CMOS 4-BIT MICROCONTROLLER

TMP47C222N, TMP47C422N
TMP47C222F, TMP47C422F

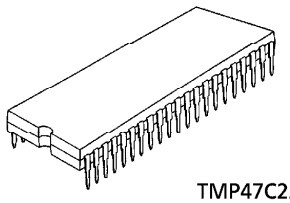
The 47C222/422 are high speed and high performance 4-bit single chip micro computers, integrating A/D converter, pulse output, zero-cross detector and LCD driver based on the TLCS-470 series.

PART No.	ROM	RAM	RACKAGE	OTP
TMP47C222N	2048 × 8-bit	192 × 4-bit	SDIP42-P-600-1.78	TMP47P422VN
TMP47C222F			QFP44-P-1414-0.80D	TMP47P422VF
TMP47C422N	4096 × 8-bit	256 × 4-bit	SDIP42-P-600-1.78	TMP47P422VN
TMP47C422F			QFP44-P-1414-0.80D	TMP47P422VF

FEATURES

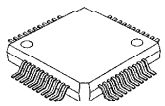
- ◆4-bit single chip microcomputer
- ◆Instruction execution time: 1.0 μs (at 8 MHz)
- ◆Low voltage operation : 2.2V (at 4.2 MHz)
- ◆92 basic instructions
  - Table look-up instructions
- ◆Subroutine nesting: 15 levels max.
- ◆6 interrupt sources (External: 2, Internal: 4)
  All sources have independent latches each, and multiple interrupt control is available.
- ◆I/O port (SDIP : 20 pins, QFP : 22 pins)
- ◆Interval Timer
- ◆Two 12-bit Timer/Counters
  Timer, event counter,and pulse width measurement mode
- ◆Watchdog Timer
- ◆Serial Interface with 8-bit buffer
  - Simultaneous transmission and reception capability
  - 8/4-bit transfer, external/internal clock, and leading/trailing edge shift mode
- ◆8-bit successive approximate type A/D converter
  - With sample and hold
  - 4 analog inputs
  - Conversion time : 24 μs (at 8 MHz)
- ◆Pulse output
  Buzzer drive/Remocon carrier
- ◆Zero-cross detector
- ◆LCD driver
  - LCD direct drive capability (max. 10-digit display at 1/4 duty LCD)
  - 1/4, 1/3, 1/2 duties or static drive are programmably selectable.
- ◆Dual-clock operation
  High-speed / Low-power-consumption operating mode
- ◆Hold function
  Battery/Capacitor back-up
- ◆Emulation pod : BM47C422

SDIP42-P-600-1.78



TMP47C222N
TMP47C422N
TMP47P422VN

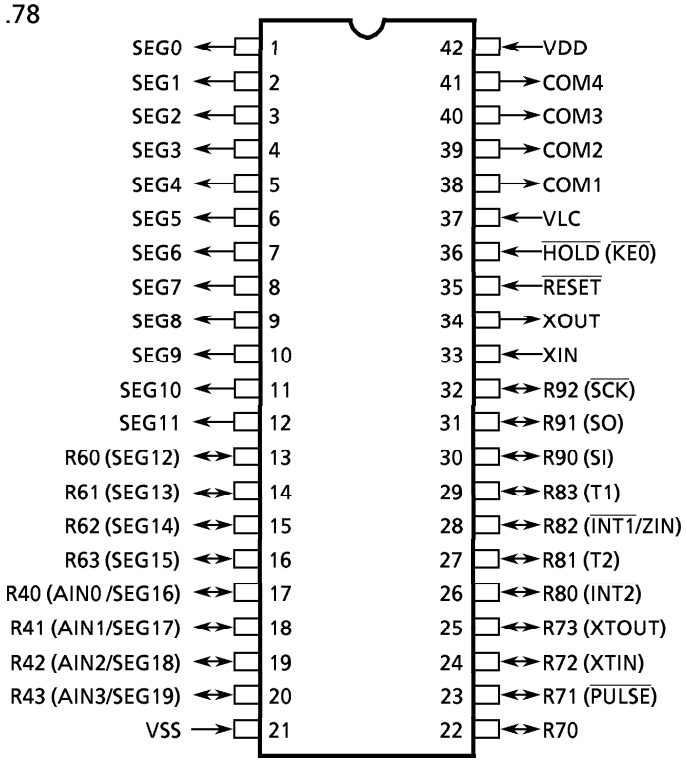
QFP44-P-1414-0.80D



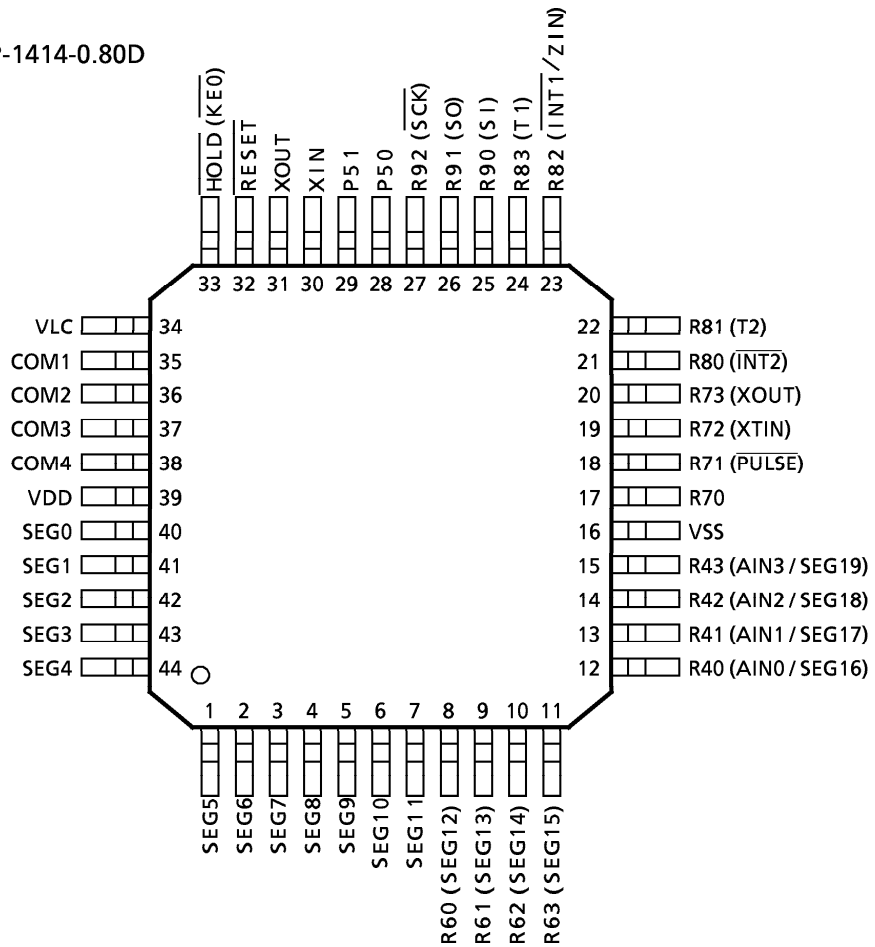
TMP47C222F
TMP47C422F
TMP47P422VF

PIN ASSIGNMENTS (TOP VIEW)

SDIP42-P-600-1.78



QFP44-P-1414-0.80D



PIN FUNCTION

PIN NAME	Input/Output	FUNCTION	
R43 (AIN3/SEG19) to R40 (AIN0/SEG16)	I/O (I /O)	4-bit I/O ports with latch (P5 port has only 2-bit). These ports can be set, cleared and tested for each bit as specified by L-register indirect addressing bit manipulation instruction.	A/D converter analog input / LCD segment drive output
P51, P50	Output		(Note)
R63 (SEG15) to R60 (SEG12)	I/O (Output)		LCD segment drive output
R73 (XTOUT)	I/O (Output)		Resonator connecting pins (Low-frequency).
R72 (XTIN)	I/O (Input)		
R71 ($\overline{\text{PULSE}}$)	I/O (Output)		Pulse output
R70	I/O		
R83 (T1)	I/O (Input)	4-bit I/O ports with latch When used as input port, external interrupt input pin, or timer/counter external input pin, the latch must be set to "1".	Timer / Counter1 external input
R82 ($\overline{\text{INT1}}$/ZIN)			External interrupt1 and zero-cross input
R81 (T2)			Timer / Counter2 external input
R80 ($\overline{\text{INT2}}$)			External interrupt2 input
R92 ($\overline{\text{SCK}}$)	I/O (I/O)	3-bit I/O ports with latch When used as input port or serial port, the latch must be set to "1".	Serial clock I/O
R91 (SO)	I/O (Output)		Serial data output
R90 (SI)	I/O (Input)		Serial data input
SEG11 to SEG0	Output	LCD segment drive output	
COM4 to COM1		LCD Common drive output	
XIN	Input	Resonator connecting pins (High-frequency). For inputting external clock, XIN is used and XOUT is opened	
XOUT	Output		
$\overline{\text{RESET}}$	Input	Reset signal input	
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	I/O (Input)	HOLD request/release signal input	sense input
VDD (VAREF)	Power Supply	+ 5 V	A/D converter analog reference voltage
VSS (VASS)		0 V (GND)	A/D converter analog reference voltage (GND)
VLC		LCD drive power supply	

Note. 47C222/422N (SDIP) do not have port P5.

OPERATIONAL DESCRIPTION

Concerning the 47C222/422 the configuration and functions of hardware are described.
The basic instruction of configuration in the 47C222/422 is the same as those of TLCS-470 series.

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)
  - a. Stack,
  - b. Stack Pointer Word (SPW),
  - c. Data Counter (DC)
- 2.5 ALU, Accumulator
- 2.6 Flags
- 2.7 System Clock Controller
- 2.8 Interrupt Controller
- 2.9 Reset Controller
- Watchdog Timer

◆PERIPHERAL HARDWARE FUNCTION

- 3.1 I/O Ports
- 3.2 Interval Timer
- 3.3 Timer/Counters (TC1, TC2)
- 3.4 Pulse output
- 3.5 Zero-cross detector
- 3.6 A/D converter
- 3.7 Serial Interface
- 3.8 LCD Driver

2. INTERNAL CPU FUNCTION

2.1 Program Counter (PC)

The program counter is a 12-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. In the 47C222/422, the long franch instruction [BSL a] should not be used.

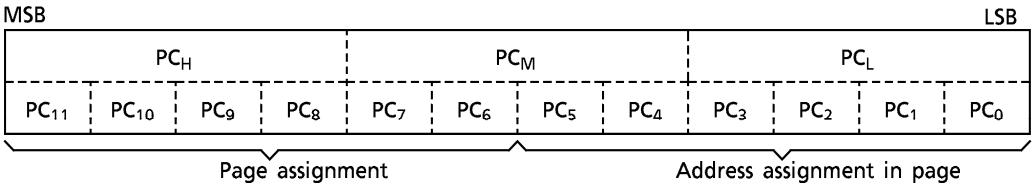


Figure 2-1. Configuration of Program Counter

The PC can directly address a 4096-byte address space. However, with the short branch and subroutine call instructions, the following points must be considered In the 47C222/422, the long branch instruction [BSL a] should not be used.

(1) 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 6 bits of the PC point the next page, so that branch is made to the next page.

(2) Subroutine call instruction [CALL a]
In [CALL a] instruction execution, the contents of the PC are saved to the stack then the value specified in the instruction is set to the PC. The address which can be specified by the instruction consists of 11 bits and the most significant bit of the PC is always "0". Therefore, the entry address of the subroutine should be within an addrress range of 000_H through $7FF_H$.

Table 2-1. Status Change of Program Counter

Instruction or Operation		Condition		Program Counter (PC)											
				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	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			0	Immediate data specified by the instruction										
	CALLS a			0	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	0	Interrupt vector			0
Reset				0	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.

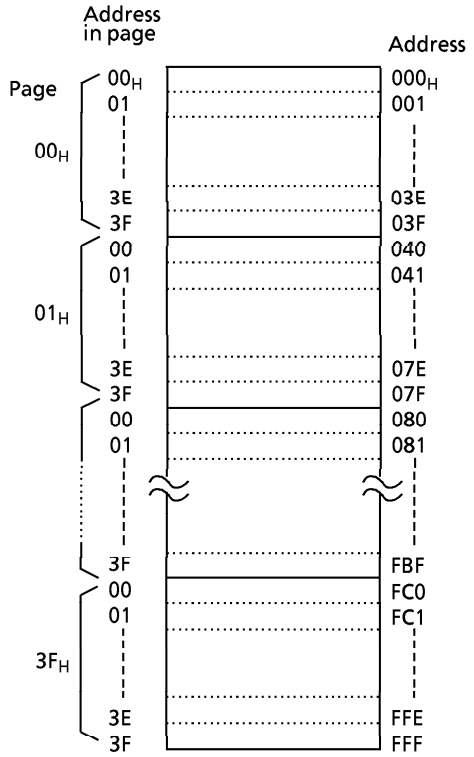


Figure 2-2. Configuration of Program Memory

2.2.1 Program Memory Capacity

Figure 2-3 shows the program memory map. Address 000_H to 086_H of the program memory are also used for special purposes.

2.2.2 Program Memory Map

The 47C222 has 2048 × 8 bits (addresses 000_H to $7FF_H$) of program memory (mask ROM), the 47C422 has 4096 × 8 bits (addresses 000_H to FFF_H).

On the 47C222, no physical program memory exists in the address range 800_H to FFF_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 000_H to $7FF_H$ are read.

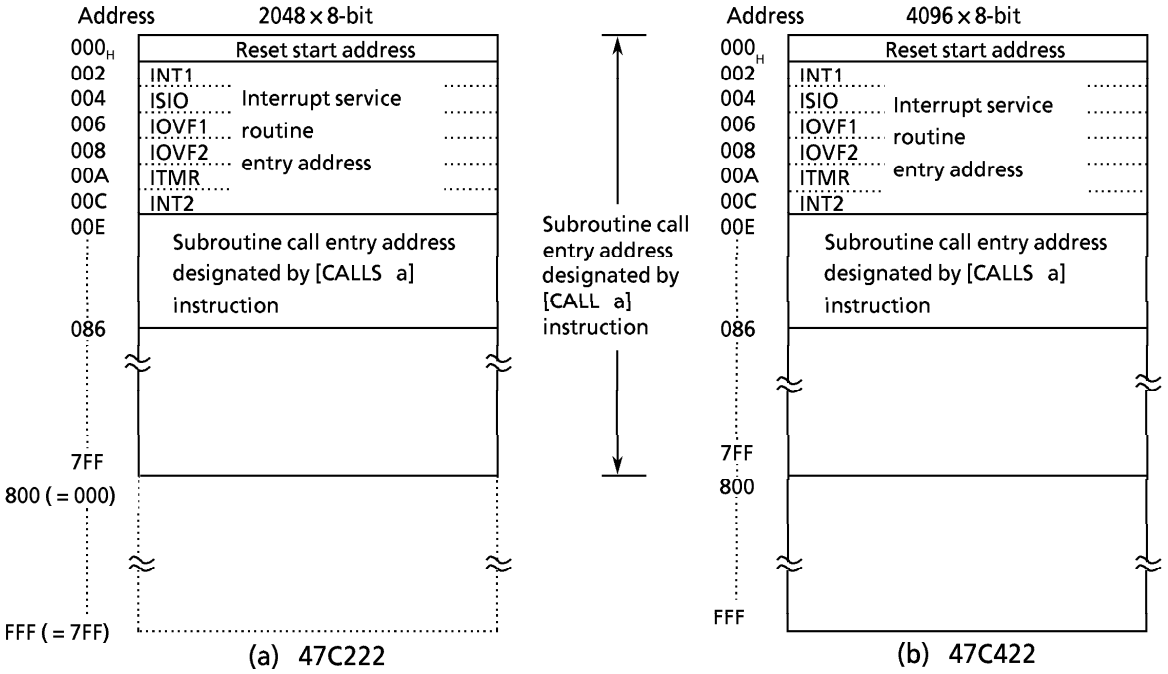


Figure 2-3. Program Memory Map

2.3 H Register and L Register

The H register and the 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 (1word = 4bits). 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).

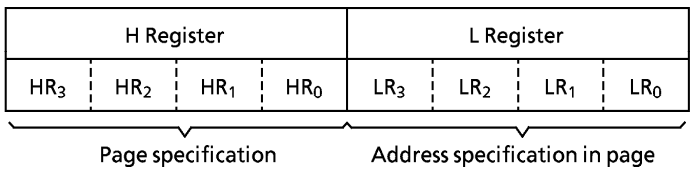


Figure 2-4. Configuration of H and L Registers

Example 1: To write immediate values "5" and "F" to data memory addresses 18_H and 11_H.

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

Example 2: The output latch of R71 pin set "1" by the L register.

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

2.4 Data Memory (RAM)

The data memory stores user-processed data. One page of this memory is 16 words long (1 word = 4 bits). It has 16 pages.

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

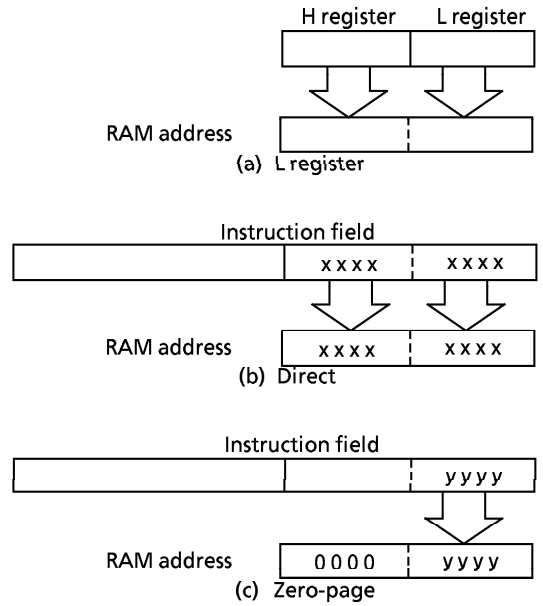
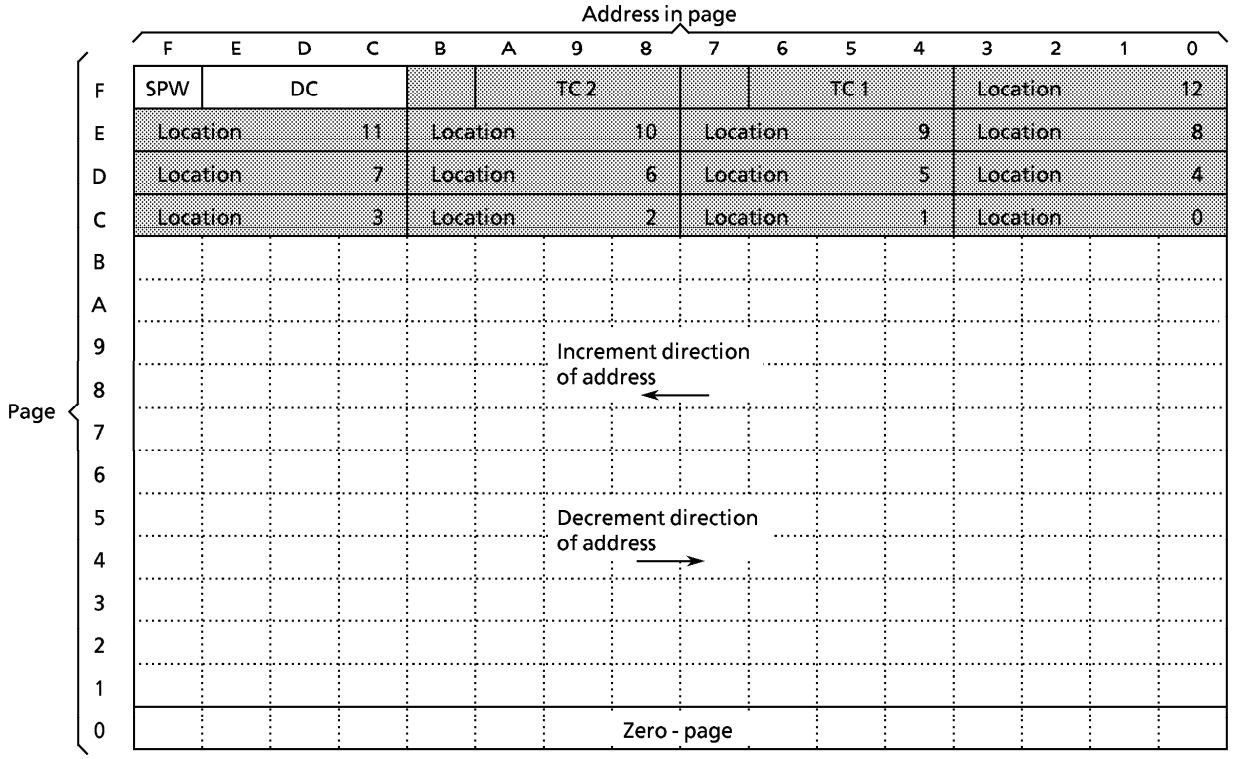


Figure 2-5. Addressing mode

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

(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 $C0_H$ through FB_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 FF_H in the data memory is called the stack pointer word, which identifies the location in the stack to be accessed (save or restore).

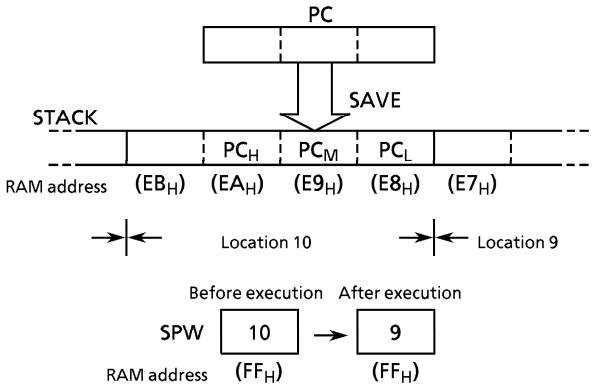
Generally, location number 0 to 12 can be set to the SPW, providing up to 13 levels of stack nesting. Locations 13 and 14 are shared with the timer/counters to be described later; therefore, when the timer/counters are not used, the stack area of up to 15 levels is available. Address FF_H is assigned 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 CF_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 CC_H through CF_H corresponding to the location 3 area is lost.)

