CMOS 4-BIT MICROCONTROLLER

TMP47C1260N, TMP47C1660N
TMP47C1260F , TMP47C1660F

The 47C1260/1660 are the high speed and high performance 4-bit single chip microcomputer, with large capacity ROM and RAM, based on the 47C660/860 (TLCS-470 series).

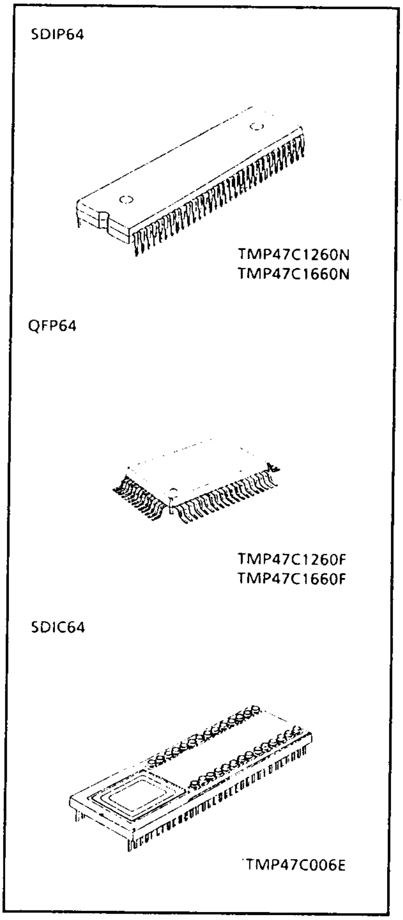
PART No	ROM	RAM	PACKAGE	PIGGYBACK
TMP47C1260N	12288 x 8-bit	768 x 4-bit	SDIP64	TMP47C006E
TMP47C1260F			QFP64	
TMP47C1660N	16384 x 8-bit		SDIP64	
TMP47C1660F			QFP64	

FEATURES

- 4-bit single chip microcomputer
- Instruction execution time: 1.3μs(at 6MHz), 244μs (at 32.8KHz)
- 105 basic instructions
- Table look-up instructions
- 5-bit to 8-bit data conversion instruction
- 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 (56 pins)

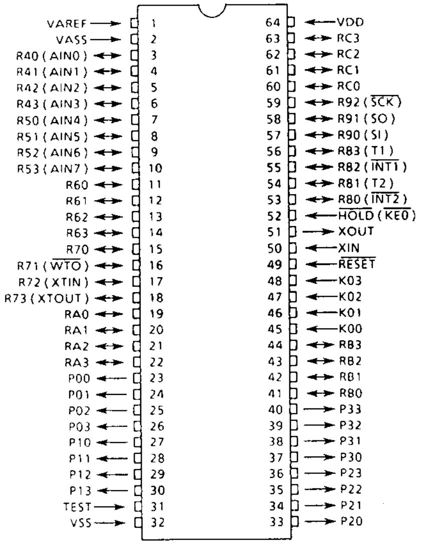
• Input	2 ports	5 pins
• Output	4 ports	16 pins
• I/O	9 ports	35 pins

- 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
  - External/internal clock, and leading/trailing edge shift, 4/8-bit mode
- 8-bit successive approximate type A/D converter
  - With sample and hold
  - 8 analog inputs
  - Conversion time : 32 μs(at 6 MHz)
- Remote control signal pre-processing capability
- High current outputs
  LED direct drive capability (typ. 20mA × 8 bits)
- Dual-clock operation
  High-speed/Low-power-consumption operating mode
- Hold function
  Battery/Capacitor back-up
- Real Time Emulator : BM47C1660

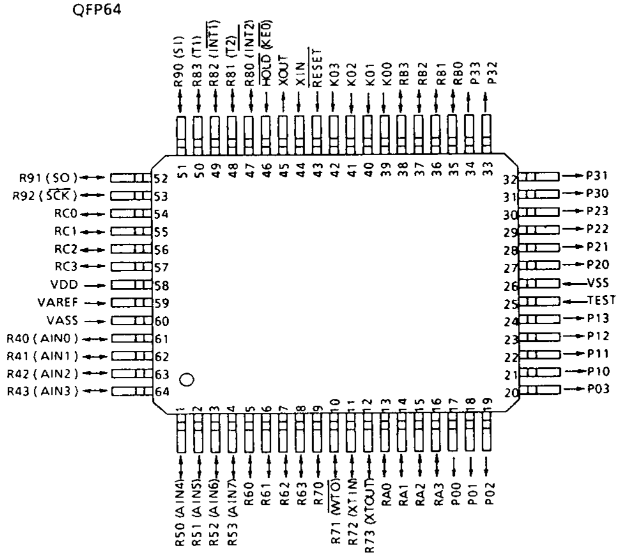


PIN ASSIGNMENT (TOP VIEW)

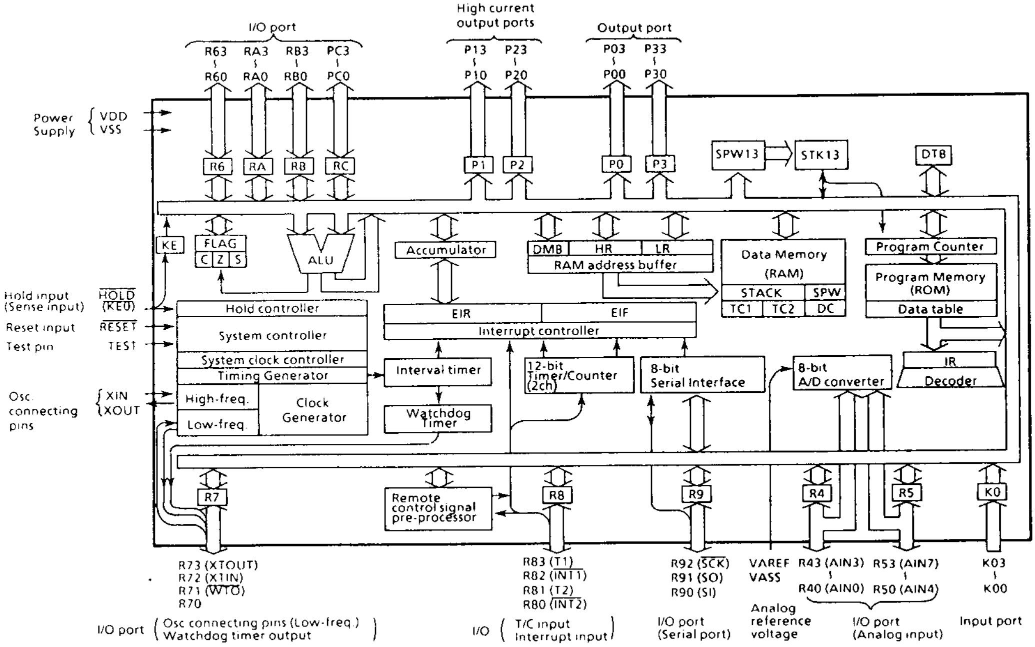
SDIP64



QFP64



BLOCK DIAGRAM



PIN FUNCTION

PIN NAME	Input/Output	FUNCTIONS	
K03 - K00	Input	4-bit input port	
P03 - P00	Output	4-bit output port with latch	
P13 - P10	Output	4-bit output port with latch. 8-bit data are output by the 5-bit to 8-bit data conversion instruction [OUTB @HL].	
P23 - P20			
P33 - P30	Output	4-bit output port with latch	
R53 (AIN7) - R40 (AIN0)	I/O (Input)	4-bit I/O port with latch. When using as input port, watchdog timer output or analog input, the latch must be set to "1". Set to Dual-clock operating mode, when R73,R72 pin use as clock generator. Can be set, cleared, and tested for each bit as specified by L register indirect addressing bit manipulation instructions.	A/D converter analog input
R63 - R60	I/O		
R73 (XTOUT)	I/O (Output)		Resonator connecting pin (Low-freq.). For inputting external clock, XTIN is used and XTOUT is opened.
R72 (XTIN)	I/O (Input)		
R71 ($\overline{\text{WTO}}$)	I/O (Output)		Watchdog timer output
R70	I/O		
R83 (T1)	I/O (Input)	4-bit I/O port with latch. When using as input port, external interrupt input pin, or timer/counter external input pin, the latch must be set to "1".	Timer/Counter 1 external input
R82 ($\overline{\text{INT1}}$)			External interrupt 1 input
R81 (T2)			Timer/Counter 2 external input
R80 ($\overline{\text{INT2}}$)			External interrupt 2 or REMO-CON input
R92 ($\overline{\text{SCK}}$)	I/O(I/O)	3-bit I/O port with latch. When using 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
RA3 - RA0	I/O	4-bit I/O port with latch. When using as input port, the latch must be set to "1".	
RB3 - RB0			
RC3 - RC0			
XIN	Input	Resonator connecting pin (High-frequency). For inputting external clock, XIN is used and XOUT is opened.	
XOUT	Output		
$\overline{\text{RESET}}$	Input	Reset signal input	
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input (Input)	HOLD request/release signal input	Sence input
TEST	Input	Test pin for out-going test. Be opened or fixed to low level.	
VDD	Power supply	+ 5V	
VSS		0V (GND)	
VAREF		A/D converter analog reference voltage (High)	
VASS		A/D converter analog reference voltage (Low)	

OPERATIONAL DESCRIPTION

1. SYSTEM CONFIGURATION

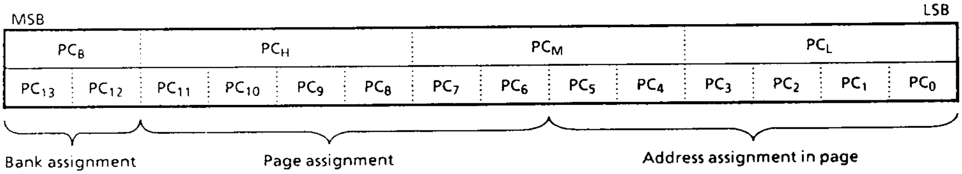
(1) Program Counter (PC)
(2) Program Memory (ROM), and Data Table Bank Selector (DTB)
(3) H Register, L Register, and Data Memory Bank Selector (DMB)
(4) Data Memory (RAM), and Stack for PC13 (STK13)
a. Stack
b. Stack Pointer Word (SPW)
c. Data Counter (DC)
(5) ALU, Accumulator
(6) Flags
(7) Clock Generator, Timing Generator
(8) System Clock Controller
(9) Input/Output Ports
(10) Interval Timer
(11) Timer/Counters (TC1, TC2)
(12) A/D converter
(13) Remote control signal pre-process circuit
(14) Serial Interface
(15) Watchdog Timer
(16) Interrupt Controller
(17) Hold Controller
(18) Reset Circuit

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

2. INTERNAL CPU FUNCTION

2.1 Program Counter (PC)

The program counter is a 14-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 acknowledged, the specified values listed in Table 2-1 are set to the PC. The PC is initialized to "0" during reset.



MSB LSB

PC_B		PC_H				PC_M				PC_L			
PC_{13}	PC_{12}	PC_{11}	PC_{10}	PC_9	PC_8	PC_7	PC_6	PC_5	PC_4	PC_3	PC_2	PC_1	PC_0

Figure 2-1. Configuration of Program Counter

The PC can directly address an 16384-byte address space. However, with the short/middle branch and subroutine call instructions, the following points must be considered:

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

(2) Middle branch instruction [BS a]
In [BS a] instruction execution, when the branch condition is satisfied, the value specified in the instruction is set to the lower 12 bits of the PC. That is, [BS a] becomes the in-bank branch instruction.When first byte or second byte of this instruction is stored at the last address of the bank, the upper 2 bits of the PC point the next bank, so that branch is made to the next bank.

(3) 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.

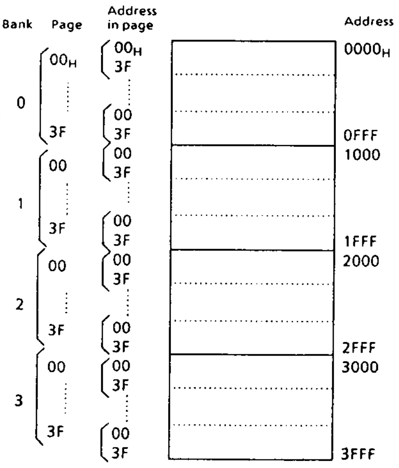


Figure 2-2. Configulation of Program

Instruction /Operation	Condition		PC13	PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
BSL a	SF = 1 (Branch condition is satisfied)		Immediate data specified by the instruction													
	SF = 0 (Branch condition is not satisfied)		+ 3													
BS a	SF = 1	Lower 12-bit address ≠ FFE, FFFH	Hold		Immediate data specified by the instruction											
		Lower 12-bit address = FFE, FFFH (The bank end address)	+ 1		Immediate data specified by the instruction											
	SF = 0		+ 2													
BSS a	SF = 1	Lower 6-bit address ≠ 3FH	Hold								Immediate data specified by the instruction					
		Lower 6-bit address = 3FH (The page end address)	+ 1								Immediate data specified by the instruction					
	SF = 0		+ 1													
CALL a			0	0	0	Immediate data specified by the instruction										
CALLS a			0	0	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 of the instruction													
Interrupt acceptance			0	0	0	0	0	0	0	0	0	0	Interrupt vector			0
RESET			0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2-1. Status Change of Program Counter

The address which can be specified by the instruction consists of 11 bits and the upper 3 bits of the PC is always "0". Therefore, the entry address of the subroutine should be within an address range of 0000_H through $07FF_H$.

2.2 Program Memory (ROM), and Data Table Bank Selector (DTB)

Programs and fixed data are stored in the program memory. The next instruction to be executed is read from the address pointed by the contents of the PC. The fixed data can be read by using the table look-up instructions or 5-bit to 8-bit data conversion instruction.

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

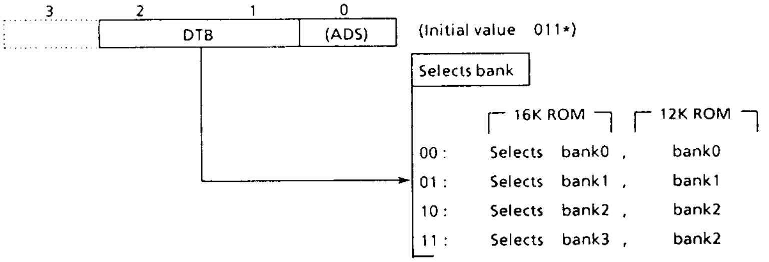
The table look-up instructions read the lower and upper 4 bits of the fixed data stored at the address specified in the data table bank selector (DTB) and the data counter (DC) to place them into the accumulator. [LDL A,@DC] instruction reads the lower 4 bits of fixed data and [LDH A,@DC +] instruction reads the upper 4 bits. The DTB is a 2-bit register to select the bank of the program memory. The Data Counter is a 12-bit register, and it can specify an address in the bank.

Example for the 47C1660 : When [LDL A,@DC] instruction is executed with the DTB value being 3, the DC value being $7A0_H$ and the contents of program memory address $37A0_H$ being 58_H, "8" is stored in the accumulator; when [LDH A,@DC +] instruction is executed "5" is stored in the accumulator and the DC value is incremented to $7A1_H$.

Note: When [LDH A, @DC +] instruction is executed with the DC value being FFF_H, the DTB value is not incremented.

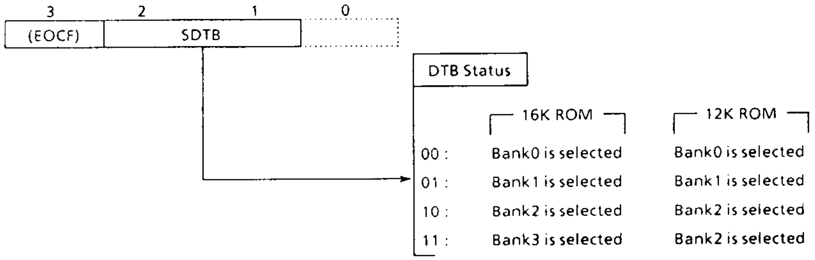
Data Table Bank Selecor (DTB)

① Command register (Port address OP13)



	16K ROM		12K ROM
00 :	Selects	bank0 ,	bank0
01 :	Selects	bank1 ,	bank1
10 :	Selects	bank2 ,	bank2
11 :	Selects	bank3 ,	bank2

② Status register (Port address IP13)



	16K ROM	12K ROM
00 :	Bank0 is selected	Bank0 is selected
01 :	Bank1 is selected	Bank1 is selected
10 :	Bank2 is selected	Bank2 is selected
11 :	Bank3 is selected	Bank2 is selected

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

The 5-bit to 8-bit data conversion instruction reads the fixed data (8 bits) from the data conversion table in the program memory to send the upper 4 bits to port P2 and the lower 4 bits to port P1. The data conversion table is located in the last 32-byte space (addresses 2FE0 throgh 2FFF$_H$ for the 47C1260, 3FE0 through 3FFF$_H$ for the 47C1660) in the program memory with the lower address consisting of the 5 bits obtained by linking the data memory contents specified by the HL register pair and the content of the carry flag. This instruction is suitable for such applications as converting BCD data into an output code to the 7-segment display element.

