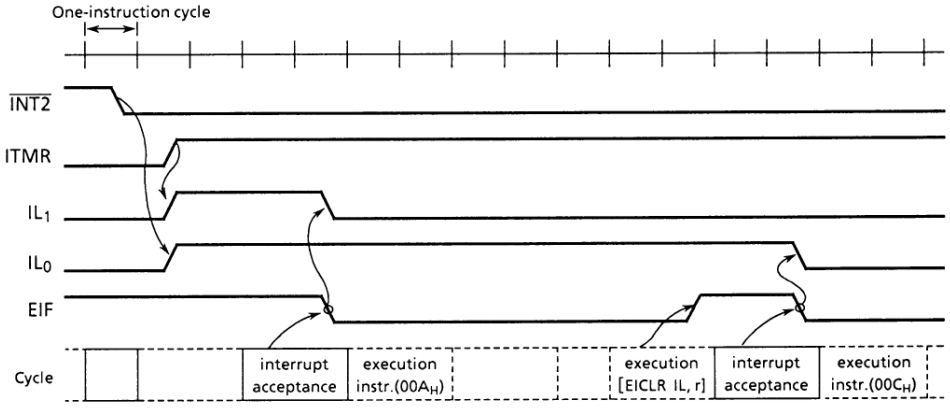
② The interrupt entry address corresponding to the interrupt source is set to the program counter.

③ The status flag is set to "1".

④ The EIF is cleared to "0", temporarily disabling the acceptance of subsequent interrupts.

⑤ The interrupt latch for the accepted interrupt source is cleared to "0".

⑥ The instruction stored at the interrupt entry address is executed. (Generally, in the program memory space at the interrupt entry address, the branch instruction to each interrupt processing program is stored.)



Notes.
1. *It is assumed that there is no other interrupt request and EIR = 0011_B.*
2. *The value r in the [EICLR IL, r] instruction is assumed as 111111_B.*
3. *[ ] denotes the execution of an instruction.*

Figure 2-22. Interrupt Timing Chart (Example)

To perform the multi-interrupt, the EIF is set to "1" in the interrupt service program, and the acceptable interrupt source is selected by the EIR. However, for the INT1 interrupt, the interrupt service is disabled under software control because it is not disabled by the EIR.

Example: The INT1 interrupt service is disabled under software control (Bit 0 of RAM [05_H] are assigned to the disabling switch of interrupt service).

```
PINT1:  TEST    05H,0   ;  Skips if RAM [05H] 0 is "1"
        B       SINT1
        RET1
SINT1:  :
```

The Interrupt return instruction [RETI] performs the following operations:

① Restores the contents of the program counter and the flags from the stack.

② Sets the EIF to "1" to provide the interrupt enable state again.

In the interrupt processing, the program counter and flags are automatically saved or restored but the accumulator and other registers are not. If it is necessary to save or restore them, it must be performed by program as shown in the following example. To perform the multi-interrupt, the saving RAM area never be overlapped.

Example: To 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 2 bits of GSAV1 should be "0's".

2.8.3 External Interrupt

When an external interrupt (INT1 or INT2) occurs, the interrupt latch is set at the falling edge of the corresponding pin input ($\overline{INT1}$ or $\overline{INT2}$). The external interrupt input is the hysteresis type, each of high and low level time requires 2 or more instruction cycles for a correct interrupt operation.

The INT1 interrupt cannot be disabled by the EIR, so that it is always accepted in the interrupt enable state (EIF="1"). Therefore, INT1 is used for an interrupt with high priority such as an emergency interrupt. When $\overline{HOLD}$ ($\overline{INT1}$) pin is used for the I/O port, the INT1 interrupt occurs at the falling edge of the pin input, so that the interrupt return [RET1] instruction must be stored at the interrupt entry address to perform dummy interrupt processing.

2.9 Reset Function

When the $\overline{RESET}$ pin is held to the low level for three or more instruction cycles when the power voltage is within the operating voltage range and the oscillation is stable, reset is performed to initialize the internal states.