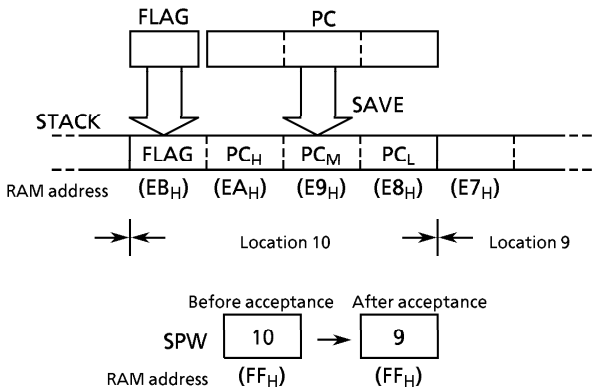
The SPW is not initialized by hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of "12" is used.

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

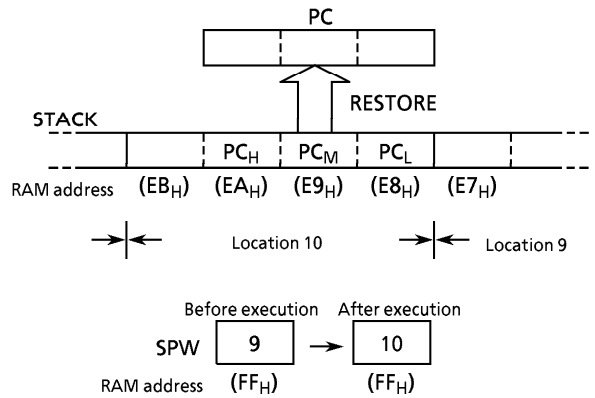
```
LD      A,#12     ;  SPW←12
ST      A,0FFH
```



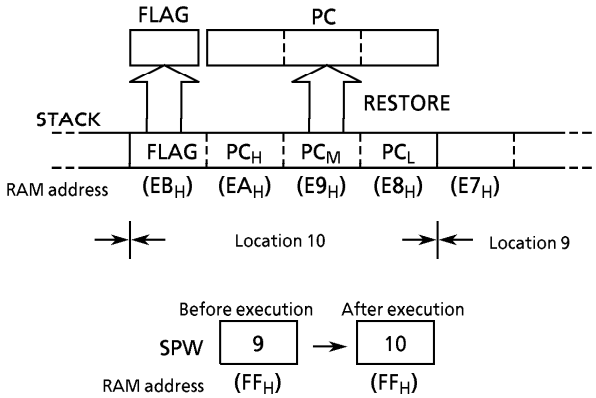
(a) At execution of the subroutine call instruction



(b) At acceptance of an interrupt



(c) At execution of the subroutine return instruction



(d) At execution of the interrupt return instruction

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

(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.

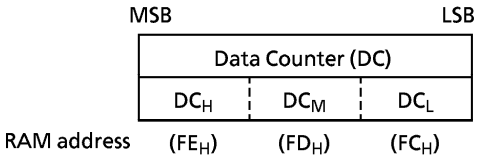


Figure 2-8. Data Counter

Example: To set the DC to 780_H.

```
LD      HL,#0FCH    ;  Sets RAM address of DCL to HL register pair.
ST      #0H,@HL+    ;  DC ← 780H
ST      #8H,@HL+
ST      #7H,@HL+
```

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

The 47C222/422 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 $F7_H$ and FB_H can be used to store the user-processed data.

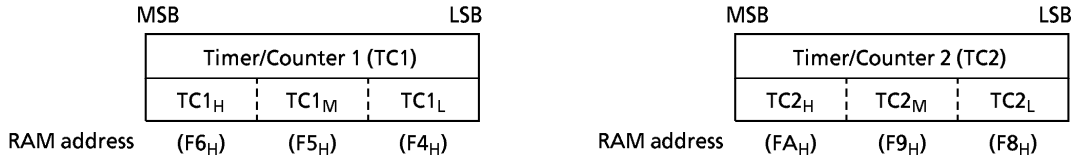


Figure 2-9. Count Registers of the Timer/Counters (TC1, TC2)

(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

The 47C422 has 256 × 4 bits (addresses 00_H to FF_H) of the data memory (RAM), and the 47C222 has 192 × 4 bits (addresses 00_H to $7F_H$, and $C0_H$ through FF_H).

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 47C222 and 47C422)

```
        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
```

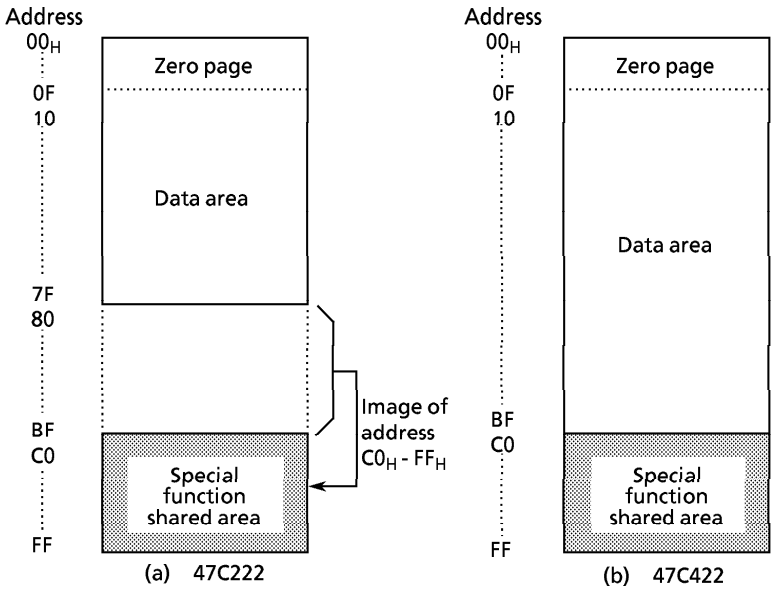


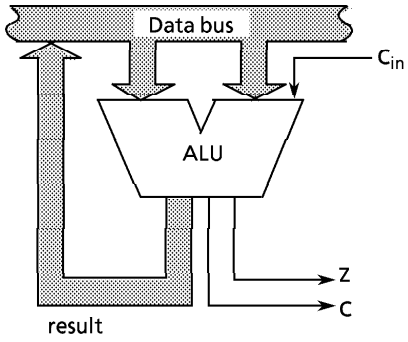
Figure 2-10. Data Memory Capacity and Address Assignment

Note: In the 47C222, the zero-page and thespecial function shared area (stack location 3 to 0) are overlapped. At programming, note that addresses 10 to 3FH are assigned to address 50 to 1FH in the 47C222. The technical data sheets for the 47P4222V shall also be referred to.

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).



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

Figure 2-11. ALU

(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".

Example: The carry information (C) and zero detect information (Z) for 4-bit additions and subtractions.

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

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.

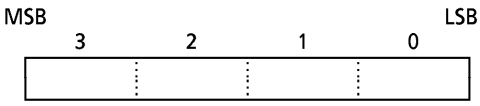


Figure 2-12. Accumulator

2.6 Flags

There are a carry flag (CF), a zero flag (ZF) and a status flag (SF), 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.

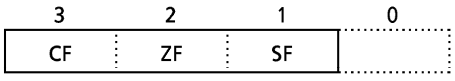


Figure 2-13. Flags

(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 instrcution, 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.

Example : When the following instructions are executed with the accumulator, H register, L register, data memory (address 07_H), 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 Clock Controller

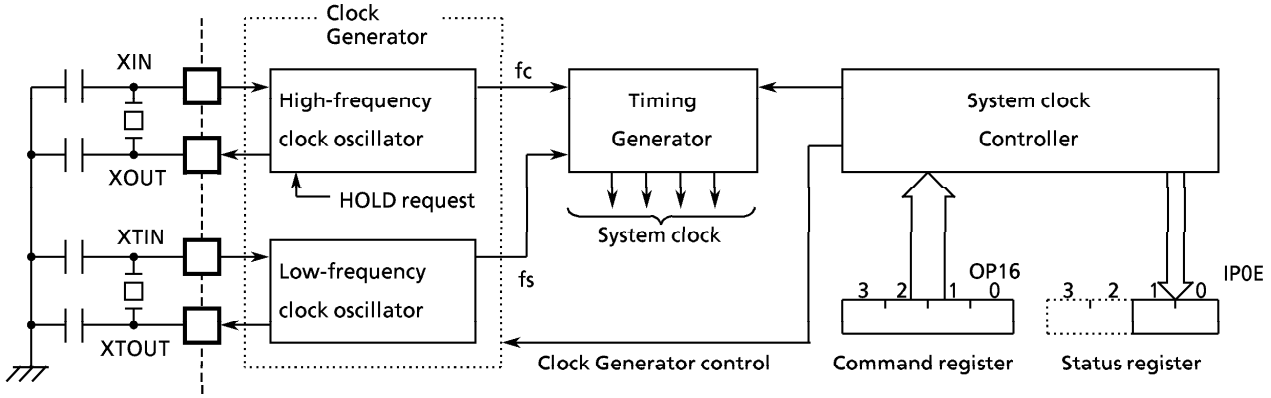


Figure 2-14. Clock Generator, Timing Generator and System Clock Controller

2.7.1 Clock Generator

The clock generator produces the basic clock pulses which provide the system clock to be supplied to the CPU and peripheral hardware. It contains two oscillators: a high-frequency clock oscillator and a low-frequency clock oscillator. Power consumption can be reduced by switching to the low power operation based on the low-frequency clock by the system clock controller. The high-frequency clock and the low-frequency clock can be easily obtained by attaching a resonator between the XIN and XOUT pins and the XTIN and XTOUT pins, respectively (At the high-frequency clock, RC oscillation is also posible, depending on the mask option). The system clock can also be obtained from the external oscillator.

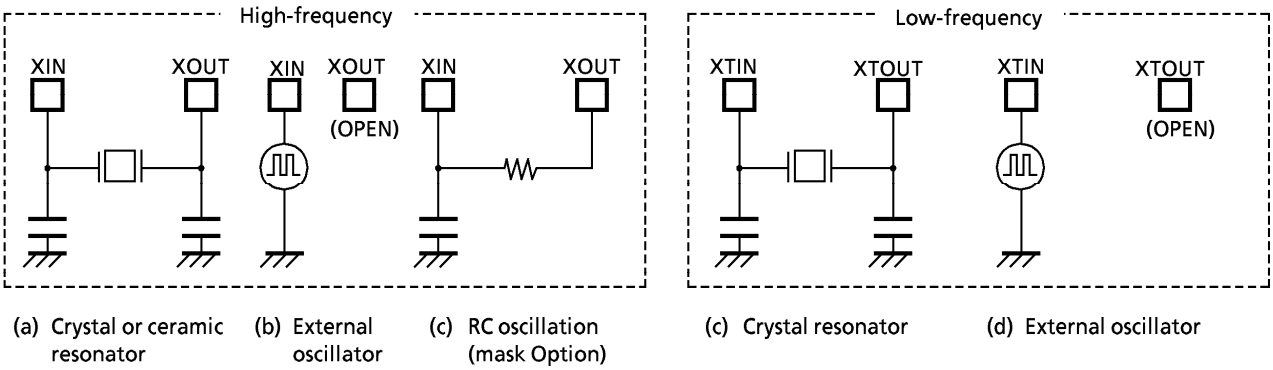
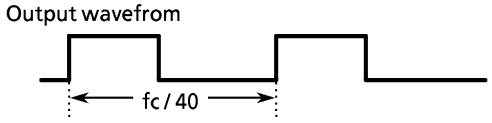


Figure 2-15. Examples of Resonator Connection

Note. Accurate adjustment of the oscillation frequency
Although no hardware to externally and directly monitor the clock pulse 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 abjust the High-frequency oscillation frequency output monitor pulse to R70 pin. (fc = 40 [Hz])

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



2.7.2 Timing Generator (TG)

The timing generator produces the system clocks from clock pulse which are supplied to the CPU and peripheral hardweare.
The timing generater consists of a 19-stage binary counter with a divide-by-3 prescaler. The source clock to the timing generater and its input stage depend on the operating mode as shown below.
During reset, the binary counter is cleared to "0". However, the prescaler is not cleared.

a. Single-clock mode
   ① Normal-1 operating mode
   The CPU and the peripheral hardware are operated on the high-frequency clock. At reset release, this mode is set. In this mode, it is necessary to clear SLCK (bit 2 of command register OP16) to "0".
   ② HOLD operating mode
   In this mode, the system operations are all stopped, holding the internal states valid immediately before the stop at the low power consumption level.

b. Dual-clock mode

① Normal-2 operating mode
In this mode, the CPU is operated on the high-frequency clock but many peripheral hardware operate on the low-frequency clock.

② SLOW operating mode
In this mode, the high-frequency clock oscillation is stopped to operate the CPU and the peripheral hardware on the low-frequency clock, thereby reducing power consumption.

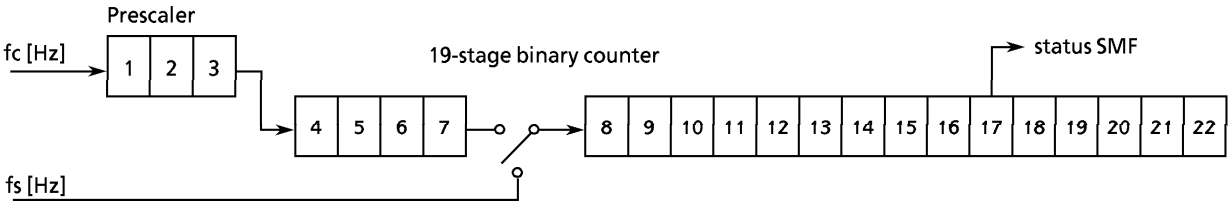


Figure 2-16. Configuration of Timing Generater

(2) The timing generater provides the following functions:

① Instruction cycle
② Internal pulse for interval timer
③ Internal pulse for timer/counters
④ Internal serial clock for a serial interface
⑤ Warm-up time at release of the hold operation
⑥ Source clock for a watchdog timer

2.7.3 System Clock Controller

The system clock controller starts or stops the high-frequency and low-frequency clock oscillator and switches between the basic clocks. The operating mode is generally divided into the single-clock mode and the dual-clock mode, which are controlled by command.

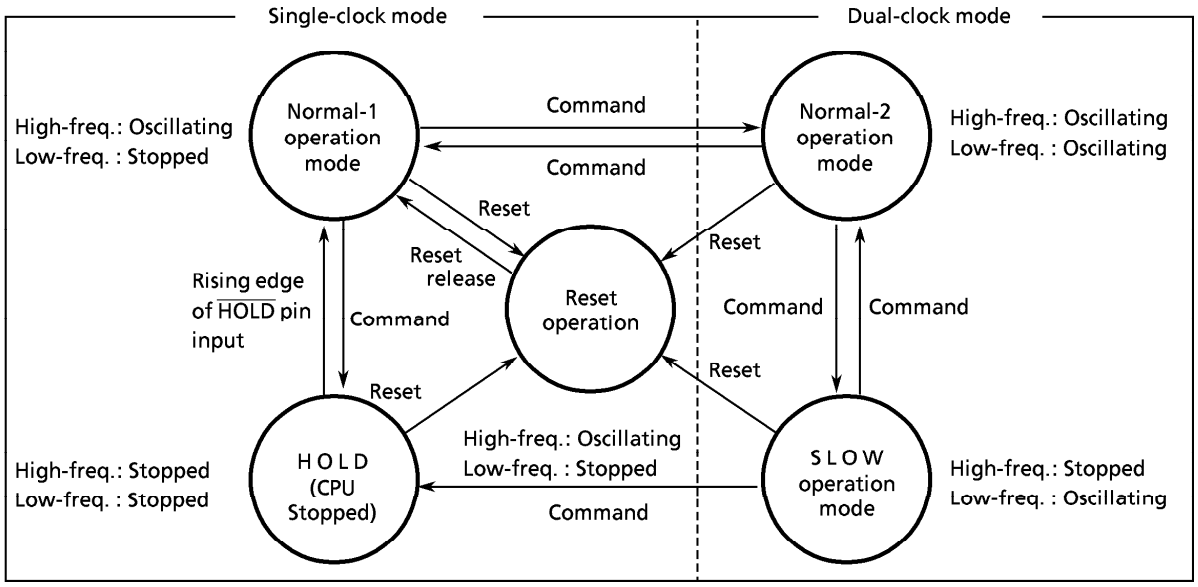


Figure 2-17. Operating Mode Transition Diagram

Note. Normal-1 and Normal-2 operating modes are sometimes referred to as the Normal operating mode collectively.

(1) System clock control

System clock control is performed by the command register (OP16). During reset, this register is initialized to "0" and the single-clock mode is selected.

Each state at operating mode switching can be read from the status register (IP0E).

System clock control command register
(Port address OP16)

3	2	1	0
DCLK3	SLCK	DCLK1	DWUT

(Initial value 0000)

DCLK3 / DCLK1	Selects operation mode	(Note 2)

0 0 : Single clock mode (Normal-1 operating mode)
0 1 : Reserved
1 0 : Dual clock mode (Normal-2 operating mode)
1 1 : Dual clock mode (SLOW operating mode)

SLCK	Selects input clock for the eighth stage of TG	(Note 3)

0 : $fc / 2^7$ [Hz]
1 : fs

DWUT	Sets the warm-up time	(Note 4)

Example : At 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)

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

SMF	Low-frequency clock oscillating state

($fc/2^{17}$ or $fc/2^{10}$ [Hz])

HOLD / SLS	$\overline{HOLD}$ pin state/operation state monitor

	Single-clock mode	Dual-clock mode
0:	$\overline{HOLD}$ pin at "H" level	In Normal operation
1:	$\overline{HOLD}$ pin at "L" level	In SLOW operation

Note 1 . fc ; High-frequency clock [Hz]
fs ; Low-frequency clock [Hz]
Note 2. the configuration of bits.
Note 3. Only Normal-2 operating mode
Note 4. Only switching from SLOW to Normal-2
Note 5. The access to command register (OP16) may cause the outputs over the eighth stage of timing generator to precede that to be expected by maximum 2^7/fc or 1/fs [s].

Figure 2-18. System Clock Control Command Register/Status Register

(2) Instruction Cycle

The instruction execution and on-chip peripheral hardware operations are performed in synchronization with the basic clock. The smallest unit of instruction execution is called the "instruction cycle". The TLCS-470 series instruction set has 2 kinds of instructions, 1-cycle instruction and 2-cycle instruction. Each instruction cycle consists of 4 states (S1 through S4). Each state consists of 2 basic clock pulses.

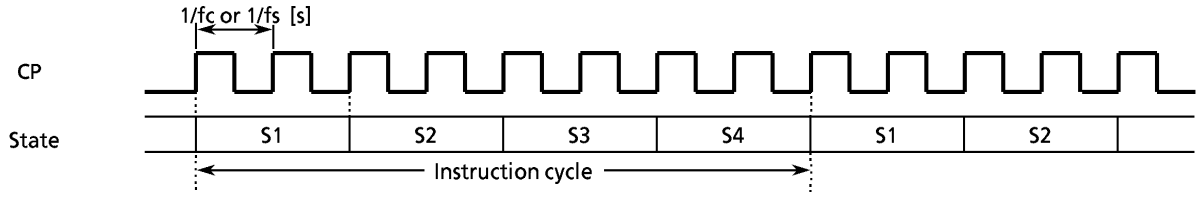


Figure 2-19. Instruction Cycle

2.7.4 Operation Mode

(1) Dual-clock mode

In this mode, the Normal-2 operation is generally performed by generating the instruction cycle from the high-frequency clock (fc). As required, the instruction cycle is generated by the low-frequency clock (fs), and the lower consumption power operation is performed by transferring SLOW operation to HOLD operation. The following describes the switching between the Normal-2, SLOW and HOLD operations in the dual clock mode. At reset, the command register is initialized to the single-clock mode. Since the low-frequency clock is not oscillated, Normal-2 operation in the dual clock mode must be set first. The low frequency clock starts oscillating by transferring to Normal-2 operation.

a. Switching from Normal-2 operation to SLOW operation

Setting DCLK1 (bit 1 of OP16) to "1" switches Normal-2 operation to SLOW operation. However it takes a few seconds to get a stable oscillation of the low-frequency clock. Therefore if there is possible to switch from Normal-2 operation to SLOW operation, wait until the low-frequency clock is stable or check the oscillation state by a program. SMF status (bit 1 of 1P0E) is available to check it.

When the high-frequency clock ($fc/2^7$) is input to the 8th stage of TG, first sets SLCK (bit 2 of OP16) to "1" ans input the low-frequency clock (fs). Then, SMF is monitored by a program. After confirming that SMF is changed "1" to "0" to "1" or "0" to "1" to "0", set DCLK1 to "1". At this time, the high-frequency clock oscillator stops.

b. Returning from SLOW operation to Normal-2 operation

Bit 2 of the command register is cleared to "0" and, at the same time,the warm-up time for return is set to DWUT. When the warm-up time has passed, the Normal-2 operation takes place. By monitoring SLS (bit 0 of the status register), the current operating mode can be known.

Note: The watchdog timer counter is used to count the warming up time.
To count certainly the warming up time, it is neccessary to initialize the counter. When switching from SLOW operation to Normal-2 operation, the watchdog timer must be cleared just before setting of the system clock control command register.

Example : Returning from SLOW operation to Normal-2 operation

```
LD      A, #0000B
OUT     A, %OP15
LD      A, #1100B
OUT     A, %OP16
```

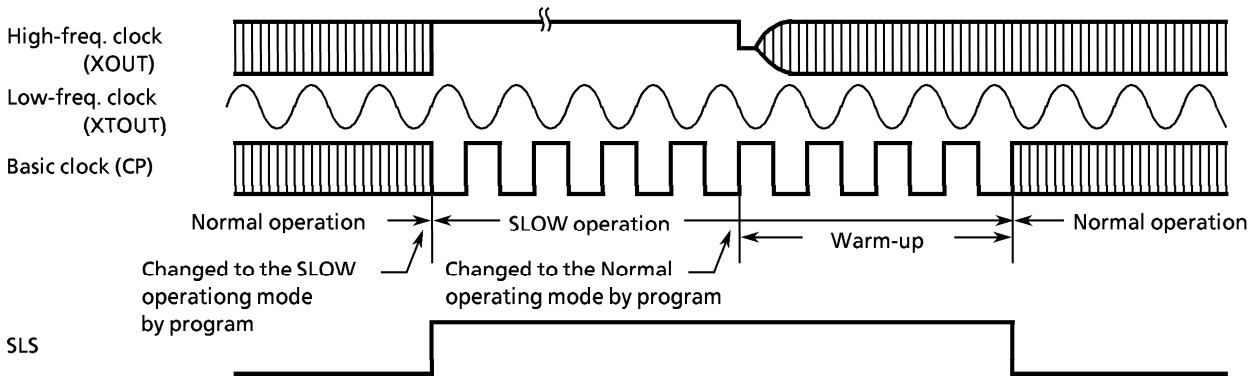


Figure 2-20. System Clock Switching Timing

c. Shifting from the SLOW operation to the HOLD operation
By setting the command in the command register (OP10), the low-frequency clock oscillation stops and the HOLD operation starts, refer to subsection "2.7.5 HOLD Operation MODE".
After being released the HOLD operation, the operation mode is NORMAL-1.