Example for the 47C1660 : The following shows that the BCD data at address 2F$_H$ in data memory is converted into the 7-segment code (anode common LED) to be output to ports P2 and P1.

```
LD      HL, #2FH ;  HL ← 2FH  (Data memory address is set)
TEST    CF       ;  CF ← 0    (The data conversion table is specified within
                               addresses 3FE0H through 3FEFH)
OUTB    @HL
  :
ORG     3FE0H    ;  Data conversion table
DATA    0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0D8H, 80H, 98H
```

2.2.1 Program Memory Map

Figure 2-3 shows the program memory map. Address 0000 through 0086$_H$ and 2FE0 through 2FFF$_H$ for the 47C1260, 0000 through 0086$_H$ and 3FE0H through 3FFF$_H$ for the 47C1660, of the program memory are also used for special purposes.

2.2.2 Program Memory Capacity

The 47C1260 contains a program memory (masked ROM) of 12288 × 8 bit (addresses 0000 through 2FFF$_H$). The 47C1660 contains a program memory (masked ROM) of 16384 × 8 bits (addresses 0000 through 3FFF$_H$).

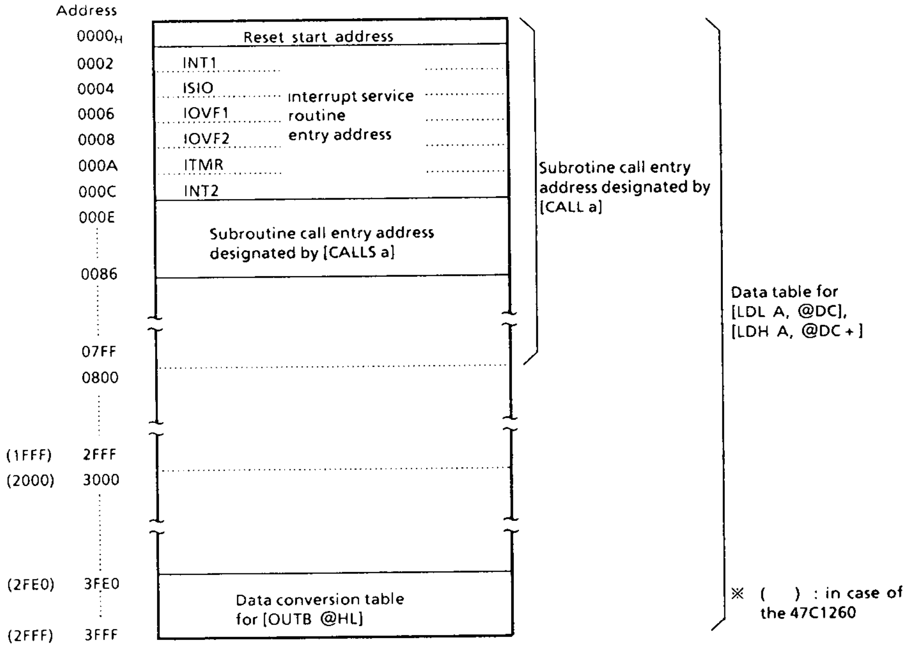

Address
0000H
0002
0004
0006
0008
000A
000C
000E
0086
07FF
0800
(1FFF) 2FFF
(2000) 3000
(2FE0) 3FE0
(2FFF) 3FFF
Reset start address
INT1
ISIO
IOVF1
IOVF2
ITMR
INT2
interrupt service routine entry address
Subroutine call entry address designated by [CALLS a]
Subrotine call entry address designated by [CALL a]
Data table for [LDL A, @DC], [LDH A, @DC+]
Data conversion table for [OUTB @HL]
※ ( ) : in case of the 47C1260

Figure 2-3. Program Memory Map

Note : *In case of the 47C1260, 5-bit to 8-bit data conversion table is at address 2FE0 through 2FFF$_H$. When the piggyback is used in order to evaluate the 47C1260, it is necessary to also place a conversion table at address 3FE0 through 3FFF$_H$.*

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

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 data memory 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 data memory consists of three banks. Bank 0, 1 and 2. The data memory bank selector (DMB) is a 2-bit register to specify a data memory bank. During reset, the DMB is initialized to "0". The DMB is set by the [CLR DMB0] , [SET DMB0] [CLR DMB1] , [SET DMB1] , [LD DMB, #K] , [MOV A, DMB] or [LD DMB, @ HL] instruction. The currently selected data memory bank can be known by executing the [TEST DMB0] or, [TESTP DMB0], [TEST DMB1] or [TESTP DMB1] instruction. The L register has the automatic 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 [SET @L], [CLR @L], or [TEST @L] instruction, the L register is also used to specify the bits corresponding to I/O port pins R73 through R40 (the indirect addressing of port bits by the L register).

Example 1 : To write immediate values "5" and "F$_H$" to data memory (bank 0) addresses 10$_H$ and 11$_H$.

```
          LD        DMB,  #0        ; DMB ← 0
          LD        HL,   #10H      ; HL ← 10H
```

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

Example 2 : The output latch of R71 pin is set to "1" by the L register indirect addrressing bit manipulation instruction.

```
LD     L, #1101B       ; Sets R71 pin address to L register
SET    @L              ; R71 ← 1
```

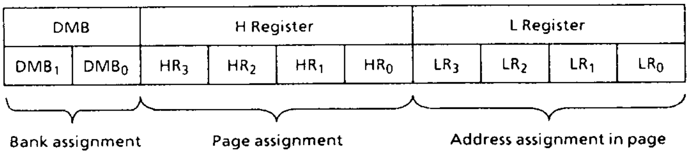


Figure 2-4. Configuration of H, L registers and DMB

2.4 Data Memory (RAM), and Stack for PC13 (STK 13)

The data memory stores user-processed data. One page of this memory is 16 words long (1 word = 4 bits). It has three banks. The data memory is addressed in one of three ways (addressing modes) :

(1) Register-indirect addressing mode
In this mode, a bank is specified by the DMB, a page 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
An address in the bank is directly specified by the 8 bits of the second byte (operand) in the instruction field. The bank is specified by the DMB.

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

(3) Zero-page addressing mode
An address in zero-page of bank 0 (addresses 00_H through $0F_H$) by the lower 4 bits of the second byte (operand) in the instruction field.

Example: ST #3, 05H ; RAM [05_H] ← 3

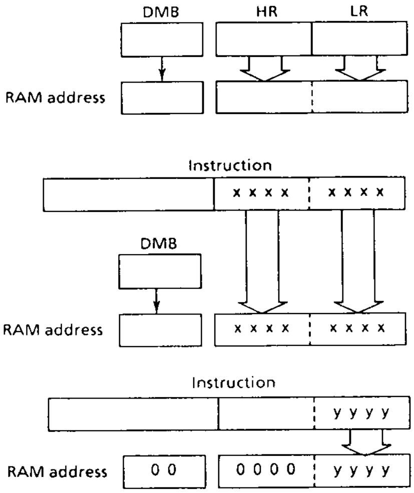
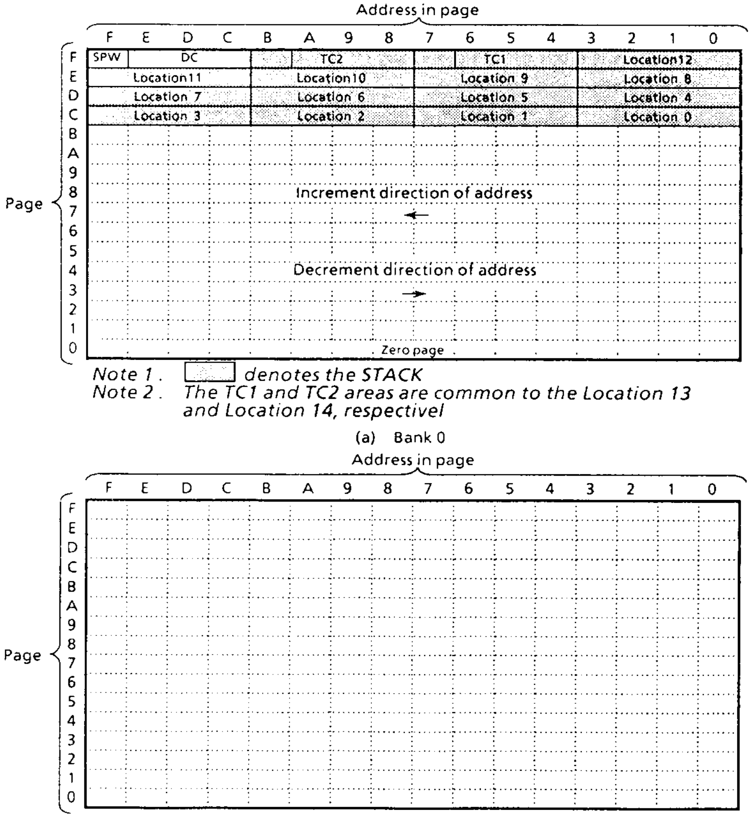


Figure 2-5. Addressing mode

2.4.1 Data Memory Map

Figure 2-6 shows the data memory map. The data memory is also used for the following special purposes: Note that this special function area is provided only on bank 0.

(1) Stack
(2) Stack pointer word (SPW)
(3) Data counter (DC)
(4) Count registers of the Timer/Counters(TC1, TC2)
(5) Zero-page



Note 1. ▨ denotes the STACK
Note 2. The TC1 and TC2 areas are common to the Location 13 and Location 14, respectivel

(a) Bank 0

(b) Bank 1 and Bank 2

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. Stacks have a maximum of 15 levels (locations 0 -14). The lower 13 bits (PC_0-PC_{12}) of the program counter and the flag area are in bank 0 (C0-FB_H) of the data memory. The area for the highest order bit (PC_{13}) of the program counter is provided in a Stack 13 (STK13). One stack level consists of a 4-word data memory and 1 bit from STK13. Locations 13 and 14 in the data memory are also used as count registers (TC1, TC2) for the Timer/Counter described later.

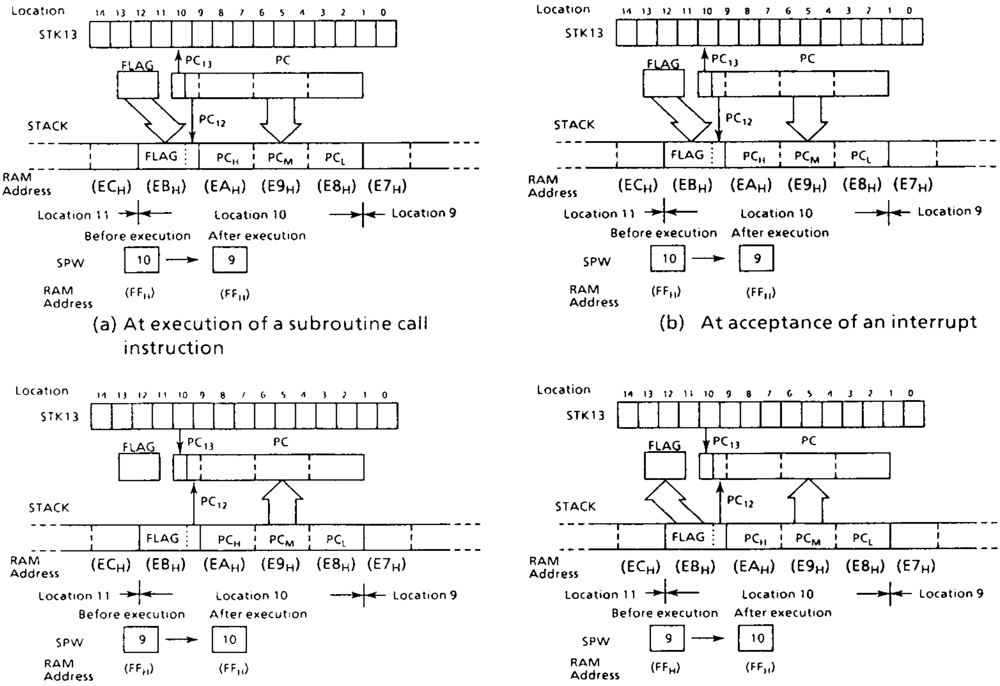
The save/restore locations in the stack are determined by the stack pointer word (SPW). The SPW is automatically decremented after save and incremented before restore. That is, the value of the stack pointer word indicates the stack location number for the next save.

(2) Stack Pointer Word (SPW)

Address FF_H in the data memory (bank 0) 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 by the count registers of 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 with 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 arera is in an address range 00_H through CF_H, up to location 4 of the stack are usable. If an interrupt is accepted with location 4 already used, the user-processed data stored in addresses CC_H through CFH corresponding to the location 3 area is lost.)The SPW is not initialized in terms of 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
LD      DMB,  #0H       ; DMB ← 0H
ST      A,   0FFH       ; RAM [FF] ← Acc
```



(a) At execution of a subroutine call instruction

(b) At acceptance of an interrupt

(c) At execution of a subroutine return

(d) At execution of an interrupt return instruction

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

(3) Stack 13 (STK13)

The Stack 13 (STK13) is a 1 bit × 15 levels (locations 0-14) register SPW13 specifies the locations when only this register is accessed individually. STK13 locations are specified according to the contents of SPW (address FF_H, bank 0 of the data memory) for subroutine call and return execution, and when interrupts are acknowledged. The contents of SPW13 do not change at this time. The following 4 commands are used to load data or to read data from specific locations in STK13.

① SET STK13

SPW13 ← Acc
STK13 [SPW13] ← 1

The contents of the accumulator are loaded into SPW13 and "1" is loaded into the location indicated by SPW13.

② CLR STK13

SPW13 ← Acc
STK13 [SPW13] ← 0

The contents of the accumulator are loaded into SPW13 and "0" is loaded into the location indicated by SPW13.

③ MOV A, SPW13

SPW13 ← Acc

The contents of the accumulator are loaded into SPW13.

④ MOV STK13, A

Acc_0 ← STK13 [SPW13], Acc_{3-1} ← 000_B

The contents of the location specified by SPW13 are loaded to the least significant bit of the accumulator. The upper 3 bits of the accumulator are cleared to "0".

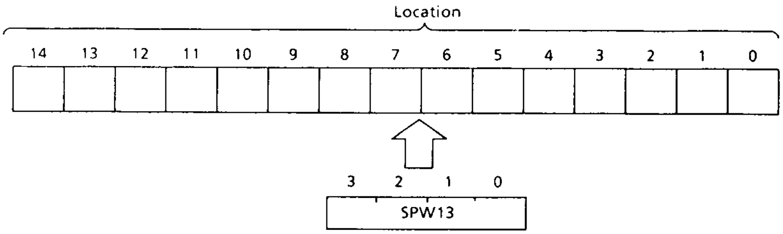


Figure 2-8. Configulation of stack 13 (STK13)

(4) Data counter (DC)

The data counter is a 12-bit counter 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 DC is assigned with a RAM address in unit of 4bits. Therefore, the RAM manipulation instruction is used to set the initial value or read the contents of the DC.

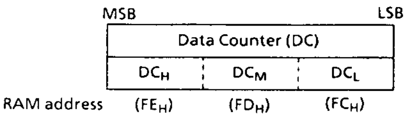


Figure 2-9. Data Counter

Example: To set the DC to 780_H

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

(5) Count registers of the Timer/Counter (TC1, TC2)

The 47C1260/1660 have two 12-bit Timer/Counters. The count register of the Timer/Counter is assigned with RAM addresses in unit of 4 bits, so that the initial value setting or contents reading is performed by using RAM instruction.The count registers are shared by 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 (bank 0) locations at addresses $F7_H$ and FB_H can be used to store the user-processed data.

	MSB		LSB
	Timer/Counter1 (TC1)		
	$TC1_H$	$TC1_M$	$TC1_L$
RAM address	($F6_H$)	($F5_H$)	($F4_H$)

MSB		LSB
Timer/Counter 2 (TC2)		
$TC2_H$	$TC2_M$	$TC2_L$
(FA_H)	($F9_H$)	($F8_H$)

Figure 2-10. Count Registers of Timer/Counters (TC1,TC2)

(6) Zero page

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

Example: To write "8" to address 09_H if bit 2 at address 04_H in the data memory (bank 0) is "1".

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

2.4.2 Data Memory Capacity

The 47C1260/1660 have 768 × 4 bit of data memory (RAM) , 256 × 4 bits (addresses 00_H through FF_H) on each of banks (bank0, bank1 and bank2). When power-on is performed, the contents of the data memory become unpredictable, so that they must be initialized by the initialize routine.

Example: To clear RAM