When the $\overline{RESET}$ pin input goes high, the reset is cleared and program execution starts from address 000_H. The $\overline{RESET}$ pin is a hysteresis input with a pull-up resistor (220k typ.). Externally attaching a capacitor and a diode implement a simplified power-on-reset operation

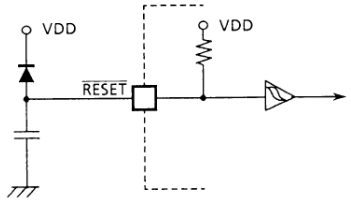


Figure 2-23. Simplified Power-On-Reset

Table 2-3. Initialization of Internal States by Reset Operation

On-chip hardware	Initial value	On-chip hardware	Initial value
Program counter (PC)	000_H	Output latch (I/O ports or Output ports)	Refer to "INPUT / OUTPUT Circuitry".
Status flag (SF)	1		
Interrupt enable master flip-flop (EIF)	0		
Interrupt enable register (EIR)	0_H	Command register	Refer to the description of each relative command register.
Interrupt latch (IL)	"0"		
Interval timer	"0"		

3. Peripheral Hardware Function

3.1 Ports

The data transfer with the external circuit and the command/status/data transfer with the internal circuit are performed by using the I/O instructions (13 kinds). There are 4 types of ports:

① I/O port	;	Data transfer with external circuit
② Command register	;	Control of internal circuit
③ Status register	;	Reading the status signal from internal circuit
④ Data register	;	Data transfer with internal circuit

These ports are assigned with port addresses (00H through IFH). Each port is selected by specifying its port address in an I/O instruction. Table 3-1 lists the port address assignments and the I/O instructions that can access the ports.

3.1.1 I/O Timing

(1) Input timing

External data is read from an input port or an I/O port in the S3 state of the second instruction cycle during the input instruction (2-cycle instruction) execution. This timing cannot be recognized from the outside, so that the transient input such as chattering must be processed by program.

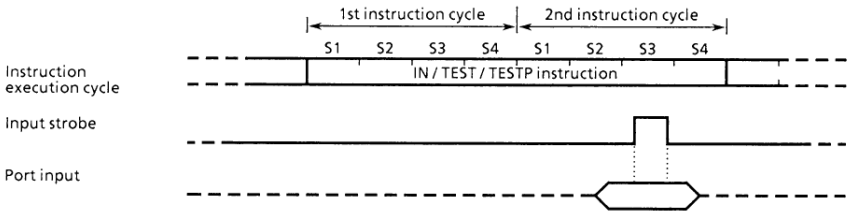


Figure 3-1. Input Timing

(2) Output timing

Data is output to an output port or an I/O port in the S4 state of the second instruction cycle during the output instruction (2-cycle instruction) execution.

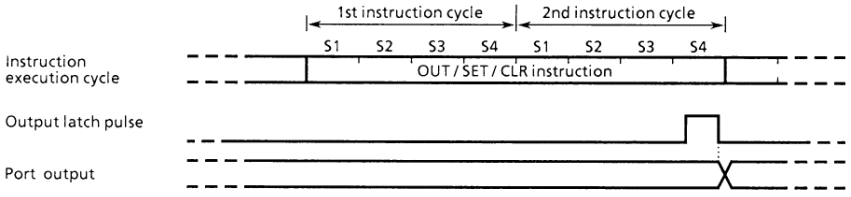


Figure 3-2. Output Timing

3.1.2 I/O Ports

The 47C101/201 have 4 I/O ports (11 pins) each as follows:

① R4, R5	;	4-bit input/output
② R8	;	2-bit input/output (shared with external interrupt input and timer/counter input)
③ KE	;	1-bit sense input (shared with hold request/release signal input)

Each output port contains a latch, which holds the output data. The input ports have no latch; therefore, it is desired to hold data externally until it is read or read twice or more before processing it.

Table 3-1. Port Address Assignments and Available I/O Instructions

Port address (**)	Port		Input / Output instructions						
	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	——	——	–	–	–	–	–	–	–
01	——	——	–	–	–	–	–	–	–
02	——	——	–	–	–	–	–	–	–
03	——	——	–	–	–	–	–	–	–
04	R4 input port	R4 output port	○	○	○	–	○	○	○
05	R5 input port	R5 output port	○	○	○	–	○	○	○
06	——	——	–	–	–	–	–	–	–
07	——	——	–	–	–	–	–	–	–
08	R8 input port	R8 output port	○	○	○	–	○	○	–
09	——	——	–	–	–	–	–	–	–
0A	——	——	–	–	–	–	–	–	–
0B	——	——	–	–	–	–	–	–	–
0C	——	——	–	–	–	–	–	–	–
0D	——	——	–	–	–	–	–	–	–
0E	HOLD status	——	○	–	–	–	–	○	–
0F	——	——	–	–	–	–	–	–	–
10_H	Undefined	Hold operating mode control	–	○	–	–	–	–	–
11	Undefined	——	–	–	–	–	–	–	–
12	Undefined	——	–	–	–	–	–	–	–
13	Undefined	——	–	–	–	–	–	–	–
14	Undefined	——	–	–	–	–	–	–	–
15	Undefined	——	–	–	–	–	–	–	–
16	Undefined	——	–	–	–	–	–	–	–
17	Undefined	——	–	–	–	–	–	–	–
18	Undefined	——	–	–	–	–	–	–	–
19	Undefined	Interval Timer interrupt control	–	○	–	–	–	–	–
1A	Undefined	——	–	–	–	–	–	–	–
1B	Undefined	——	–	–	–	–	–	–	–
1C	Undefined	Timer / Counter 1 control	–	○	–	–	–	–	–
1D	Undefined	Timer / Counter 2 control	–	○	–	–	–	–	–
1E	Undefined	——	–	–	–	–	–	–	–
1F	Undefined	——	–	–	–	–	–	–	–

