CMOS 4-BIT MICROCONTROLLER

TMP47C800N
TMP47C800F

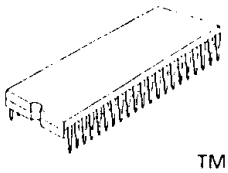
The 47C800 is a high speed and high performance 4-bit single chip microcomputer, integrating ROM, RAM, input/output ports, timer/counters, a serial interface, and two clock generators on a chip.
The 47C800 is the standard type device in the TLCS-470 series, and provides high current output capability for LED direct drive.

PART No.	ROM	RAM	PACKAGE	OTP	PIGGYBACK
TMP47C800N	8192 × 8-bit	512 × 4-bit	SDIP42	TMP47P800N	TMP47C980E
TMP47C800F			QFP44	TMP47P800F	

FEATURES

- ◆4-bit single chip microcomputer
- ◆Instruction execution time: 1.3μs (at 6MHz), 244μs (at 32.8kHz)
- ◆92 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 (36 pins)
  - Input 2 ports 5 pins
  - Output 2 ports 8 pins
  - I/O 6 ports 23 pins
- ◆Interval Timer
- ◆Two 12-bit Timer/Counters
  Timer, event counter, and pulse width measurement mode
- ◆Watchdog Timer
- ◆Serial Interface with an 8-bit buffer
  External/internal clock, leading/trailing edge shift, and 4/8-bit mode
- ◆High current outputs
  LED direct drive capability (typ. 20 mA × 8 bits).
- ◆Dual-clock operation
  High-speed/Low-power-consumption operating mode
- ◆Hold function
  Battery/Capacitor back-up
- ◆Real Time Emulator : BM47C800A

SDIP42

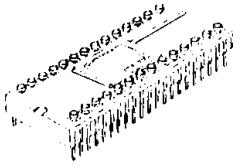


TMP47C800N
TMP47P800N

QFP44

TMP47C800F
TMP47P800F

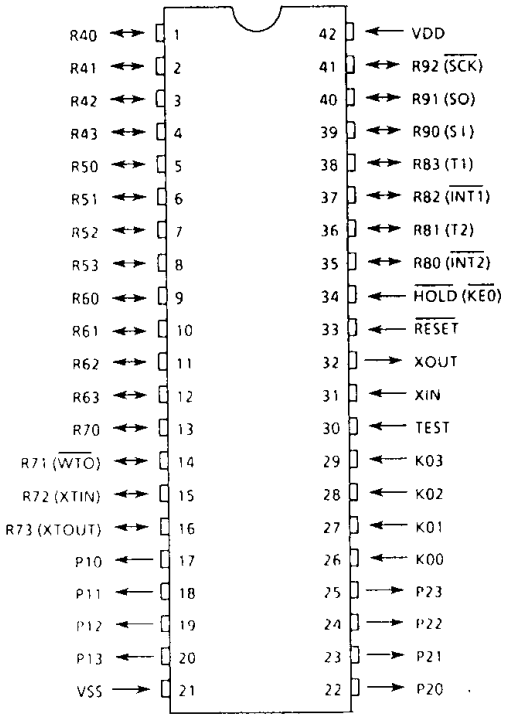
SDIC42



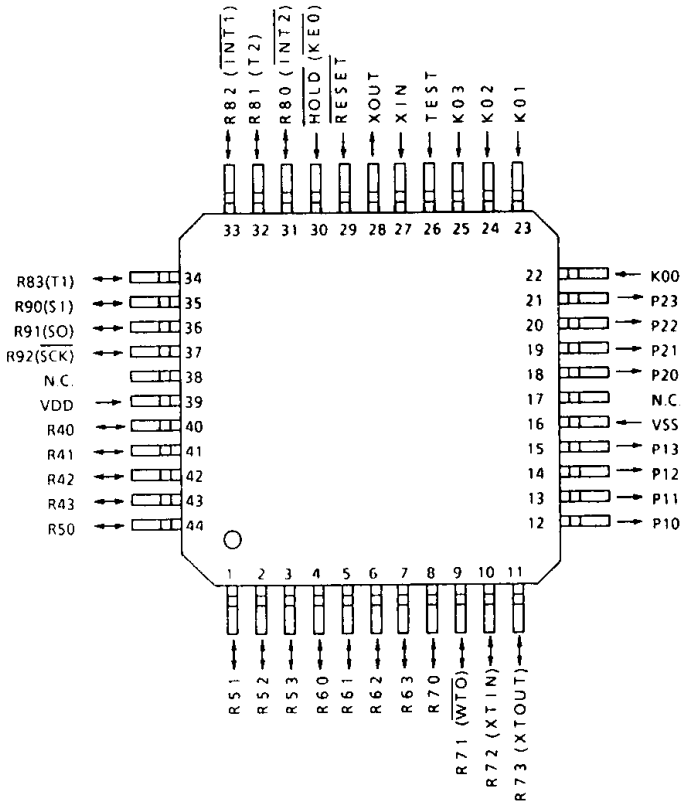
TMP47C980E

PIN ASSIGNMENTS (TOP VIEW)

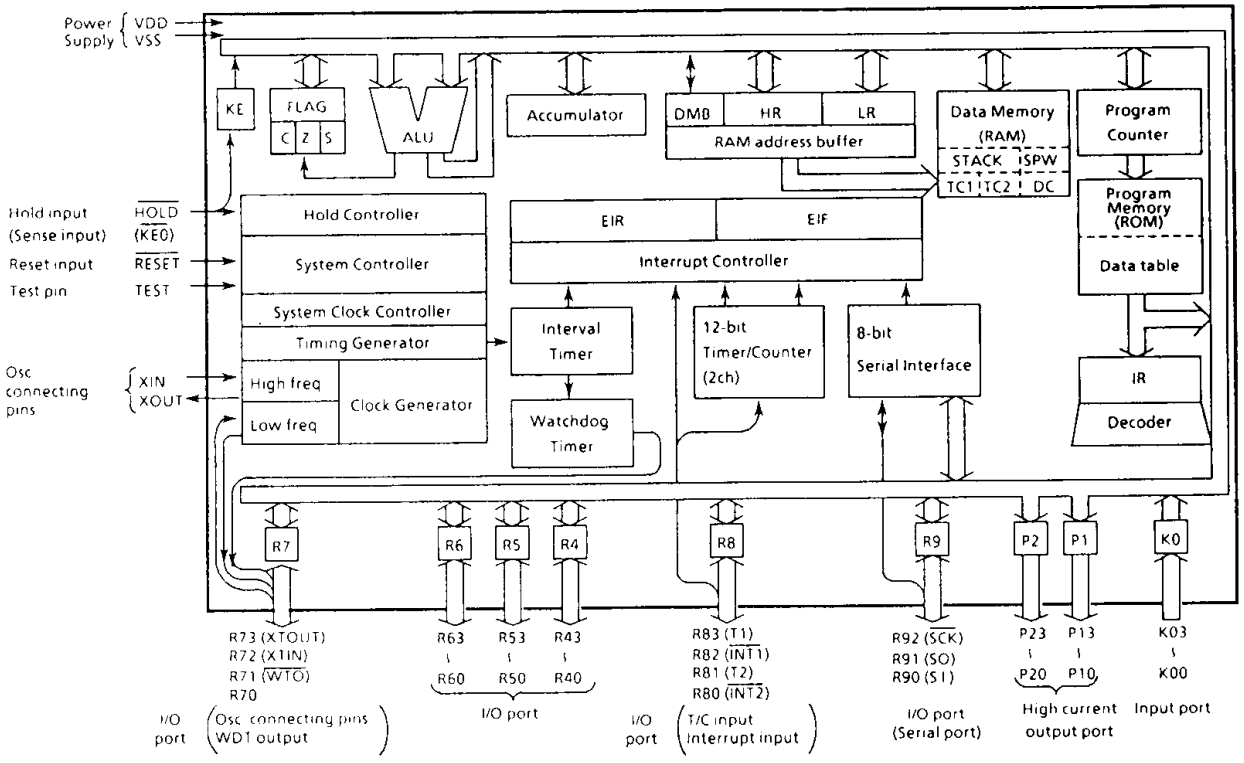
(1) SDIP42



(2) QFP44



BLOCK DIAGRAM



PIN FUNCTION

PIN NAME	Input/Output	FUNCTION	
K03 - K00	Input	4-bit input port	
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			
R43 - R40	I/O	4-bit I/O port with latch. When used as the input port, the latch must be set to "1".	
R53 - R50			
R63 - R60			
R73 (XTOUT)	I/O (Output)	4-bit I/O port with latch. When used as the input port or watchdog timer output pin, the latch must be set to "1".	Resonator connecting pins (Low-frequency)
R72 (XTIN)	I/O (Input)		
R71 ($\overline{WTO}$)	I/O (Output)		Watchdog timer output
R70	I/O		
R83 (T1)	I/O (Input)	4-bit I/O port with latch. When used as the input port, external interrupt input pin, or timer/counter input pin, the latch must be set to "1".	Timer/Counter 1 external input
R82 ($\overline{INT1}$)			External interrupt 1 input
R81 (T2)			Timer/Counter 2 external input
R80 ($\overline{INT2}$)			External interrupt 2 input
R92 ($\overline{SCK}$)	I/O (I/O)	3-bit I/O port with latch. When used as the 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
XIN	Input	Resonator connecting pins (High-frequency). For inputting external clock, XIN is used and XOUT is opened.	
XOUT	Output		
$\overline{RESET}$	Input	Reset signal input	
$\overline{HOLD}$ ($\overline{KE0}$)	Input (Input)	HOLD request/release signal input	Sense input
TEST	Input	Test pin for shipping. Be fixed to low level.	
VDD	Power Supply	+ 5V	
VSS		0V(GND)	

OPERATIONAL DESCRIPTION

1. SYSTEM CONFIGURATION

(1) 1Program Counter (PC)
(2) Program Memory (ROM)
(3) H Register, L Register, and Data Memory Bank Selector (DMB)
(4) Data Memory (RAM)
  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) Serial Interface
(13) Watchdog Timer
(14) Interrupt Controller
(15) Hold Controller
(16) Reset Circuit

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

2. INTERNAL CPU FUNCTION

2.1 Program Counter (PC)

The program counter is a 13-bit binary counter which indicates the address of the program memory storing the next instruction to be executed. Normally, the PC is incremented by the number of bytes of the instruction every time it is fetched. When a branch instruction or a subroutine instruction has been executed or an interrupt has been accepted, the specified values listed in Table 2-1 are set to the PC. The PC is initialized to "0" during reset.