```
                LD     HL,   #00H        ; HL ← 00H
      SCLR0 :   LD     DMB,   #0         ; DMB ← 0
      SCLR1 :   ST     #0,   @HL+        ; RAM [HL] ← 0,  LR ← LR+1
                B      SCLR1
                LD     DMB,   #1         ; DMB ← 1
      SCLR2 :   ST     #0,   @HL+        ; RAM [HL] ← 0,  LR ← LR+1
                B      SCLR2
                LD     DMB,   #2         ; DMB ← 2
      SCLR3 :   ST     #0,   @HL+        ; RAM [HL] ← 0,  LR ← LR+1
                B      SCLR3
                ADD    H,   #1           ; HR ← HR+1
                B      SCLR0
```

2.5 ALU and Accumulator

2.5.1 Arithmetic/Logic Unit (ALU)

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

(1) Carry information (C)
The carry information indicatets a carry-out from the most significant bit in an addtion. 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".

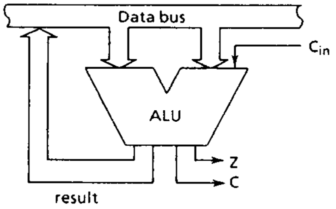


Figure 2-11. ALU

Note. Cin indicates the carry input specified by instruction

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

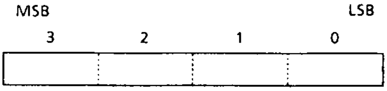


Figure 2-12. Accumulator

2.6 Flags

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

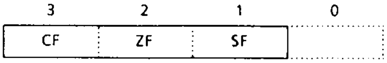


Figure 2-13. Flags

(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 (Cin) to the ALU to hold the carry information.

(2) Compare instructions [CMPR A,@HL],[CMPR A,#k]

The CF holds the carry information (non-borrow).

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

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

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

(3) Status flag (SF)
The SF provides the branch condition for a branch instruction. Branch is performed when the SF 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 instruction, this instruction becomes a conditional branch instruction. During reset, the SF is initialized to "1", other flags are not affcted.

Example : When the following instructions are executed with the accumulator, H register, L register, data memory address 07_H, and carry flag are "C_H", "0_H", "7_H", "5_H", 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 Clock Generator, Timing Generator and System Clock Controller

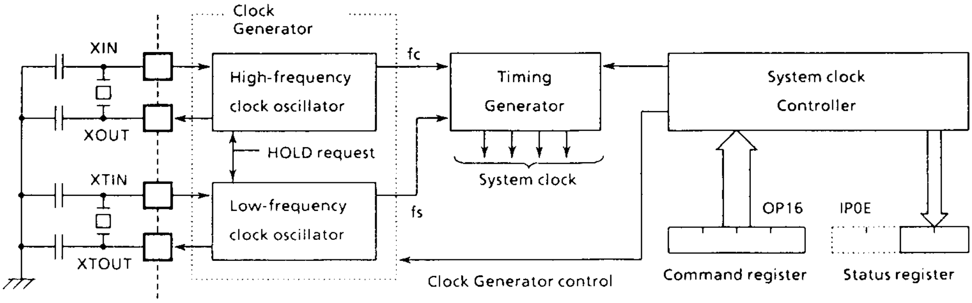
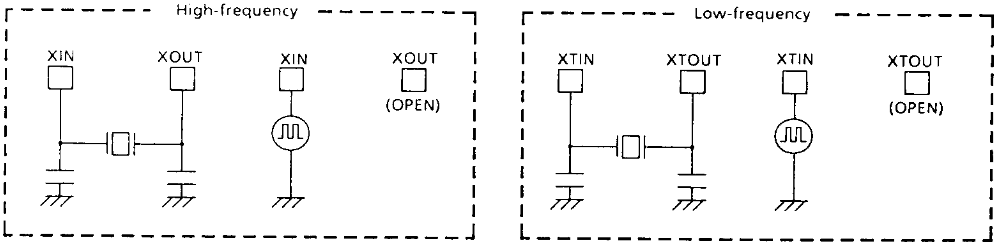


Figure 2-14. Clock Generator, Timing Generator and System Clock Controller

2.7.1 Clock Generator (CG)

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. The system clock can also be obtained from the external oscillator.



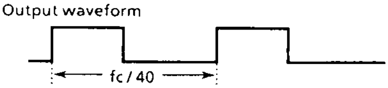
(a) Crystal or ceramic resonator (b) External oscillator (c) Crystal (d) External

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 interrupt disabled and timer/counters stopped and monitoring this pulse. With a system requiring the oscillation frequency adjustment, the adjusting program must be created beforehand.

Example : To output the high-frequency oscillation frequency adjusting monitor pulse to R70 pin.

```
SFCCHK :    SET     %OP07, 0
            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.

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 2-16 shows the operating mode transition diagram. Figure 2-17 shows the command register. Figure 2-18 shows the status register.

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-frerquency 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.

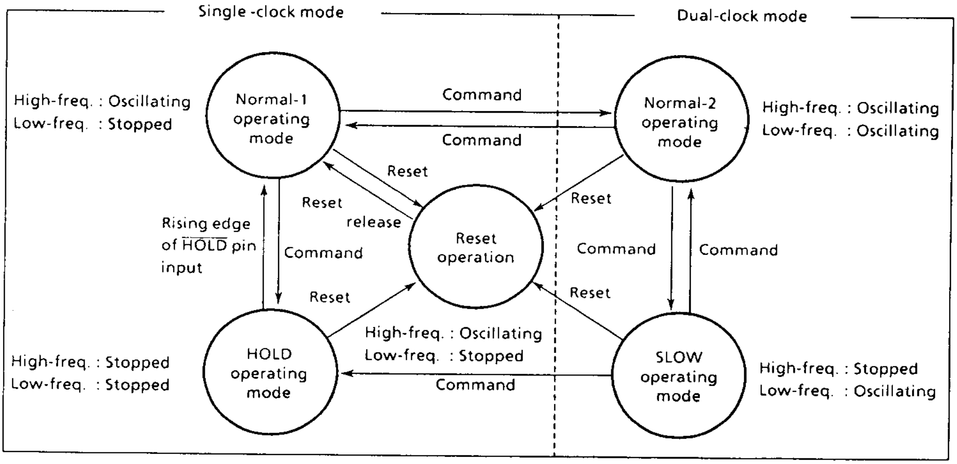


Figure 2-16. Operating Mode Transition Diagram

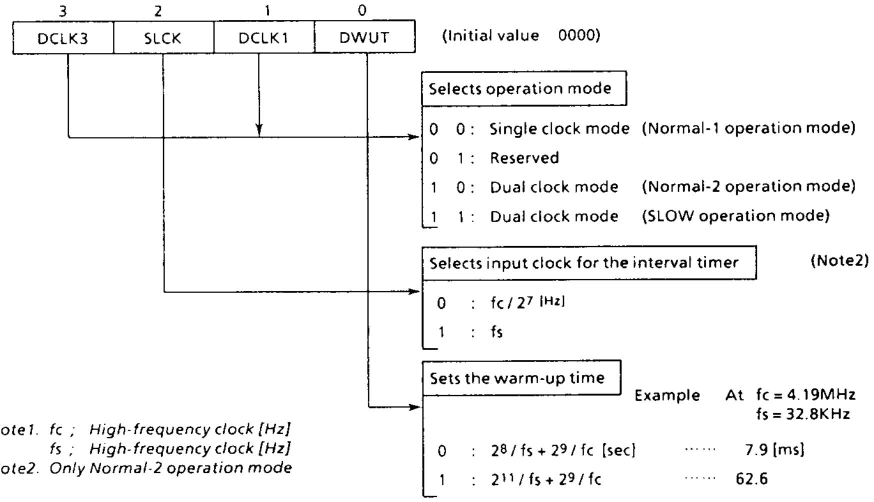
Note 1. In the HOLD and SLOW operating modes, the power consumed by the oscillator and the internal hardware is reduced. However, the power for the pin interface (depending on the external circuitry and program) is not directly associated with the low-power consumption operation. This must be considered in system design as well as interface circuit design.

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

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).

Note. 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 [sec].

System clock control command register (Port address OP16)


3
2
1
0
DCLK3
SLCK
DCLK1
DWUT
(Initial value 0000)
Selects operation mode
0 0 : Single clock mode (Normal-1 operation mode)
0 1 : Reserved
1 0 : Dual clock mode (Normal-2 operation mode)
1 1 : Dual clock mode (SLOW operation mode)
Selects input clock for the interval timer (Note2)
0 : fc / 2^7 [Hz]
1 : fs
Sets the warm-up time
Example At fc = 4.19MHz
fs = 32.8KHz
0 : 2^8 / fs + 2^9 / fc [sec] ...... 7.9 [ms]
1 : 2^11 / fs + 2^9 / fc 62.6

Note1. fc ; High-frequency clock [Hz]
fs ; High-frequency clock [Hz]
Note2. Only Normal-2 operation mode

Figure 2-17. System Clock Control Command Register

System clock control status register (Port address IP0E)

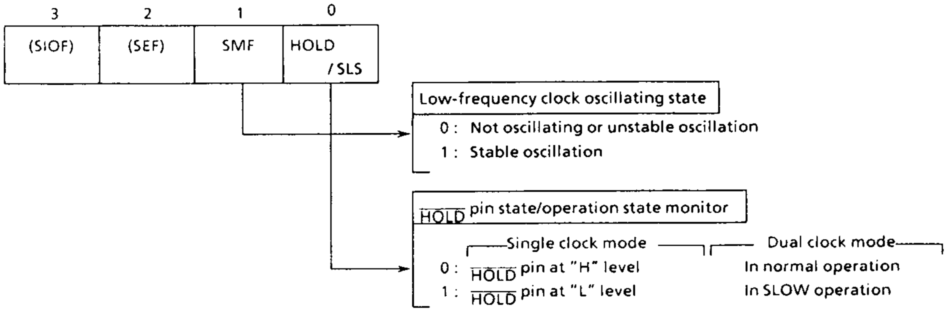

3
2
1
0
(SIOF)
(SEF)
SMF
HOLD / SLS
Low-frequency clock oscillating state
0 : Not oscillating or unstable oscillation
1 : Stable oscillation
HOLD pin state/operation state monitor
Single clock mode
Dual clock mode
0 : HOLD pin at "H" level
In normal operation
1 : HOLD pin at "L" level
In SLOW operation

Figure 2-18. System Clock Control Status Register

(1) Single-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 "5.1 HOLD Operating Mode".

(2) Dual-clock mode

In this mode, the Normal-2 operation is generally performed by generating the system clock from the high-frequency clock (fc). As required, the SLOW operation can be performed by generating the system clock from the low-frequency clock (fs). In the SLOW operation, the high-frequency clock oscillation automatically stops,enabling the low-power voltage operation or the low-power consumption operation. Instruction execution does not stop when the operation speed switching is pereformed. However, some peripheral hardware capabilities may be affected. For details, refer to the description of the relevant operation.

(3) System clock switching control

The following describes the switching between the Normal-2 and SLOW operations in the dual clock mode. During reset, the command register is initialized to the single-clock mode. It must be set to the Normal-2 operation of the dual-clock mode.

a. Switching from Normal-2 operation to SLOW operation

SMF (bit 1 of the status register) is monitored by program. First, Sets SLCK(bit 2 of OP16) to "1". Next, when it has been confirmed that the low-frequency clock oscillation is stable and SMF is changed "1"→"0"→"1" or "0"→"1"→"0", bit 1 of the command register is set to "1". At this time, the high-frequency clock oscillator stops.

b. Returning from SLOW operation to Normal-2 operation

Bit 1 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.

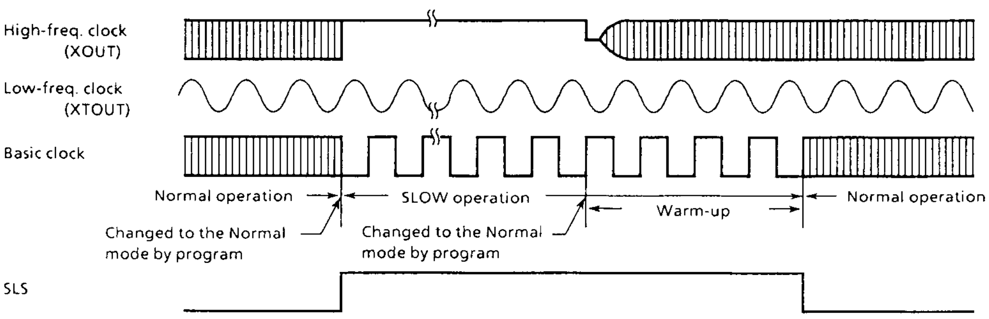


Figure 2-19. System Clock Switching Timing

2.7.4 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-470A series instruction set has 3 kinds of instructions, 1-cycle instruction to 3-cycle instruction. Each instruction cycle consists of 4 states (S1 through S4). Each state consists of 2 basic clock pulses.

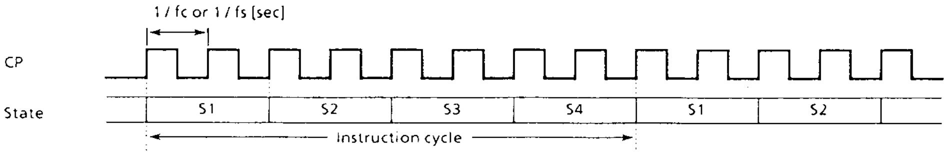


Figure 2-20. Instruction Cycle

3.PERIPHERAL HARDWARE FUNCTION

3.1 Ports

47C1260/1660 have 15 I/O ports (56pins) each as follows:

① K0 ; 4-bit input
② P0, P3 ; 4-bit output
③ P1, P2 ; 4-bit output
④ R4, R5 ; 4-bit input/output (shared with A/D converter analog inputs)
⑤ R6 ; 4-bit input/output
⑥ R7 ; 4-bit input/output (shared with the low-frequency resonator connecting pins and the watchdog timer output)
⑦ R8 ; 4-bit input/output (shared with external interrupt request input and timer/counter input)
⑧ R9 ; 3-bit input/output (shared with serial port)
⑨ RA, RB, RC ; 4-bit input/output
⑩ KE ; 1-bit sense input (shared with hold request/release signal input)

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.

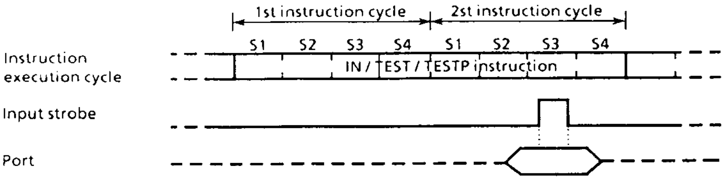


Figure 3-1. Input Timing

(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.

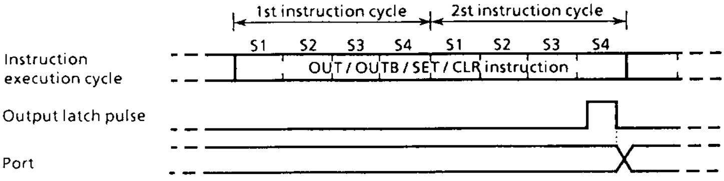


Figure 3-2. Output Timing

Port address (**)	Port		Input/Output instruction						
	Input (IP**)	Output (OP**)	IN %p, A IN %p, @HL	OUT A, %p OUT @HL,%p	OUT #k, %p	OUTB @HL	SET %p, b CLR %p, b	TEST %p, b TESTP %p, b	SET @L CLR @L TEST @L
00_H	K0 input port	P0 output port	○	○	○	–	–	○	–
01	P1 output latch	P1 output port	○	○	○	○ (Note2)	○	○	–
02	P2 output latch	P2 output port	○	○	○	○	○	○	–
03	P3 output latch	P3 output port	○	○	○	–	○	○	–
04	R4 input port (Analog input)	R4 output port	○	○	○	–	○	○	○
05	R5 input port (Analog input)	R5 output port	○	○	○	–	○	○	○
06	R6 input port	R6 output port	○	○	○	–	○	○	○
07	R7 input port	R7 output port	○	○	○	–	○	○	○
08	R8 input port	R8 output port	○	○	○	–	○	○	–
09	R9 input port	R9 output port	○	○	○	–	○	○	–
0A	RA input port	RA output port	○	○	○	–	○	○	–
0B	RB input port	RB output port	○	○	○	–	○	○	–
0C	RC input port	RC output port	○	○	○	–	○	○	–
0D	REMO-CON count value	REMO-CON offset value	○	○	○	–	–	–	–
0E	Status input (Note 4)	REMO-CON control	○	○	○	–	–	○	–
0F	Serial receive buffer	Serial transmit buffer	○	○	○	–	–	–	–
10_H	HOLD pin status	Hold operating mode control	○	○	–	–	–	–	–
11	——	——	–	–	–	–	–	–	–
12	A / D coverted value	A / D analog input selector	○	○	–	–	–	–	–
13	A / D, DTB status input	A / D start register, DTB selector	○	○	–	–	–	–	–
14	——	——	–	–	–	–	–	–	–
15	——	Watchdog timer control	–	○	–	–	–	–	–
16	——	System clock control	–	○	–	–	–	–	–
17	——	——	–	–	–	–	–	–	–
18	——	——	–	–	–	–	–	–	–
19	——	Interval Timer interrupt control	–	○	–	–	–	–	–
1A	——	——	–	–	–	–	–	–	–
1B	——	——	–	–	–	–	–	–	–
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 : The 5-bit to 8-bit data conversion instruction [OUTB @HL], automatic access to ports P1 and P2.