Note. *"——"means the reserved state. Unavailable for the user programs.*

(1) Ports R4 (R43 to R40), R5 (R53 to 50)

These ports are 4-bit I/O ports with a latch. When used as an input port, the latch must be set to “1”. The latch is initialized to “1” during reset. Port R4 can directly drive LEDs.

These 2 ports (8 pins) can be set, cleared, and tested for each bit as specified by L register indirect addressing bit manipulation instructions ([SET @L], [CLR @L], and [TEST @L]). Table 3-1 lists the pins (I/O ports) that correspond to the contents of L register.

Example: To clear R43 output as specified by the L register indirect addressing bit manipulation instruction.

```
LD    L, #00011B    ;  Sets R43 pin address to L register
CLR   @L            ;  R43 0
```

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

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

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

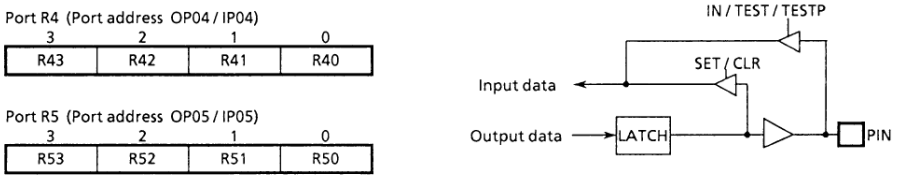


Figure 3-3. Ports R4, R5

(2) Port R8 (R81 to R80) and Port KE

Port R8 is a 2-bit I/O port with a latch.When used as an input port, the latch must be set to “1”. The latch is initialized to “1” during reset.

Port R8 is shared with the external interrupt input pin and the timer/counter input pin. To use this port for one of these functional pins, the latch should be set to “1”. To us it for an ordinary I/O port, the acceptance of external interrupt should be disabled or the event counter/pulse width measurement modes of the timer/counter should be disabled.

R82, R83 pins do not exist actually but R82, R83 has the latch. And R82 is wired to $\overline{\text{HOLD}}$ ($\overline{\text{INT1}}$) pin, internally.

Port KE (KEO) is a 1-bit sense input port shared with the hold request/release signal input in ($\overline{\text{HOLD}}$). This input port is assigned to the least significant bit of port address IPOE and is processed as the data with inverted polarity. For example, if an input instruction is executed with the pin on the high level, ”0” is read. The bit1 to bit3 of port KE, and undefined value is read when an input instruction is executed.

Note: When $\overline{\text{HOLD}}$ ($\overline{\text{INT1}}$) pin is used for an I/O port, external interrupt 1 occurs upon detection of the falling edge of pin input, and if the interrupt enable master flip-flop is enabled, the interrupt request is always accepted. So that a dummy interrupt processing must be performed (only the interrupt return instruction [RETI] is executed).

With R80 ($\overline{\text{INT2}}$) pin, external interrupt 2 occurs like $\overline{\text{HOLD}}$ ($\overline{\text{INT1}}$) in but bit 0 of the interrupt enable register (EIR_0) is only kept at “0”not accepting the interrupt request.