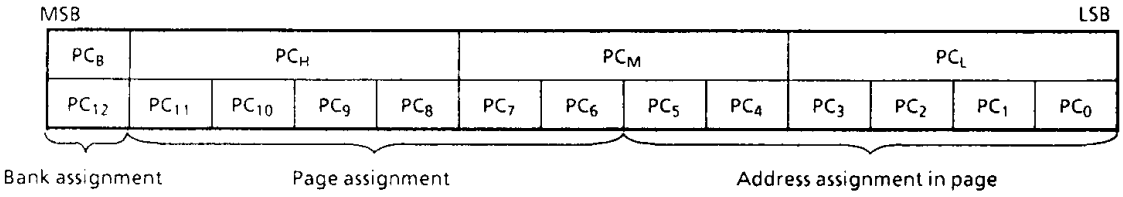


Figure 2-1. Configuration of Program Counter

The PC can directly address an 8192-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 status flag is "1", 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 7 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.

Instruction or Operation		Condition		Program Counter (PC)
				PC_{12} PC_{11} PC_{10} PC_9 PC_8 PC_7 PC_6 PC_5 PC_4 PC_3 PC_2 PC_1 PC_0
Instruction execution	BSL a	SF = 1 (Branch condition is satisfied)		Immediata data specified by the instruction
		SF = 0 (Branch condition is not satisfied)		+ 3
	BS a	SF = 1	Lower 12-bit address ≠ FFE, FFF_H	Hold / Immediata data specified by the instruction
			Lower 12-bit address = FFE, FFF_H (Last address in bank)	+ 1 / Immediata data specified by the instruction
		SF = 0		+ 2
	BSS a	SF = 1	Lower 6-bit address ≠ $3F_H$	Hold / Immediata data specified by the instruction
			Lower 6-bit address = $3F_H$ (Last address in page)	+ 1 / Immediata data specified by the instruction
		SF = 0		+ 1
	CALL a			0 0 / Immediata data specified by the instruction
	CALLS a			0 0 0 0 0 / The value generated by the immediate data specified by the instruction / 1 1 0
	RET			The return address restored from stack
	RETI			The return address restored from stack
	Others			Incremented by the number of bytes in the instruction
Interrupt acceptance				0 0 0 0 0 0 0 0 0 / Interrupt vector / 0
Reset				0 0 0 0 0 0 0 0 0 0 0 0 0

Table 2-1. Status Change of Program Counter

When first byte or second byte of this instruction is stored at the last address of the bank, the most significant bit 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 by the instruction is set to the PC. The address which can be specified by the instruction consists of 11 bits and the upper 2 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)

The 47C800 has 8192 × 8bits (addresses 0000_H through $1FFF_H$) of program memory (mask ROM).

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

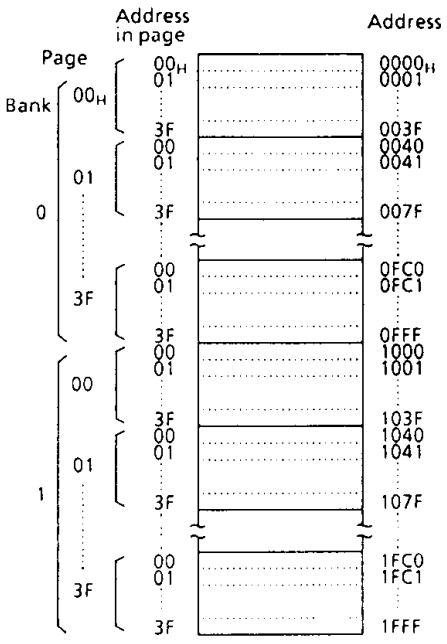


Figure 2-1. Configuration of Program Memory

(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 counter (DC) to place them into the accumulator. [LDL A,@DC] instruction reads the lower 4 bits of fixed data and [LDH A,@DC+] instruction reads the upper 4 bits.The DC is a 12-bit register, and it can specify an address within the range of 1000_H through $1FFF_H$ of the program memory.

(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 output the upper 4 bits to port P2 and the lower 4 bits to port P1. The table is located in the last 32-byte space (addresses $1FE0_H$ through $1FFF_H$) 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 elements.

Example : The following shows that the BCD data at address $2F_H$ in data memory is converted into the 7-segment code (e.g.,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 table is specified at addresses 1FE0H - 1FEFH)
OUTB    @HL
  :
ORG     1FE0H       ; 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_H through 0086_H and $1FE0_H$ through $1FFF_H$ of the program memory are also used for special purposes.

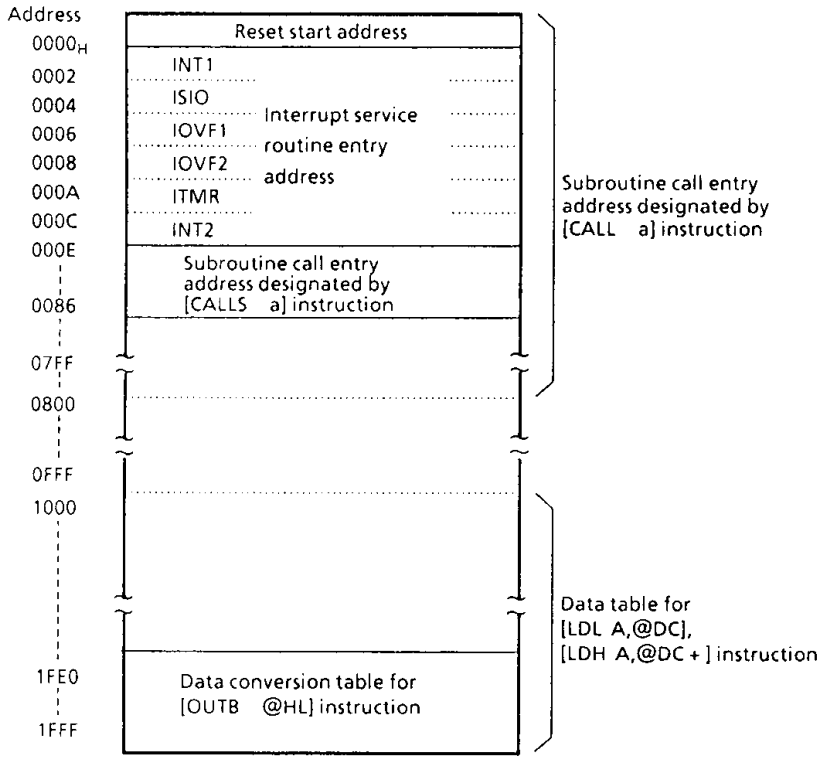


Figure 2-3. Program Memory Map

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 two banks (bank0 and bank1). The data memory bank selector (DMB) is a 1-bit register to specify a data memory bank. During reset, the DMB is initialized to "0". The DMB is set or cleared by the [CLR DMB] or [SET DMB] instructions. The currently selected data memory bank can be known by executing the [TEST DMB] or [TESTP DMB] 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.

```
CLR     DMB       ;  DMB ← 0
LD      HL,#10H   ;  HL ← 10H
ST      #5,@HL +  ;  RAM [10H] ← 5, ← 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
```

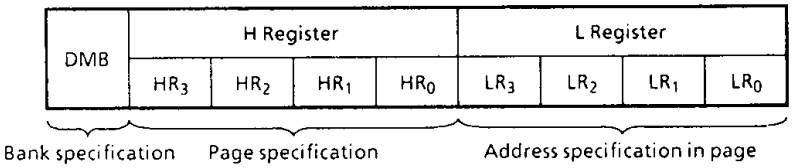


Figure 2-4. Configuration of H, L registers and DMB

2.4 Data Memory (RAM)

The 47C800 has a total of 512 × 4 bits of data memory (RAM), 256 × 4 bits (addresses 00_H through FF_H) on each of banks (bank0 and bank1).