Note 3 : As concerns the port address "00", IN and TEST instructions operate port K0, and OUT instruction operates port P0.

Note 4 : The status input of serial interface, clock generator, and HOLD (KE0) pin.

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

3.1.2 I/O Ports

(1) Port K0 (K03 - K00)

Port K0 is a 4-bit input-only port. A pull-up or pull-down resistor can be contained by the mask option.

Port K0 (Port address IP00)

3	2	1	0
K03	K02	K01	K00

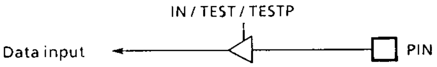


Figure 3-3. Port K0

(2) Ports P0 (P03-P00) and P3 (P33-P30)

Ports P0 and P3 are 4-bit output ports with latch. The latch is initialized to "0" during reset. When an input instruction is executed, the latch data are read as the port P3 but the port P0.

(3) Ports P1 (P13 - P10) and P2 (P23 - P20)

Ports P1 and P2 are 4-bit high current output ports which can directly drive LEDs, with 4-bit latches. When an input instruction is executed, the latch data is read in these ports. The latch is initialized to "1" during reset. They can be accessed separately at port addresses OP01/IP01 and OP02/IP02. Additionally,8-bit data can be set to these ports (the upper 4 bits to port P2, the lower 4 bits to port P1) by using the 5-bit to 8-bit data conversion instruction [OUTB @HL].

Example 1: To output immediate value "5" to port P1.

```
        OUT      #5, %OP01  ;  Port P1←5
```

Example 2: To read the latch data from port P2 to store it in the accumulator

```
        IN       %IP02, A   ;  Acc←Port P2
```

Example 3: To read, from ROM, the 8-bit data corresponding to the 5 bits obtained by linking the content (1 bit) of the carry flag with the contents (4 bits) of at address 90_H in RAM to output the 8-bit data to ports P2 and P1.

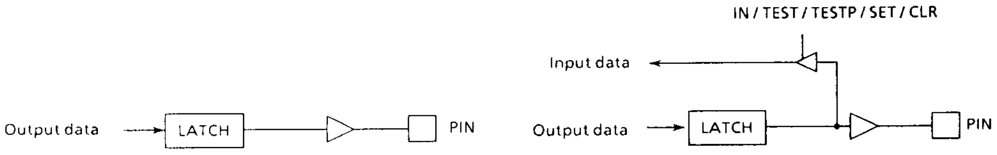
```
        LD       HL, #90H   ;  HL←90H (Sets the RAM address)
        OUTB     @HL        ;  Port P2, P1←ROM data
```

Port P0 (Port address OP00)

3	2	1	0
P03	P02	P01	P00

Port P3 (Port address OP03 / IP03)

3	2	1	0
P33	P32	P31	P30



(a) Port P0

(b) Port P3

Figure 3-4. Ports P0, P3

Port P2 (Port address OP02 / IP02)

3	2	1	0
P23	P22	P21	P20

Port P1 (Port address OP01 / IP01)

3	2	1	0
P13	P12	P11	P10

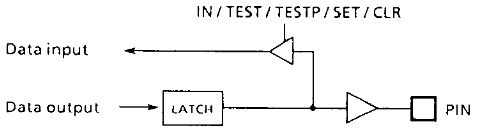


Figure 3-5. Ports P1, P2

(4) Ports R4 (R43 - R40), R5 (R53 - R50), R6 (R63 - R60), R7 (R73 - R70)

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

These 4 ports (16 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 3-2 lists the pins (I/O ports) that correspond to the L register contents.

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

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

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

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

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

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

Table 3-2. Relationship between L register contents and I/O port bits

a. Ports R4 (R43-R40) and R5 (R53-R50)

Ports R4 and R5 are 4-bit I/O ports with latch shared by the analog inputs for A/D converter. When used as an input ports or analog inputs, 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.

Port R5 (Port address OP05 / IP05)

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

Port R4 (Port address OP04 / IP04)

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

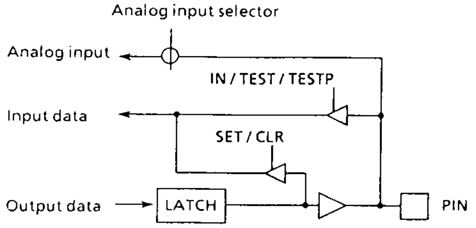


Figure 3-6. Ports R4, R5

b. Port R6 is 4-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 R6 (Port address OP06/IP06)

3	2	1	0
R63	R62	R61	R60

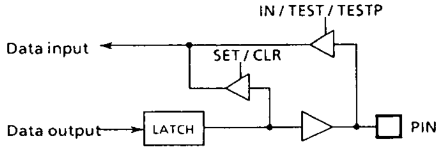


Figure 3-7. Port R6

c. Port R7 is shared by the low-frequency resonator connection pins (XTIN, XTOUT) and the watchdog timer output pin ($\overline{\text{WTO}}$). For the dual-clock mode operation, the low-frequency resonator (32.768kHz) 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 watchdog timer is used, R71 ($\overline{\text{WTO}}$) becomes the watchdog timer output pin. The watchdog timer output is the logical AND output with the port R71 output latch. To use the R71 pin for an ordinary I/O port, the watchdog timer must be disabled (with the watchdog timer output set to "1").

Port R7 (Port address OP07 / IP07)

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

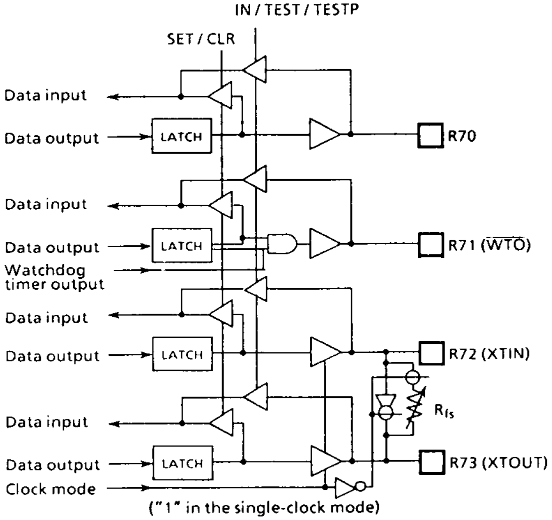


Figure 3-8. Port R7

(5) Port R8 (R83 - 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 by the external interrupt request 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 must be disabled or the event counter/pulse width measurement modes of the timer/counter must be disabled.

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 but bit 0 of the interrupt enable register is only kept at "0", not accepting the interrupt request.

Port R8 (Port address OP08 / IP08)

3	2	1	0
R83 (T1)	R82 ($\overline{\text{INT1}}$)	R81 (T2)	R80 ($\overline{\text{INT2}}$)

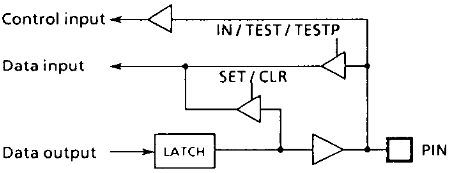


Figure 3-9. Port R8

(6) Port R9 (R92 - R90)

Port R9 is a 3-bit I/O port with 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". To use port R9 for an ordinary I/O port, the serial port must be disabled. Although R93 pin does not exist actually, execution of the set or clear instruction for R93 ([SET %OP09,3] or [CLR %OP09,3]) affects the operation of the internal CPU. Therefore, these instructions should not be execution on R93. However, other instructions may be used, in which an uncertain value is read upon execution of an input instruction.

Port R9 (Port address OP09 / IP09)

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

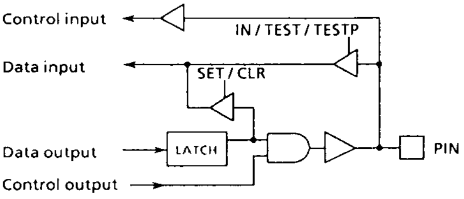


Figure 3-10. Port R9

(7) Ports RA (RA3-RA0), RB (RB3-RB0), RC (RC3-RC0)

These ports are 4-bit I/O ports with latch. The latch is initialized to "1" during reset.
When used as input port, the latch should be set to "1".

R*port (Port address OP0* / IP0*)

3	2	1	0
R*3	R*2	R*1	R*0

* ; A_H, B_H, C_H

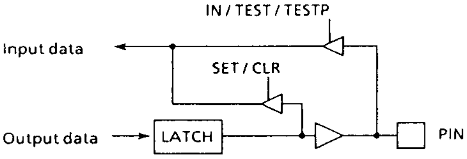


Figure 3-11. Ports RA, RB, RC

(8) Port KE ($\overline{\text{KE0}}$), Port K10 ($\overline{\text{K103}}$)

Port KE or Port K10 is a 1-bit sense input port shared by the hold request/release signal input pin ($\overline{\text{HOLD}}$). 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 $\overline{\text{KE0}}$ port cannot be used in the dual-clock mode

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

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

Port KE (Port address IP0E)

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

Port K10 (Port address IP10)

3	2	1	0
K103 (HOLD)			

Usable in Single clock mode only.

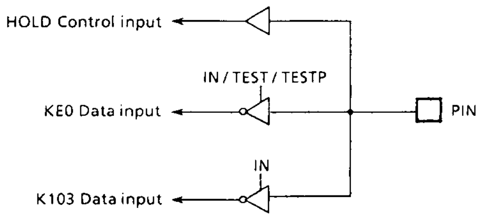


Figure 3-12. Port KE

3.2 Interval Timer

3.2.1 Configuration of Interval Timer

The interval timer consists of a 19-stage binary counter with a divide-by-3 prescaler. The source clock to the interval timer 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.

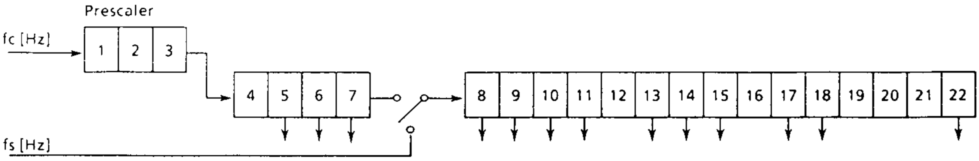


Figure 3-13. Configuration of Interval Timer

3.2.2 Functions of Interval Timer

The interval timer provides the following functions:

(1) The timer to generate an interrupt of fixed frequency
(2) Generation of internal pulse for timer/counters
(3) Generation of internal serial clock for a serial interface
(4) Generation of warm-up time at release of the hold operating mode
(5) Generation of source clock for a watchdog timer

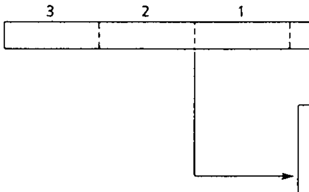
3.2.3 Interval Timer Interrupt (ITMR)

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)



(Initial value 0000)

	Single clock mode Normal-1	Dual clock mode Normal-2	 SLOW
00** :	Interrupt disabled	Interrupt disabled	Interrupt disabled
0100 :	Interrupt frequency fc / 2^{11} [Hz]	fs / 2^{4} [Hz]	Reserved
0101 :	Interrupt frequency fc / 2^{13}	fs / 2^{6}	Reserved
0110 :	Interrupt frequency fc / 2^{15}	fs / 2^{8}	Reserved
0111 :	Interrupt frequency fc / 2^{17}	fs / 2^{10}	fs / 2^{10} [Hz]
1*** :	Reserved	Reserved	Reserved

*Note1. * ; don't care*
Note2. fc ; High-frequency clock [Hz],
fs ; Low-frequency clock [Hz]

(a) Command register

Single clock mode	Dual clock mode	At fc = 4.194304MHz, fs = 32.768KHz
fc / 2^{11} [Hz]	fs / 2^{4} [Hz]	2048 Hz
fc / 2^{13}	fs / 2^{6}	512
fc / 2^{15}	fs / 2^{8}	128
fc / 2^{17}	fs / 2^{10}	32

(b) Example of interrupt frequency

Figure 3-14. Interval Timer Interrupt Command Register

3.3 Timer/Counters (TC1, TC2)

The 47C1260/1660 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.

	MSB		LSB
	Timer/Counter1 (TC1)		
	$TC1_H$	$TC1_M$	$TC1_L$
RAM address	($F6_H$)	($F5_H$)	($F4_H$)

	MSB		LSB
	Timer/Counter2 (TC2)		
	$TC2_H$	$TC2_M$	$TC2_L$
RAM address	(FA_H)	($F9_H$)	($F8_H$)

Figure3-15. 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)

3	2	1	0
TC1MS		IPR1	

(Initial value 0000)

Mode select

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

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)

3	2	1	0
TC2MS		IPR2	

(Initial value 0000)

Mode select

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

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]

Figure 3-16. 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 3-17. Note that counting continues if there is a count request after overflow occurrence.

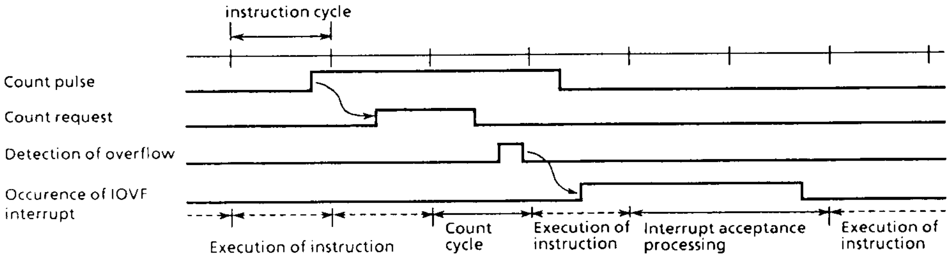


Figure 3-17. Timer/Counter Overflow Interrupt Timing

(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 2μs drops to 3.64μs.

Example: To operate TC2 in the event counter mode.