Note. In the HOLD and SLOW operating modes, the power consumed by the oscillator and the internal hardware is reduced. However, the power for the pin interface (depending on the external circuitry and program) is not directly associated with the low-power consumption operation. This must be considered in system design as well as interface circuit design.

(2) singl-clock mode
In this mode, only the high-frequency clock oscillator is used. Pins R72 (XTIN) and R73 (XTOUT) become the ordinary I/O port. The HOLD operating mode is available for reducing power consumption. It is controlled by the command register (OP10). In this mode, therefore, the system clock control command register (OP16) need not be manipulated. For the details of the HOLD operation, refer to Subsection "2.7.5 HOLD Operating Mode".

2.7.5 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 (IP0E or IP10).
Bit 0 of IP0E can be used in the single-clock mode. However the bit 0 of IO0E can not be used in the dual-clock mode, because the bit is used as the operating state monitor (SLS). To monitor the state of HOLD pin input in the dual-clock mode, use bit 3 of IP10. Bit 3 of IP10 can be used in the single-clock mode.

(1) Starts the HOLD operating mode
Setting the command to the command register starts HOLD operation in Normal-1 or SLOW mode. The following states keep during the HOLD operation.
① Oscillator stops and the systemś internal operations are all held up.
② The interval timer 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.

HOLD oprerating mode command register
(Port address : OP10)

3	2	1	0
HLDMS		HWUT	

(Initial value : *0**)

HLDMS	Sets mode and starts HOLD operation

01: Starts HOLD operation in edge sensitive release mode
11: Starts HOLD operation in level sensitive release mode
*0: Reserved

HWUT	Sets the warm-up time at release of HOLD operating mode

Example : At fc = 4 MHz

00: 2^{18}/fc [s] ····· 65.5 [ms]
01: 2^{14}/fc ····· 4.1
10: Reserved
11: 2^{6}/fc ····· 0.016

HOLD operation mode status register
(Port address IP0E)

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

(Port address IP10)

HOLD (K103)			

HOLD	$\overline{\text{HOLD}}$ pin input state

0 : $\overline{\text{HOLD}}$ pin is high
1 : $\overline{\text{HOLD}}$ pin is low (HOLD operation request)

Note 1. * ; *don't care*
Note 2. * ; *Do not access the OP10 when HOLD mode is not used.*

Figure 2-21. HOLD Operating Mode Command Register / Status Register

The HOLD operating mode consists of the level-sensitive release mode and the edge-sensitive release mode.

a. Level-sensitive release mode

In this mode, the HOLD operating is released by setting the $\overline{\text{HOLD}}$ pin to the high level. This mode is used for the capacitor backup with the main 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 clear sequence (warm-up) sets in immediately. Therefore, to start the HOLD operation in the level-sensitive mode, that the $\overline{\text{HOLD}}$ pin input is low (the HOLD operation request) must be recognized in program. This recognition is one of the two ways below:

① Testing $\overline{\text{HOLD}}$ (bit 0 of the status register)
② Applying the $\overline{\text{HOLD}}$ pin input also to the $\overline{\text{INT1}}$ pin to generate the external interrupt 1 request.

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

```
SHOLDH:  TEST   %IP0E, 0        ;Waits until HOLD pin input goes low.
         B      SHOLDH
         LD     A, #1101B       ;OP10←1101B
         OUT    A, %OP10
```

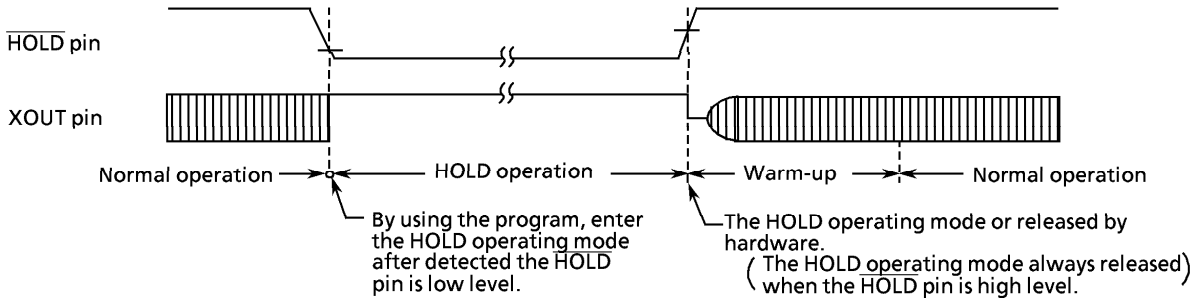


Figure 2-22. Level-sensitive Release Mode

b. Edge-sensitive release 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 release 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
```

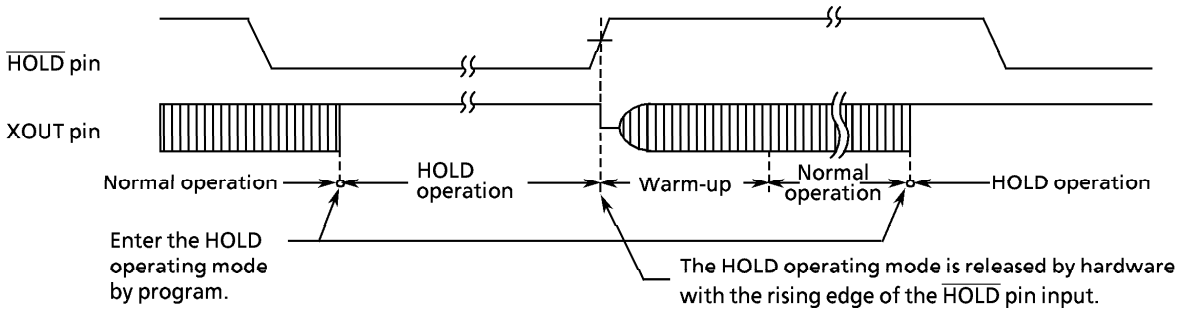


Figure 2-23. Edge-sensitive Release Mode

Note. 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. In the CMOS circuitry, little current flows 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 flow across the port's input transistor, requiring to fix the level by pull-up or other means.

(2) Releases the HOLD operating mode

The HOLD operating mode is released in the following sequence:

① The high-freg. 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 instruction which starts the HOLD operation.

※ *The warm-up time is obtained by dividing the basic clock by the interval timer, so that, if the frequency at clearing the HOLD operation is unstable, the warm-up time shown in Figure 2-23 includes an error. Therefore, the warm-up time must be handled as an approximate value.*

The HOLD operation is also released by setting the $\overline{RESET}$ 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 considerred:
To release the HOLD operation, the power voltage needs to be raised to the operating voltage level. If this is done, the $\overline{\text{RESET}}$ pin input, which is at the high level, also rises with the power voltage. In this case, if a time constant circuit or the like is externally attached, the rise of the $\overline{\text{RESET}}$ pin input voltage goes behind the rise of the power voltage. At this time, if the voltage level of the $\overline{\text{RESET}}$ pin input drops below the non-inverted high level input voltage of the $\overline{\text{RESET}}$ pin input (hysteresis input), the reset operation may occur.

2.8 INTERRUPT FUNCTION

(1) Interrupt Controller

There are 6 interrupt sources (2 external and 4 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	Serial Interface Interrupt	(ISIO)	2	IL_4	EIF = 1, EIR_3 = 1	004_H
	TC1 overflow Interrupt	(IOVF1)	3	IL_3	EIF = 1, EIR_2 = 1	006_H
	TC2 overflow Interrupt	(IOVF2)	4	IL_2	EIF = 1, EIR_1 = 1	008_H
	Interval Timer Interrupt	(ITMR)	5	IL_1		$00A_H$
External	External Interrupt 2	(INT2)	(lowest) 6	IL_0	EIF = 1, EIR_0 = 1	$00C_H$

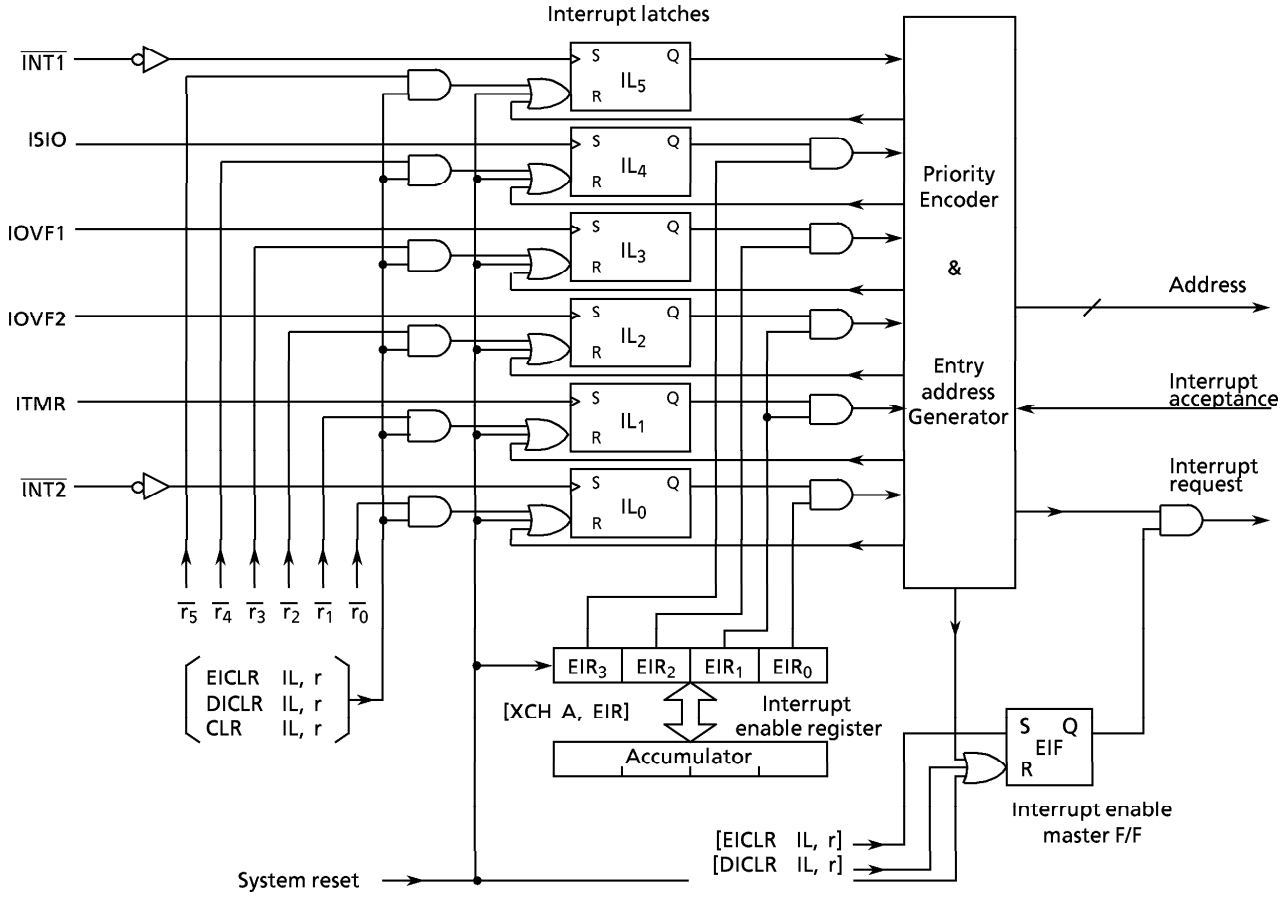


Figure 2-24. Interrupt Controller Block Diagram

a. 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 instructions [EICLR IL, r] and [DICLR IL, r], respectively. The EIF is initialized to "0" during reset.

b. 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 [XCH A, EIR] instruction. The EIR is initialized to "0" during reset.

c. Interrupt latch (IL)

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 ISIO to "0".

```
EICLR   IL,010000B  ;  EIF←1, IL5←0, IL3 – IL0←0
```

(2) Interrupt Processing

An interrupt request is held until the interrupt is accepted or the IL is cleared by the reset or 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].

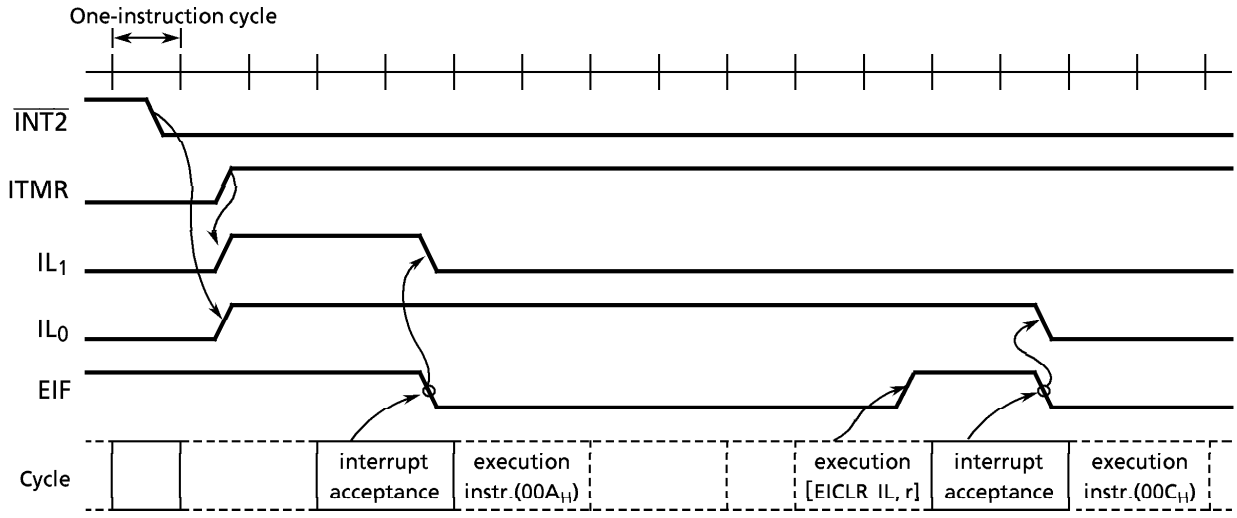
The interrupt acknowledge processing consists of the following sequence:

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.)

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
        RETI
SINT1:   ⋮
```



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 2-25. Interrupt Timing Chart (Example)

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.

Note. When the time required for the interrupt service is longer than that foe the interrupt request, only the interrupt service program is executed without executing the main program.

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".

(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 $\overline{INT2}$).

Because 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 R82 ($\overline{INT1}$) 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 [RETI] instruction must be stored at the interrupt entry address to perform dummy interrupt processing.

The INT2 interrupt can be enable/disable by the EIR. Therefor, the INT2 interrupt occurs at the falling edge of the pin input when R80 (INT2) pin is used for the I/O port.

But bit 0 of the EIR is only kept at "0" not accepting the interrupt request.

2.9 RESET FUNCTION

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

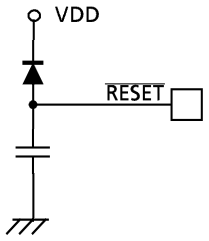


Figure 2-26. Simplified Power-On-Reset Circuit

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"		

2.9.1 Watchdog Timer (WDT)

The watchdog timer capability is provided to quickly detect the CPU malfunction such as endless looping caused by noises or the like, and restore the CPU to the normal state. The WDT is disabled during reset.

The WDT consists of 10 binary counters, a flip-flop, and a controller. Source input clock of binary counters is fc/2^{15}[Hz]. The flip-flop is set to "1" during reset, and cleared to "0" on the rising edge of the binary counter output. The WDT is controlled by the command register (OP15). The command register is initialized to "1000_B" during reset.

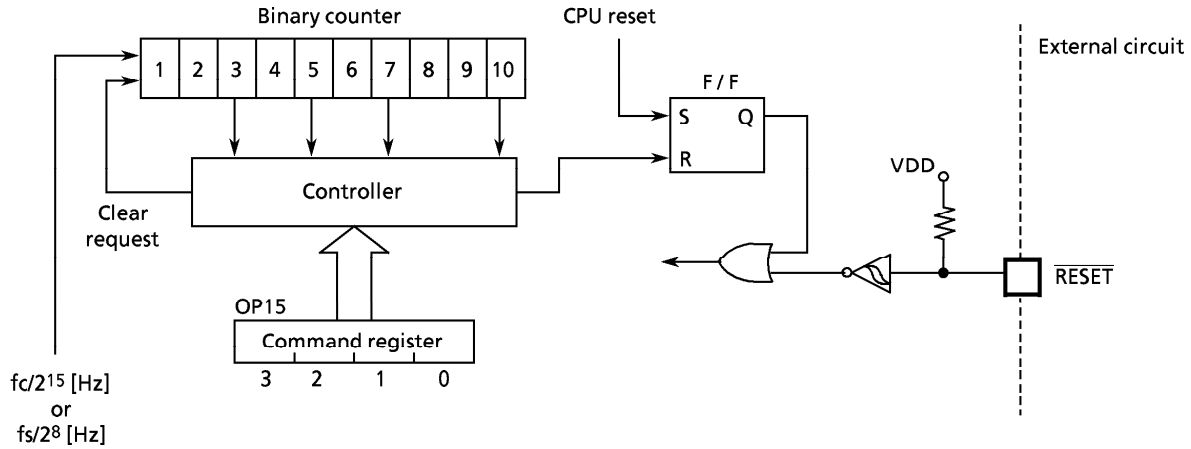


Figure 2-27. Configuration of Watchdog Timer

To detect the CPU malfunction by the WDT:

① Set the WDT detection time, and clear the binary counters.
② Enable the WDT.
③ Clear the binary counters within WDT detection time that was set in ①. If a CPU malfunction occurs, preventing the binary counters from being cleared, the flip-flop is clearred to "0" on the rising edge of the binary counter output, making the malfunction detection signal active.

Example : To enable the with detection time of $63 \times 2^{15}/fc$ [s]

```
        LD      A, #0010B       ; OP15 ← 0010B  (Set WDT detection time,
        OUT     A, %OP15                        clear binary counters)
        LD      A, #0110B       ; OP15 ← 0110B  (Enable WDT)
        OUT     A, %OP15
          :
          :
          :
        LD      A, #0110B       ; OP15 ← 0110B  (Clear binary counters)
        OUT     A, %OP15
          :
          :
```

Within WDT detection timer

Note. It is necessary to clear the binary counter prior to enabling watchdog timer. Further, the Watchdog Timer should be disable by program during warm-up time from SLOW operating mode to Normal-2 operating mode.

Watchdog timer control command register
(port address OP15)

3	2	1	0	
RWT	EWT	TWT		(Initial value 1000)

RWT	Clears Binary counter

0 : Binary counter cleared
(after clear, it is automatically set to "1")

EWT	Watchdog timer enable/disable

0 : Disable
1 : Enable

TWT	Set Watchdog timer detection time	Example At fc = 4.19 MHz	
00 :	$3 \times 2^{15}/fc$ [s]	·····	23 [ms]
01 :	$15 \times 2^{15}/fc$	·····	117
10 :	$63 \times 2^{15}/fc$	·····	493
11 :	$511 \times 2^{15}/fc$	·····	3996

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

(a) Command Register

TWT	Normal-1 operation	Normal-2 operation		SLOW	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

(b) WDT detection timer

Figure 2-28. Watchdog Timer Control Command Register

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 (00_H through $1F_H$). Each port is selected by specifying its port address in an I/O instruction. Table 3-2 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.

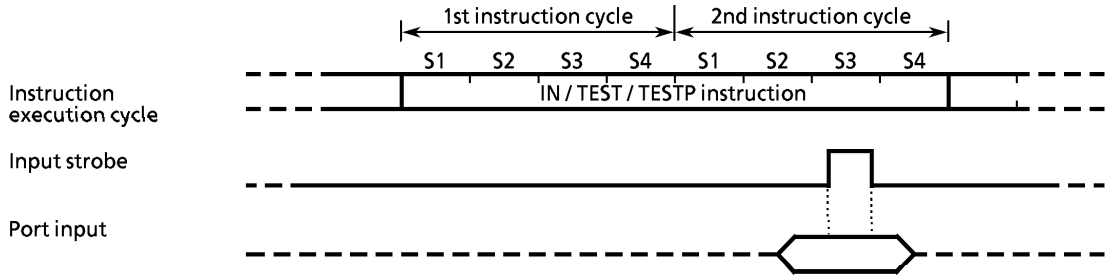


Figure 3-1. Input Timing

(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.

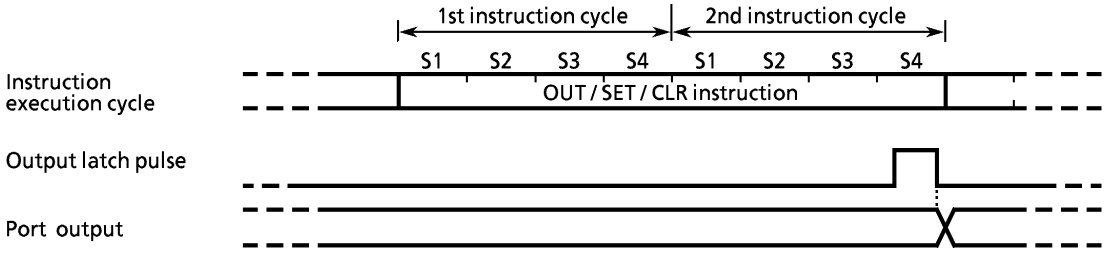


Figure 3-2. Output Timing

3.1.2 I/O Ports

47C222/422 have 7 I/O ports (22 pins) each as follows:

① R4 ; 4-bit input/output (shared with A/D converter analog inputs)
② P5 ; 2-bit output
③ R6 ; 4-bit input/output
④ R7 ; 4-bit input/output (shared with low-frequency resonator connecting pins and pulse output)
⑤ R8 ; 4-bit input/output (shared with zero-cross input, external interrupt input and timer/counter input)
⑥ R9 ; 3-bit input/output (shared with serial port)
⑦ 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.

(1) Ports R4, R5, R6 and R7

These ports are 4-bit I/O ports with a latch (Port R5 is 2-bit). When used as an input port, the latch must be set to "1". The latch is initialized to "1" during reset.