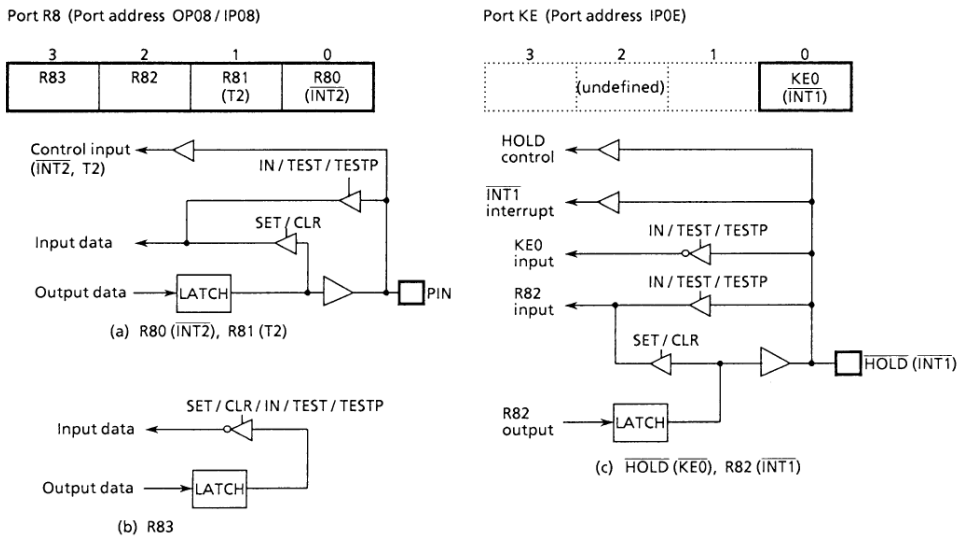


Figure 3-4. Port R8 and $\overline{\text{HOLD}}$ ($\overline{\text{INT1}}$) Pin

3.2 Interval Timer

The interval timer can be used to generate an interrupt with a fixed frequency. For an interval timer interrupt (ITMR), one of 4 frequencies can be selected by command. The command register (OP19) is initialized to "0" during reset. An interval timer interrupt is generated at the first rising edge of the binary counters output after the command has been set. The interval timer is not cleared by command, so that the first interrupt may occur earlier than the preset interrupt period.

Example: To set the interval timer interrupt frequency to $fc/2^{12}$ [Hz].

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

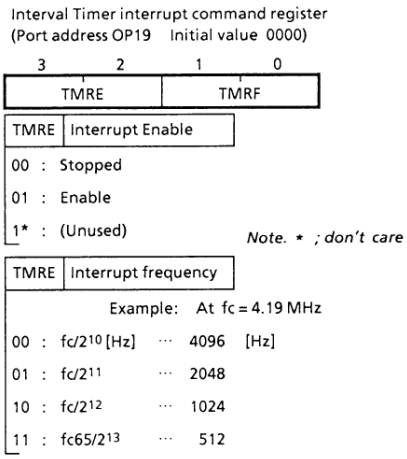


Figure 3-5. Command Register

3.3 Timer/Counters (TC1,TC2)

The 47C101/201 contain two 12-bit timer/counters (TC1, TC2). RAM addresses are assigned to the count register in unit of 4 bits, permitting the initial value setting and counter reading through the RAM manipulation instruction. When the timer/counter is not used, the mode selection may be set to "stopped" to use the RAM at the address corresponding to the timer/counter for storing the ordinary user-processed data.

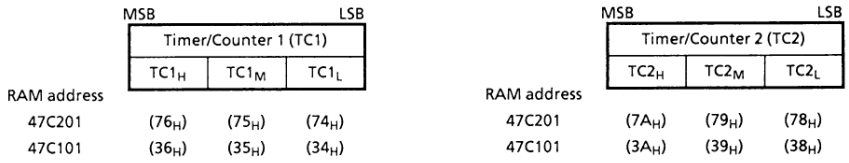


Figure 3-16. Count Registers of the Timer/Counters (TC1, TC2)

3.3.1 Functions of Timer/Counters

The timer/counters provide the following functions:

① Event counter

② Programmable timer

③ Pulse width measurement

3.3.2 Control of Timer/Counters

The timer/counters are controlled by the command registers. The command register is accessed as port address OP1C for TC1 and port address OP1D for TC2. These registers are initialized to "0" during reset.

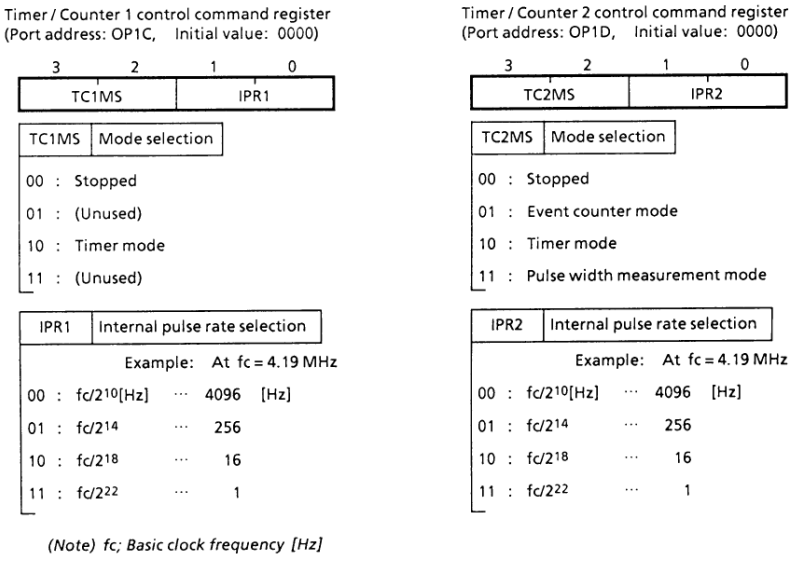


Figure 3-7. Timer/Counter Control Command Registers

The timer/counter increments at the rising edge of each count pulse. Counting starts with the first rising edge of the count pulse generated after the command has been set. Count operation is performed in one instruction cycle after the current instruction execution, during which the execution of a next instruction and the acceptance of an interrupt are delayed. If counting is requested by both TC1 and TC2 simultaneously, the request by TC1 is preferred. The request by TC2 is accepted in the next instruction cycle. Therefore, during count operation, the apparent instruction execution speed drops as counting occurs more frequently.