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

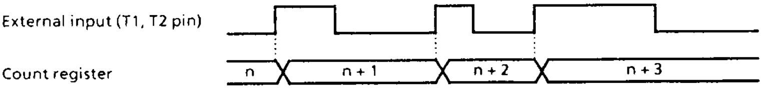


Figure 3-18. Event Counter Mode Timing Chart

(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 2μs drops to 2.016μs. 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,#1010B       ;  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
```

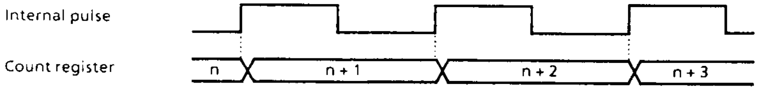


Figure 3-19. Timer Mode Timing Chart

※ 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:

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

Internal pulse rate	Max.setting time	At fc = 4.194304MHz	
		Internal pulse rate	Max.setting time
fc / 2^6 [Hz]	2^{18} / fc [sec]	65536 [Hz]	0.0625 [sec]
fc / 2^8	2^{20} / fc	16384	0.25
fc / 2^{10}	2^{22} / fc	4096	1
fc / 2^{14}	2^{26} / fc	256	16
fc / 2^{18}	2^{30} / fc	16	256
fc / 2^{22}	2^{34} / fc	1	4096

Table 3-3. Internal Pulse Rate Selection

(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 3-20, 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.

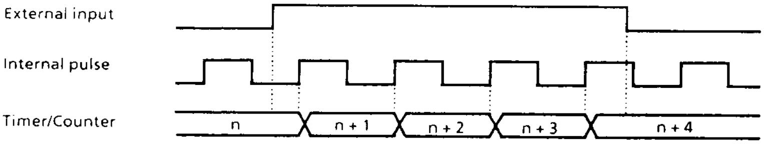


Figure 3-20. Pulse Width Measurement Mode Timing Chart

3.4 A/D Converter

47C1260/1660 have a 8-bit successive approximate type A/D converter and is capable of processing 8 analog inputs.

3.4.1 Circuit Configuration

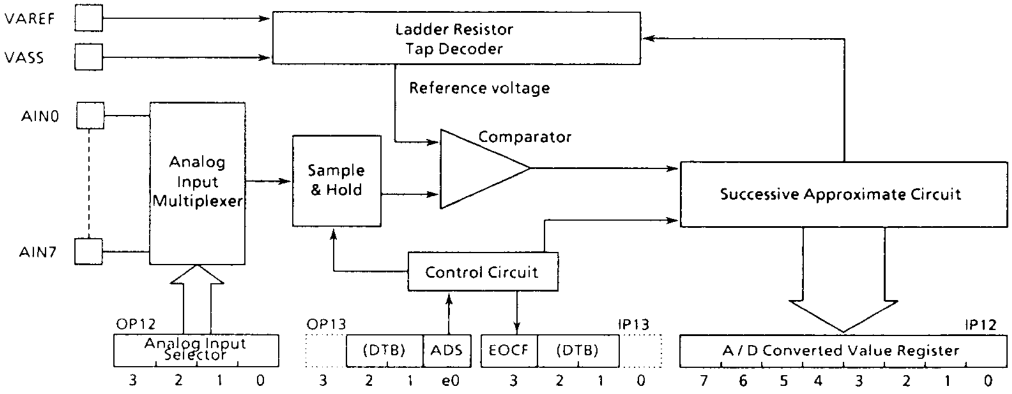


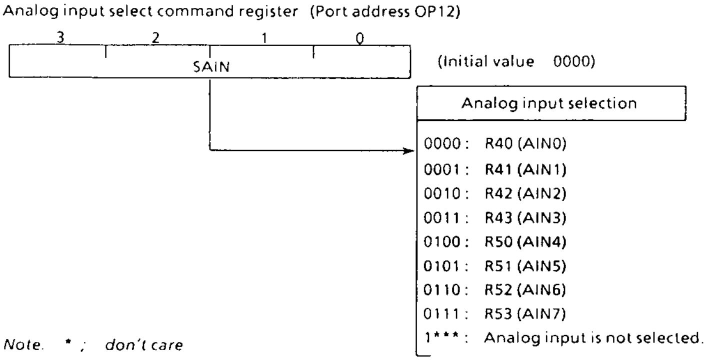
Figure 3-21. Block Diagram of A/D Converter

3.4.2 Control of A/D converter

The operation of A/D converter is controlled by a command. The command register is accessed as port addresses OP12 and OP13. A/D converted value and end of conversion flag (EOCF) can be known by accessing port addresses IP12 and IP13.

(1) Analog input selector

Analog inputs (AIN0 through AIN7) are selected by this register.



Analog input selection	
0000 :	R40 (AIN0)
0001 :	R41 (AIN1)
0010 :	R42 (AIN2)
0011 :	R43 (AIN3)
0100 :	R50 (AIN4)
0101 :	R51 (AIN5)
0110 :	R52 (AIN6)
0111 :	R53 (AIN7)
1*** :	Analog input is not selected.

*Note. * ; don't care*

Figure 3-22. Analog input selector

(2) Start of A/D conversion

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.

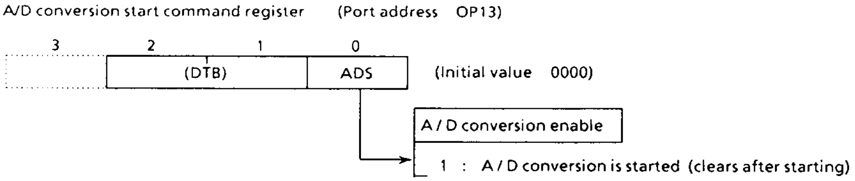

A/D conversion start command register (Port address OP13)
3
2
1
0
(DTB)
ADS
(Initial value 0000)
A / D conversion enable
1 : A / D conversion is started (clears after starting)

Figure 3-23. A/D conversion start register

(3) A/D converted value register

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).

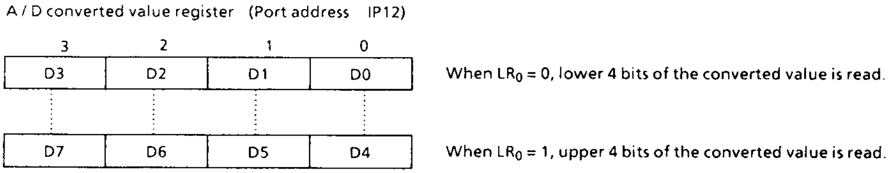

A / D converted value register (Port address IP12)
3
2
1
0
D3
D2
D1
D0
When LR_0 = 0, lower 4 bits of the converted value is read.
D7
D6
D5
D4
When LR_0 = 1, upper 4 bits of the converted value is read.

Figure 3-24. A/D converted value register

(4) A/D converter status register

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".

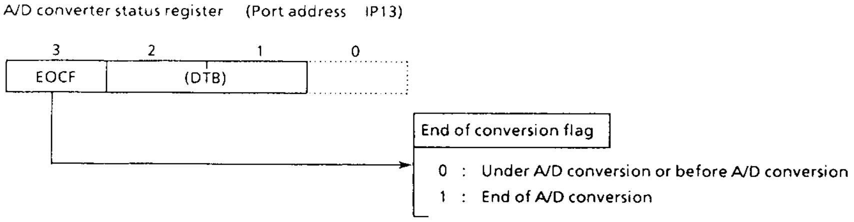

A/D converter status register (Port address IP13)
3
2
1
0
EOCF
(DTB)
End of conversion flag
0 : Under A/D conversion or before A/D conversion
1 : End of A/D conversion

Figure 3-25. A/D converter status register

3.4.3 How to use A/D converter

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

(1) Start of A/D conversion

Prior to conversion, select one of the analog input AIN0 through AIN7 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".

(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 (AIN4), 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, #4H         ; Selecting analog input (AIN4)
          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
```

3.5 Remote Control Pre-processor

The remote control pre-processor counts the edge-to-edge time of the input pulse and generates an interrupt request each time that the switching edge of an input pulse is detected. Remote control signal waveform which has been rectified by the receiver is inputed to the R80 ($\overline{INT2}$) pin.

3.5.1 Circuit configuration

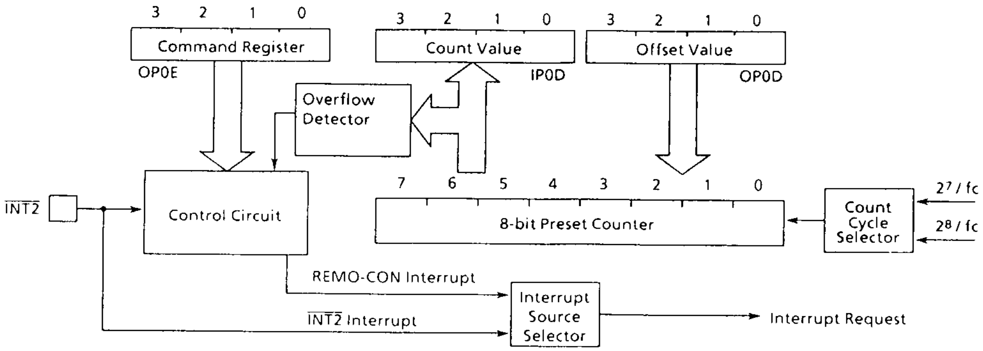
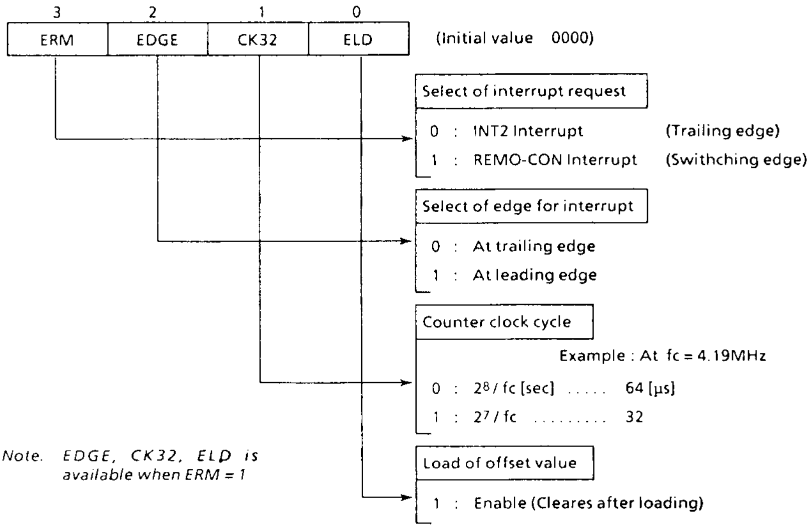


Figure 3-26. Remote Control Pre-processor

3.5.2 Control of Remote Control Pre-processor

The remote control pre-processor is controlled by the command register (OP0E) Also, external interrupt 2 is used to interrupt source. At reset, INT2 is selected, so a remote control discrimination interrupt is selected by command. Interrupt enable master F/F (EIF) and interrupt enable register (EIR) are used to enable/disable remote control discrimination interrupt. The interrupt priority is the same as for external interrupt 2.

REMO-CON pre-processor command register (Port address OP0E)


3
2
1
0
ERM
EDGE
CK32
ELD
(Initial value 0000)
Select of interrupt request
0 : INT2 Interrupt (Trailing edge)
1 : REMO-CON Interrupt (Swithching edge)
Select of edge for interrupt
0 : At trailing edge
1 : At leading edge
Counter clock cycle
Example : At fc = 4.19MHz
0 : 2⁸/fc [sec] ..... 64 [μs]
1 : 2⁷/fc 32
Load of offset value
1 : Enable (Cleares after loading)

Note. EDGE, CK32, ELD is available when ERM = 1

Figure 3-27. Command Register

(1) 8-bit preset counter

This a binary counter which counts the time from edge to next edgd. A value of offset value register (OP0D) are loaded to the lower 4 bits and the value of the upper 4 bits are stored to the count value register (IP0D).

Offset value is loaded in the following cases :

① When ELD (bit 0 of the OP0E) is set to "1".
② When there is an overflow of the lower 4 bits.
③ When interrupt request occurs.

Count value is stored in the following cases :

① When the upper 4 bits are all "1111_B".
② When interrupt request occurs.

(2) Reference time and detection time

The reference time must be created before measuring the pulse width of the remote control signal waveform. Reference time is set the value shown in Table 3-4. by the count cycle of the preset counter selected with the OP0E and the offset value set to the OP0D. Therefore, the product of the reference time and the count value read from the IP0D becomes the detecion time.
However, a worst error of the detection time is equivalent to reference time.

Count clock cycle	Offset value (HEX)	Reference time (n = 0~15)
2^7 / fc [sec]	0 to F	$(2^{11} - n \times 2^7)$ / fc [sec]
2^8 / fc	0 to F	$(2^{12} - n \times 2^8)$ / fc

Table 3-4. Set value of reference time

(3) Remote control pre-processing operation

First, set the offset value for creating the reference time to the OP0D. The offset value is loaded to the lower 4 bits of preset counter by setting ELD to "1". ERM can also be set to "1" at the same time. Setting ERM starts counting. ELD is cleared after loading but it is automatically loaded during the count by an overflow of the lower 4 bits.
After storing the count value in the IP0D by detecting the switching edge (leading/trailing) of the input pulse, the preset counter again loads the offset value to the lower 4 bits, clears to "0" the upper 4 bits and restarts counting.
An interrupt request is generated at this time. Therefore, read the count value in the interrupt service routine. The next interrupt is disabled until the stored count value is read. Switching edges can select in one by one using EDGE.
If the upper 4 bits of counter becomes "1111_B" before the next edge is accepted, it is judgeg to be an overflow and the interrupt is generated.
In this case, IP0D is set to "F_H" and it can be identified to be the overflow interruption. The input pulse width of both "H" and "L" levels must be more than 192/fc [sec]. If any shorter pulse than 192/fc [sec] may be expected, put the dummy instruction [IN %IP0D, A] reading out from count value register in the main routine.
Even if ERM is "0", when an INT2 interrupt source is enabled (determined by EIF, EIR), an interrupt request is generated by a falling edge of input pulse.
Figure3-35. shows the operation timing the remoto control pre-processor.

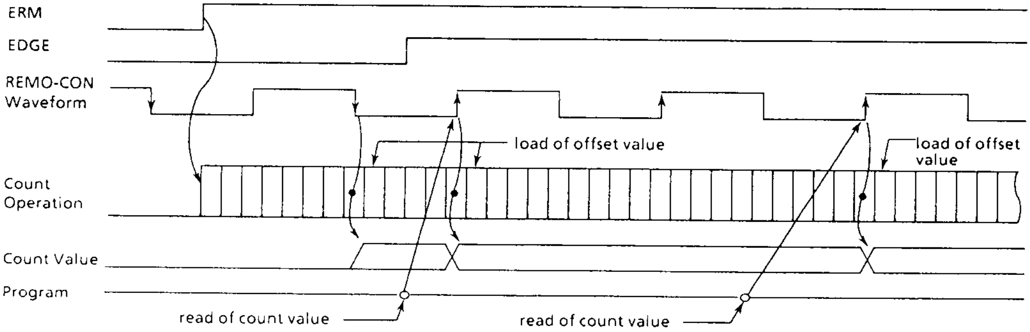
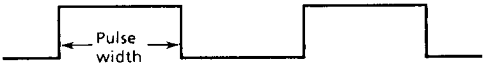


Figure 3-28. Timing chart

Example : To store a count value to RAM [10_H] that pulse width of input pulse is measured.



```
Main Routine
          OUT    #5, %OPOD        ;  Offset value register set to "5".
          OUT    #0FH, %OP0E      ;  Interrupt enable with setting count cycle to 32
                                     µsec, leading edge detection mode.
          CLR    0, 0             ;  Clear to RAM [00H] 0 as interrupt status flag.
          LD     A, #0001B        ;  Interrupt enable
          XCH    A, EIR
          EICLR  IL, #111110B
           :

INT2 service routine
          TEST   0, 0             ;  Skip, if interrupt status flag is "0".
          B      ST0
          LD     HL, #10H         ;  HL ← 10H
          IN     %IPOD, @HL       ;  RAM [10H] ← Count value
          CLR    0, 0             ;  Clearing interrupt status flag.
          OUT    #0EH, %OP0E      ;  Set to leading edge detection mode.
          B      SEND
ST0 :     SET    0, 0             ;  Setting interrupt status flag.
          OUT    #0AH, %OP0E      ;  Set to trailing edge detection mode.
          IN     %IPOD, A         ;  Dummy
SEND :     :
```

※ Calculation of pulse width with the above program when the count value stored to the RAM [10_H] is "9".
When the clock frequency is 32µs (fc = 4.19MHz) and the offset value is "5", the reference time will be as follows :

$$(2^{11} - 5 \times 2^{7}) / fc = 336 \, [\mu s]$$

Thus, the detection time (pulse width) will be 336 × 9 = 3.02[ms] (including an error of 336µs max.)

Note : When the low-frequency clock (fs) is input to the interval timer in the normal 2 operating mode, count cycle $2^8/fc$ operation cannot be used. Also, remote control pre-processor cannot be used in the SLOW operating mode.

3.6 Serial Interface (SIO)

The 47C1260/1660 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 ($\overline{\text{SCK}}$), 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.

3.6.1 Configuration of Serial Interface

Figure 3-29. shows configuration of serial interface.

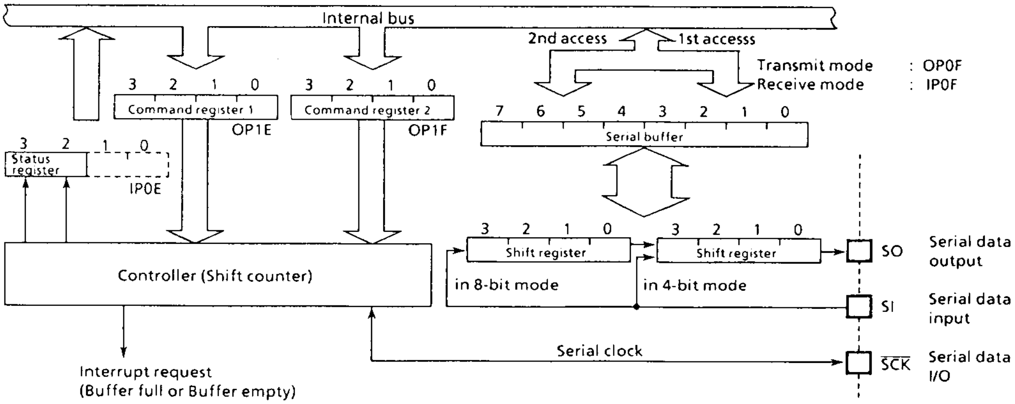
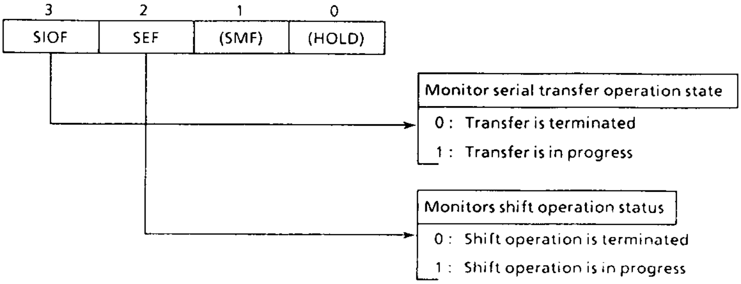


Figure 3-29. Configuration of Serial Interface

3.6.2 Control of Serial Interface

The serial interface is controlled by command registers (OP1E, OP1F). The operating states of the serial interface can be monitored by the status register (IP0E).