These 4 ports (14 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 addrressing bit manipulation instruction.

```
LD      L, #0011B       ; Sets R43 pin address to L register
CLR     @L              ; R43←0
```

Table 3-1. 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	P50
0 1 0 1	P51

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

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

Port R4 (Port address OP04 / IP04)

3	2	1	0
R43 (AIN3) (SEG19)	R42 (AIN2) (SEG18)	R41 (AIN1) (SEG17)	R40 (AIN0) (SEG16)

Port P5 (Port address OP05 / IP05)

3	2	1	0
		P51	P50

Port R6 (Port address OP06 / IP06)

3	2	1	0
R63 (SEG15)	R62 (SEG14)	R61 (SEG13)	R60 (SEG12)

Port R7 (Port address OP07 / IP07)

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

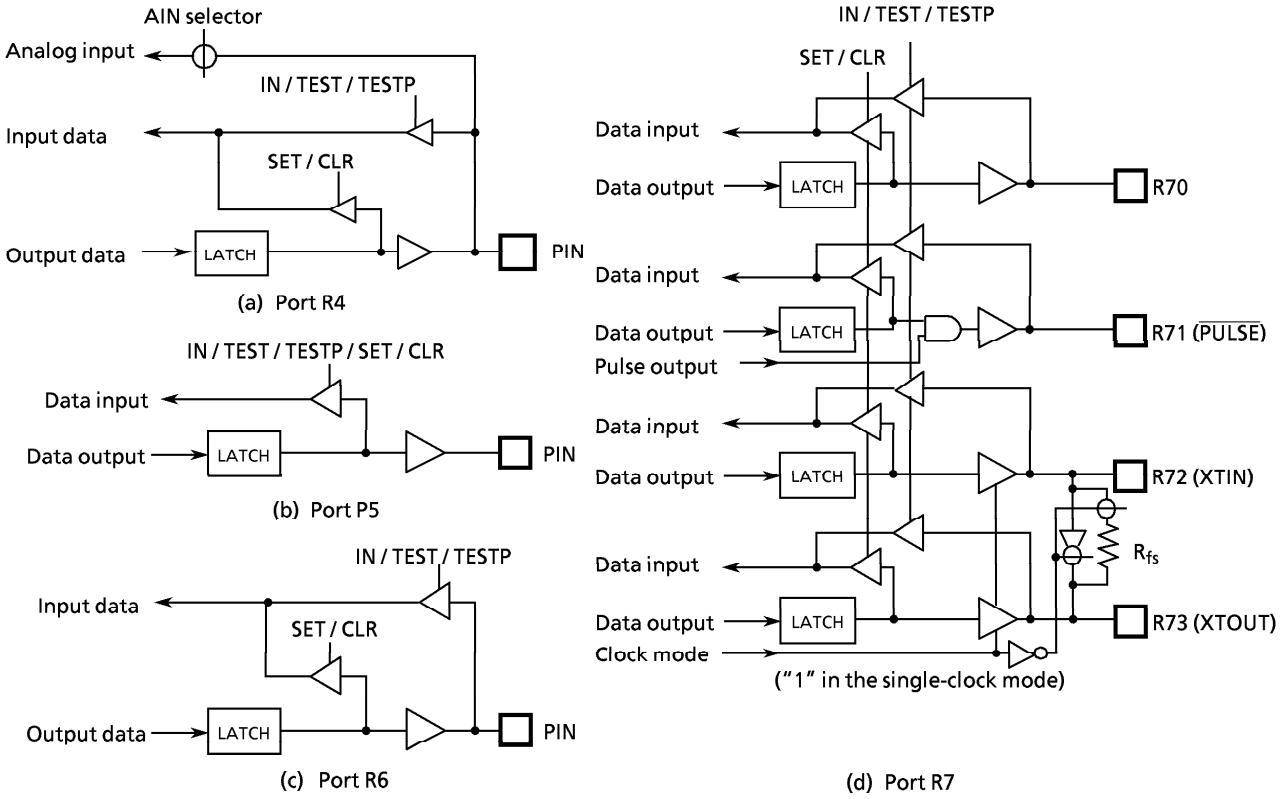


Figure 3-3. Ports R4, R5, R6 and R7

(a) Port R4 (R43 to R40)

Port R4 is 4-bit I/O port with latch shared by the analog inputs for A/D converter and LCD segment output. When used as an analog inputs or segment output, the latch should be set to "1". If other port is used as an output, be careful not to execute the output instruction for any port during A/D conversion in order to keep accuracy of conversion. The latch is initialized to "1" and analog input is selected R40 (AIN0) pin during reset.

(b) Port P5 (P51, P50)

Port P5 is 2-bit output port with a latch. When an input instruction is executed, the latch data is read. The latch is initialized to "1" during reset. Port P5 is on-chip only QFP package type.

(c) Port R6 (R63 to R60)

Port R6 is 4-bit I/O port with latch shared by the LCD segment output. When used as segment output, the latch should be set to "1". The latch is initialized to "1" during reset.

(d) Port R7 (R73 to R70)

Port R7 is shared by the low-frequency resonator connection pins (XTIN, XTOUT) and the pulse output pin ($\overline{\text{PULSE}}$). For the dual-clock mode operation, the low-frequency resonator (32.768 kHz) is connected to R72 (XTIN) and R73 (XTOUT) pins. For the single-clock mode operation, R72 and R73 pins are used for the ordinary I/O ports. When the pulse output is used, R71 ($\overline{\text{PULSE}}$) becomes the pulse output pin. The pulse output is the logical AND output with the port R71 output latch. To use the R71 pin for an ordinary I/O port, the pulse output must be disabled.

(2) Port R8 (R83 to R80)

Port R8 is a 4-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 use 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.

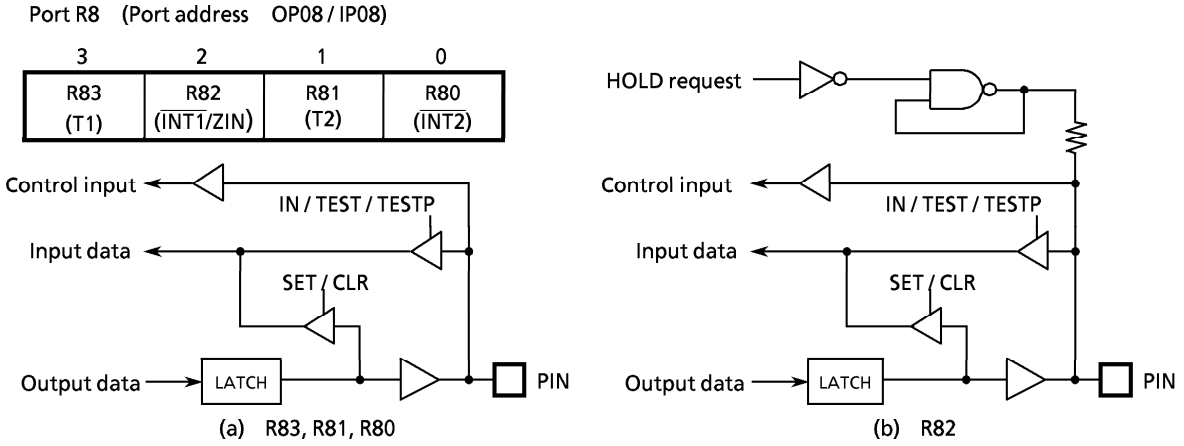


Figure 3-4. Port R8

Note: When R82 ($\overline{\text{INT1}}$) 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 R82 in but bit 0 of the interrupt enable register (EIR_0) is only kept at "0", not accepting the interrupt request.

(3) Port R9 (R92 to R90)

Port R9 is a 3-bit I/O port with a latch. When used as an input, the latch must be set to "1". The latch is initialized to "1" during reset. Port R9 is shared with the serial port. To use port R9 for the serial port, the latch should be set to "1".

Although R93 pin does not exist actualy. However, other instructions can be used, in which an undefined value is read upon execution of an input instruction.

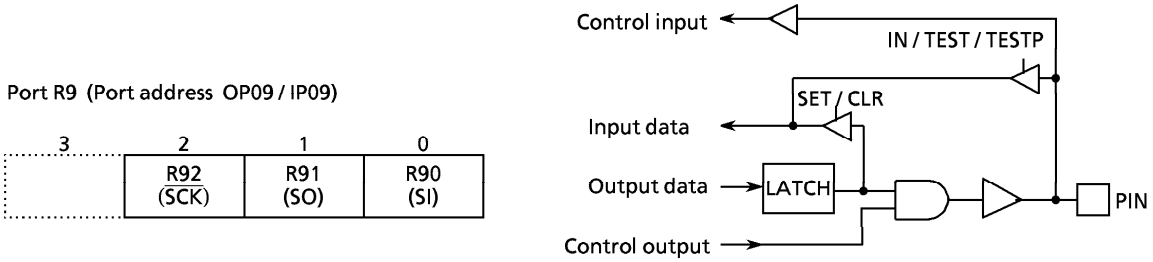


Figure 3-5. Port R9

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

Port KE is a 1-bit sense input port shared by the hold request/release signal input pin ($\overline{HOLD}$). This input port is assigned to the least significant bit of port address IP0E or the most significant bit of port address IP10 and both input data 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. Note that $\overline{KE0}$ input cannot be used in the dual-clock mode because KE0 input monitors the states of SLOW operation mode. To monitor the states of HOLD pin K103 input.

Example: To wait until $\overline{KE0}$ pin goes low. (in single clockmode.)

```
SWAIT :   TEST    %IP0E,  0 ;  Waits if KE0 pin = "L".
          B       SWAIT
```

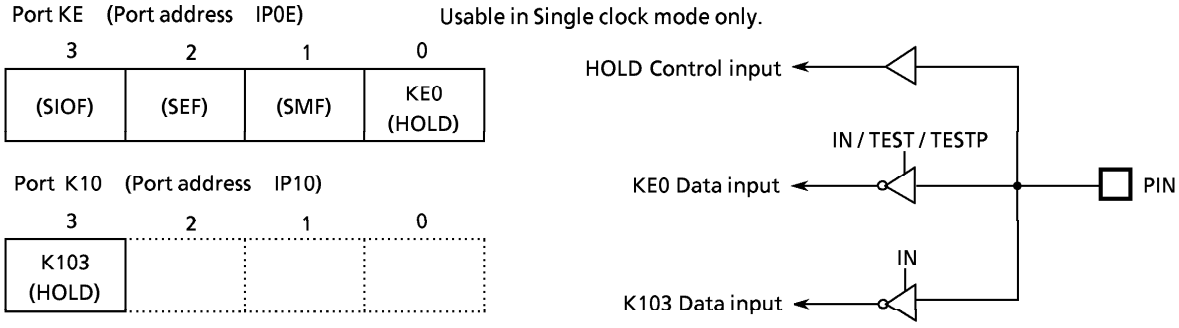


Figure 3-6. Port KE

Table 3-2. Port Address Assignments and Available I/O Instructions

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	——	——	–	–	–	–	–	–	–
01	——	——	–	–	–	–	–	–	–
02	——	——	–	–	–	–	–	–	–
03	——	——	–	–	–	–	–	–	–
04	R4 input port (Analog input)	R4 output port	○	○	○	–	○	○	○
05	P5 output latch	P5 output port	○	○	○	–	○	○	○
06	R6 input port	R6 output port	○	○	○	–	○	○	○
07	R7 input port	R7 output port	○	○	○	–	○	○	○
08	R8 input port	R8 output port	○	○	○	–	○	○	–
09	R9 input port	R9 output port	○	○	○	–	○	○	–
0A	——	——	–	–	–	–	–	–	–
0B	——	——	–	–	–	–	–	–	–
0C	——	——	–	–	–	–	–	–	–
0D	——	——	–	–	–	–	–	–	–
0E	SIO, HOLD status	LCD output resistance selector	○	○	○	–	–	○	–
0F	Serial receive buffer	Serial transmit buffer	○	○	○	–	–	–	–
10_H	——	Hold operating mode control	○	○	–	–	–	–	–
11	——	——	–	–	–	–	–	–	–
12	A / D coverted value	A / D analog input selector	○	○	–	–	–	–	–
13	A / D status input	A / D start register	○	○	–	–	–	–	–
14	——	——	–	–	–	–	–	–	–
15	——	Watchdog timer control	–	○	–	–	–	–	–
16	——	System clock control	–	○	–	–	–	–	–
17	——	Pulse output control	–	○	–	–	–	–	–
18	——	——	–	–	–	–	–	–	–
19	——	Interval Timer interrupt control	–	○	–	–	–	–	–
1A	——	LCD driver control 1	–	○	–	–	–	–	–
1B	——	LCD driver control 2	–	○	–	–	–	–	–
1C	——	Timer/Counter 1 control	–	○	–	–	–	–	–
1D	——	Timer/Counter 2 control	–	○	–	–	–	–	–
1E	——	Serial interface control 1	–	○	–	–	–	–	–
1F	——	Serial interface control 2	–	○	–	–	–	–	–

Note 1 : "——" means the reserved state. Unavailable for the user programs.

Note 2 : Do not use the set and clear instructions with bit 3 of the OP09.

3.2 Interval Timer

The interval timer can be used to generate an interrupt with a fixed frequency. An interval timer interrupt is controlled by the command register (OP19) . And the command resister (OP19) is initialized to "0" during reset. An interval timer interrupt is generated at the first rising edge of the binary counter 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 $fc/2^{15}$[Hz](Single clock mode)

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

Interval timer interrupt control command register (Port address OP19)

3	2	1	0
TMRE		TMRF	

(Initial value 0000)

TMRE	Interrupt enable/disabled
00 :	Stopped
01 :	Enabled
1* :	Reserved

Note. * ; *don't care*

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 $fs/2^{10}$

(a) Comand register

TMRF	Normal-1 operation	Normal-2 operation		SLOW	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]	Reserved	2048 [Hz]
01	$fc/2^{13}$		$fs/2^6$	"	512
10	$fc/2^{15}$		$fs/2^8$	"	128
11	$fc/2^{17}$		$fs/2^{10}$	$fs/2^{10}$	32

(b) Example of interrupt frequency

Figure 3-7. Interval Timer Interrupt Command Register

3.3 Timer/Counters (TC1, TC2)

The 47C222/422 contains two 12-bit timer/counters. RAM addresses are assigned to the count register in unit of 4 bits, permitting the initial value setting and counter reading through the RAM mauipulation 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 use-processed data.

RAM address	MSB		LSB
	Timer/Counter1 (TC1)		
	$TC1_H$	$TC1_M$	$TC1_L$
47C422	($F6_H$)	($F5_H$)	($F4_H$)
47C222	(76_H)	(75_H)	(74_H)

RAM address	MSB		LSB
	Timer/Counter2 (TC2)		
	$TC2_H$	$TC2_M$	$TC2_L$
47C422	(FA_H)	($F9_H$)	($F8_H$)
47C222	($7A_H$)	(79_H)	(78_H)

Figure3-8. The 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 timer/counter 1, and port address OP1D for timer/counter 2. These registers are initialized to "0" during reset.

Timer/counter 1 control command register
(port address OP1C) (Initial value 0000)

3	2	1	0
TC1MS		IPR1	

TC1MS	Mode select
00 :	Stopped
01 :	Event counter mode
10 :	Timer mode
11 :	Pulse width measurement mode

IPR1	Internal pulse rate (interval timer output) select		
	Single clock mode	Dual clock mode	
	Normal 1	Normal 2	SLOW
00 :	$fc / 2^6$ [Hz]	$fc / 2^6$ [Hz]	Reserved
01 :	$fc / 2^8$	$fs / 2$	Reserved
10 :	$fc / 2^{10}$	$fs / 2^3$	Reserved
11 :	$fc / 2^{14}$	$fs / 2^7$	$fs / 2^7$ [Hz]

Timer/counter 2 control command register
(port address OP1D) (Initial value 0000)

3	2	1	0
TC2MS		IPR2	

TC2MS	Mode select
00 :	Stopped
01 :	Event counter mode
10 :	Timer mode
11 :	Pulse width measurement mode

IPR2	Internal pulse rate (interval timer output) select		
	Single clock mode	Dual clock mode	
	Normal 1	Normal 2	SLOW
00 :	$fc / 2^{10}$ [Hz]	$fs / 2^3$ [Hz]	Reserved
01 :	$fc / 2^{14}$	$fs / 2^7$	$fs / 2^7$ [Hz]
10 :	$fc / 2^{18}$	$fs / 2^{11}$	$fs / 2^{11}$
11 :	$fc / 2^{22}$	$fs / 2^{15}$	$fs / 2^{15}$

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

(a) Comand register

Normal-1 operation	Normal-2 operation		SLOW	Max. setting at times mode [s] (fc = 4.194304 MHz, fs = 32.768 kHz)
	SLCK = 0 ($fc / 2^7$)	SLCK = 1 (fs)		
$fc / 2^6$ [Hz]		$fc / 2^6$ [Hz]	Reserved	$2^{18} / fc$ (0.0625)
$fc / 2^8$		$fs / 2$	〃	$2^{20} / fc$ or $2^{13} / fs$ (0.25)
$fc / 2^{10}$		$fs / 2^3$	〃	$2^{22} / fc$ or $2^{15} / fs$ (1)
$fc / 2^{14}$		$fs / 2^7$	$fs / 2^7$	$2^{26} / fc$ or $2^{19} / fs$ (16)
$fc / 2^{18}$		$fs / 2^{11}$	$fs / 2^{11}$	$2^{30} / fc$ or $2^{23} / fs$ (256)
$fc / 2^{22}$		$fs / 2^{15}$	$fs / 2^{15}$	$2^{34} / fc$ or $2^{27} / fs$ (4096)

(b) Internal palse rate and Max. setting time at timer mode

Figure 3-9. Timer/Counter Control Command Register

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 1 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 from TC2 is accepted in the next instruction cycle. Therefore, during a 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 during the interrupt enabled state and the overflow interrupt is accepted immediately after its occurrence, the interrupt is processed in the sequence shown in Figure 3-10. Note that counting continues if there is a count request after overflow occurrence.

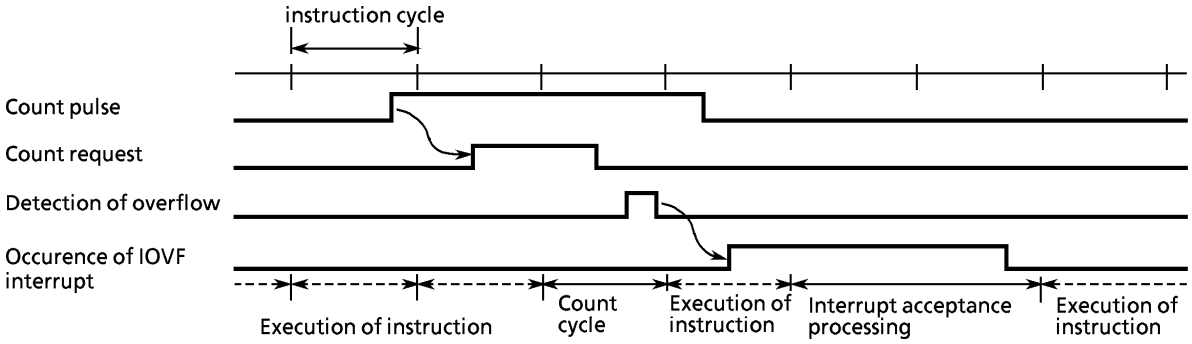


Figure 3-10. Timer/Counter Overflow Interrupt Timing

(1) Event counter mode

In the event counter mode, the timer/counter increments at each rising edge of the external pin (T1, T2) input. T1, T2 pins are shared by R83, R81 pins. Output latch of R83, R81, are set to "1" when used as timer/counter input. Also output latch is initialized "1" during reset. The maximum applied frequency of the external pin input is fc/32 for the 1-channel operation; for the 2-channel operation, the frequency is fc/32 for TC1 and fc/40 for TC2. The apparent instruction execution speed drops most to (9/11) × 100 = 82 % when TC1 and TC2 are operated at the maximum applied frequency because the count operation is inserted once every 4 instruction cycles for TC1 and every 5 instruction cycles for TC2. For example, the instruction execution speed of 2 μs drops to 3.64 μs.

Example: To operate TC2 in the event counter mode.

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

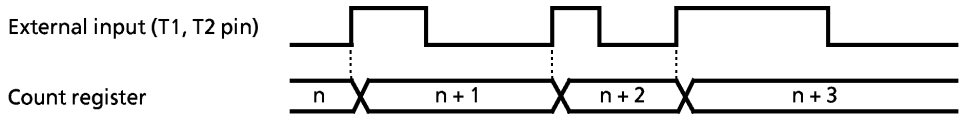


Figure 3-11. Event Counter Mode Timing Chart

(2) Timer mode

In the timer mode, the timer/counter increments at the rising edge of the internal pulse generated from the interval timer. 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 crerate 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) × 100 =

0.8%. For example, the instruction execution speed of 2 μs drops to 2.016 μs. In the timer mode, R83 (T1) and R81 (T2) pins provide the ordinary I/O ports.

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

```
LD      HL,#0F4H        ;   TC1 ← E79H (Setting of count register)
ST      #9,@HL+
ST      #7,@HL+
ST      #0EH,@HL+
LD      A,#1000B        ;   OP1C ← 1010B (Timer mode rate fc/2^10)
OUT     A,%OP1C
LD      A,#0100B        ;   EIR ← 0100B (Enables interrupt)
XCH     A,EIR
EICLR   IL,110111B      ;   EIF ← 1, IL3 ← 0
```

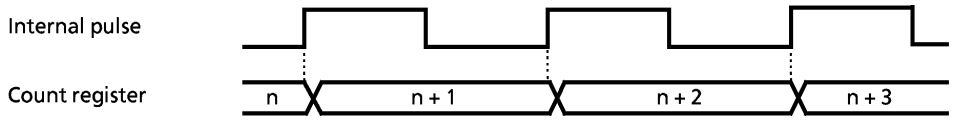


Figure 3-12. Timer Mode Timing Chart

※ The apparent execution rate is calculated as following.

$$1 \div \{ \frac{\text{(Fundamental clock frequency)} / 8}{\text{(Internal pulse rate)}} - 1 \} \times 100 \quad [\%]$$

※ Calculating the initial value of the count register

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

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

$2^{12} - (100 \times 10^{-3}) \times (4 \times 10^{6} / 2^{10}) = 3705\ (E79_H)$

(3) Pulse width measurement mode

In the pulse width measurement mode, the timer/counter increments with the pulse obtained by sampling the external pins (T1,T2) by the internal pulse. As shown in Figure 3-13, 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 by program. Normally, a frequency sufficifient slower than the internal pulse ratesetting is applied to the external pin.