The timer/counter causes an interrupt upon occurrence of an overflow (a transition of the count value from FFF_H to 000_H). If the timer/counter is in the interrupt enabled state and the overflow interrupt is accepted immediately after its occurrence, the interrupt is processed in the sequence shown in Figure 3-8. Note that counting continues if there is a count request after overflow occurrence.

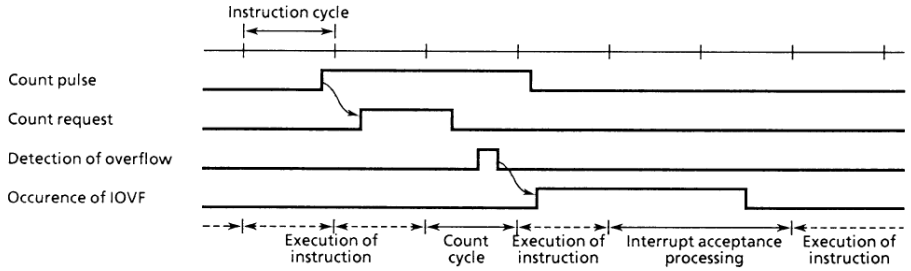


Figure 3-8. Timer/Counter Overflow Interrupt Timing

(1) Event counter mode

In the event counter mode, the timer/counter increments at each rising edge of the external pin (T2) input. The maximum applied frequency of the external pin input is fc/32. The apparent instruction execution speed drops most to (1/3) x 100 = 33% when TC2 is operated at the maximum applied frequency because the count operation is inserted once every 4 instruction cycles. For example, the instruction execution speed of 2μs drops to 2.66μs

Example: To operate TC2 in the event counter mode.

```
LD      A, #0100B    ; OP1D←01**B
OUT     A, %0P1D
```

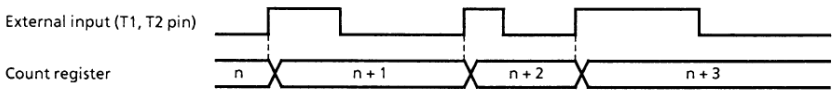


Figure 3-9. Event Counter Timing Chart

(2) Timer mode

In the timer mode, the timer/counter increments at the rising edge of the internal pulse generated from the timing generator. One of 4 internal pulse rates can be selected by the command register. The selected rate can be initially set to the timer/counter to generate an overflow interrupt in order to create a desired time interval.

When an internal pulse rate of $fc/2^{10}$ is used, a count operation is inserted once every 128 instruction cycles, so that the apparent instruction execution speed drops by (1/127) x 100 = 0.8%. For example, the instruction execution speed of 2μs drops to 2.016μs.

Example: To generate an overflow interrupt (at fc = 4MHz) by the TC1 after 100ms.

```
LD      HL, #0F4H    ;  TC1←E79H (setting of the count register)
ST      #9, @HL+
ST      #7, @HL+
ST      #0EH, @HL+
LD      A, #1000B    :  OP1C←1000B
OUT     A, %OP1C
LD      A, #0100B    ;  EIR←0100B (enables interrupt)
A       EIR
EICLR   IL, 110111B  ;  EIF←1, IL3←0
```

Calculating the preset value of the count register

The preset value of the count register is obtained from the following relation

2^{12} - (interrupt setting time) x (internal pulse rate)

For example, to generate an overflow interrupt after 100ms at fc = 4MHz with the internal pulse rate of fc/2^{10}, set the following value to the count register as the preset value:

$2^{12} - (100 \times 10^{-3}) \times (4 \times 106/2^{10}) = 3705 = E79_H$

Table 3-3. Internal Pulse Rate Selection

Internal Pulse Rate	Max. Setting Time	Example: At fc = 4.194304MHz	
		Internal Pulse Rate	Max. Setting Time
fc/2^{10} [Hz]	2^{22}/fc [s]	4096 [Hz]	1 [s]
fc/2^{14}	2^{26}	256	16
fc/2^{18}	2^{30}	16	256
fc/2^{22}	2^{34}	1	4096

(3) Pulse width measurement mode

In the pulse width measurement mode, the timer/counter increments with the pulse obtained by sampling the external pins (T2) by the internal pulse. As shown in Figure 3-11, the timer/counter increments only while the external pin input is high. The maximum applied frequency to the external pin input must be one that is enough for analyzing the count value. Normally, a frequency sufficiently slower than the internal pulse rate setting is applied to the external pin.