Serial interface status register (Port address IP0E).



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

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 3-30. Serial Interface Status Register

Serial interface control command register1 (port address OP1E)

3	2	1	0
INH	SBIT	CKR	

(Initial value 0001)

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

CKR → Select serial clock frequency

	Single clock mode Normal 1	Dual clock mode 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]

Note. fc ; High-frequency clock [Hz]

Serial interface control command register 2 (port address OP1F)

3	2	1	0
ESIO	RM	LM	ECKM

(Initial value 0000)

ESIO → Instructs serial transfer start/end

0 : Instructs serial transfer end

1 : Instructs serial transfer start

RM → Select transfer mode

	4 bit transfer	8 bit transfer
0 :	Transmit mode	Transmit mode
1 :	Receive mode	Transmit/Receive mode

LM → Select shift edge

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 $\overline{\text{SCK}}$ pin)

1 : External clock (input from $\overline{\text{SCK}}$ pin)

Note. When setting the transfer mode, ESIO must be "0"

Figure 3-31. Serial Interface Control Command Register

(1) Serial clock

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

a. Clock source selection

① Internal clock

The serial clock frequency is selected by command register.

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).

② 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.

b. Shift edge selection

① Leading edge

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

② 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.

(2) Transfer bit number

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

a. 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.

b. 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) Transfer modes

Selection between the transmit mode and the receive mode is performed by RM (bit 2 of the command register 2).

a. Transmit mode

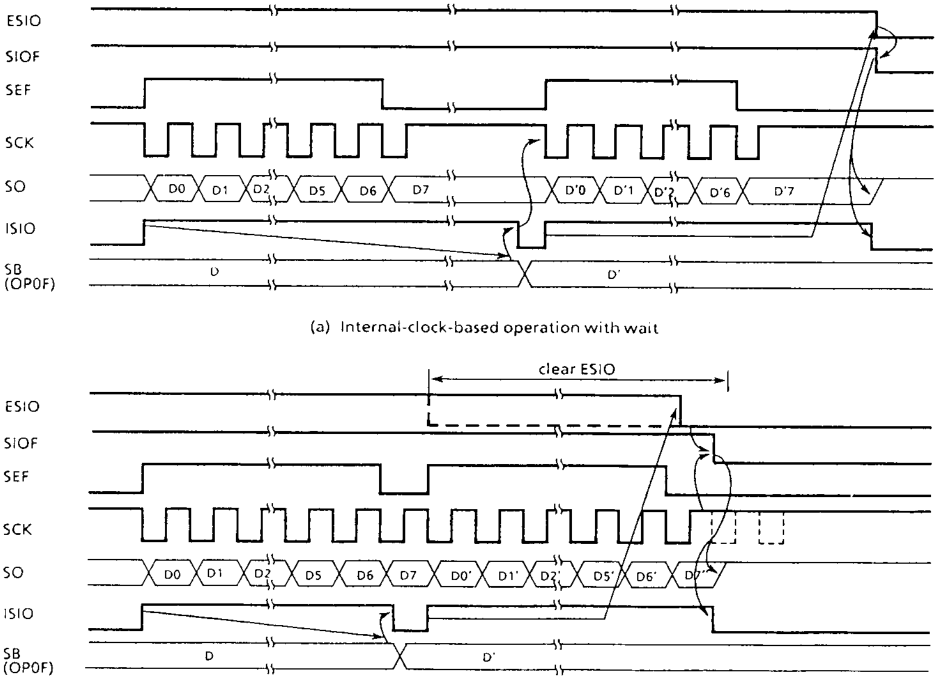
The transmit mode is set to the command register than writes the first transmit data (4 bits or 8 bits) to the buffer register (OP0F). (If the transmit mode is not set, the data is not written to the buffer register). In the 8-bit transfer mode, the 8-bit data is written by accessing the buffer register (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 next transmit data is written 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).

Example: To transmit (8-bit serial transfer) data stored in data memory (its address is specified by the HL register pair and the DMB) in synchronization with the internal clock (fc/27).

```
LD        A,   #0101B    ; OP1E ← 0101B (Sets the 8-bit serial transfer)
OUT       A,   %OP1E
LD        A,   #0010B    ; OP1F ← 0010B (Sets the transmit mode)
OUT       A,   %OP1F
OUT       @HL,   %OP0F   ; OP0F ← RAM [HL] (Writes the lower 4-bit data)
INC       L
OUT       @HL,   %OP0F   ; OP0F ← RAM [HL] (Writes the upper 4-bit data)
LD        A,   #1010B    ; ESIO ← 1 (Instructs transmission start)
OUT       A,   %OP1F
```



(a) Internal-clock-based operation with wait

(b) External-clock-based operation

Figure 3-32. Transmit Mode

b. Receive mode

At RM = 1 ; 4-bit receive mode is set when SBIT is cleared to "0", 8-bit simultaneous transmit/receive mode is set when SBIT is set to "1".

- 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 (IPOF),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:

① 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.

② 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).

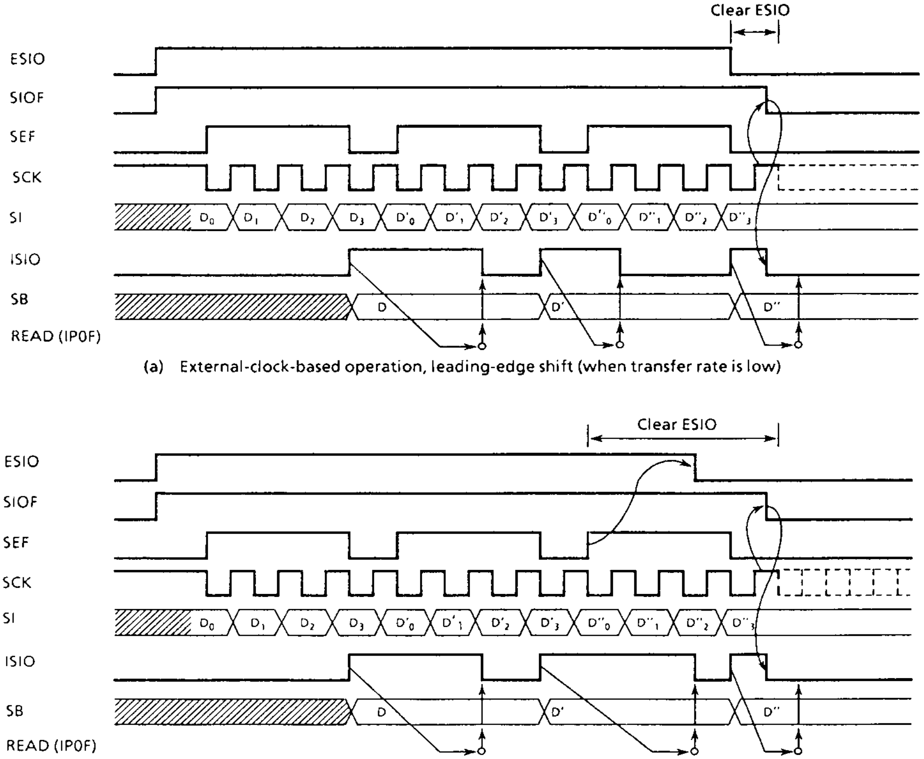
```
SSEF0 :   TEST    %IP0E,  2      ; Waits until SEF = "1"
          B       SSEF0
          LD      A,  #0110B     ; ESIO ← 0
          OUT     A,  %OP1F
          IN      %IP0F,  A      ; Acc ← IP0F (Reads received data)
```

③ 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) External-clock-based operation, leading-edge shift (when transfer rate is low)

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

Figure3-33. 4-bit Receive Mode

- 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.

Example 1 : To write data to be transmitted and to instruct the transmit/receive start.

```
LD      A, #0110B      ; Sets the 8-bit transfer and serial clock frequency.
OUT     A, %OP1E
LD      A, #0110B      ; Sets the transmit/receive mode of internal clock
                         operation
OUT     A, %OP1F
LD      HL, #20H       ; OP0F←RAM[20H] (Writes lower 4-bit data to be
                         transmitted)
OUT     @HL, %OP0F
INC     L              ; OP0F←RAM[21H] (Writes upper 4-bit data to be
                         transmitted)
OUT     @HL, %OP0F
LD      A, #1110B      ; ESIO ← 1 (Instructs serial transfer start)
OUT     A, %OP1F
 :                     ; Data transfer
```

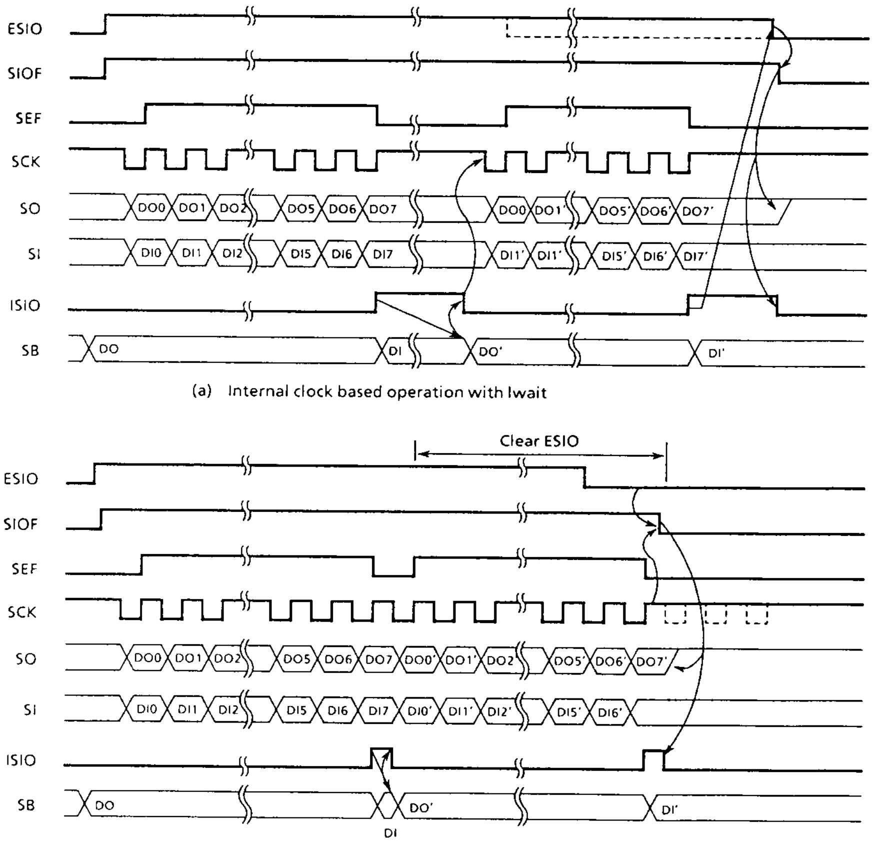
Example 2 : To read data received and to write next data to be transmitted.

```
LD      HL, #30H       ; Stores lower 4-bit data received in RAM[30H].
IN      %IP0F, @HL
INC     L              ; Stores upper 4-bit received in RAM[31H].
IN      %IP0F, @HL
LD      HL, #22H       ; Writes next lower 4-bit data to be transmitted.
OUT     @HL, %OP0F
INC     L              ; Writes next upper 4-bit data to be transmitted.
OUT     @HL, %OP0F
```

(4) 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 Iwait

(b) External clock based operation

Figure3-34. 8-bit Transmit/Receive Mode

3.7 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 watchdog timer output appears on R71 ($\overline{\text{WTO}}$) pin as the malfunction detect signal. To use the WDT, the output latch of R71 must be set to "1" (during reset, it is set to "1"). Note that, the WDT is disabled during reset. Connecting the $\overline{\text{WTO}}$ pin and the $\overline{\text{RESET}}$ pin resets malfunction.

3.7.1 Configuration of Watchdog Timer

The WDT consists of 10 binary counters, a flip-flop, and a controller. Source input clock of binary counters is fc/2^7[Hz] or fs[Hz]. Table 3-8. shows watchdog timing detection time and operating mode (detection time of watchdog timer) .

The flip-flop is set to "1" during reset, and cleared to "0" on the rising edge of the binary counter output.

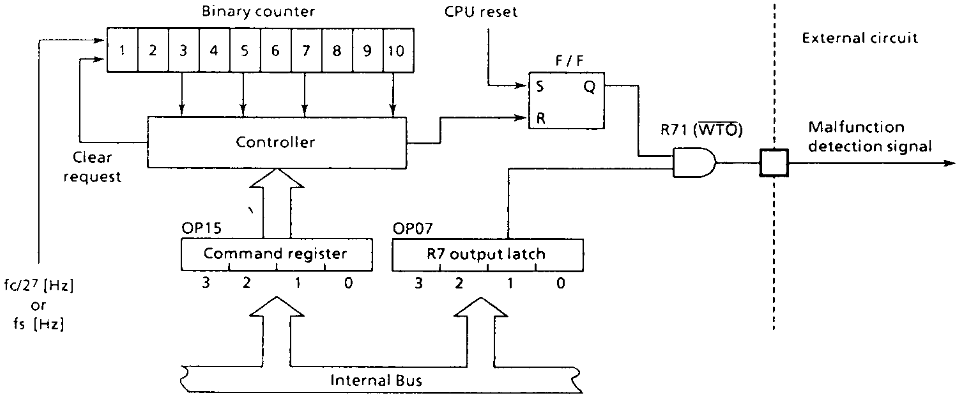


Figure 3-35. Configuration of Watchdog Timer

3.7.2 Control of Watchdog Timer

The WDT is controlled by the command register (OP15). The command register is initialized to "1000_B" during reset.

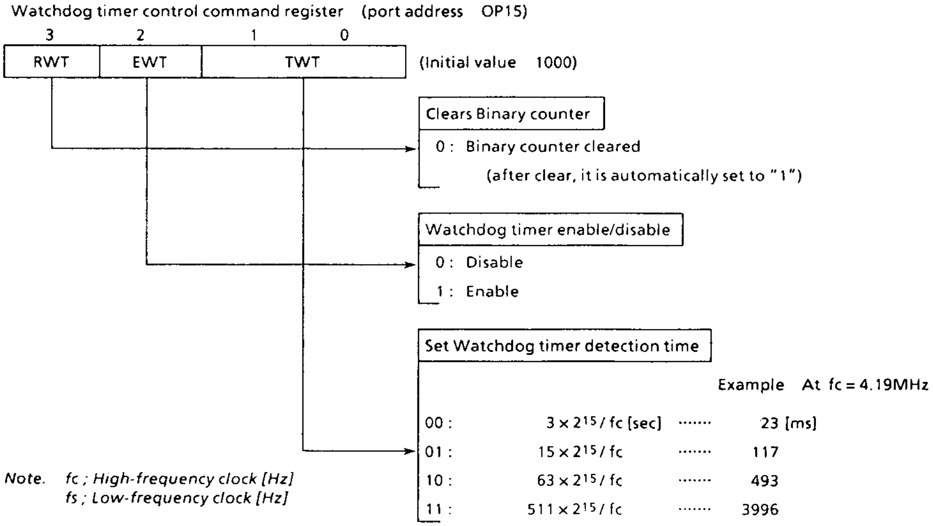
To detect the CPU malfunction by the WDT:

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.
If the output latch of R71 is "1" at this time, the $\overline{WTO}$ output goes low.

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

```
                              LD    A, #0010B   ; OP15 ← 0010B  (Set WDT detection time, clear
                              OUT   A, %OP15                     binary counters)
                              LD    A, #1110B   ; OP15 ← 1110B  (Enable WDT)
                           ┌─ OUT   A, %OP15
                           │    :
Within WDT detection timer │    :
                           │    :
                           │  LD    A, #0110B   ; OP15 ← 0110B  (Clear binary counters)
                           └─ OUT   A, %OP15
                                :
                                :
```

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.*



Clears Binary counter

0 : Binary counter cleared

(after clear, it is automatically set to "1")

Watchdog timer enable/disable

0 : Disable

1 : Enable

Set Watchdog timer detection time

Example At fc = 4.19MHz

00 : $3 \times 2^{15} / fc$ [sec] ······· 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]

Figure3-36. Watchdog Timer Control Command Register

TWT	Normal 1 operation mode	Normal 2 operation mode (Interval timer input)		SLOW operation mode
		$fc / 2^7$ [Hz]	fs [Hz]	
00	$3 \times 2^{15} / fc$ [sec]		$3 \times 2^8 / fs$ [sec]	
01	$15 \times 2^{15} / fc$		$15 \times 2^8 / fs$	
10	$63 \times 2^{15} / fc$		$63 \times 2^8 / fs$	
11	$511 \times 2^{15} / fc$		$511 \times 2^8 / fs$	

Table 3-5. Watchdog Timing detection time

4. INTERRUPT FUNCTION

4.1 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.