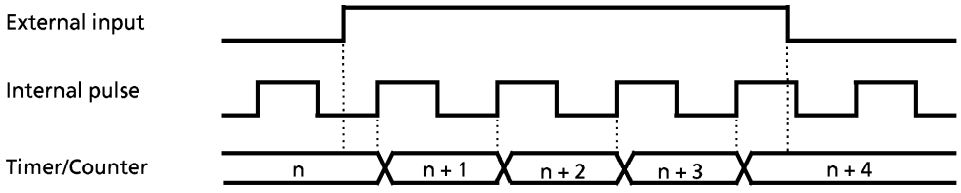


Figure 3-13. Pulse Width Measurement Mode Timing Chart

3.4 Pulse output

Pulse output is used for buzzer drive and remote control carrier. Pulse output is shared with the R71 pin. Pulse output is asynchronous.

3.4.1 Circuit Configuration

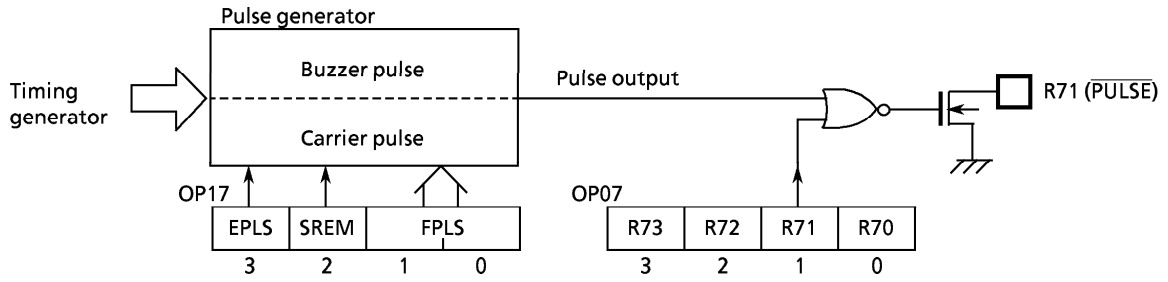


Figure 3-14. Pulse Generator

3.4.2 Control of pulse Output

The pulse output is controlled by the command register (OP17) and R71 output latch data (bit 1 of OP07). At reset, the OP17 is initialized to "0000_B" and pulse output is disabled. To use the pulse output, instruct start/stop of pulse after pulse output is enabled by the OP17.

Also, pulse output is "L" level (the OP17 is cleared to "0000_B") during the HOLD operating mode. External LED and so foth may be destroyed if HOLD operation is executed during output of pulse. Therefore, HOLD operating mode should be execute after pulse is stopped (after R71 output latch set to "1").

Note. It is necessary to disable the pulse output circuit prior to enabling fs output mode.

Example: Buzzer pulse of 2 kHz is output (fc = 4 MHz)

```
LD      A, #1001B
OUT     A, %OP17 ;  OP17←1001B
  ⋮
CLR     %OP07, 1 ;  Pulse start
```

Pulse output control
(Port address : OP17) (Initial value :0000)

3	2	1	0
EPLS	SREM	FPLS	

EPLS	Pulse enable/disable

0 : Disable
1 : Enable

SREM	Selects output mode

0 : Buzzer pulse (1/2 duty)
1 : Remo-com carrier (1/3 duty)

PFLS	Selects output frequency

00 : $fc/2^{12}$, $fc/(2^2 \times 3)$ [Hz]
01 : $fc/2^{11}$, $fc/(2^4 \times 3)$
10 : $fc/2^{10}$, $fc/(2^5 \times 3)$
11 : $fc/2^9$, fs

Buzzer Carrier

Figure 3-15. Pulse Output Control

Table 3-3. Pulse Output Frequency

FPLS	Buzzer pulse		Carrier pulse	
	Pulse rate	at fc = 4.19 MHz	Pulse rate	frequency
00	$fc / 2^{12}$ [Hz]	1.024 [kHz]	$fc / (2^2 \times 3)$ [Hz]	37.9 [kHz] (fc = 455 kHz)
01	$fc / 2^{11}$	2.048	$fc / (2^4 \times 3)$	37.5 (fc = 1.8 MHz)
10	$fc / 2^{10}$	4.096	$fc / (2^5 \times 3)$	37.5 (fc = 3.6 MHz)
11	$fc / 2^9$	8.192	fs	–

(1) Buzzer pulse

The buzzer pulse can be selected one of the four pulse rates by the program. The buzzer pulse is output only when the R71 output latch is "0". "H" level is output when the output latch is "1".

Note. When a piezoelectric buzzer is connected to the pin, voltage may be generated by the buzzer due to thermal or mechanical shock. In such cases, there is danger of the pin being destroyed so a zener diode should always be connected for protection.

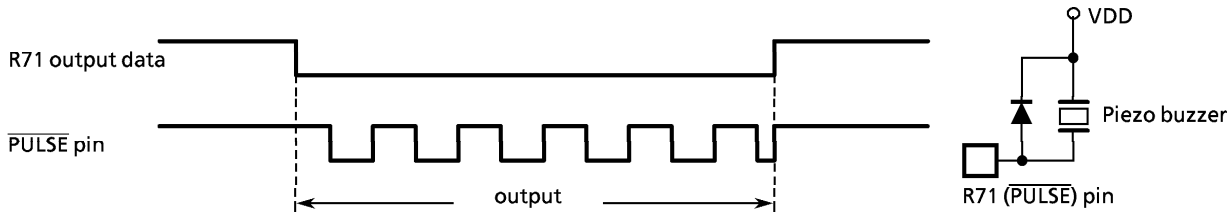


Figure 3-16. Circuit Example of Buzzer Pulse and Timing

(2) Carrier pulse for remote control signal transmitter

The remote control transmitting carrier has a frequency in table 3-3. The basic clock (fc) divided by 12, 48 and 96 is output respectively at 1/3 duty. Also, the remote control transmitting carrier is output only when the R71 output latch is "0". "H" level is output when the output latch is "1". When fs is selected as the output frequency. fs [Hz] is output at 1/2 duty.

Example : To output fs at the remote control transmitting carrier mode (FPLS to "11")

```
LD   A,#0111B
OUT  A,%OP17   ; OP17←0111B
LD   A,#1111B
OUT  A,%OP17   ; OP17←1111B
  ⋮
CLK  %OP07, 1  ; Pulse start
  ⋮
SET  %OP07, 1  ; Pulse stop
```

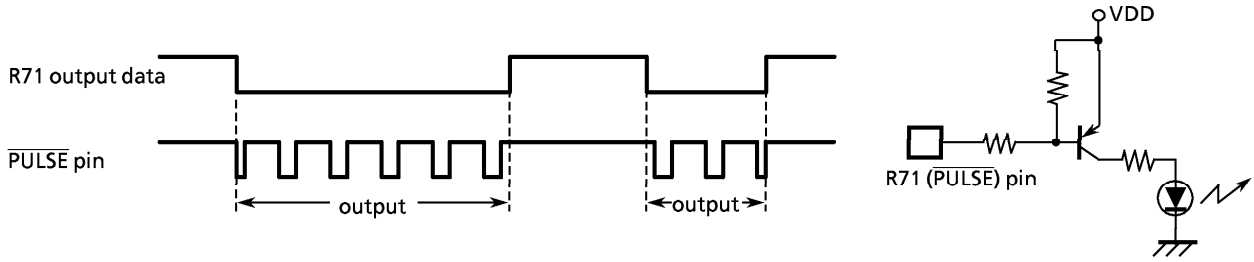


Figure 3-17. Circuit Example of Carrier Pulse and Timing

3.5 Zero-cross detector

R82 pin is used for zero-cross detection input (ZIN) and zero-cross detection can be performed by connecting and external capacitor. To use the zero-cross detector, the R82 output latch must be set to "1" (it is set to "1" during reset).

This function can be used for commercial power supply frequency input, and time base or triac control. ZIN pin is shared by the external interrupt 1. The INT1 interrupt occurs at the falling edge of the pin input by setting interrupt enable master flip-flop (EIF) to "1".

The zero-cross detector is disable and R82 pin is set to "H" level during the hold operating mode. When driving R82 pin directly without using an external capacitor, R82 is used for normal digital input or interrupt input.

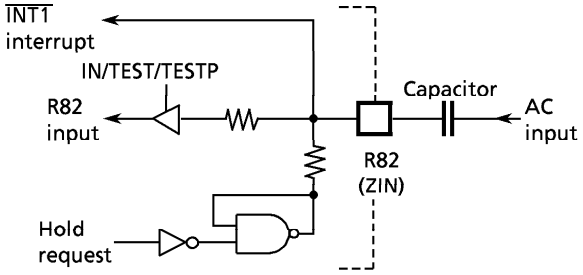


Figure 3-18. Zero-cross Detector

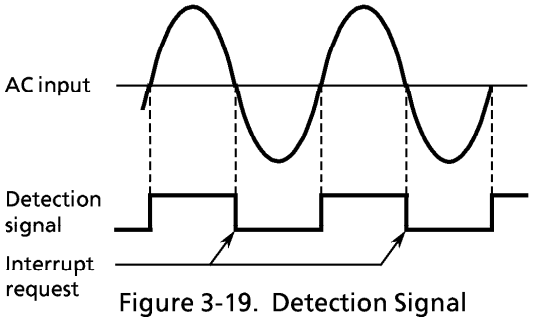


Figure 3-19. Detection Signal

3.6 A/D Converter

47C222/422 have a 8-bit successive approximate type A/D converter and is capable of processing 4 analog inputs. Analog reference voltage can be cut off by command register (bit 3 of OP12). In the hold mode, analog reference voltage is cut off automatically.

3.6.1 Circuit Configuration

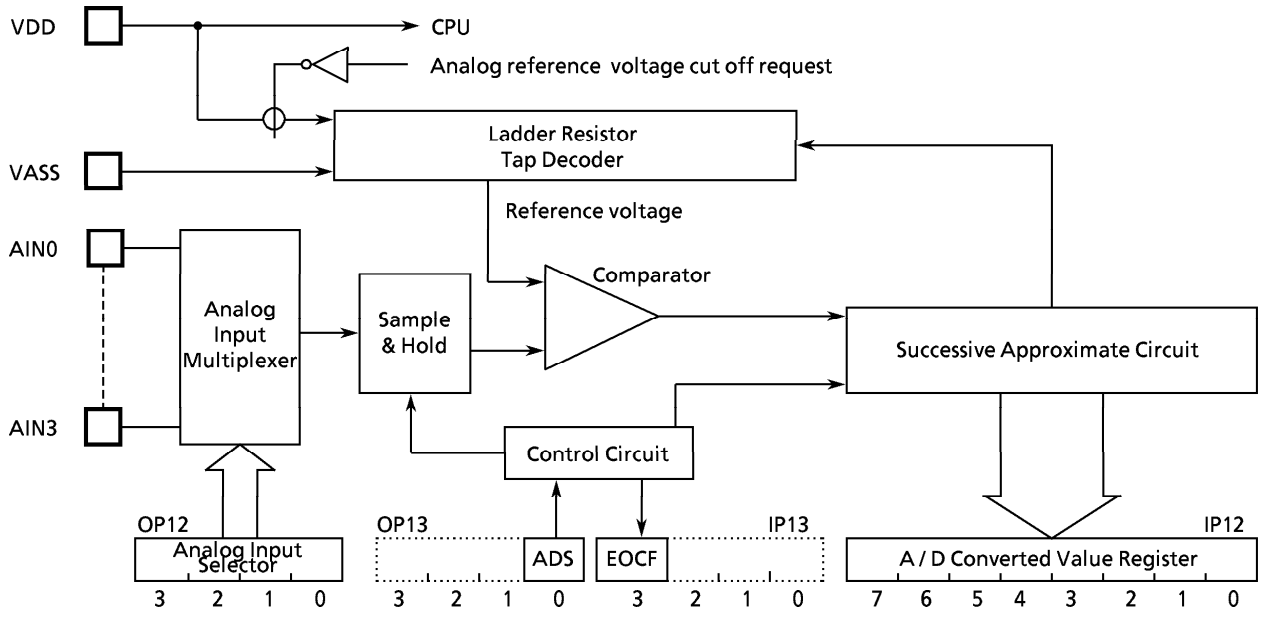


Figure 3-20. Block Diagram of A/D Converter

3.6.2 Control of A/D converter

The operation of A/D converter is controlled by a command register (OP12, OP13, IP12, IP13).

(1) Analog input selector (OP12)
Analog inputs (AIN0 through AIN3) are selected by this register.

Analog input select command register
(Port address OP12)

3	2	1	0	
ICUT	SAIN			(Initial value 0000)

ICUT	Ladder resistor connect/cutoff
0 :	Ladder resistor connect to VDD.
1 :	Ladder resistor cut off from VDD.

SAIN	Analog input selection
000 :	R40 (AIN0)
001 :	R41 (AIN1)
010 :	R42 (AIN2)
011 :	R43 (AIN3)
1** :	Analog input is not selected

Figure 3-21. Analog Input Selector

(2) Start of A/D conversion (OP13)

A/D conversion is started when ADS is set to "1". After the conversion is started, ADS is cleared by hardware. If the restart is requested during the conversion, the conversion is started again at the time. Analog input voltage is hold by the sample hald circuit.

A/D conversion start command register
(Port address OP13)

3	2	1	0	
			ADS	(Initial value 0000)

ADS	A / D conversion enable

1 : A / D conversion is started (clears after starting)

Figure 3-22. A/D conversion start register

(3) A/D converter end freg (IP13)

End of conversion flag (EOCF) is a single bit flag showing the end of conversion and is set to "1" when conversion ended. When both upper 4 bits and lower 4 bits of a converted value are read or A/D conversion is started, EOCF is cleared to "0".

A/D converter status register
(Port address IP13)

3	2	1	0
EOCF			

EOCF	End of conversion flag

0 : Under A/D conversion or before A/D conversion
1 : End of A/D conversion

Figure 3-23. A/D converter status register

(4) A/D converted value register (IP12)

An A/D converted value is read by accessing port address IP12. An A/D converted value is read by splitting into upper 4 bits and lower 4 bits by a value of LR_0 (LSB of the L registers).

A / D converted value register
(Port address IP12)

3	2	1	0
D3 / D7	D2 / D6	D1 / D5	D0 / D4

D7 to D0	A/D converted value is read

When $LR_0 = 0$, lower 4 bits of the converted value is read.
When $LR_0 = 1$, upper 4 bits of the converted value is read.

Figure 3-24. A/D converted value register

3.6.3 How to use A/D converter

Apply positive of analog reference voltage to the VDD pin and negative to the VSS pin. The A/D conversion is carried out by splitting reference voltage between VDD and VSS to bit corresponding voltage by a ladder resistor and making a judgement by comparing it with analog input voltage.

(1) Start of A/D conversion

Prior to conversion, select one of the analog input AIN0 througn AIN3 by the analog input selector. Place output of the analog input, which is to be A/D converted, in the high impedance state by setting "1". If other port is used as an output port, be careful not to execute the output instruction for any port during conversion in order to keep accuracy of conversion.

A/D conversion is started by setting ADS (bit1 of the A/D conversion start register). When conversion ends after 24 instruction cycles, EOCF showing the end of conversion is set to "1".

Analog input voltage is sampled during the following 2 instruction cycles after setting A/D conversion enable.

Note. The sample and hold circuit has capacitor (C_A = 12 pF typ.) with resistor (RA = 5 kΩ typ.). See I/O circuitry table. This capacitor should be charged or discharged within 2 instruction cycles.

(2) Reading of an A/D converted value

After the end of conversion, read an A/D converted value is read by splitting into lower 4 bits and upper 4 bits by the A/D converted value register (IP12). Lower 4 bits of the A/D converted value can be read when LR_0 = 0 and upper 4 bits when LR_0 = 1. Usually an A/D converted value is stored in RAM by an instruction [IN %p, @HL]. Further, if an A/D converted value is read during conversion, it becomes an indefinite value.

(3) A/D conversion with HOLD operation

When the HOLD operation is started during the conversion, the conversion is terminated and an A/D converted value becomes indefinite. Therefore, EOCF is kept clear to "0" after release from the HOLD operation. However, if the HOLD operation is started after the end of conversion (after EOCF has been set), A/D converted value and status of EOCF are held.

Example: Selecting analog input (AIN3), starting A/D conversion, monitoring EOCF, and storing lower 4 bits and upper 4 bits of a converted value to RAM [10_H] and RAM [11_H] respectively.

```
        LD      A, #3H          ; Selecting analog input (AIN3)
        OUT     A, %OP12
        LD      A, #1H          ; Start of A/D conversion
        OUT     A, %OP13
SLOOP : IN      %IP13, A        ; To wait until EOCF goes to "1"
        TEST    A, 3
        B       SLOOP
        LD      HL, #10H        ; HL ← 10H
        IN      %IP12, @HL      ; RAM [10H] ← Lower 4 bits
        INC     L               ; Increment of L registers
        IN      %IP12, @HL      ; RAM [11H] ← Upper 4 bits
```

3.7 Serial Interface (SIO)

The 47C222/422 have a serial interface with an 8-bit buffer. 4-bit/8-bit tramsfer mode can be selected. In the 8-bit transfer mode, data may be transmitted and received simultaneously. The serial interface is connected to the exterenal device via 3 pins (the serial port): R92 ($\overline{SCK}$), R91 (SO), and R90 (SI). The serial port is shared by port R9. For the serial port, the output latch of port R9 must be set to "1". In the transmit mode, R90 pin provides the I/O port; in the receive mode, R91 pin provides the I/O port.

3.7.1 Configuration of Serial Interface

Figure 3-25. shows configuration of serial interface.

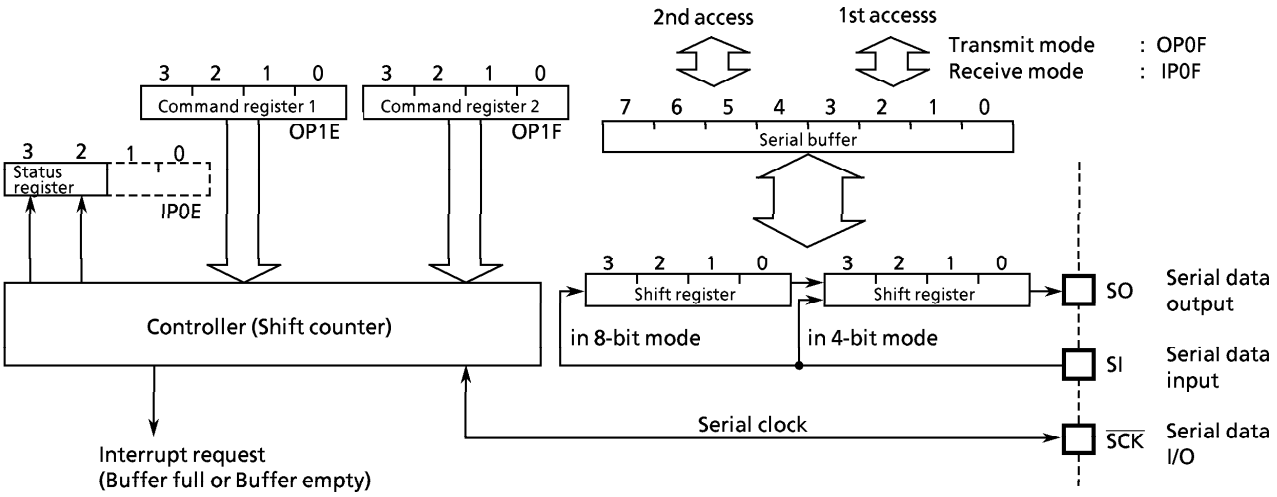


Figure 3-25. Configuration of Serial Interface

3.7.2 Control of Serial Interface

The serial interface is controlled by command registers (OP1E, OP1F) and the status register (IP0E).

Serial interface status register
(Port address IP0E)

3	2	1	0
SIOF	SEF		(HOLD)

(Initial value : 0000)

SIOF	Monitor serial transfer operation state

0 : Transfer is terminated
1 : Transfer is in progress

SEF	Monitors shift operation status

0 : Shift operation is terminated
1 : Shift operation is in progress

Figure 3-26. Serial Interface Status Register

Serial interface control command register1
(port address OP1E) (Initial value 0001)

3	2	1	0
INH	SBIT	CKR	

INH	Forcible stop of serial transfer
0	: Transfer continue
1	: Automatically clearing after stopped

SBIT	Transfer bit number
0	: 4-bit serial transfer
1	: 8-bit serial transfer

SKR	Select serial clock frequency		
	Single clock mode	Dual clock mode	
	Normal 1	Normal 2	SLOW
00	: fc / 2^6 [Hz]	fc / 2^6 [Hz]	Reserved
01	: fc / 2^7	fc / 2^7	Reserved
10	: fc / 2^9	fs / 2^2	Reserved
11	: fc / 2^{12}	fs / 2^5	fs / 2^5 [Hz]

Serial interface control command register 2
(port address OP1F) (Initial value 0000)

3	2	1	0
ESIO	RM	LM	ECKM

ESIO	Instructs serial transfer start/end
0	: Instructs serial transfer end
1	: Instructs serial transfer start

RM	Select transfer mode	(Note 1)	
		4 bit transfer	8 bit transfer
0	:	Transmit mode	Transmit mode
1	:	Receive mode	Transmit/Receive mode

LM	Select shift edge	(Note 2)
0	: Shift at the trailing edge of serial clock	
1	: Shift at the leading edge of serial clock	

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

Note 1. When setting the transfer mode, ESIO must be "0".
Note 2. Transmit/Receive mode : don't care
Transmit mode : LM = 1

Figure 3-27. Serial Interface Control Command Register

3.7.3 Serial clock

For the serial clock, one of the following can be selected according to the contents of the command registers:

(1) Clock source selection

a. Internal clock

The serial clock frequency is selected by command register1.

The serial clock is output on the $\overline{SCK}$ pin. Note that the start of transfer, the $\overline{SCK}$ pin output goes high. This device provides the wait function in which the shift is not occurred until these processings are completed.

The highest transfer rate based on the internal clock is 93750 bits/second (at fc = 6 MHz).

b. External clock

The signal obtained by the clock supplied to the $\overline{SCK}$ pin from the outside is used for the serial clock. In this case, the output latch of R92 ($\overline{SCK}$) must be set to "1" beforehand. For the shift operation to be performed correctly, each of the serial clock high and low levels needs 2 instruction cycles or more to be completed.

(2) Shift edge selection

a. Leading edge

Date is shifted at the leading edge (the falling edge of $\overline{SCK}$ pin input) of the serial clock.

b. Trailing edge

Data is shifted at the trailing edge (the rising edge of $\overline{SCK}$ pin input) of the serial clock. However, in the transmit mode, the trailing-edge shift is not supported.

3.7.4 Transfer bit number

SBIT (bit 2 of the command register 1) can select 4-bit/8-bit serial transfer.