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

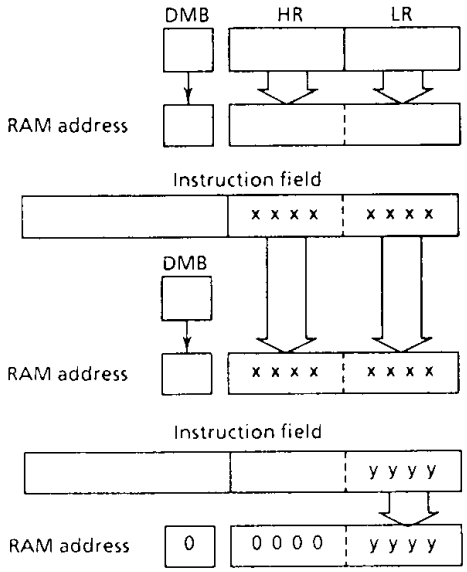


Figure 2-5. Addressing mode

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

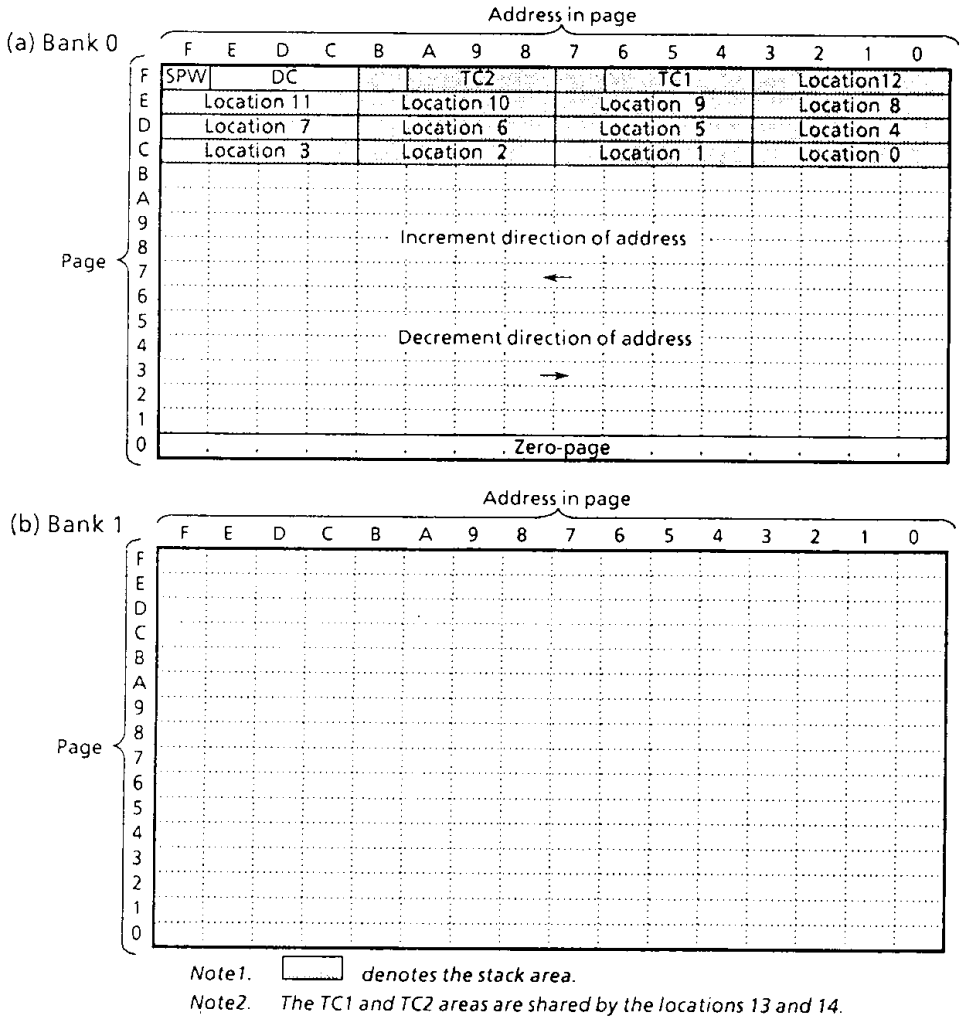
Example: To clear RAM

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

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



Note1. ▭ *denotes the stack area.*

Note2. *The TC1 and TC2 areas are shared by the locations 13 and 14.*

Figure 2-6. Data Memory Map

(1) Stack

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

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

The stack consists of up to 15 levels (locations 0 through 14) which are provided in the bank 0 of data memory (addresses $C0_H$ through FB_H). Each location consists of 4-word data memory. Locations 13 and 14 are shared by the count registers of the timer/counters (TC1, TC2) to be described later.

The save/restore locations in the stack are determined by the stack pointer word (SPW). The SPW is automatically decremented after save and incremented before restore. That is, the value of the 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 area 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 CF_H corresponding to the location 3 area is lost.)

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

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

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

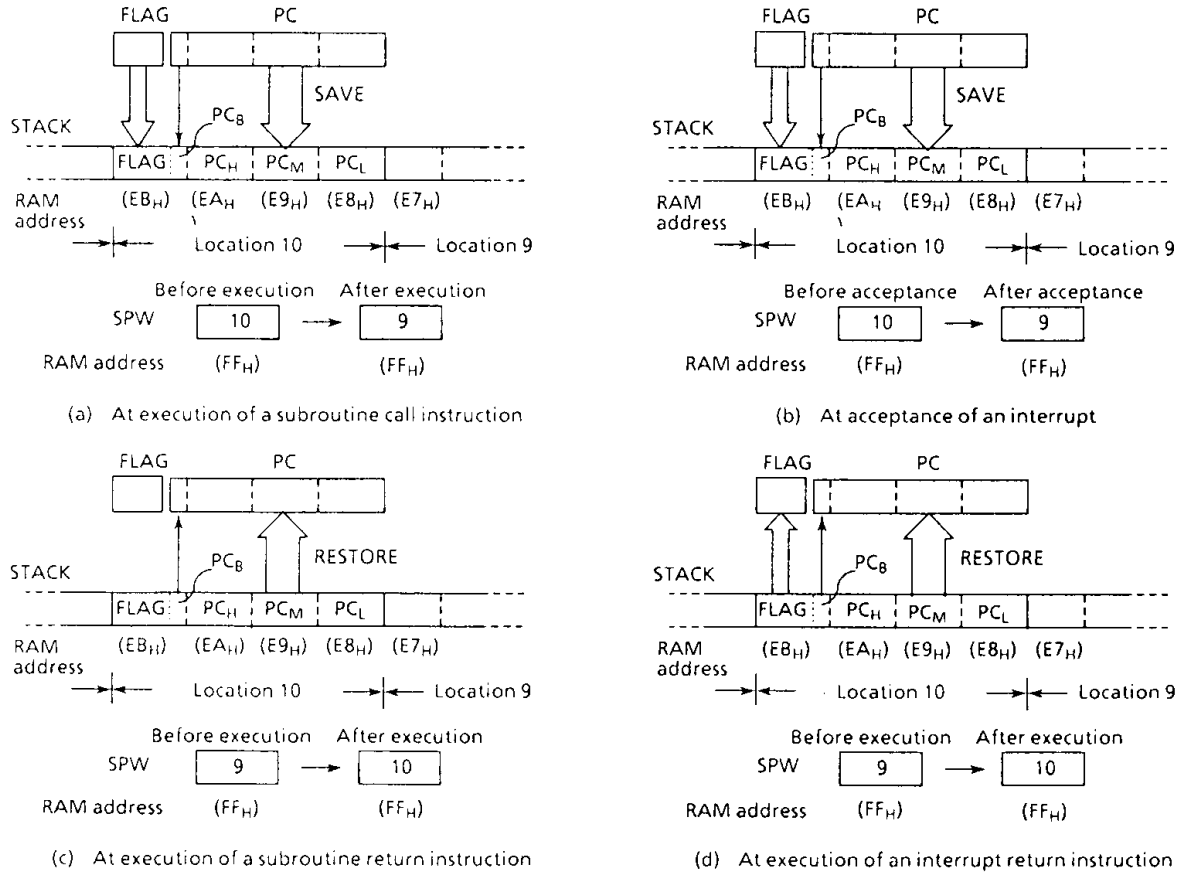


Figure 2-7. Accessing Stack (Save/Restore)

(3) 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 data tables are set in the program memory area between addresses 1000_H and $1FFF_H$. The DC is assigned with a RAM address in unit of 4 bits.

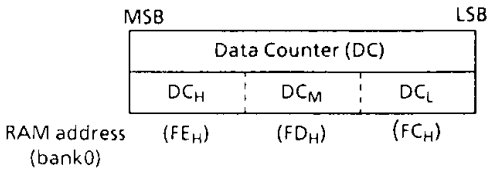


Figure 2-8. Data Counter

Therefore, the RAM manipulation instruction is used to set the initial value or read the contents of the DC.

Example: To set the DC to 780_H

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

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

The 47C800 has 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 manipulation 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.