Sources			Interrupt priority by hardware	Interrupt latch	Pertmit conditions by program	Entry address
External	External interrupt 1	(INT1)	(High rank) 1	IL_5	EIF = 1	0002_H
Internal	Serial interface interrupt	(ISIO)	2	IL_4	EIF = 1, EIR_3 = 1	0004_H
	Timer/Counter 1 overflow interrupt	(IOVF1)	3	IL_3	EIF = 1, EIR_2 = 1	0006_H
	Timer/Counter 2 overflow interrupt	(IOVF2)	4	IL_2	EIF = 1, EIR_1 = 1	0008_H
	Interval timer interrupt	(ITMR)	5	IL_1		$000A_H$
External	External interrupt 2	(INT2)	(Low rank) 6	IL_0	EIF = 1, EIR_0 = 1	$000C_H$

Table 4-1. Interrupt Sources

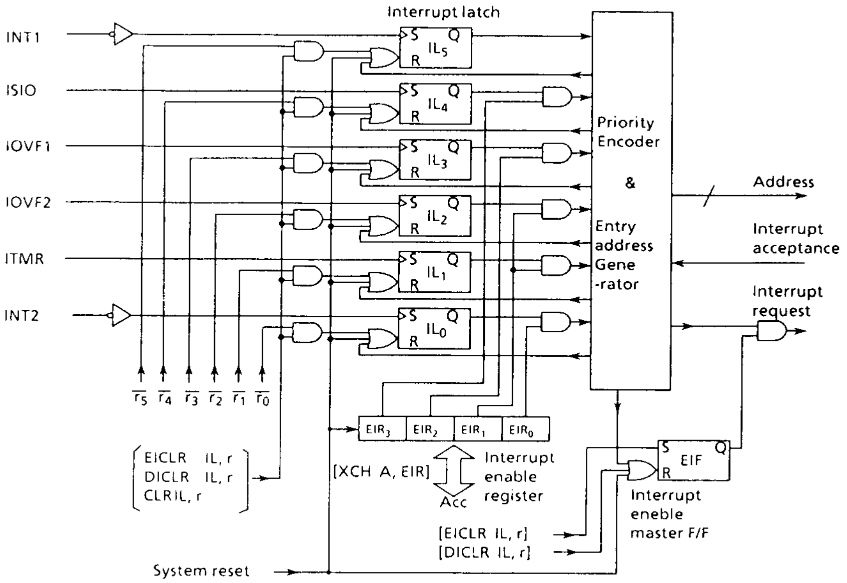


Figure 4-1. Interrupt Controller Block Diagram

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

The EIF controls the enable/disable of all interrupts. When this flip-flop is cleared to "0", all interrupts are disabled; when it is set to "1", the interrupts are enabled. When an interrupt is accepted, the EIF is cleared to "0", temporarily disabling the acceptance of subsequent interrupts. When the interrupt service program has been executed, the EIF is set to "1" by the execution of the interrupt return instruction [RETI], being put in the enabled state again. Set or clear of the EIF in program is performed by instructions [EICLR IL,r] and [DICLR IL,r], respectively. The EIF is initialized to "0" during reset.

(2) Interrupt enable register (EIR)

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

(3) Interrupt latches (IL_5 to IL_0)

An interrupt latch (IL) is provided for each interrupt source. It is set to "1" when an interrupt request is made to ask the CPU for accepting the interrupt. Each latch is cleared to "0"upon acceptance of the interrupt. It is initialized to "0" during the reset. The interrupt latches 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 IL is cleared; when the value is "1", the IL is held. Note that the interrupt latches 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
```

4.2 Interrupt Processing

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

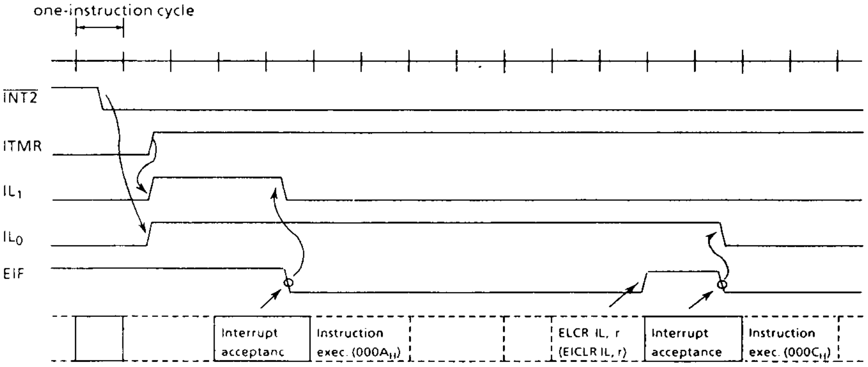
⑤ The IL for the accepted interrupt source is cleared to "0".

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. Note that the interrupt entry address is assigned every 2-byte, so that the long branch instruction can not be stored in the program normally. The interrupt service program is assigned to the memory locations 0000 to $0FFF_H$.)

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 disabled switch of interrupt service).

```
PINT1 :    TEST    05H, 0        ; RAM[05H]0 = 1, returns without the interrupt
                                   service
B          SINT1
           RETI
SINT1 :      :
```



Note1. It is assumed that there is no other interrupt request and EIR = 0011$_B$.
Note2. The value r in the [EICLR IL, r] instruction is assumed as 11111$_B$.
Note3. ⬚ denotes the execution of an instruction.

Figure 4-2. 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. That is the next interrupt request is accepted immediately after execution of the interrupt return instruction.

Note. *When the time required for the interrupt service is longer than that for 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 (H or L register, DMB, DC, etc.) 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 1 : To save/restore accumlator 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".

Example 2 : To save DMB to the lower 2 bits of address GSAV2 in data memory

```
                    CLR     GSAV3, 0        ;  RAM [GSAV3] 0 ← 0
                    CLR     GSAV3, 1        ;  RAM [GSAV3] 1 ← 0
                    TEST    DMB0            ;  DMB0 = 0, skip
                    B       SKIPS0
                    SET     GSAV3, 0        ;  RAM [GSAV3] 0 ← 1
         SKIPS0 :   TEST    DMB1            ;  DMB1 = 0, skip
                    B       SKIPS1
                    SET     GSAV3, 1        ;  RAM [GSAV3]1←1
         SKIPS1 :
```

Example 3 : Restore DMB from bit 0 of address GSAV2 in the RAM.

```
                    LD      DMB, #0000B     ;  DMB ← 0
                    TEST    GSAV3, 0        ;  RAM [GSAV3] 0 = 0, skip
                    B       SKIPR0
                    SET     DMB0            ;  DMB0 ← 1
         SKIPR0 :   TEST    GSAV3, 1        ;  RAM [GSAV3]1 = 0, skip
                    B       SKIPR1
                    SET     DMB1            ;  DMB1←1
         SKIPR1 :
```

4.3 External Interrupt

When an external interrupt (INT1 or INT2) occurs, the interrupt latch is set at the falling edge of the corresponding pin input ($\overline{\text{INT1}}$ or $\overline{\text{INT2}}$) . 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 an emergency interrupt. When R82 ($\overline{\text{INT1}}$) pin is used for the I/O port, the INT1 interrupt occurs at the falling edge of the pin input, so that the interrupt return [RET1] instruction must be stored at the interrupt entry address to perform dummy interrupt processing.

The INT2 interrupt can be enabled/disabled by the EIR.

Therefore, the INT2 interrupt occurs at the falling edge of the pin input when R80 ($\overline{\text{INT2}}$) pin is used for the I/O port.

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

Because the external interrupt input is the hysteresis type, each of high and low level operation requires 2 or more instruction cycles for a correct interrupt operaion.

※ The external interrupt through the rising edge.

The external interrupt pin through the rising edge is not prepared under hardware. When Timer/Counter are not used, the interrupt request signal can be applied to the Timer/Counter input pin by the event counter mode (count register is set to FFF_H) .

5. POWER SAVING FUNCTION

The 47C1260/1660 provide the HOLD operating mode and the SLOW operating mode to implement the low-power-consumpting operations.

5.1 HOLD Operating Mode

The HOLD feature stops the system and holds the systems internal states active before stop with a low power. The $\overline{\text{HOLD}}$ 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). The $\overline{\text{HOLD}}$ pin is shared with the $\overline{\text{KE0}}$ pin.

5.1.1 Starts the HOLD operating mode

The HOLD operation is started when the command is set and holds the following states during the HOLD operation:

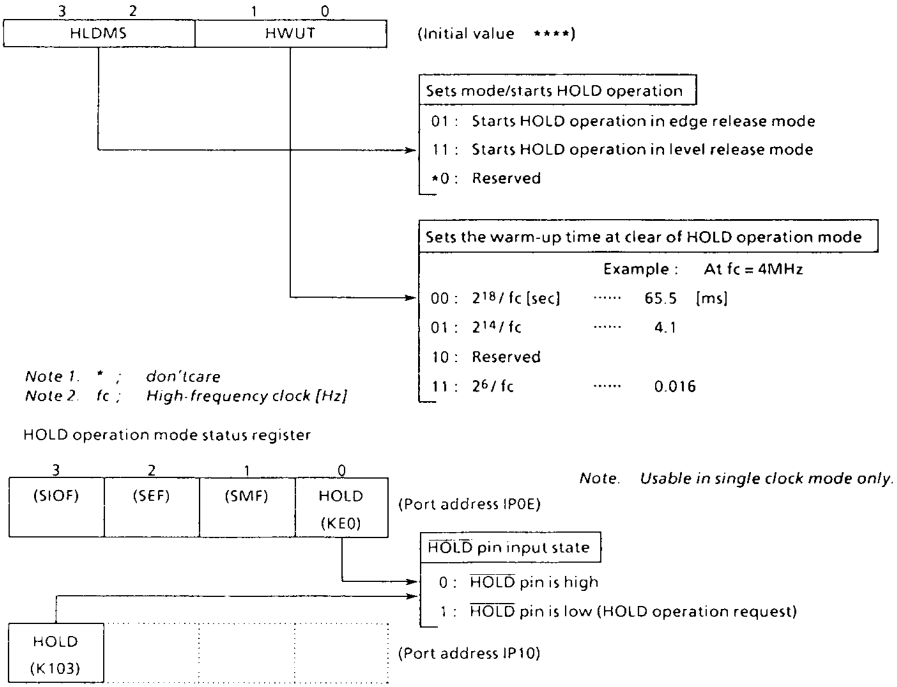
① Oscillator stops and the systems internal operations are all held up.

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 operation mode command register (Port address OP10)


3 2 1 0
HLDMS
HWUT
(Initial value ****)
Sets mode/starts HOLD operation
01 : Starts HOLD operation in edge release mode
11 : Starts HOLD operation in level release mode
*0 : Reserved
Sets the warm-up time at clear of HOLD operation mode
Example : At fc = 4MHz
00 : 2^18/fc [sec] ...... 65.5 [ms]
01 : 2^14/fc 4.1
10 : Reserved
11 : 2^6/fc 0.016
HOLD operation mode status register
3 2 1 0
(SIOF)
(SEF)
(SMF)
HOLD (KE0)
(Port address IP0E)
Note. Usable in single clock mode only.
HOLD pin input state
0 : HOLD pin is high
1 : HOLD pin is low (HOLD operation request)
HOLD (K103)
(Port address IP10)

*Note 1. * ; don'tcare*
Note 2. fc ; High-frequency clock [Hz]

Figure 5-1. HOLD Operating Mode Command Register/Status Register

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

(1) Level-sensitive release mode

In this mode, the $\overline{\text{HOLD}}$ 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 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 release 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:

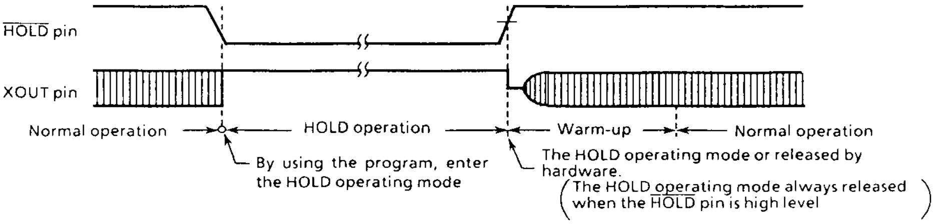
① Testing HOLD

② Applying the HOLD pin input also to the $\overline{\text{INT1}}$ pin to generate the external interrupt 1 request.

In single clock mode.

Example : To test 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
```



By using the program, enter the HOLD operating mode

The HOLD operating mode or released by hardware.
(The HOLD operating mode always released when the $\overline{\text{HOLD}}$ pin is high level)

Figure 5-2. Level-sensitive Release Mode

(2) Edge-sensitive release mode

In this mode, the HOLD operation is released at the rising edge of the $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ pin input is high, the HOLD operation is performed.

In single clock mode.