(1) 4-bit serial transfer

In this mode, transmission/reception is performed on 4-bit basis. ISIO interrupt is generated every 4-bit transfer. Transmit/receive data is written/read by accessing the buffer register (OP0F/IP0F) respectively.

(2) 8-bit serial transfer

In this mode, transmission/reception is performed on 8-bit basis. ISIO interrupt is generated every 8-bit transfer. Transmit /receive data is written / read by accessing the buffer register (OP0F / IP0F) twice.

At the first access after setting transfer mode or generating the interrupt request, the write/read operation of lower 4-bit is performed to from the buffer register. At the second access, that of upper 4-bit is performed.

3.7.5 Transfer modes

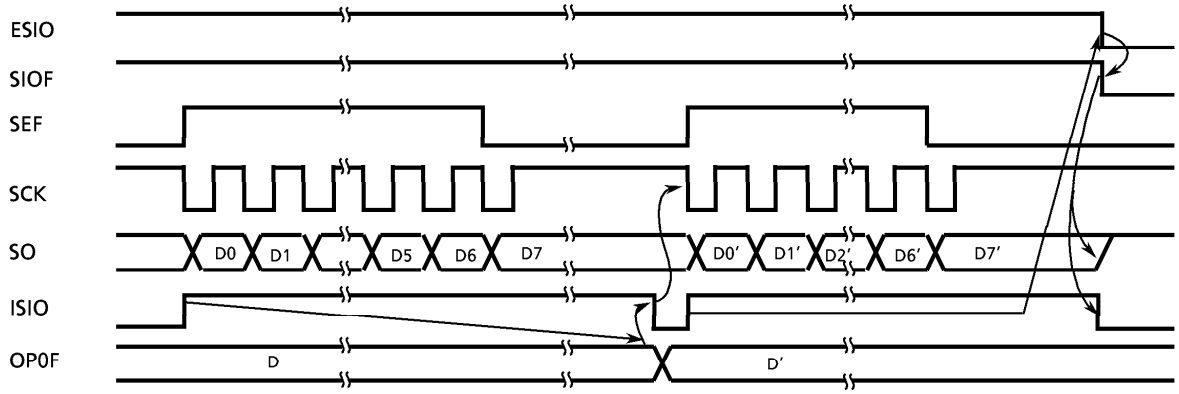
Selection between the transmit mode, the receive mode (at transferring 4 bit) and the transmit and receive mode (at transferring 8 bit) is performed by RM (bit 2 of the command register 2). Switching the transfer modes should be implemented after specifying the end of transferring (clears ESIO to "0") and conferring the end of transferring (ISOF).

(1) Transmit mode

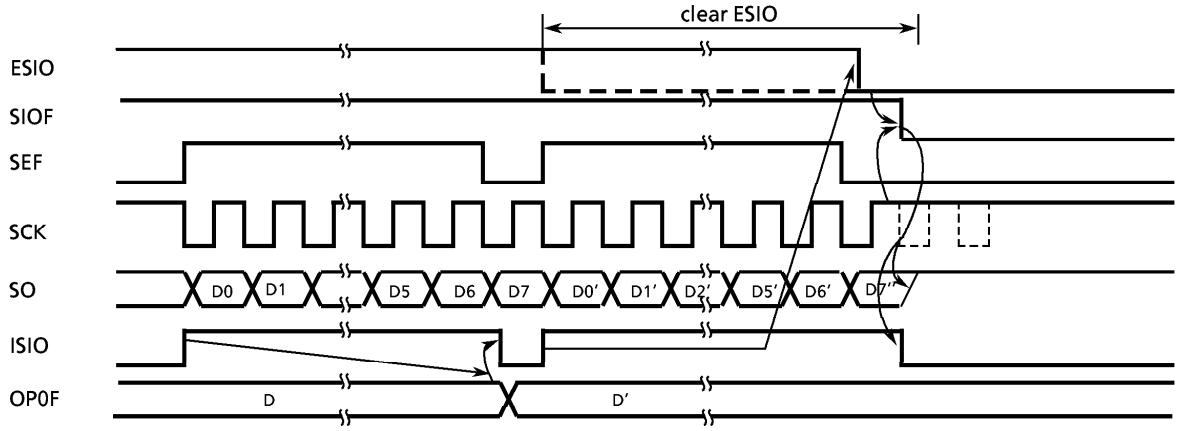
The transmit mode is set to the command register than writes the first transmit data (4 bits or 8 bits) is written to the buffer register (OP0F). (If the transmit mode is not set, the data is not written to the buffer register). In the 8-bit transfer mode, the 8-bit data is wirtten by accessing the buffer register (OP0F) twice. The transmit data is written after the 8-bit transfer mode is set or an interrupt request occurs: the lower 4 bits are written by the first access and the upper 4 bits by the next access. Then, setting ESIO to "1" starts transmission. The transmit data is output to the SO pin in synchronization with the serial clock from the LSB side sequentially. When the LSB is output, the transmit data is moved from the buffer register to the shift register. When the buffer register becomes empty, the buffer empty interrupt (ISIO) to request for the next transmit data is generated. In the interrupt service program, when the nexttransmit data tis written o the buffer register, the interrupt request is reset.

In the operation based on the internal clock, if no more data is set after the transmission of the 4-bit or 8-bit data, the serial clock is stopped and the wait state sets in. In the operation based on the external clock, the data must be set in the buffer register by the time the next data shift operation starts. Therefore, the transfer rate is determined by the maximum delay time between the occurrence of the interrupt request and the writing of data to the buffer register by the interrupt serviced program.

To end transmission, ESIO is cleared to "0" instead of writing the next transmit data by the buffer empty interrupt service program. When ESIO is cleared,transmission stops upon termination of the currently shifted-out data. The transmission end can be known by the SIOF state (SIOF goes "0" upon transmission end). In the operation based on the external clock, ESIO must be cleared to "0" before the next data is shifted out. If ESIO is not cleared before, the transmission stops upon sending the next 4-bit or 8-bit data(dummy).



(a) Internal-clock-based operation with wait



(b) External-clock-based operation

Figure 3-28. Transmit Mode

(2) 4-bit receive mode

Data can be received when ESIO is set to "1" after setting the receive mode to the command register. The data is put from the SI pin to the shift register in synchronization with the serial clock. Then the 4/8-bit data is transferred from the shift register to the buffer register (IP0F),upon which the (buffer full) interrupt (ISIO) to reguest for readingreceived data is generated. The receive data is read from the buffer register by the interrupt service program. When the data has been read, the interrupt request is reset and the next data is put in the shift register to be transferred to the buffer register.

In the operation based on the internal clock, if the previous receive data has not been read from the buffer register at the end of capturing the next data, the serial clock is stopped and the wait operation is performed until the data has been read. In the operation based on the external clock, the shift operation is performed in synchronization with the externally-suppliedclock, so that the data must be read from the buffer register before the next receive data is transferred to it. The maximum transfer rate in the external-clock-based operation is determined by the maximum delay time between the generation of interrupt request and the reading of receive data. In the receive mode, the shift operation may be performed at either the leading edge or the trasiling edge. In the leadingedge shift operation,data is captured at the leading edge of the serial clock, so that the first shift data must be put in the SI pin before the first serial clock is applied at the start of transfer.

Example : To instruct the receive start operation with the 4-bit serial transfer, internal clock and leadingedge shift (with the interrupt enable register already set).

```
LD      A,  #0000B      ;  OP1E ← 0000B (Sets the 4-bit serial transfer)
OUT     A,  %OP1E
LD      A,  #0110B      ;  OP1F ← 0110B (Sets the receive mode)
OUT     A,  %OP1F
EI                      ;  EIF ← 1 (Enables interrupt)
LD      A,  #1110B      ;  ESIO ← 1 (Instructs reception start)
OUT     A,  %OP1F
```

To end the receive operation, ESIO must be cleared to "0". When ESIO is cleared, the completion of the transfer of the current 4-bit data to the buffer register terminates the receive operation. To confirm the end of the receive operation by program, SIOF (bit 3 of the status register) must be sensed. SIOF goes "0" upon the end of receive operation.

Note: If the transfer modes are changed, the contents of the buffer register are lost. Therefore, the modes should not be changed until the last received data is read even after the end of reception is instructed (by clearing ESIO to "0").

The receive operation can be terminated in one of the following approaches determined by the transfer rate:

a. When the transfer rate is sufficiently low (the external-clock-based operation):
If ESIO can be cleared to "0" before the next serial clock is applied upon occurrence of buffer full interrupt in the external-clock-based operation, ESIO is cleared to "0" by the interrupt service program, then the last received data is read.

Example : To instruct reception end when transfer rate is low (leading-edge shift).

```
LD      A,  #0111B      ;  ESIP ← 0 (Instruct reception end)
OUT     A,  %OP1E
IN      %IP0F,A         ;  Acc ← IP0E (Reads received data)
```

b. When the transfer rate is high (the internal/external clock-based operation):
If the transfer rate is high and,therefore, it is possible that the capture of the next data starts before ESIO is cleared to "0" upon acceptance of any interrupt, ESIO must be cleared to "0" by confirming that SEF (bit 2 of the status register) is set at reading the data proceeding the last data. Then, the data is read. In the interrupt serevicing following the reception of the last data, no operation is needed for termination; only the reading of the received data is performed. This method is generally employed for the internal-clock-based operations. For an external-clock-based operation, ESIO must be cleared and the received data must be read before the last data is transferred to the buffer register.

Example: To instruct reception end when transfer rate is high (the internal clock, leading-edge shift).

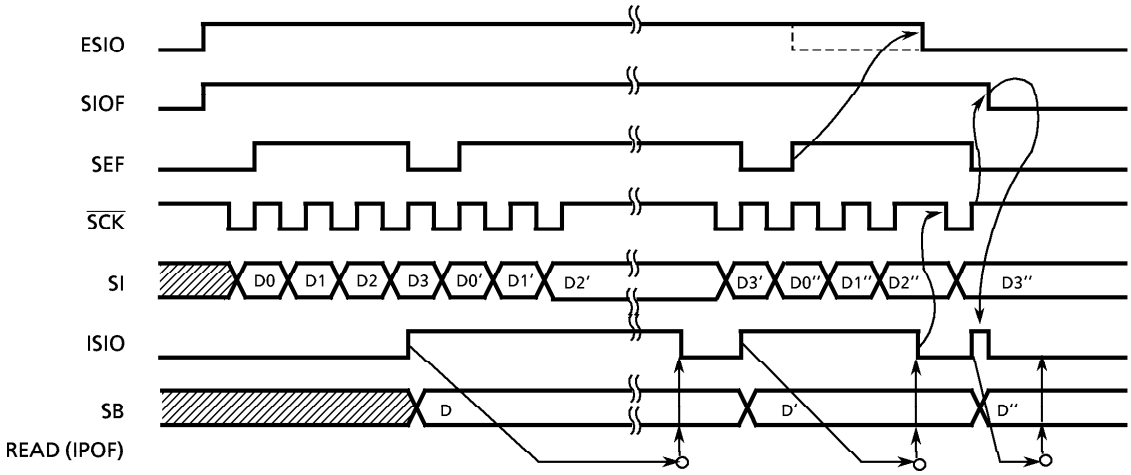
```
SSEF0 :  TEST    %IP0E,  2      ; Waits until SEF = "1"
         B       SSEF0
         LD      A,  #0110B     ; ESIO ← 0
         OUT     A,  %OP1F
         IN      %IP0F,  A      ; Acc ← IP0F (Reads received data)
```

c. One-word reception

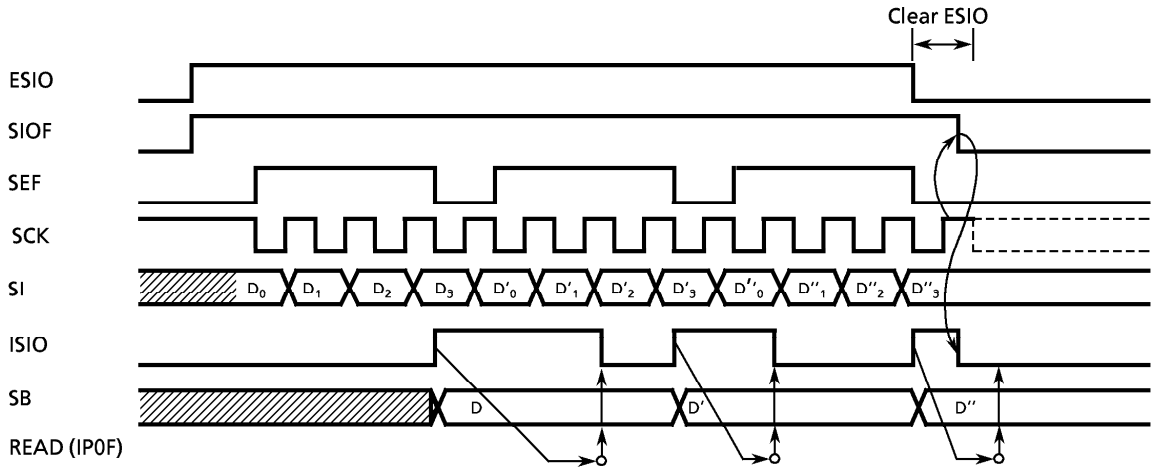
When receiving only 1 word, ESIO is set to "1" then it is cleared to "0" after confirming that SEF has gone "1". In this case, buffer full interrupt is caused only once, so that the received data is read by the interrupt service program.

Example: To instruct the start/end of 1-word reception
(the internal clock, the trailing edge shift).

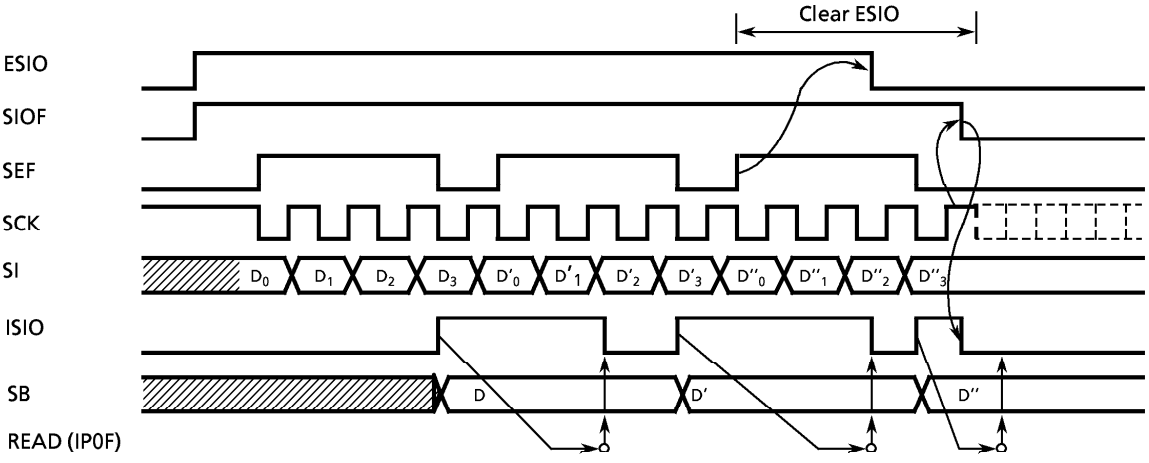
```
        LD      A,  #0100B  ; OP1F ←0100B (Sets in the receive mode)
        OUT     A,  %OP1F
        EI                  ; EIF ← 1 (Enables interrupt)
        LD      A,  #1110B  ; ESIO ← 1 (Instructs reception start)5
        OUT     A,  %OP1F
SSEF0 : TEST    %IP0E,  2   ; Confirms that SEF = "1"
        B       SSEF0
        LD      A,  #0110B  ; ESIO ← 0 (Instructs reception end)
        OUT     A,  %OP1F
```



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



(b) External-clock-based operation, leading-edge shift (when transfer rate is low)



(c) Internal-clock-based operation, leading-edge-shift (when transfer rate is high)

Figure3-29. 4-bit Receive Mode

(3) 8-bit Transmit/Receive Mode

After setting the transmition/reception mode to the command register, write first transmit data into the buffer register. Then, when "1" is set to ESIO, data transmition/reception becomes possible. The transmit data is output to the SO pin at the leading edge of serial clock and the receive data is input from the SI pin at the trailing edge. If the shift register is filled with the receive data, the data is transferred to the buffer register and ISIO (buffer full) interrupt is generated to request data read. The received data is read from the buffer register by the interrupt service program, and then write the transmit data to the buffer register.

Lower order 4 bits of both transmit and receive data are read/written from/into the buffer register by first access after setting of transmition/reception mode or generation of ISIO and higher 4 bits by next access.

In the operation based on the internal clock, SIO becomes the wait state until the received data are read out and the next data to be transmitted are written.

In the operation based on the external clock, the shift operation is synchronized with the external clock ; therefore, it is necessary to read the data received and to write data to be sent next before starting the next shift operation. The maximum transfer rate using an external clock is determined by the maximum delay time between the generation of the interrupt request and the writing of the data to be transmitted after the reading of the received data.

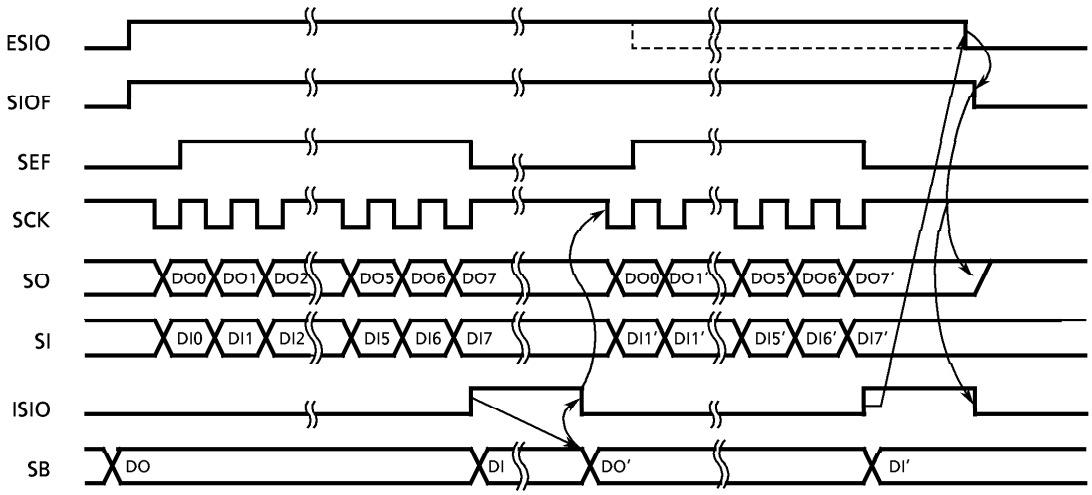
Also, the buffer register is used for both transmission and reception, therefore, the data must be written after reading 8 bits of receive data.

This operation is ended by clearing ESIO to "0". When ESIO is cleared, this operation is ended after transfer of the current 8 bits od data to the buffer register is completed. Programs can confirm that the operation has been completed by sensing SIOF (bit 3 of the status register) because SIOF is cleared to "0" when the operation is completed.

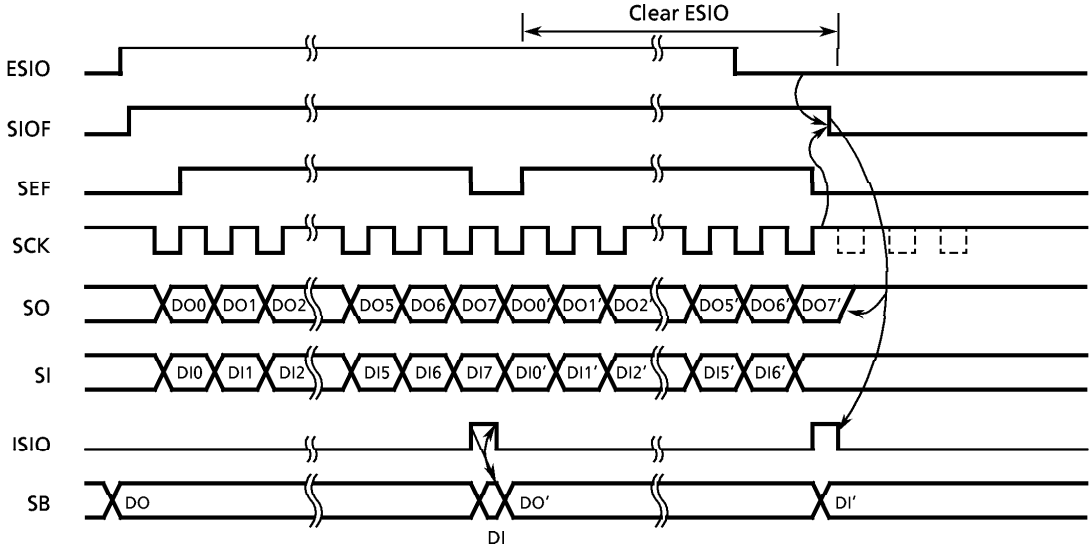
3.7.6 Stopping serial transfer

A serial transfer operation can be stopped forcibly.

It is stopped by setting INH (bit 3 of command register 1) to "1", clearing the shift counter. When the serial transfer is over, INH is automatically cleared to "0" with no other bits of command register affected. In the transmit mode of this case, $\overline{SCK}$ and SO output are initialized to "H" level whereas the shift register is not cleared. Therefore, after the resumption of transmit, SO holds the data just before forcible stop via the shift register until the 1st shift data comes to SO.



(a) Internal clock based operation with wait



(b) External clock based operation

Figure3-30. 8-bit Transmit/Receive Mode

3.8 LCD Driver

The 47C222/422 have the circuit that directly drives the liquid crystal display (LCD) and its control circuit.

The 47C222/422 have the following connecting pins with LCD.

① Segment output port 12 pins (SEG11 to SEG0)
② Segment output port (shared with I/O port) 8 pins (SEG19 to SEG12)
③ Common output port 4 pins (COM4 to COM1)

In addition, VLC pin is provided as the drive power pin.

The devices that can be directly driven is selectable from LCD of the following drive methods.