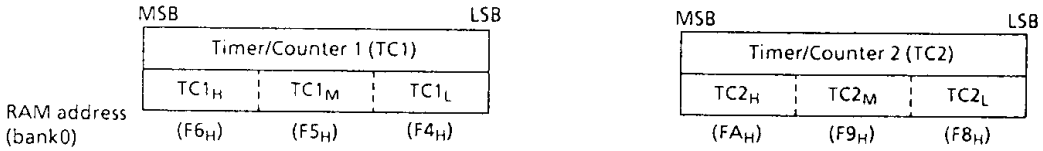


Figure 2-9. Count Registers of Timer/Counters (TC1,TC2)

(5) 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 is "0".
        B       SKIP
        ST      #8, 09H     ; Writes "8" to address 09H in the RAM.
SKIP:
```

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)

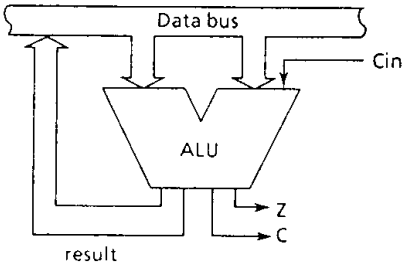
The carry information indicatets a carry-out from the most significant bit in an addtion. A subtraction is performed as addition of two's complement, so that, with a subtraction, the carry information indicates that there is no borrow to the most significant bit. With a rotate instruction, the information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

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



Note. Cin indicates the carry input specified by instruction.

Figure 2-10. ALU

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.

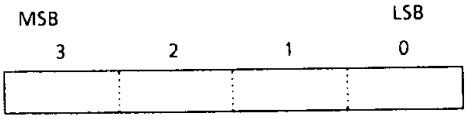


Figure 2-11. Accumulator

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.

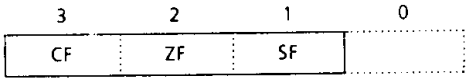


Figure 2-12. Flag

(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

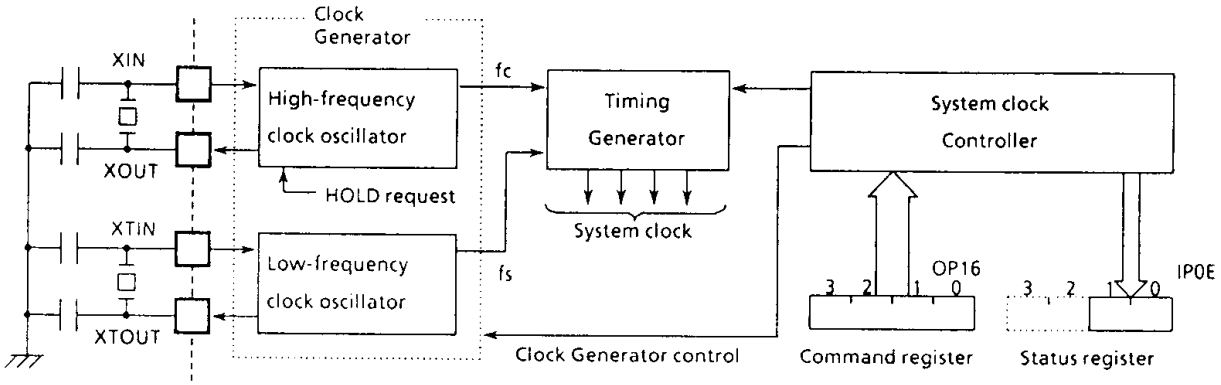


Figure 2-13. Clock Generator, Timing Generator and System Clock Controller

2.7.1 Clock Generator

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

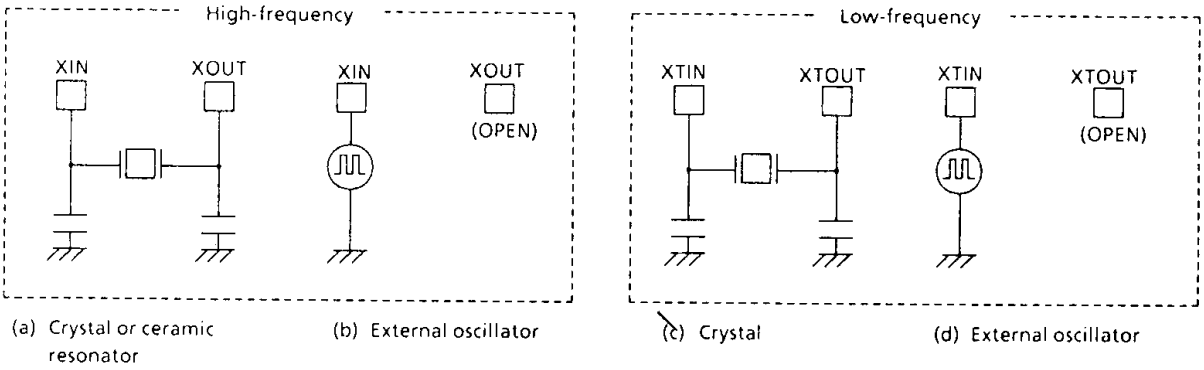


Figure 2-14. Examples of Resonator Connection

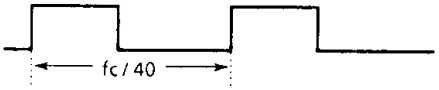
Note. Accurate adjustment of the oscillation frequency

Although no hardware to externally and directly monitor the clock pulse is not provided, the oscillation frequency can be adjusted by making the program to output the pulse with a fixed frequency to the port with the all interrupts disabled and timer/counters stopped and monitoring this pulse. With a system requiring the oscillation frequency adjustment, the adjusting program must be created beforehand.

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

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

Output wavefrom



2.7.2 Timing Generator

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 2-15 shows the operating mode transition diagram. Figure 2-16 shows the command and status registers.

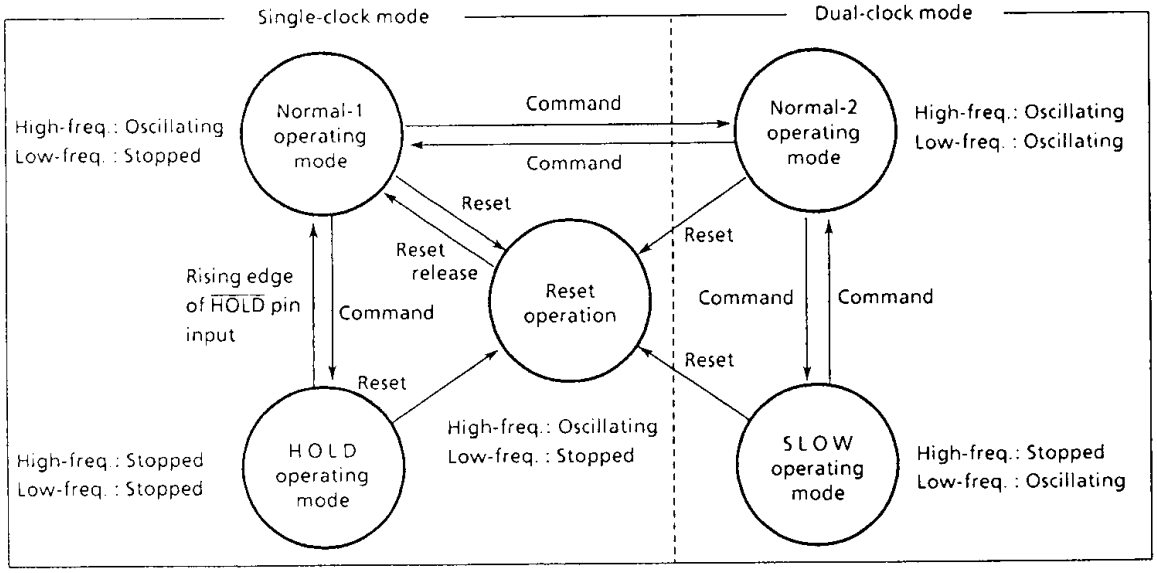


Figure 2-15. Operating Mode Transition Diagram

a. Single-clock mode

(1) Normal-1 operating mode

In this mode, the CPU and the peripheral hardware are operated on the high-frequency clock. At reset release, this mode is set.

(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.

b. Dual-clock mode

(1) Normal-2 operating mode

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

(2) SLOW operating mode

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

Notes.

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.*
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).

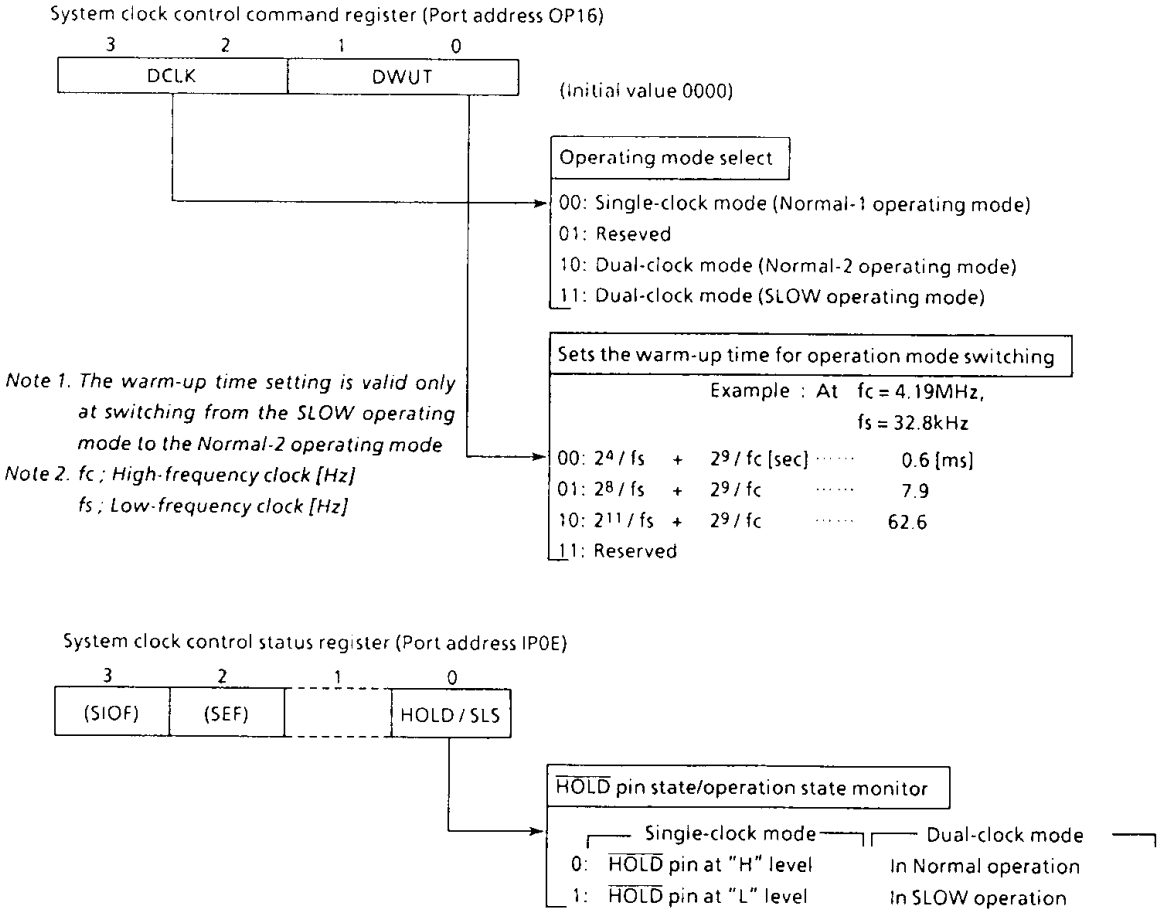


Figure 2-16. System Clock Control Command Register/Status Register

(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 performed. 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

Setting bit 2 of command register (OP16) to "1" stops the high-frequency clock oscillation and switches the system clock into low-frequency clock. Note that low-frequency clock oscillator takes a few seconds from the start of its oscillation to reach the stable oscillation. Therefore, switch to the SLOW operation mode immediately after the transition from Normal-1 to Normal-2 should be inhibited by program until stable low-frequency clock is supplied (for a few seconds as mentioned above).

b. Returning from SLOW operation to Normal-2 operation

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

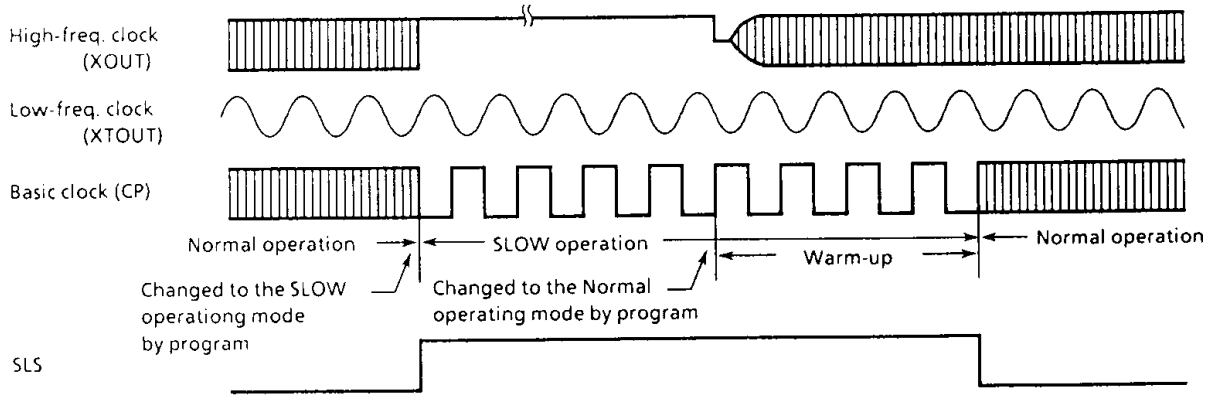


Figure 2-17. System Clock Switching Timing

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-470 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.

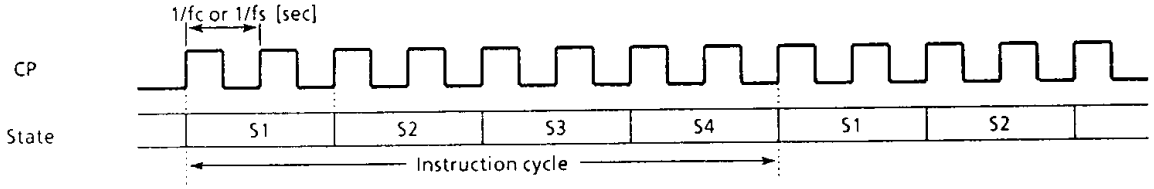


Figure 2-18. Instruction Cycle

3. PERIPHERAL HARDWARE FUNCTION

3.1 Ports

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

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

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

3.1.1 I/O Ports

The 47C800 has 10 I/O ports (36 pins) each as follows:

① K0	;	4-bit input
② P1, P2	;	4-bit output
③ R4, R5, R6	;	4-bit input/output
④ R7	;	4-bit input/output (shared with the low-frequency reasonator connecting pins and the watchdog timer output)
⑤ R8	;	4-bit input/output (shared with external interrupt request input and timer/counter input)
⑥ R9	;	3-bit input/output(shared with serial port)
⑦ KE	;	1-bit sense input (shared with hold request/release signal input)

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

Port Address (**)	Port		Input/Output instructions						
	Input (IP**)	Output (OP**)	IN %p, A IN %p, @HL	OUT A,%p OUT @HL,%p	OUT #k, %p	OUTB @HL	SET %p, b CLR %p, b	TEST %p,b TESTP %p,b	SET @L CLR @L TEST @L
00_H	K0 input port	——	○	–	–	–	–	○	–
01	P1 output latch	P1 output port	○	○	○	○ (Note 2)	○	○	–
02	P2 output latch	P2 output port	○	○	○	○	○	○	–
03	——	——	–	–	–	–	–	–	–
04	R4 input port	R4 output port	○	○	○	–	○	○	○
05	R5 input port	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	——	——	–	–	–	–	–	–	–
0B	——	——	–	–	–	–	–	–	–
0C	——	——	–	–	–	–	–	–	–
0D	——	——	–	–	–	–	–	–	–
0E	Status register (Note 3)	——	○	–	–	–	–	○	–
0F	Serial receive buffer	Serial transmit buffer	○	○	○	–	–	–	–
10_H	Undefined	Hold operating mode control	–	○	–	–	–	–	–
11	Undefined	——	–	–	–	–	–	–	–
12	Undefined	——	–	–	–	–	–	–	–
13	Undefined	——	–	–	–	–	–	–	–
14	Undefined	——	–	–	–	–	–	–	–
15	Undefined	Watchdog Timer control	–	○	–	–	–	–	–
16	Undefined	System clock control	–	○	–	–	–	–	–
17	Undefined	——	–	–	–	–	–	–	–
18	Undefined	——	–	–	–	–	–	–	–
19	Undefined	Interval Timer interrupt control	–	○	–	–	–	–	–
1A	Undefined	——	–	–	–	–	–	–	–
1B	Undefined	——	–	–	–	–	–	–	–
1C	Undefined	Timer/Counter 1 control	–	○	–	–	–	–	–
1D	Undefined	Timer/Counter 2 control	–	○	–	–	–	–	–
1E	Undefined	Serial interface control 1	–	○	–	–	–	–	–
1F	Undefined	Serial interface control 2	–	○	–	–	–	–	–

Note 1. *"——" means the reseved state. Unavilable 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. *The status input of serial interface, clock generator, and $\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$) pin.*

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

(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.

Figure 3-1. Port K0

(2) 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
```