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
```

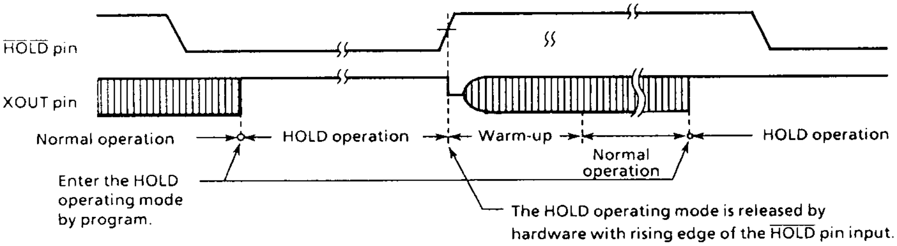

$\overline{\mathrm{HOLD}}$ pin
XOUT pin
Normal operation
HOLD operation
Warm-up
Normal operation
HOLD operation
Enter the HOLD operating mode by program.
The HOLD operating mode is released by hardware with rising edge of the $\overline{\mathrm{HOLD}}$ pin input.

Figure 5-3. 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 0.5V), 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 ports input transistor, requiring to fix the level by pull-up or other means.*

5.1.2 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 in single clock mode from the instruction next to the instruction which starts the HOLD operation. At this time, the interval timer starts from the reset state "0".

※ *The warm-up time is obtained by dividing the basic clock by the interval timer, so that, if the frequency at clearing the HOLD operation is unstable, the warm-up time shown in Figure 5-1 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 $\overline{RESET}$ 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 $\overline{RESET}$ pin input voltage goes behind the rise of the power voltage. At this time, if the voltage level of the $\overline{RESET}$ pin input drops below the non-inverted high level input voltage of the $\overline{RESET}$ pin input (hysteresis input), the reset operation may occur.

5.2 SLOW Operating Mode

In the SLOW operating mode, the power consumption is reduced by operating the system on the low-frequency clock. For the details of this mode, refer to Subsection "2.7.3 System Clock Controller".

6. RESET FUNCTION

When the RESET pin is held to the low level for at 3 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 0000_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.

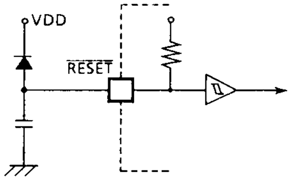


Figure 6.1 Simplified power-on-reset circuit

On-chip hardware	Initial value	On-chip hardware	Initial value
Program counter (PC)	0000_H	Interval timer	"0"
Status flag (SF)	1	Output latch (I/O port or output ports)	Refer to "I/O" circuitry
Data memory bank selecter (DMB)	0		
Interrupt enable master flip-flop (EIF)	0	Command register	Refer to the description of each relative command register
Interrupt enable register (RIR)	0_H		
Interrupt latch (IL)	"0"		

Table 6-1. Initialization of Internal States by Reset Action

ELECTRICAL CHARACTERISTICS

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

PARAMETER	SYMBOL	PINS	RATING	UNIT
Supply Voltage	V_{DD}		– 0.3 to 7	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT1}	Ports R4, R5, R7, Push-pull	– 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	Ports P1, P2, R6, R8, R9	– 0.3 to 10	
Output Current (per 1 pin)	I_{OUT1}	Ports R	3.2	mA
	I_{OUT2}	Ports P1, P2	30	
	I_{OUT3}	Ports P0, P3	15	
Output Current (total)	ΣI_{OUT1}	Ports P0, P1	120	mA
	ΣI_{OUT2}	Ports P2, P3	120	
Power Dissipation [T_{opr} = 70°C]	PD		600	mW
Soldering Temperature (time)	T_{sld}		260 (10sec)	°C
Storage Temperature	T_{stg}		– 55 to 125	°C
Operating Temperature	T_{opr}		– 40 to 70	°C

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V, T_{opr} = – 40 to 70°C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Max.	UNIT
Supply Voltage	V_{DD}		In the Normal mode	4.5	6.0	V
			In the SLOW mode	2.7		
			In the HOLD mode	2.0		
Input High Voltage	V_{IH1}	Except Hysteresis Input	$V_{DD} \geqq 4.5V$	$V_{DD} \times 0.7$	V_{DD}	V
	V_{IH2}	Hysteresis Input		$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5V$	$V_{DD} \times 0.9$		
Input Low Voltage	V_{IL1}	Except Hysteresis Input	$V_{DD} \geqq 4.5V$	0	$V_{DD} \times 0.3$	V
	V_{IL2}	Hysteresis Input			$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5V$		$V_{DD} \times 0.1$	
Clock Frequency	fc	XIN, XOUT		0.4	6.0	MHz
	fs	XTIN, XTOUT		30	34	KHz

Note 1. Input voltage V_{IH3}, V_{IL3} : in the SLOW or HOLD mode

D.C. CHARACTERISTICS (V$_{SS}$ = 0V, T$_{opr}$ = −40 to 70°C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Typ.	Max.	UNIT
Hysteresis Voltage	V_{HS}	Hysteresis Input		–	0.7	–	V
Input Current	I_{IN1}	Port K0, TEST, $\overline{RESET}$, $\overline{HOLD}$	V_{DD} = 5.5V, V_{IN} = 5.5V / 0V	–	–	±2	μA
	I_{IN2}	Ports R (open drain)					
Low Input Current	I_{IL}	Ports R (push-pull)	V_{DD} = 5.5V, V_{IN} = 0.4V	–	–	−2	mA
Input Resistance	R_{IN1}	Port KO with pull-up/pull-down		30	70	150	KΩ
	R_{IN2}	$\overline{RESET}$		100	220	450	
Output Leakage Current	I_{LO}	Ports (open drain)	V_{DD} = 5.5V, V_{OUT} = 5.5V	–	–	2	μA
Output High Voltage	V_{OH}	Ports (push-pull)	V_{DD} = 4.5V, I_{OH} = −200μA	2.4	–	–	V
Output Low Voltage	V_{OL}	Except XOUT, ports P	V_{DD} = 4.5V, I_{OL} = 1.6mA	–	–	0.4	
Output Low Current	I_{OL2}	Ports P1, P2	V_{DD} = 4.5V, V_{OL} = 1.0V	–	20	–	mA
	I_{OL3}	Ports P0, P3		–	7	–	
Supply Current (in the Normal mode)	I_{DD}		V_{DD} = 5.5V fc = 4MHz	–	3	6	mA
Supply Current (in the SLOW mode)	I_{DDS}		V_{DD} = 3.0V fs = 32.768KHz	–	30	60	μA
Supply Current (in the HOLD mode)	I_{DDH}		V_{DD} = 5.5V	–	0.5	10	μA

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

Note 2. Input Current I_{IN1} ;The current through resistor is not included, when the input resistor (pull-up/pull-down) is contained.

Note 3. Supply Current I_{DD}, I_{DDH} ; V_{IN} = 5.3V/0.2V
The K0 port is open when the input resistor is contained. The voltage applied to the R port is within the valid range.

Note 4. Supply Current I_{DDS} ; V_{IN} = 2.8V/0.2V
Low frequency clock is only osillated (connecting XTIN, XTOUT).

A / D CONVERSION CHARACTERISTICS (T$_{opr}$ = −40 to 70°C)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Analog Reference Voltage	V_{AREF}		V_{DD} − 1.5	–	V_{DD}	V
	V_{ASS}		V_{SS}	–	1.5	
Analog Reference Voltage Range	ΔV_{AREF}	$V_{AREF} - V_{ASS}$	2.5	–	–	V
Analog Input Voltage	V_{AIN}		V_{ASS}	–	V_{AREF}	V
Analog Supply Current	I_{REF}		–	0.5	1.0	mA
Nonlinearity Error		V_{DD} = 5.0V, V_{SS} = 0.0V V_{AREF} = 5.000V V_{ASS} = 0.000V	–	–	±1	LSB
Zero Point Error			–	–	±1	
Full Scale Error			–	–	±1	
Total Error			–	–	±2	

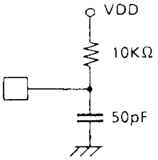
A. C. CHARACTERISTICS

(V_{SS} = 0V, V_{DD} = 4.5 to 6.0V, T_{opr} = −40 to 70°C)

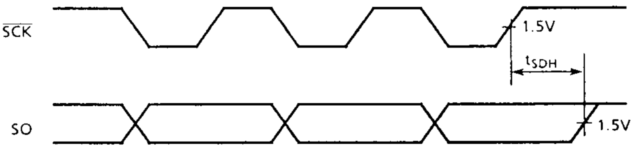
PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Instruction Cycle Time	t_{cy}	In the Normal mode	1.9	–	20	ns
		In the SLOW mode	235	–	267	
High level Clock pulse Width	t_{WCH}	External clock mode	80	–	–	ns
Low level Clock pulse Width	t_{WCL}					
A / D Sampling Time	t_{AIN}	fc = 4MHz	–	4	–	μs
Shift Data Hold Time	t_{SDH}		$0.5t_{cy} - 300$	–	–	ns

Note. Shift Data Hold Time

External circuit for $\overline{SCK}$ pin and SO pin



Serial port (completion of transmission)



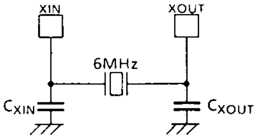
RECOMMENDED OSCILLATING CONDITIONS

(V_{SS} = 0V, V_{DD} = 4.5 to 6.0V, T_{opr} = −40 to 70°C)

(1) 6MHz

Ceramic Resonator

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



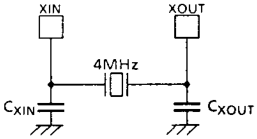
(2) 4MHz

Ceramic Resonator

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

Crystal Oscillator

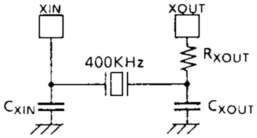
204B-6F 4.0000	(TOYOCOM)	$C_{XIN} = C_{XOUT} = 20pF$



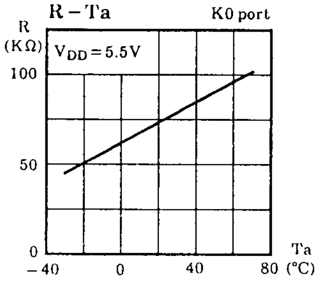
(3) 400KHz

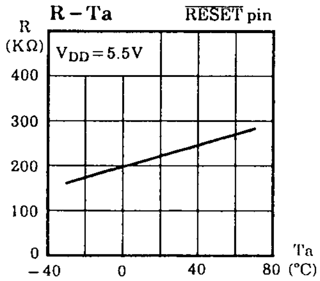
Ceramic Resonator

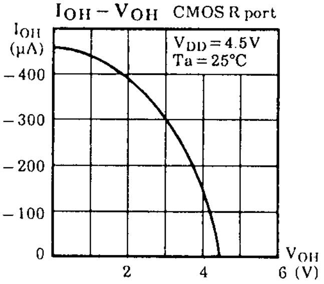
CSB400B	(MURATA)	$C_{XIN} = C_{XOUT} = 220pF$, $R_{XOUT} = 6.8K\Omega$
KBR-400B	(KYOCERA)	$C_{XIN} = C_{XOUT} = 100pF$, $R_{XOUT} = 10K\Omega$

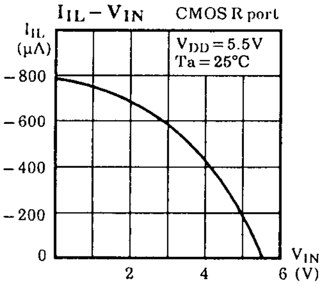


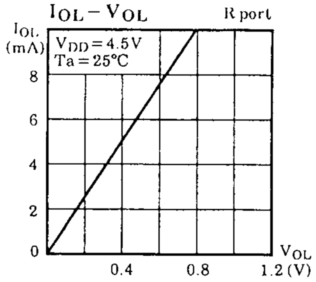
TYPICAL CHARACTERISTICS

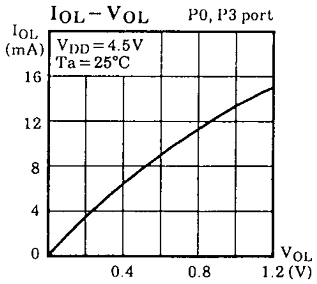


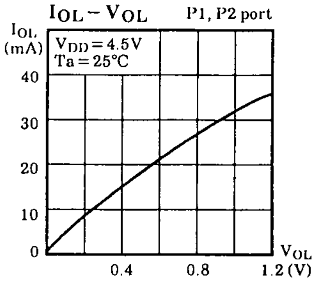


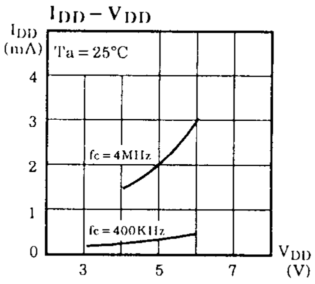


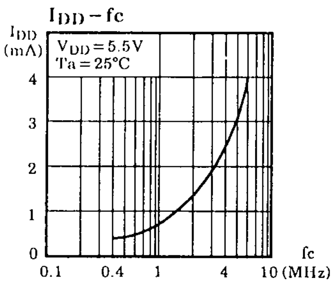








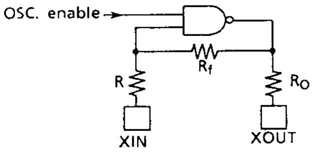
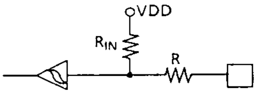
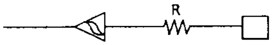
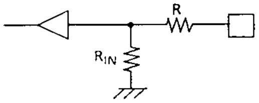




INPUT/OUTPUT CIRCUITRY

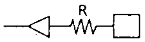
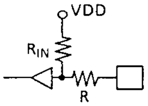
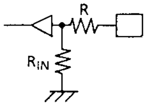
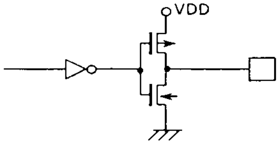
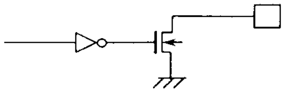
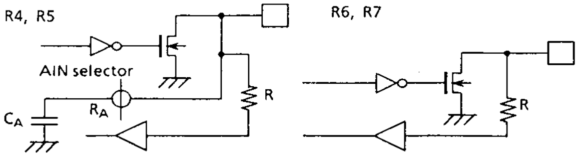
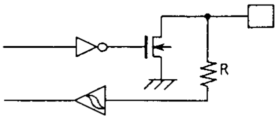
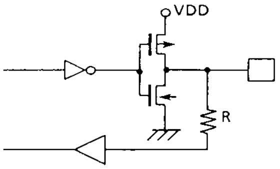
(1) Control pins

Input / Output circuitries of the 47C1260 / 1660 control pins are shown below.

CONTROL PIN	I/O	CIRCUITRY	REMARK
XIN XOUT	Input Output		Resonator connecting pins (R = 1KΩ typ.) (R_f = 1.5MΩ typ.) (R_O = 2KΩ typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Contained pull-up resistor (R_{IN} = 220KΩ typ.) (R = 1KΩ typ.)
$\overline{\text{HOLD}}$ (KE0)	Input (Input)		Hysteresis input (Sense input) (R = 1KΩ typ.)
TEST	Input		Contained pull-down resistor (R_{IN} = 70KΩ typ.) (R = 1KΩ typ.)

(2) I/O ports

The input/output circuitries of the 47C1260/1660 I/O ports are shown below, any one of the circuitries can be chosen by a code (IA-IC) as a mask option.

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE			REMARKS
		IA	IB	IC	
K0	Input				Pull-up/pull-down resistor R_{IN} = 70KΩ (typ.) R = 1KΩ (typ.)
P0 P3	Output				Push-pull output Initial "Low" High current I_{OL} = 7mA (typ.)
P1 P2	Output				Sink open drain output Initial "Hi-Z" High current I_{OL} = 20mA (typ.)
R4 R5 R6 R7	I/O				Sink open drain output Initial "Hi-Z" R = 1KΩ (typ.) Analog inputs (R4, R5) R_A = 5KΩ (typ.) C_A = 12pF (typ.)
R8 R9	I/O				Sink open drain output Initial "Hi-Z" Hysteresis input R = 1KΩ (typ.)
RA RB RC	I/O				Push-pull output Initial "High" R = 1KΩ (typ.)

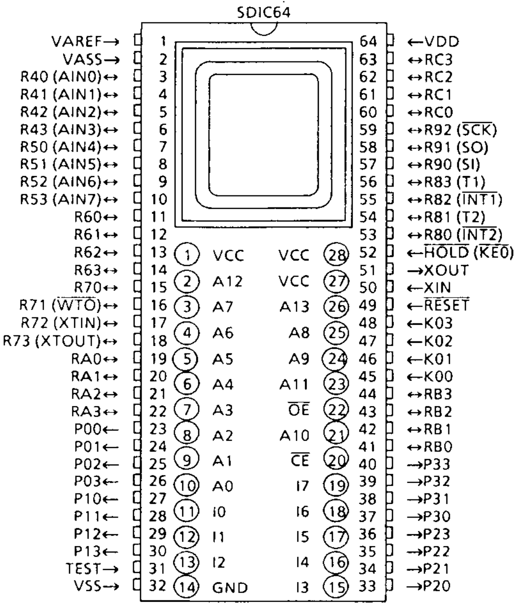
—This Page Intentionally Left Blank—

CMOS 4-BIT MICROCONTROLLER

TMP47C006E

The 47C006, which is equipped with an EPROM as program memory, is a piggyback type evaluator chip used for development and operational confirmation of the 47C1260/1660 application systems (programs).
The 47C006 is pin compatible with the 47C1260/1660 which are mask-programed ROM devices.

PIN ASSIGNMENT (TOP VIEW)



PIN FUNCTION (Top of the package)

PIN NAME	Input / Output	FUNCTIONS
A13 - A0	Output	Program memory address output
I7 - I0	Input	Program memory data input
$\overline{\text{CE}}$	Output	Chip enable signal output
$\overline{\text{OE}}$		Output enable signal output
VCC	Power supply	+ 5V (connected with VDD)
GND		0V (connected with VSS)

A.C. CHARACTERISTICS

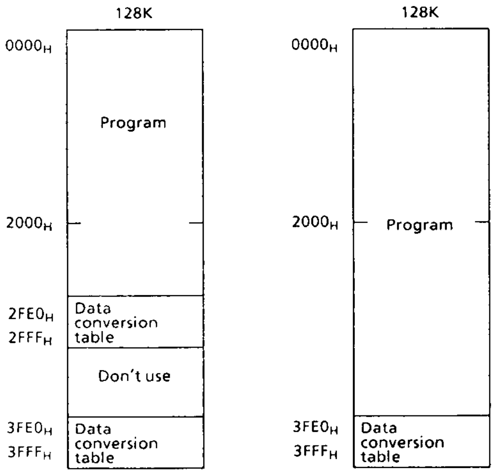
PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Address Delay Time	t_{AD}	V_{SS} = 0V, V_{DD} = 4.5~6.0V	–	–	150	ns
Data Setup Time	t_{IS}	C_L = 100pF	150	–	–	ns
Data Hold Time	t_{IH}	Topr = −40~70°C	50	–	–	ns

NOTES FOR USE

(1) Program memory

The program area are as shown in Figure 1.

When this chip is used as evaluator of the 47C1260, data conversion table for [OUTB @HL] instruction must be allocated at two areas and they must be the same contents as shown in Figure 1.



(a) 47C1260 (b) 47C1660

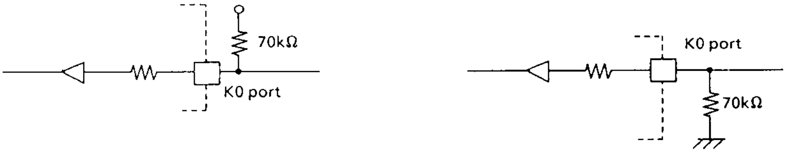
Figure1. Program area

(2) Data memory

47C006 contains three 256 x 4 bit data memory banks (bank0, bank1 and bank2).

(3) I/O ports

Input/Output circuitries of I/O ports in the 47C006 are similar to the code IA of the 47C1260/1660. When this chip is used as evaluator with other I/O code (IB and IC), it is necessary to provide the external resistors.



(a) Code : IB (b) Code : IC

Figure 2. I/O code and external circuitry