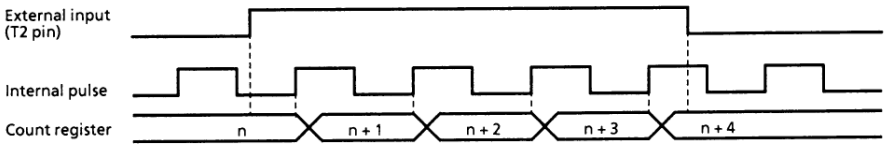
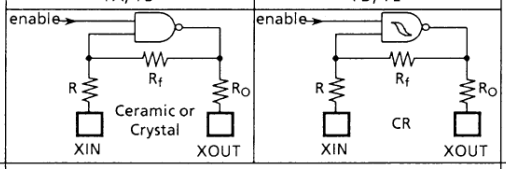
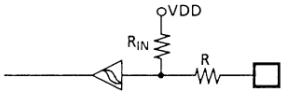
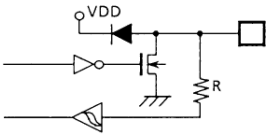


Figure 3-11. Pulse Width Measurement Mode Timing Chart

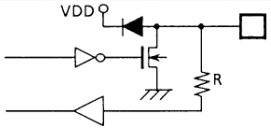
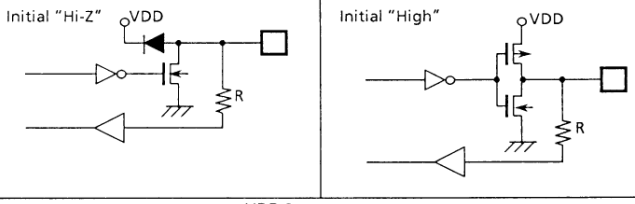
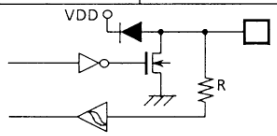
Input/Output Circuitry

The input/output circuitries of the 47C101/201 control pins are shown below, any one of the circuitries can be chosen by a code (FA, FB, FD or FE) as a mask option.

(1) Control pins

CONTROL PIN	I/O	CIRCUITRY and CODE	REMARKS
XIN XOUT	Input Output	FA, FB / FD, FE 	Resonator connecting pins R = 1 kΩ (typ.) Rf = 1.5 MΩ (typ.) RO = 2 kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor RIN = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{HOLD}}$ ($\overline{\text{INT1}}$)	Input (I/O)		Sink open drain output Initial "Hi-Z" Hysteresis input R = 1 kΩ (typ.)

(2) I/O Ports

PORT	I/O	INPUT / OUTPUT CIRCUITRY and CODE	REMARKS
R4	I/O		Sink open drain output Initial "Hi-Z" R = 1 kΩ (typ.)
R5	I/O	FA, FD / FB, FE 	Sink open drain or push-pull output R = 1 kΩ (typ.)
R80 R81	I/O		Sink open drain output Initial "Hi-Z" Hysteresis input R = 1 kΩ (typ.)

Electrical Characteristics

Absolute Maximum Ratings (V_{SS} = 0V)

Parameter	Symbol	Pins		Rating	Unit
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}			-0.3 to V_{DD} + 0.3	V
Output Current (Per 1 pin)	I_{OUT1}	Port R4		30	mA
	I_{OUT2}	Ports R5, R8, $\overline{HOLD}$		3.2	
Output Current (Total)	ΣI_{OUT1}	Port R4		120	mA
Power Dissipation [T_{opr} = 70°C]	PD		DIP	600	mW
			SOP	600	
Soldering Temperature (time)	T_{sld}			260 (10s)	°C
Storage Temperature	T_{stg}			-55 to 125	°C
Operating Temperature	T_{opr}			-30 to 70	°C

Recommended Operating Conditions (V_{SS} = 0V, T_{opr} = -430 to 70°C)