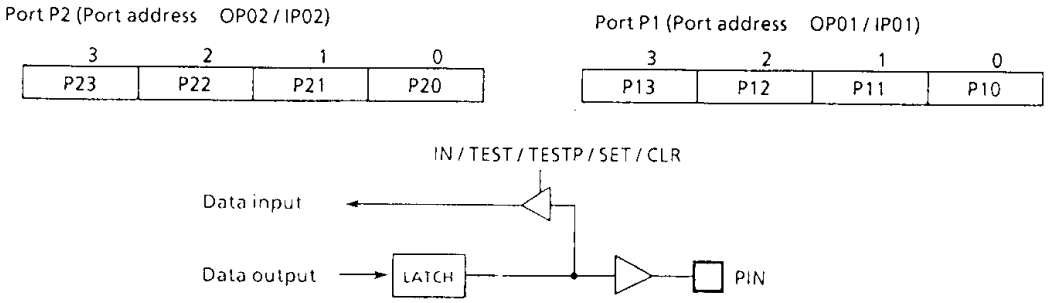


Figure 3-2. Ports P1, P2

(3) 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

Port R7 is shared by the low-frequency resonator connection pins (XTIN, XTOUT) and the watchdog timer output pin ($\overline{WTO}$). 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 ($\overline{WTO}$) 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").