① 1 / 4 Duty (1 / 3 Bias) LCD ····· Max. 80 Segment (10 digits × 8 segments)
② 1 / 3 Duty (1 / 3 Bias) LCD ····· Max. 60 Segment (7 digits × 8 segments)
③ 1 / 2Duty (1 / 2 Bias) LCD ····· Max. 40 Segment (5 digits × 8 segments)
④ Static LCD ····· Max. 20 Segment (2 digits × 8 segments)

3.8.1 Configuration of LCD driver

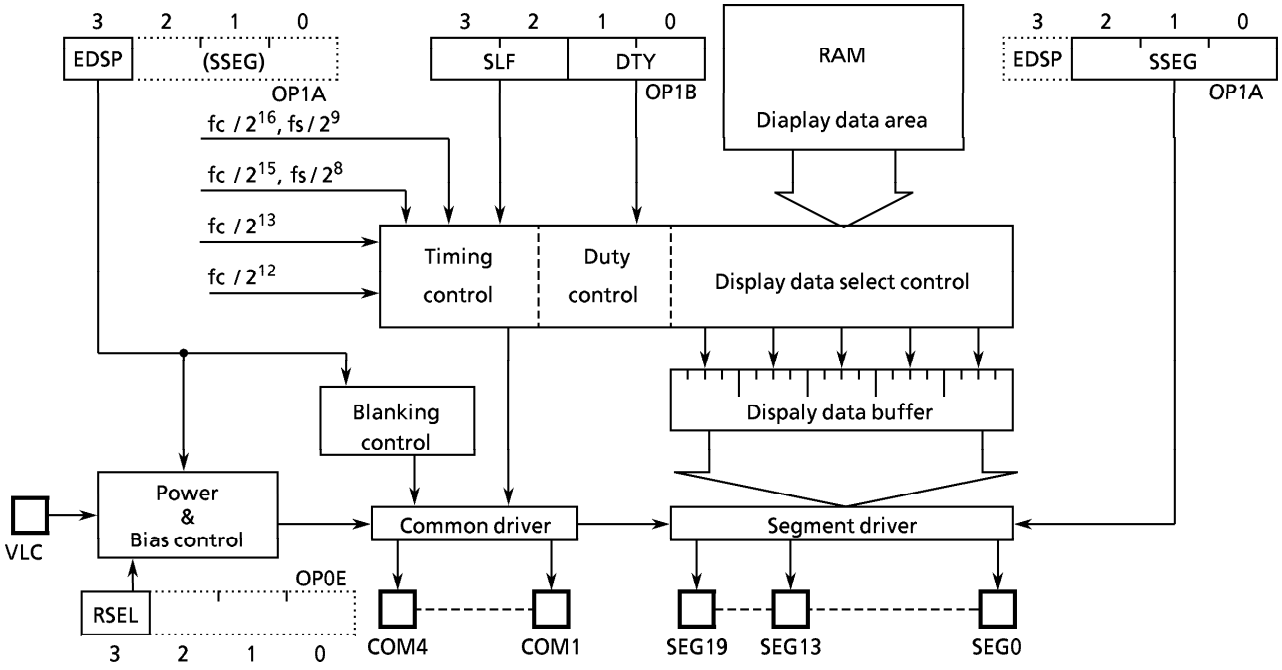


Figure 3-31. LCD Driver

3.8.2 LCD output resistance

LCD output resistance can be selected with using LCD display. Selecting of high / low resistance is executed by RSEL (bit 3 of OP0E). Output resistance is set to 20/200 kΩ during reset.

Selection of output resistance
(Port address : OP0E) (Initial value : 0000)

3	2	1	0
RSEL			

RSEL	LCD output resistance
0 :	Output resistance1 (20/200 kΩ)
1 :	Output resistance2 (10/70 kΩ)

Note. Bit 2 to 0 must be set to "0"

Figure 3-32. Command Register

Table 3-5. Selection of LCD output resistance

RSEL	Output low resistance		Output high resistance	
	Segment R_{OS1}	Common R_{OC1}	Segment R_{OS2}	Common R_{OC2}
0	20 kΩ		200 kΩ	
1	10 kΩ		70 kΩ	

Note. The output resistance shows Typ. values (Topr = 25 °C, VDD = 5 V)

3.8.3 Control of LCD driver circuit

The LCD driver is controlled by the command register 1,2 (OP1A, OP1B). Further, when the command register 2 is accessed, the most significant bit of the command register 1 must be set to "0" (Blanking).

LCD Driver control command register 1
(Port address : OP1A) (Initial value : 0000)

3	2	1	0
EDSP	SSEG		

LCD Display Control

0: Blanking

1: Enables LCD display (Blanking is released)

SSEG | Selection of port/segment

Pin / SSEG	13	14	15	16	17	18	19	20
000	P	P	P	P	P	P	P	P
001	S	P	P	P	P	P	P	P
010	S	S	P	P	P	P	P	P
011	S	S	S	P	P	P	P	P
100	S	S	S	S	P	P	P	P
101	S	S	S	S	S	P	P	P
110	S	S	S	S	S	S	P	P
111	S	S	S	S	S	S	S	S

P : port, S : segment

LCD Driver control command register 2
(Port address : OP1B) (Initial value : 0000)

3	2	1	0
SLF		DTY	

Selection of LCD drive frequency

	Normal Mode	SLOW Mode
00:	$fc / 2^{12}$ [Hz]	Unused
01:	$fc / 2^{13}$	Unused
10:	$fc / 2^{15}$	$fs / 2^{8}$ [Hz]
11:	$fc / 2^{16}$	$fs / 2^{9}$

Selection of driving methods

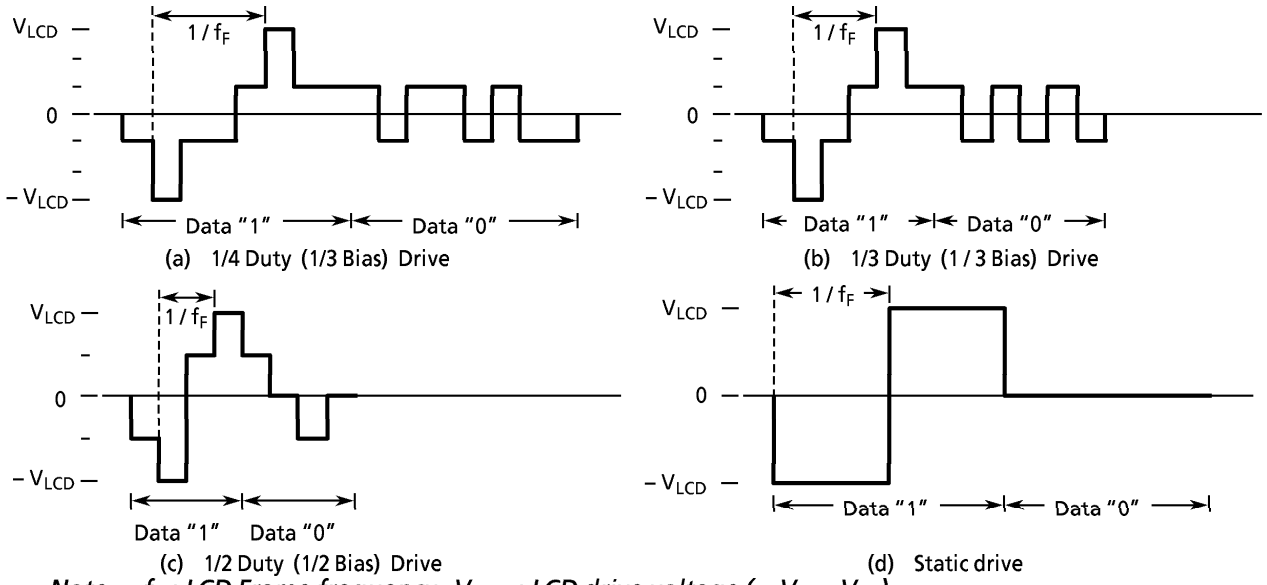
00:	1 / 4 Duty	(1 / 3 Bias)
01:	1 / 3 Duty	(1 / 3 Bias)
10:	1 / 2 Duty	(1 / 2 Bias)
11:	Static	

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

Figure 3-33. LCD Driver control Command Register

(1) Driving methods of LCD driver

Driving methods of LCD is selected 4 kind of DTY (bit 1 to 0 of command register 2). The drive method is initialized according to LCD used in the initial program.



Note. f_F *; LCD Frame frequency* V_{LCD} *; LCD drive voltage (* $= V_{DD}-V_{LC}$*)*

Figure 3-34. LCD drive wareform (Voltage COM-SEG Pins)

(2) Frame frequency

Frame frequency (f_F) is set according to the drive method and base frequency as shown in the following table 3-6. The base frequency is selected by SLF (the lower 2 bits of the command register) according to the reference clock frequency fc and fs.

Table 3-6. Setting of LCD Frame Frequency

a. At the single clock mode

SLF	BASE FREQUENCY [Hz]	FRAME FREQUENCY [Hz]			
		1 / 4 DUTY	1 / 3 DUTY	1 / 2 DUTY	STATIC
11	$\frac{fc}{2^{16}}$	$\frac{fc}{2^{16}}$	$\frac{4}{3} \cdot \frac{fc}{2^{16}}$	$\frac{4}{2} \cdot \frac{fc}{2^{16}}$	$\frac{fc}{2^{16}}$
	(fc = 4 MHz)	61	81	122	61
10	$\frac{fc}{2^{15}}$	$\frac{fc}{2^{15}}$	$\frac{4}{3} \cdot \frac{fc}{2^{15}}$	$\frac{4}{2} \cdot \frac{fc}{2^{15}}$	$\frac{fc}{2^{15}}$
	(fc = 4 MHz)	122	163	244	122
01	$\frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$	$\frac{4}{3} \cdot \frac{fc}{2^{13}}$	$\frac{4}{2} \cdot \frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$
	(fc = 4 MHz)	488	651	977	488
00	$\frac{fc}{2^{12}}$	$\frac{fc}{2^{12}}$	$\frac{4}{3} \cdot \frac{fc}{2^{12}}$	$\frac{4}{2} \cdot \frac{fc}{2^{12}}$	$\frac{fc}{2^{12}}$
	(fc = 400 kHz)	98	130	195	98

Note . fc ; High-frequency clock [Hz]

b. At the dual clock mode

SLF	BASE FREQUENCY [Hz]	FRAME FREQUENCY [Hz]			
		1 / 4 DUTY	1 / 3 DUTY	1 / 2 DUTY	STATIC
10	$\frac{fs}{2^{9}}$	$\frac{fs}{2^{9}}$	$\frac{4}{3} \cdot \frac{fs}{2^{9}}$	$\frac{4}{2} \cdot \frac{fs}{2^{9}}$	$\frac{fs}{2^{9}}$
	(fs = 32 kHz)	63	83	125	63
01	$\frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$	$\frac{4}{3} \cdot \frac{fs}{2^{8}}$	$\frac{4}{2} \cdot \frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$
	(fs = 32 kHz)	125	167	250	125

Note . fs ; Low-frequency clock [Hz]

(3) LCD drive voltage

The LCD drive voltage (V_{LCD}) is given by the difference in potential (V_{DD}-V_{LC}) between pins VDD and VLC. Therefore, when the CPU operating voltage and LCD drive voltage are the same, the VLC pin is connected to the VSS pin.

The LCDs light only when the difference in potential between the segment output and common output is $\pm V_{LCD}$, and turn off at all other times.

During reset, the power switch of the LCD driver is turned off automatically, shutting off the VLC voltage.

Both the segment output and common output become V_{DD} level at this time and the LCDs turn off.
The power switch is turned on to supply VLC voltage to the LCD driver by setting EDSP (bit 3 of the command register 1) to "1_B". After that, the power switch will not turn off even during blanking (setting EDSP to "0") and the VLC voltage continues to flow.
The power switch is turned off during hold operation low power consumption by turning off the LCD. When hold operation is released the status in effect immediately before the hold operation is reinstated.

3.8.4 LCD display operation

(1) Display data setting

Display data are stored to the display data area (Max 20 words) in the data memory.
The display data stored to the display data area (address 20 to 33_H) are read automatically and sent to the LCD driver by the hardware.
The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Therefore, display patterns can be changed by only overwriting the contents of the display data area with a program. The table look up instruction is mainly used for this overwriting.
Figure 3-37 shows the correspondence between the display data area and the SEG/COM pins. The LCD light when the display data is "1" and turn off when "0".
The number of segment which can be driven differs depending on the LCD drive method therefore, the number of display data area bits used to store the data also differs. Consequently, data memory not used to store display data and data memory for which the addresses are not connected to LCD can be used to store ordinary user's processing data.

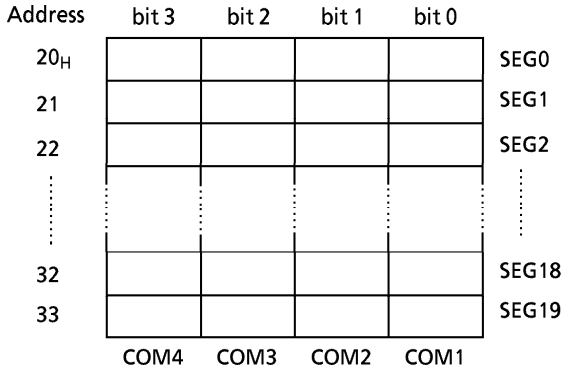


Figure 3-35. LCD Display Data Area

Table 3-7. Drivinig Method and Bit for Display Data

Driving methods	bit 3	bit 2	bit 1	bit 0
1 / 4 Duty	COM4	COM3	COM2	COM1
1 / 3 Duty	–	COM3	COM2	COM1
1 / 2 Duty	–	–	COM2	COM1
Static	–	–	–	COM1

Note. – ; This bit is not used for display data.

(2) Blanking

Blanking is applied by setting EDSP to "0" and turns off the LCD by outputting the non light operation level to the COM pin.
The SEG pin continuously outputs the signal level in accordance with the display data and drive method.
With static drive, no voltage is applied between the COM and SEG pins when the LCD is turned off by data (display data cleared to "0"), but the COM pin output becomes constant at the $V_{LCD}/2$ level when turning off the LCD by blanking, so the COM and SEG pins are then driven by $V_{LCD}/2$.

3.8.5 Control method of LCD driver

(1) Initial Setting

Flow chart of initial setting are as shown in Figure 3-36.

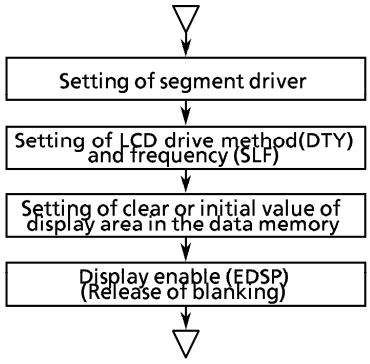


Figure 3-36. Initial setting of LCD driver

Example : When operating the 47C422 with 1/4 duty LCD using a from frequency of fc/2^{15} [HZ] .

```
LD      A, #0111    ; Blanking and all segment
OUT     A, %OP1A
LD      A, #1000    ; Setting 1/4 duty drive and frame
                      frequency
OUT     A, %OP1B
  ⋮                 ; Setting of clear or intial value of
  ⋮                   display area in the data memory
LD      A, #1111B   ; Display enable
OUT     A, %OP1A
  ⋮
```

(2) Store of display data

Normally, display data are kept permanently in the program memory and are then stored to the display data area by the table look-up instruction.

This can be explained using numerical display with 1/4 duty LCD as an example. The COM and SEG connections to the LCD are the same as those shown in Figure 3-37 and the display data are as shown in Table 3-8. Programming example for displaying numerals corresponding to BCD data stored at address 10_H in the data memory is shown below.

```
        LD      HL,  #0FCH      ; To set the DC
        LD      A,   10H
        ST      A,   @HL+
        ST      #DTBL / 16,@HL+
        ST      #DTBL / 256,@HL+
        LD      HL,  #20H       ; Store of display data
        LDL     A,   @DC
        ST      A,   @HL+
        LDH     A,   @DC+
        ST      A,   @HL+
         ⋮
DTBL  :DATA     11011111B,   00000110B,
                11100011B,   10100111B,
                00110110B,   10110101B,
                11110101B,   00010111B,
                11110111B,   10110111B
```

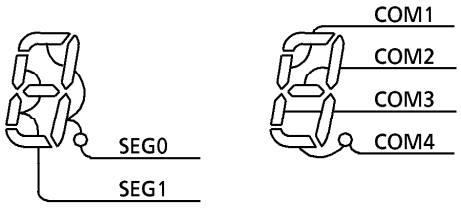


Figure 3-37. Example of COM and SEG connections

Table 3-8. Example of display data (1 / 4 Duty LCD)

Numeral	Display	Display data		Numeral	Display	Display data	
		Upper	Lower			Upper	Lower
0		1101	1111	5		1011	0101
1		0000	0110	6		1111	0101
2		1110	0011	7		0001	0111
3		1010	0111	8		1111	0111
4		0011	0110	9		1011	0111

Table 3-9 shows the same numerical display used in Table 3-8, but using 1/2 duty LCD. The connections of the COM and SEG pins to the LCD are the same as those shown in Figure 3-40.

Table 3-9. Example of display data (1 / 2 Duty LCD)

Numeral	Display data				Numeral	Display data			
	Upper		Lower			Upper		Lower	
0	**01	**11	**01	**11	5	**11	**10	**01	**01
1	**00	**10	**00	**10	6	**11	**11	**01	**01
2	**10	**10	**01	**11	7	**01	**10	**00	**11
3	**10	**01	**01	**11	8	**11	**11	**01	**11
4	**11	**10	**00	**10	9	**11	**10	**01	**11