Parameter	Symbol	Pins		Conditions		Min.	Max.	Unit
Supply Voltage	V_{DD}		Normal mode	Crystar or ceramic	fc = 6.0MHz	4.5	5.5	V
					fc = 4.2MHz	2.7		
				RC	fc = 2.5MHz	2.2		
			HOLD mode	–	–	2.0		
Input High Voltage	V_{IH1}	Except Hysteresis Input		In the normal operating area		V_{DD} x 0.7	V_{DD}	V
	V_{IH2}	Hysteresis Input				V_{DD} x 0.75		
	V_{IH3}			In the HOLD mode		V_{DD} x 0.9		
Input Low Voltage	V_{IL1}	Except Hysteresis Input		In the normal operating area		0	V_{DD} x 0.3	V
	V_{IL2}	Hysteresis Input					V_{DD} x 0.25	
	V_{IL3}			In the HOLD mode			V_{DD} x 0.1	
Clock Frequency	fc	XIN, XOUT		V_{DD} = 4.5 to 5.5V		0.4	6.0	MHz
				V_{DD} = 2.7 to 5.5V			4.2	
				V_{DD} = 2.2 to 5.5V (RC)			2.5	

DC Characteristics (V_{SS} = 0V, T_{opr} = -30 to 70°C)

Parameter	Symbol	Pins	Conditions	Min.	typ.	Max.	Unit
Hysteresis Voltage	V_{HS}	Hysteresis Input		–	0.7	–	V
Input Current	I_{IN1}	$\overline{RESET}$, $\overline{HOLD}$	V_{DD} = 5.5V, V_{IN} = 5.5V/0V	–	–	±2	μA
	I_{IN2}	Open drain output ports					
Input Resistance	R_{IN}	$\overline{RESET}$		100	220	450	kΩ
Input Low Current	I_{IL}	Push-pull output ports	V_{DD} = 5.5V, V_{IN} = 0.4V	–	–	-2	mA
Output Leakage Current	I_{LO}	Open drain output ports	V_{DD} = 5.5V, V_{OUT} = 5.5V	–	–	2	μA
Output High Voltage	V_{OH}	Push-pull output ports	V_{DD} = 4.5V, I_{OH} = -200μA	2.4	–	–	V
			V_{DD} = 2.2V, I_{OH} = -5μA	2.0	–	–	
Output Low Voltage	V_{OL}	Except XOUT and port R4	V_{DD} = 4.5, I_{OL} = 1.6mA	–	–	0.4	V
			V_{DD} = 2.2V, I_{OL} = 20μA	–	–	0.1	
Output Low Current	I_{OL1}	Port R4	V_{DD} = 4.5V, V_{OL} = 1.0V	–	20	–	mA
Supply Current (in the Normal operating mode)	I_{DD}		V_{DD} = 5.5V, fc = 4MHz	–	2	4	mA
			V_{DD} = 5.5V, fc = 4MHz	–	1	2	
			V_{DD} = 3.0V, fc = 400kHz	–	0.5	–1	
Supply Current (in the HOLD operating mode)	I_{DDH}		V_{DD} = 5.5V	–	0.5	10	μA

Note 1. Typ. values show those at T_{opr} = 25 °C, V_{DD} = 5V.
Note 2. Input Current I_{IN1}; The current through resistor is not included.
Note 3. Supply Current V_{IN} = 5.3 V/0.2V (V_{DD} = 5.5V) or 2.8V/0.2V (V_{DD} = 3.0V)

AC Characteristics (V_{SS} = 0V, T_{opr} = -30 to 70°C)

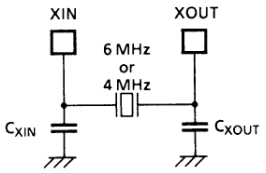
Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Instruction Cycle Time	t_{cy}		V_{DD} = 4.5 to 5.5V	1.3	–	20	μs
			V_{DD} = 2.7 to 5.5V	1.9			
			V_{DD} = 2.2 to 5.5V	3.2			
High Level Clock Pulse Width	t_{WCH}	For external clock operation	$V_{DD} \geq$ 2.7V	80	–	–	ns
			V_{DD} < 2.7V	160			
Low Level Clock Pulse Width	t_{WCL}		$V_{DD} \geq$ 2.7V	80			
			V_{DD} < 2.7V	160			

Recommended Oscillating Conditions (V_{SS} = 0V, V_{DD} = 2.7 to 5.5V, T_{opr} = -30 to 70°C)

(1) 6MHz

Ceramic Resonator

CSA6.00MGU (MURATA)	$C_{XIN} = C_{XOUT}$ = 30 pF
KBR-6.00MS (KYOCERA)	$C_{XIN} = C_{XOUT}$ = 30 pF
EFOEC6004A4 (NATIONAL)	$C_{XIN} = C_{XOUT}$ = 30 pF