Ports R4-R6 (Port address OP0•/IP0•)

3	2	1	0
R•3	R•2	R•1	R•0

•; 4 to 6

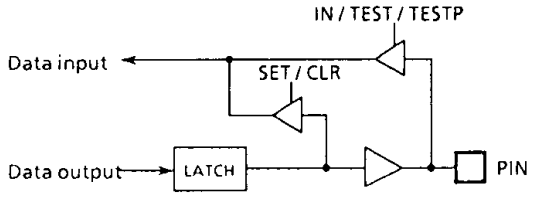


Figure 3-3. Ports R4, R5, R6

Port R7 (Port address OP07/IP07)

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

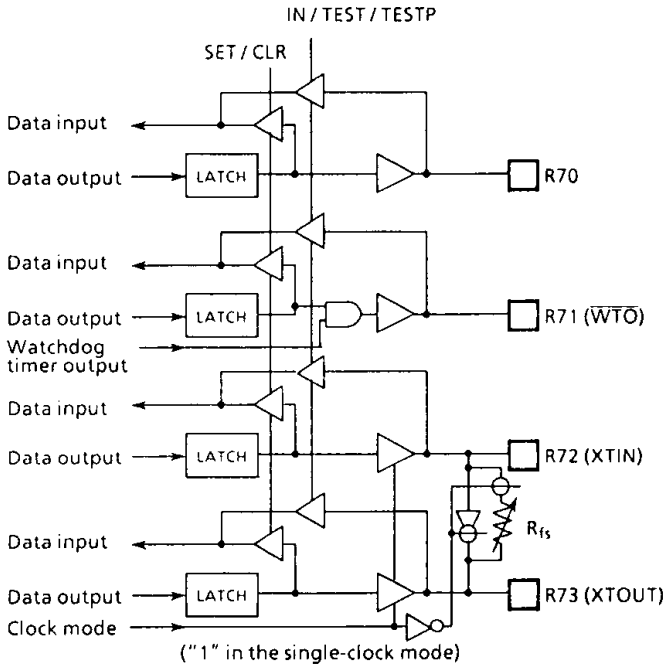


Figure 3-4. Port R7

(4) 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 ($\overline{INT1}$) 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 ($\overline{INT2}$) 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 ($\overline{INT1}$)	R81 (T2)	R80 ($\overline{INT2}$)

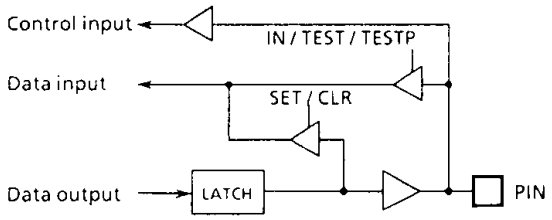


Figure 3-5. Port R8

(5) Port R9 (R92 - R90)

Potr 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 ($\overline{SCK}$)	R91 (SO)	R90 (SI)

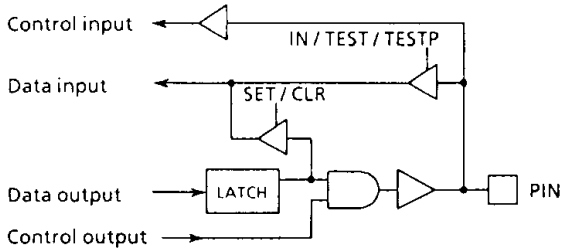


Figure 3-6. Port R9

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

Port KE is a 1-bit sense input port shared by the hold request/release signal input pin ($\overline{HOLD}$). This input port is assigned to the least significant bit of port address IP0E and is processed as the data with inverted polarity. For example, if an input instruction is executed with the pin on the high level, "0" is read. Note that $\overline{KE0}$ pin cannot be used in the dual-clock mode.

Example: To wait until $\overline{KE0}$ pin goes low.

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

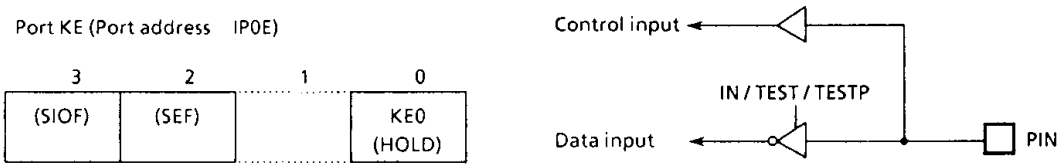


Figure 3-7. Port KE

3.1.2 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.

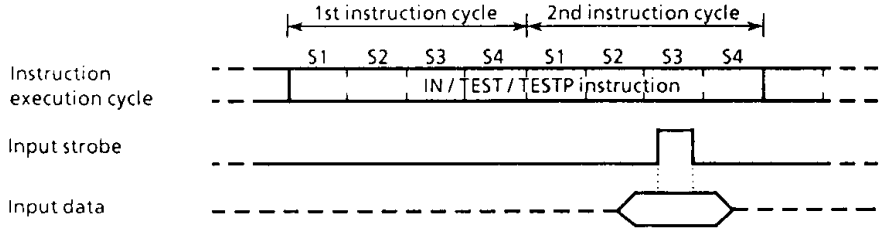


Figure 3-8. Input Timing

(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.

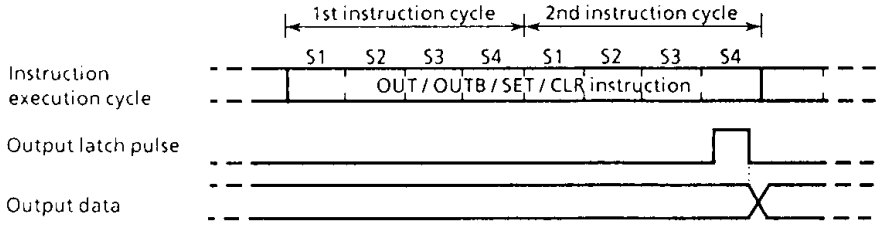


Figure 3-9. Output Timing

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-8 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.

① In the single-clock mode

The high-frequency clock (fc) is applied to the first stage of the interval timer.

② In the dual-clock mode

In this mode, the interval timer is split between stages 7 and 8. The low-frequency clock (fs) is applied to the 8th stage. The high-frequency clock (fc) is applied to the first stage of the interval timer. In the SLOW operating mode, the high-frequency clock oscillator stops, so that the divided outputs of the interval timer stage 1 through 7 also stop.

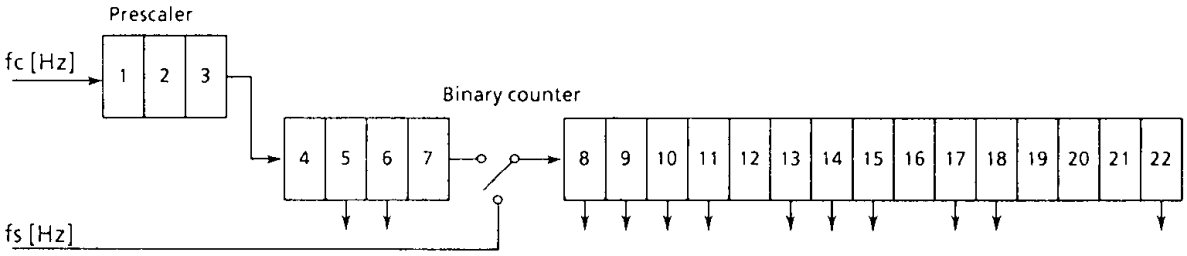


Figure 3-10. Configuration of Interval Timer

3.2.2 Functions of Interval Timer

The interval timer provides the following functions:

① Generation of instruction cycle
② Generation of an interrupt with a fixed frequency (the interval timer interrupt)
③ Generation of internal source clock for timer/counters
④ Generation of internal serial clock for a serial interface
⑤ Generation of warm-up time at release of the HOLD operating mode
⑥ 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 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 addres OP19)

3	2	1	0

(initial value 0000)

	Single-clock mode: Normal-1		Dual-clock mode: Normal-2	Dual-clock mode: 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

Note 1. • ; *don't care*
Note 2. *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 3-11. Interval Timer Interrupt Control Command Register

3.3 Timer/Counters (TC1, TC2)

The 47C800 has 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 user-processed data.

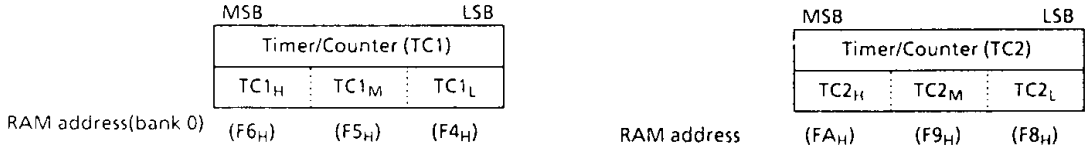


Figure 3-12. Count Registers of the Timer/Counters (TC1, TC2)

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.