*Note. * ; don't care*

(3) Example of drive output

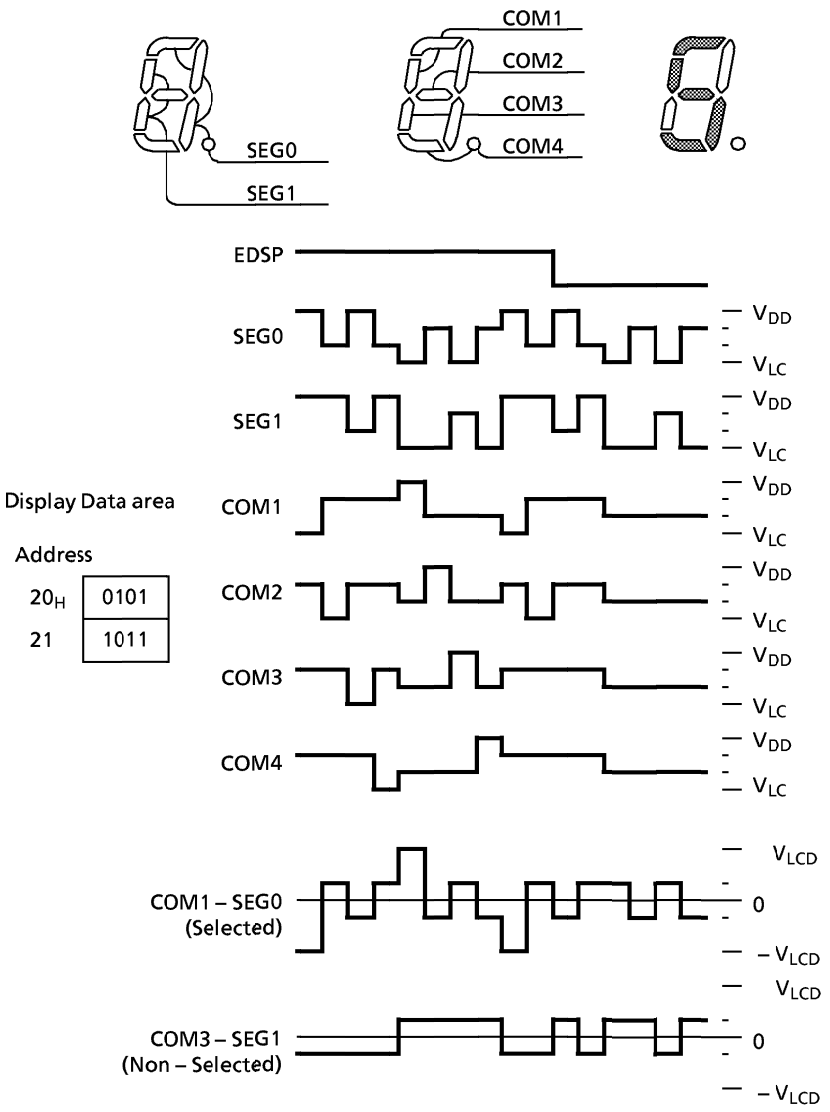


Figure 3-38. 1/4 Duty (1/3 Bias) Drive

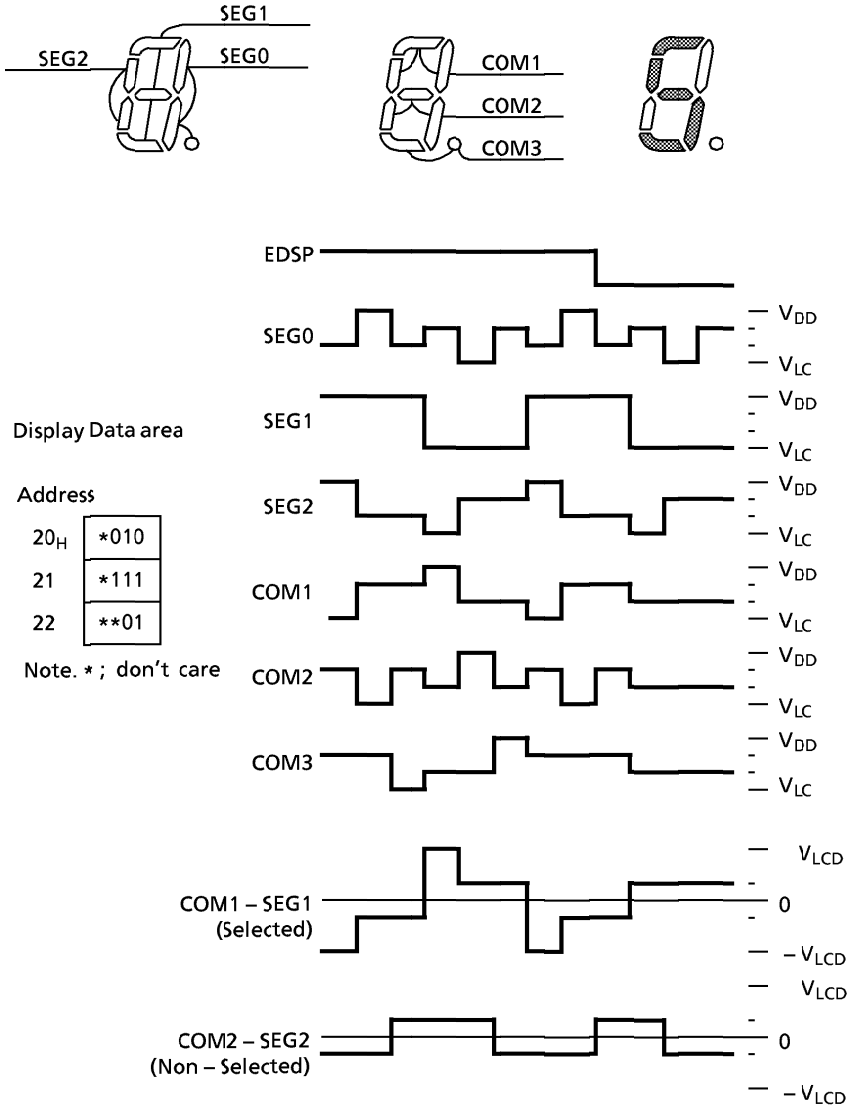


Figure 3-39. 1/3 Duty (1/3 Bias) Drive

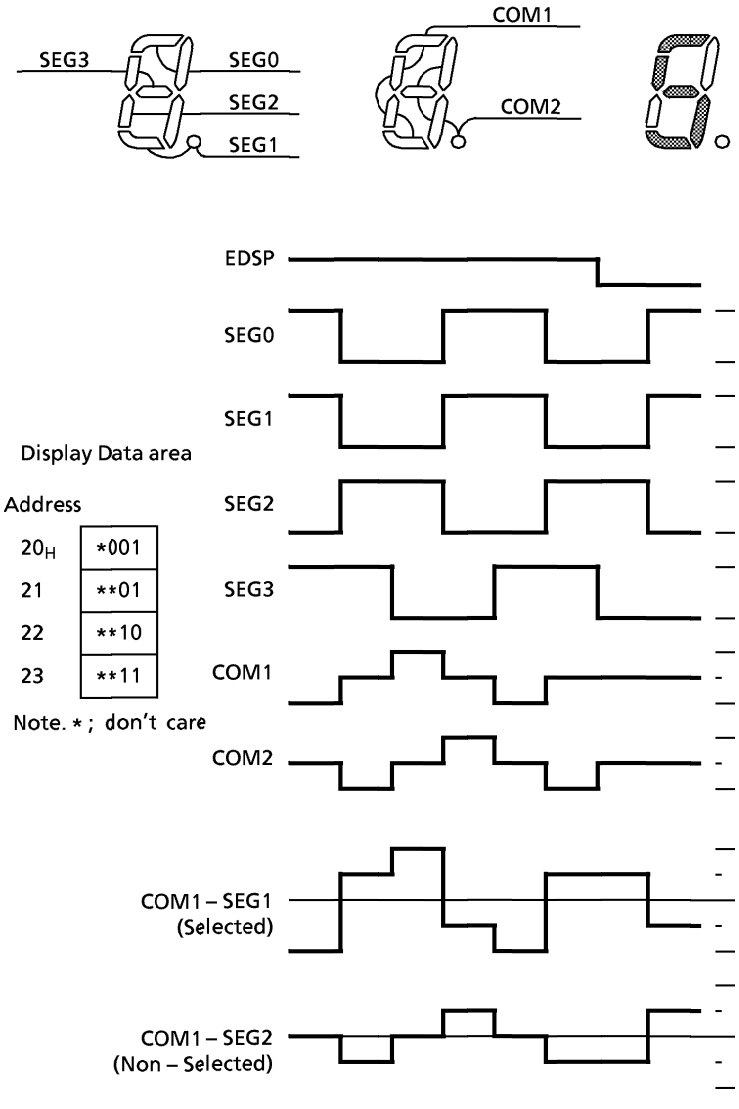


Figure 3-40. 1/2 Duty (1/2 Bias) Drive

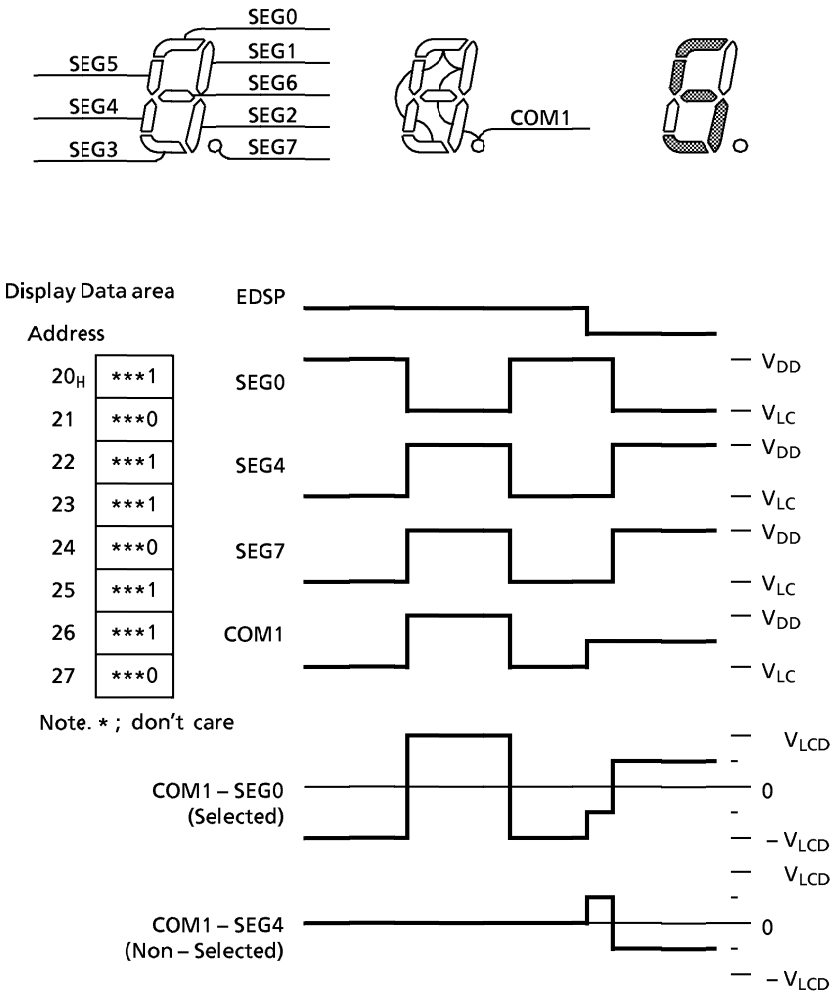
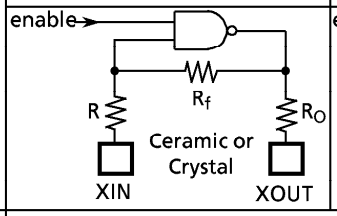
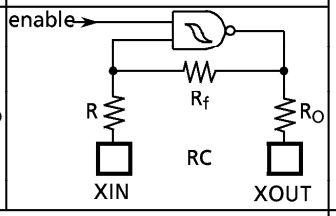
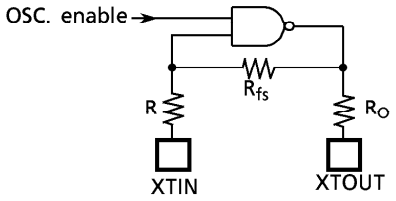
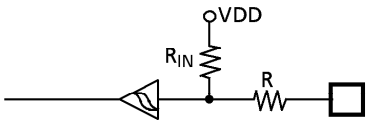
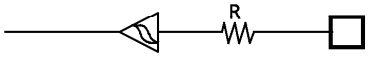


Figure 3-41. Static Drive

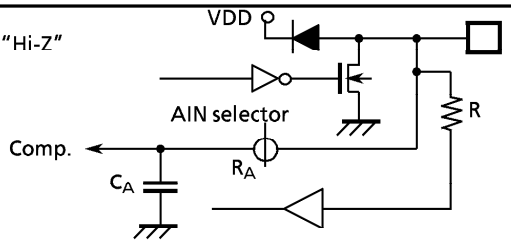
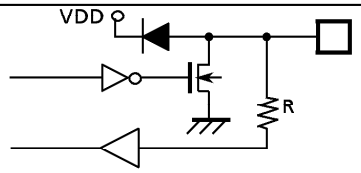
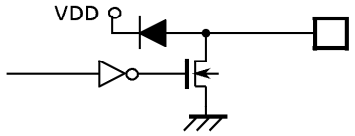
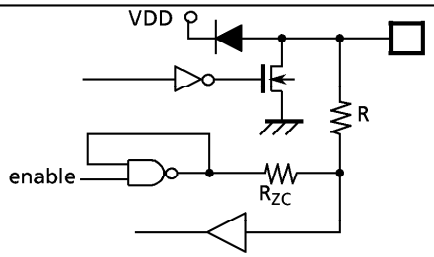
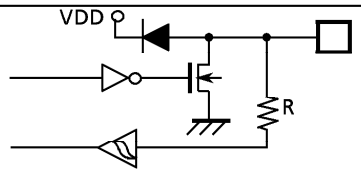
INPUT / OUTPUT CIRCUITRY

The input/output circuitries of the 47C222/422 are shown as below, any one of the circuitries can be chosen by a code (SA, SD) as a mask option.

(1) Control pins

CONTROL PIN	I/O	CIRCUITRY and CODE	REMARKS
XIN XOUT	Input Output	SA:  SD: 	Resonator connecting pins R = 1 kΩ (typ.) R_f = 1.5 MΩ (typ.) R_O = 2 kΩ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (R = 1 kΩ typ.) (R_{fs} = 6 MΩ typ.) (R_O = 220 kΩ typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor R_{IN} = 220 kΩ (typ.) R = 1 kΩ (typ.)
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input (Input)		Hysteresis input R = 1 kΩ (typ.)

(2) I/O ports

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE	REMARKS
R4	I/O	Initial "Hi-Z" 	Sink open drain output R = 1 kΩ (typ.) Analog input R_A = 5 kΩ (typ.) C_A = 12 pF (typ.)
R6 R7	I/O	Initial "Hi-Z" 	Sink open drain output R = 1 kΩ (typ.)
P5	Output	Initial "Hi-Z" 	Sink open drain output
R82	I/O	Initial "Hi-Z" 	Sink open drain output R = 1 kΩ (typ.) Zero-cross input R_{ZC} = 1 MΩ (typ.)
R80 R81 R83 R9	I/O	Initial "Hi-Z" 	Sink open drain output Hysteresis input R = 1 kΩ (typ.)

Note. 47C222 / 422 (SDIP) do not have port P5.

ELECTRICAL CHARACTERISTICS

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

PARAMETER	SYMBOL	PINS	RATINGS	UNIT
Supply Voltage	V_{DD}		– 0.3 to 6.5	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT}		– 0.3 to V_{DD} + 0.3	V
Output Current (Per 1 pin)	I_{OUT1}	Port R4, R7	30	mA
	I_{OUT2}	Port R5, R6, R8, R9	3.2	
Power Dissipation [T_{opr} = 70 °C]	ZI_{OUT1}	Port R4, R7	120	mW
	PD		400	
Soldering Temperature (time)	T_{sld}		260 (10 s)	°C
Storage Temperature	T_{stg}		– 55 to 125	°C
Operating Temperature	T_{opr}		– 30 to 70	°C

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0 V, T_{opr} = – 30 to 70 °C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Max.	UNIT
Supply Voltage	V_{DD}		fc = 8.0 MHz	2.7	5.5	V
			fc = 4.2 MHz	2.2		
			In the SLOW mode	2.2		
			In the HOLD mode	2.0		
Input High Voltage	V_{IH1}	Except Hysteresis Input	In the normal operating area	V_{DD} × 0.7	V_{DD}	V
	V_{IH2}	Hysteresis Input		V_{DD} × 0.75		
	V_{IH3}		In the HOLD mode	V_{DD} × 0.9		
Input Low Voltage	V_{IL1}	Except Hysteresis Input	In the normal operating area	0	V_{DD} × 0.3	V
	V_{IL2}	Hysteresis Input			V_{DD} × 0.25	
	V_{IL3}		In the HOLD mode		V_{DD} × 0.1	
Clock Frequency	fc	XIN, XOUT	V_{DD} = 2.7 to 5.5 V	0.4	8.0	MHz
			V_{DD} = 2.2 to 5.5 V		4.2	
			In the RC oscillation		2.5	
	fs	XTIN, XTOUT	V_{DD} = 2.2 to 5.5 V	30	34	kHz

D.C. CHARACTERISTICS (V_{SS} = 0 V, 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} = 5.5 V, V_{IN} = 5.5V / 0V	–	–	± 2	μA
	I_{IN2}	Open drain output ports					
Input Resistance	R_{IN}	$\overline{RESET}$		100	220	450	kΩ
Output Leakage Current	I_{LO}	Open drain output ports	V_{DD} = 5.5 V, V_{OUT} = 5.5 V	–	–	2	μA
Output Low Current	I_{OL}	Port R4, R7	V_{DD} = 4.5 V, V_{OL} = 1.0 V	7	10	–	mA
Output Low Voltage	V_{OL}	Port P5, R6, R8, R9	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	–	–	0.4	V
			V_{DD} = 2.2 V, I_{OL} = 20 μA	–	–	0.1	
Segment Output Low Resistance	R_{OS1}	SEG pin	V_{DD} = 5 V, V_{DD} – V_{LC} = 3 V	—	10 or 20	—	kΩ
Common Output Low Resistance	R_{OC1}	COM pin					
Segment Output High Resistance	R_{OS2}	SEG pin		—	70 or 200	—	kΩ
Common Output High Resistance	R_{OC2}	COM pin					
Segment/Common Output Registance	$V_{O2/3}$	SEG / COM pin		3.8	4.0	4.2	V
	$V_{O1/2}$			3.3	3.5	3.7	
	$V_{O1/3}$			2.8	3.0	3.2	
Supply Current (in the Normal mode)	I_{DD}		V_{DD} = 5.5 V, fc = 4 MHz	–	2	4	mA
			V_{DD} = 3.0 V, fc = 4 MHz	–	1	2	
			V_{DD} = 3.0 V, fc = 400 kHz	–	0.5	1	
Supply Current (in the SLOW mode)	I_{DDS}		V_{DD} = 3.0 V, fs = 32.768 kHz	–	20	40	μA
Supply Current (in the HOLD mode)	I_{DDH}		V_{DD} = 5.5 V	–	0.5	10	μA

Note 1. Typ. values show those at T_{opr} = 25 °C, V_{DD} = 5 V.

Note 2. Input Current I_{IN1} : The current through resistor is not included.

Note 3. Output Resistance R_{os}, R_{oc} ; Shows on-resistance at the level switching.

Note 4. $V_{O2/3}$; Shows 2/3 level output voltage, when the 1/4 or 1/3 duty LCD is used.
$V_{O1/2}$; Shows 1/2 level output voltage, when the 1/2 duty or static LCD is used.
$V_{O1/3}$; Shows 1/3 level output voltage, when the 1/4 or 1/3 duty LCD is used.

Note 5. Supply Current I_{DD}, I_{DDH} : V_{IN} = 5.3 V / 0.2 V (V_{DD} = 5.5 V), 2.8 V / 0.2 V (V_{DD} = 3.0 V)
Supply Current I_{DDS} ; V_{IN} = 2.8 V / 0.2 V. Low frequency clock is only osillated .

Note 6. When using LCD, it is necessary to consider values of Ros 1/2 and Roc 1/2.

Note 7. Times fou SEG/COM output switching on ;Ros1, Roc1 : 2/fc (s)
Ros2, Roc2 : 1/(n · f_F) (1/n ; duty, f_F : frame frequency)

A / D CONVERSION CHARACTERISTICS (T_{opr} = – 30 to 70 °C)

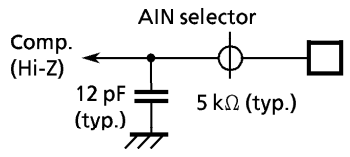
PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Analog Reference Voltage Range	$\triangle V_{AREF}$	$V_{DD} - V_{SS}$	2.7	–	–	V
Analog Input Voltage	V_{AIN}		V_{SS}	–	V_{DD}	V
Analog Supply current	I_{REF}		–	0.5	1.0	mA
Nonlinearity Error		V_{DD} = 2.7 V to 5.5 V V_{SS} = ± 0.000 V	–	–	± 1	LSB
Zero Point Error			–	–	± 1	
Full Scale Error			–	–	± 1	
Total Error			–	–	± 2	

A.C. CHARACTERISTICS (V_{SS} = 0 V, Topr = – 30 to 70 °C)

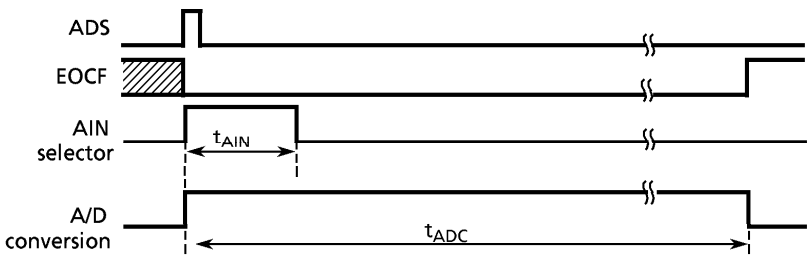
PARAMETER	SYMBOL	CONDITIONS		Min.	Typ.	Max.	UNIT
Instruction Cycle Time	tcy	In the normal mode	V_{DD} = 2.7 to 5.5V	1.0	–	20	μs
			V_{DD} = 2.2 to 5.5V	1.9			
			RC oscillation	3.2			
		In the SLOW mode		235		267	
High level clock pulse width	t_{WCH}	For external clock (XIN input)	$V_{DD} \geqq 2.7$ V	60	–	–	ns
			$V_{DD} < 2.7$ V	120			
Low level clock pulse width	t_{WCL}		$V_{DD} \geqq 2.7$ V	60			
			$V_{DD} < 2.7$ V	120			
A/D Conversion Time	t_{ADC}			–	24 tcy	–	μs
A/D Sampling Time	t_{AIN}			–	2 tcy	–	
Shift data Hold Time	t_{SDH}			0.5 tcy – 300	–	–	ns

Note 1 A/D conversion timing :

Internal circuit for pins AIN0 to 7

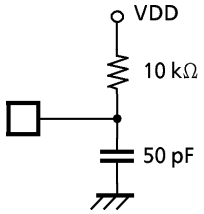


A/D conversion timing

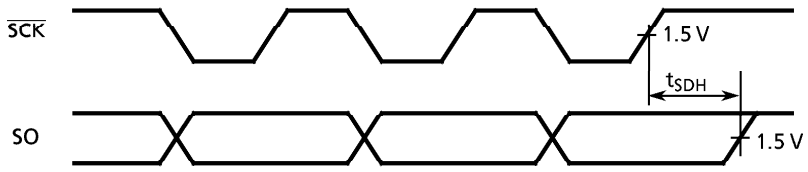


* Electrical change inust be loaded into the buit-in condensen during t_{AIN} for normal A/D conversion.

Note2 Shift data Hold Time :
External circuit for pins
$\overline{SCK}$ and SO



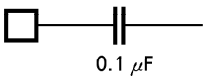
Serial port (completed of transmission)

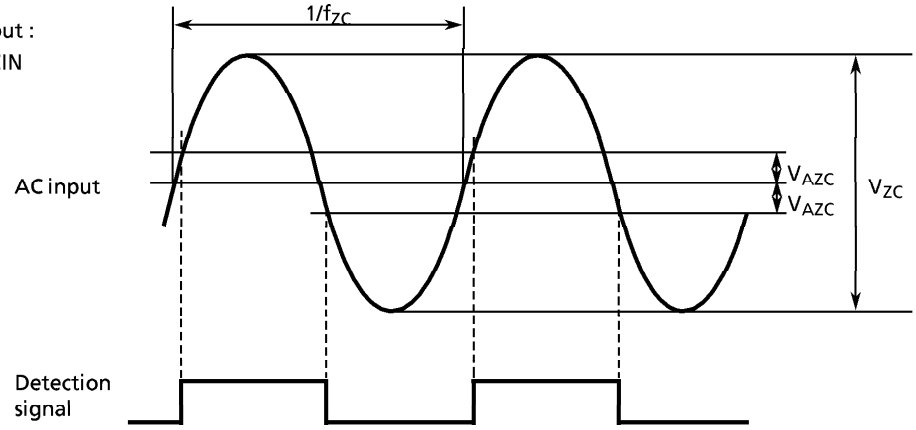


ZERO-CROSS DETECTION CHARACTERISTICS (Vss = 0V, Topr = – 30 to 70℃)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Zero-cross Input Voltage	V_{ZC}	AC coupling (C = 0.1 μF)	1.0	—	3.0	V_{P-P}
Zero-cross Accuracy	V_{AZC}	f_{ZC} = 50 to 60 Hz (sine curve)	—	—	± 135	mV
Zero-cross input frequensy	f_{ZC}		40	—	1000	Hz

Note 3 Zero-cross detection input :
External circuit for pin ZIN





RECOMMENDED OSCILLATING CONDITIONS (V_{SS} = 0 V, V_{DD} = 2.2 to 5.5 V, T_{opr} = – 30 to 70 °C)

(1) 6 MHz

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) 4 MHz

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)	$C_{XIN} = C_{XOUT}$ = 20 pF

(3) 400 kHz

Ceramic Resonator

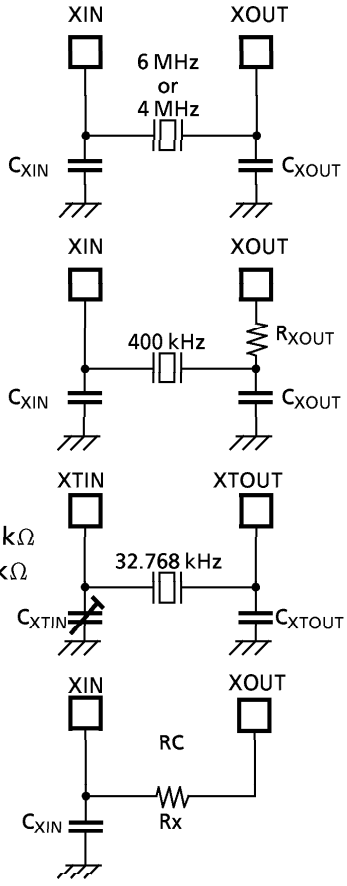
CSB400B (MURATA)	$C_{XIN} = C_{XOUT}$ = 220 pF, R_{XOUT} = 6.8 kΩ
KBR-400B (KYOCERA)	$C_{XIN} = C_{XOUT}$ = 100 pF, R_{XOUT} = 10 kΩ
EFOA400K04B (NATIONAL)	$C_{XIN} = C_{XOUT}$ = 470 pF, R_{XOUT} = 0 Ω

(4) 32.768 kHz

Crystal Oscillator C_{XTIN}, C_{XTOUT} ; 10 to 33 pF (Note)

(5) RC Oscillation (V_{SS} = 0 V, V_{DD} = 5.0 V, T_{opr} = 25 °C)

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



Note : In order to get the accurate oscillation frequency, the adjustment of capacitors must be required.

TYPICAL CHARACTERISTICS