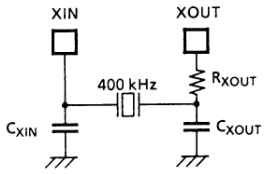
(2) 4MHz

Ceramic Resonator

CSA4.00MG (MURATA)	$C_{XIN} = C_{XOUT}$ = 30 pF
KBR-4.00MS (KYOCERA)	$C_{XIN} = C_{XOUT}$ = 30 pF
EFOEC4004A4 (NATIONAL)	$C_{XIN} = C_{XOUT}$ = 30 pF

Crystal Oscillator

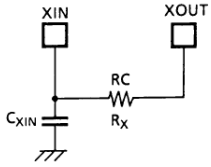
204B-6F 4.0000 (TOYOCOM)	$C_{XIN} = C_{XOUT}$ = 20pF



(3) 400kHz

Ceramic Resonator

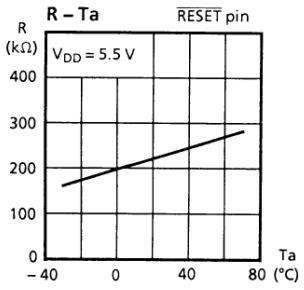
CSB400B (MURATA)	$C_{XIN} = C_{XOUT}$ = 220pF, R_{XOUT} = 6.8kΩ
KBR-400B (KYOCERA)	CXIN = CXOUT = 100pF, R_{XOUT} = 10kΩ
EFOA400K04B (NATIONAL)	$C_{XIN} = C_{XOUT}$ = 470pF, R_{XOUT} = 0Ω

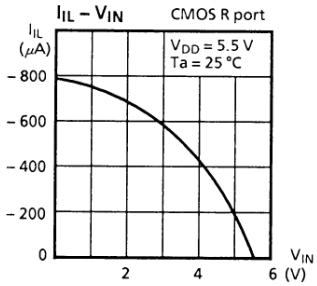


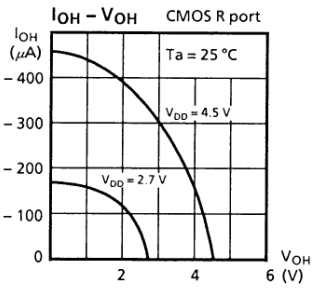
(4) RC Oscillation (V_{SS} = 0V, V_{DD} = 5.0V, Topr = 25°C)

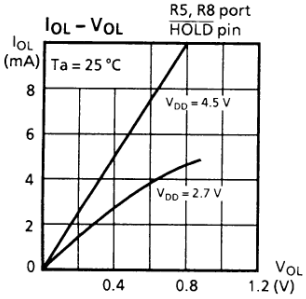
2MHz (Typ.)	C_{XIN} = 33pF, R_X = 10kΩ
400kHz (Typ.)	C_{XIN} = 100pF, R_X = 30kΩ

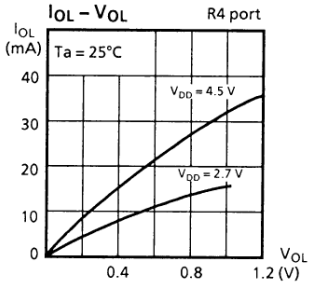
Typical Characteristics

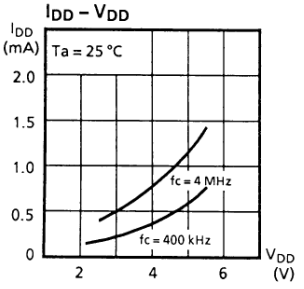


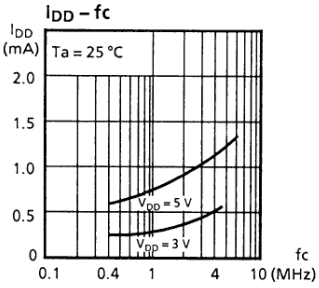


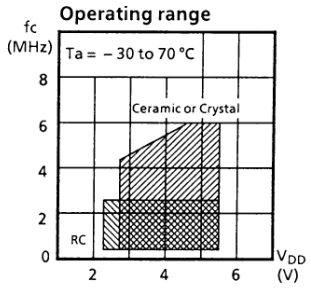


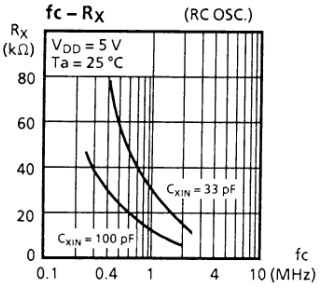


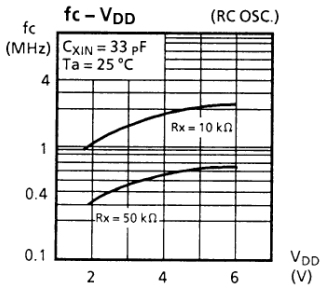












Notes