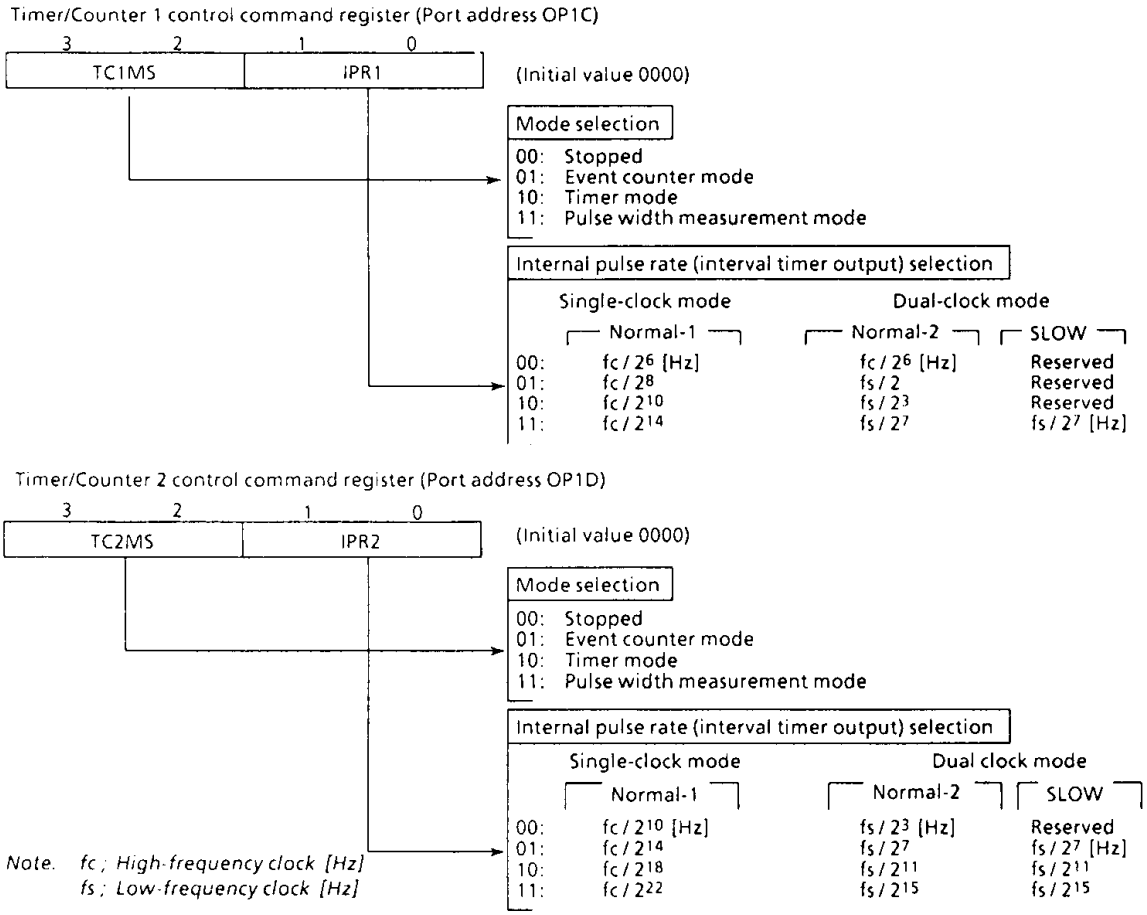


Figure 3-13. Timer/Counter Control Command Register

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 one instruction cycle after the current instruction execution, during which the execution of a next instruction and the acceptance of an interrupt are delayed. If counting is requested by both TC1 and TC2 simultaneously, the request by TC1 is preferred. The request 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 3-14. Note that counting continues if there is a count request after overflow occurrence.

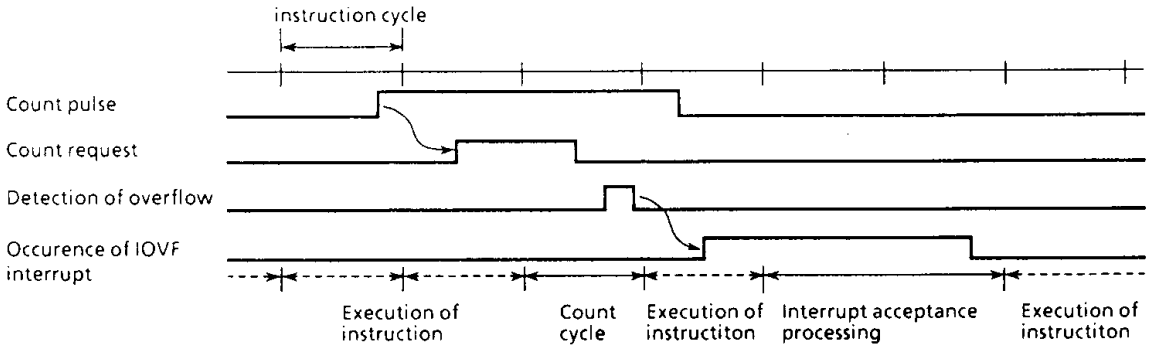


Figure 3-14. Timer/Counter Overflow Interrupt Timing

(1) Event counter mode

In the event counter mode, the timer/counter increments at each rising edge of the external pin (T1, T2) input. 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 3.6μs.

Example: To operate TC2 in the event counter mode.

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

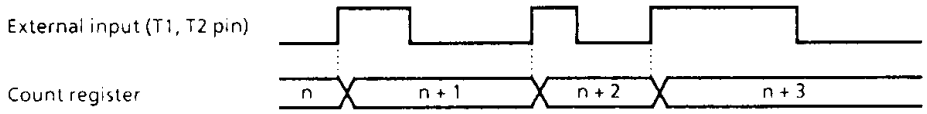


Figure 3-15. Event Counter Mode Timing Chart

(2) Timer mode

In the timer mode, the timer/counter increments as 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 create a desired time interval.

When an internal pulse rate of $fc/2^{10}$ is used, a count operation is inserted once every 128 instruction cycles, so that the apparent instruction execution speed drops by (1/127) × 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)
CLR     DMB
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
```

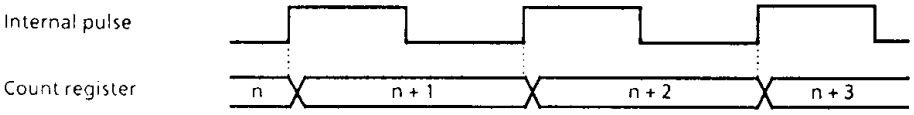


Figure 3-16. Timer Mode Timing Chart

Calculating the preset value of the count register

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

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

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

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

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 3-17 the timer/counter increments only while the external pin input is high. The maximum applied frequency to the external pin input must be one that is enough for analyzing the count value. Normally, a frequency sufficient slower than the internal pulse rate setting is applied to the external pin.

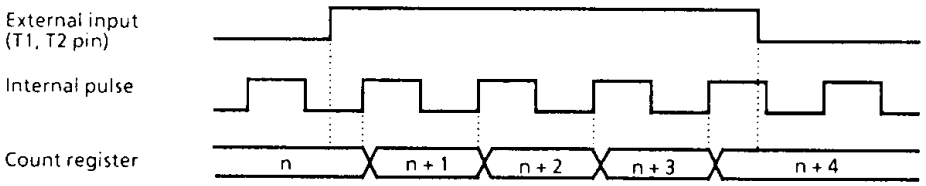


Figure 3-17. Pulse Width Measurement Mode Timing Chart

3.4 Serial Interface (SIO)

The 47C800 contains a serial interface with an 8-bit buffer. The serial interface is connected to the exterenal device via 3 pins (the serial port): R92 ($\overline{SCK}$), R91 (SO), and R90 (S I). 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.

3.4.1 Configuration of Serial Interface

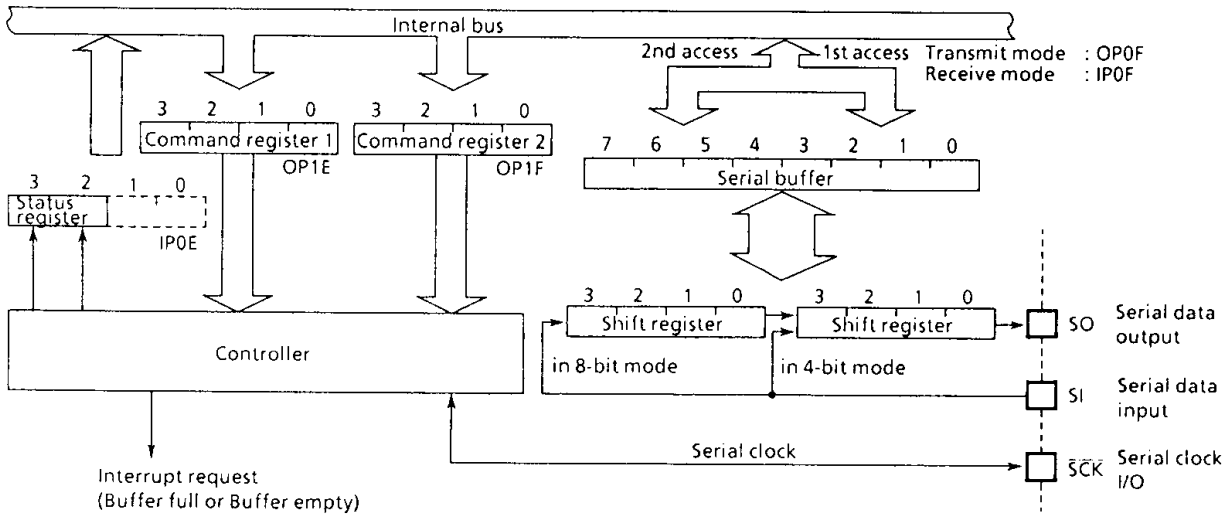


Figure 3-18. Configuration of Serial Interface

3.4.2 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).

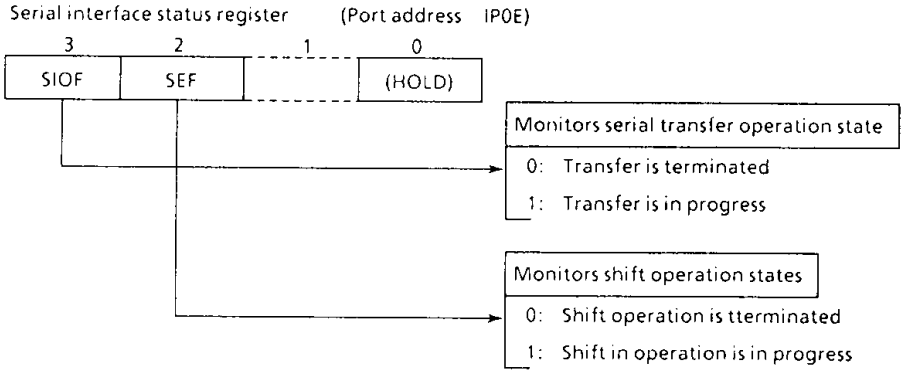


Figure 3-19. Serial Interface Status Register

Serial interface control command register 1 (Port address OP1E)

3	2	1	0
INH	SBIT	SCKR	

(Initial value 0001)

Forcible stop of serial transfer
- 0: Transfer continues
- 1: Forcible stop (Automatically clear, after stopped)

Selects transfer bit number
- 0: 4-bit serial transfer
- 1: 8-bit serial transfer

Selects serial clock frequency

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

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

Serial interface control command register 2 (Port address OP1F)

3	2	1	0
ESIO	RM	LM	ECKM

(Initial value 0000)

Instructs serial transfer start/terminate
- 0: Instructs serial transfer terminate
- 1: Instructs serial transfer start

Selects transfer mode
- 0: Transmit mode
- 1: Receive mode

Selects shift edge
- 0: Shift at the trailing edge of serial clock
- 1: Shift at the leading edge of serial clock

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

Note. When setting the transfer mode, ESIO must be "0".
When Transmit mode, LM must be "1"

Figure 3-20. 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 register:

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's high and low levels needs 2 instruction cycles or more to be completed.

b. Shift edge selection

① Leading edge

Data 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).

a. Transmit mode

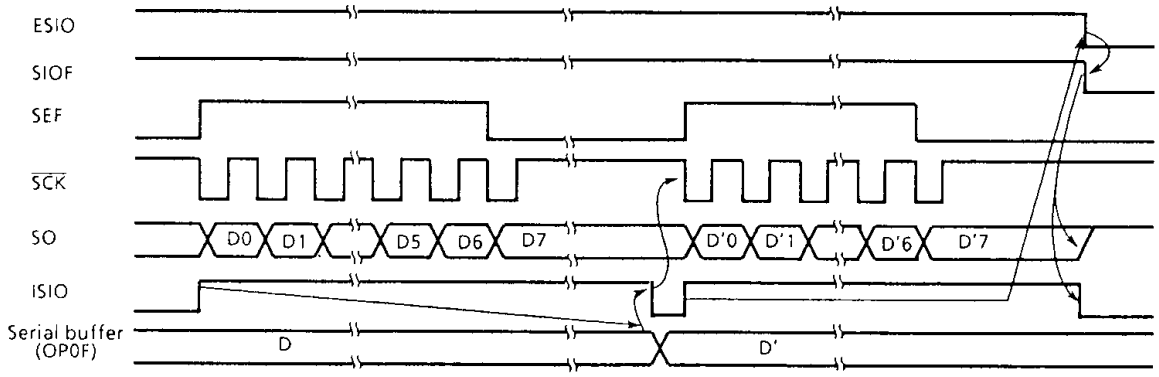
The transmit mode is set to the command register then writes the first transmit data (4bits or 8bits) to the buffer register (OP0F). If the transmit mode is not set, the data is not written to the buffer register.

In the 8-bit transfer mode, the 8-bit data is written by accessing the buffer register (OP0F) twice. 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 to the buffer register, the interrupt request is reset.

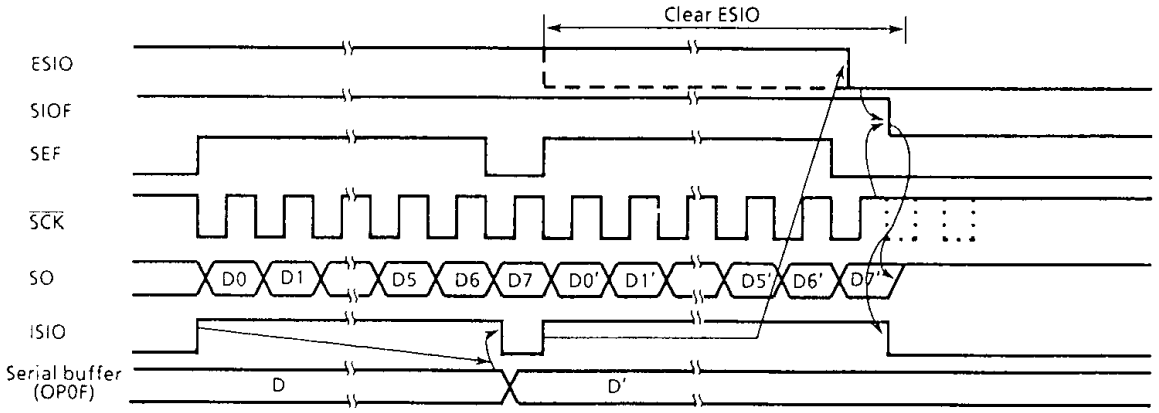
In the operation based on the internal clock, if no more data is set after the transmission of the 4-bit or 8-bit data, the serial clock is stopped and the wait state sets in. In the operation based on the external clock, the data must be set in the buffer register by the time the next data shift operation 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 transmit 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 of the
                              operation based on internal clock)
        OUT     A,  %OP1F
        OUT     @HL, %OP0F    ; OP0F←RAM[HL] (Writes the transmit data of
                              lower 4-bits)
        INC     L
        OUT     @HL, %OP0F    ; OP0F←RAM[HL] (Writes the transmit data of
                              upper 4-bits)
        LD      A,  #1010B    ; ESIO←1 (Instructs transmission start)
        OUT     A,  %OP1F
```



(a) Internal-clock-based operation with wait



(b) External-clock-based operation

Figure 3-21. Transmit Mode

b. Receive mode

Data can be received when ESIO is set to "1" after setting the receive mode to the command register. The data is put from the SI pin to the shift register in synchronization with the serial clock. Then the 4/8-bit data is transferred from the shift register to the buffer register (IP0F), upon which the buffer full interrupt (ISIO) to request for reading received data is generated. The receive data is read from the buffer register by the interrup service program. In the 8-bit transfer mode, the received data is read after an interrupt request occurs: the lower 4 bits are read by the first access and the upper 4 bits by the next access. 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-supplied clock, 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 trailing edge. In the leading edge 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 8-bit serial transfer, internal clock and leading edge shift (with the interrupt enable register already set).

```
LD      A,  #0100B      ;  OP1E←0100B (Sets the 8-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 or 8-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" at the end of receive operation.

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.

Example : To instruct the receive end with sufficient low transfee rate (the leading edge shift).

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

② When the transfer rate is sufficiently high (the external/internal 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.

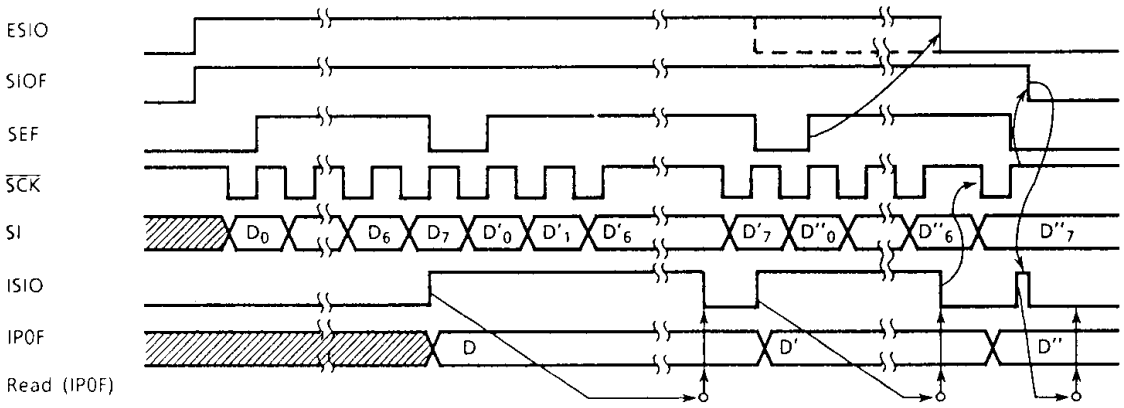
Example : To instruct the start/end of 1-word reception (the internal clock, the trailing edge shift).

```
        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)
        OUT     A,  %OP1F
SSEF0 : TEST    %IP0E,  2       ; Confirms that SEF = "1"
        B       SSEF0
        LD      A,  #0110B      ; ESIO←0 (Instructs reception end)
        OUT     A,  %OP1F
```

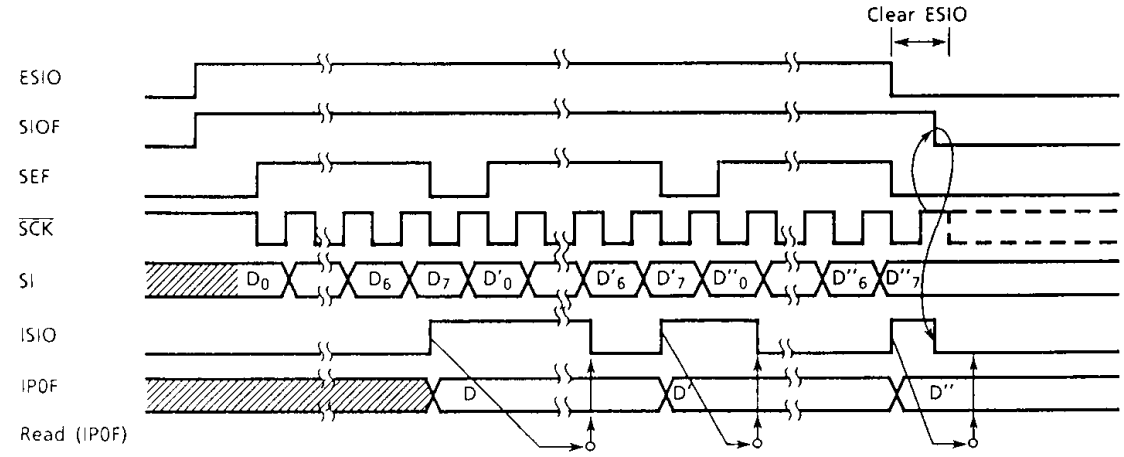
(4) Stopping serial transfer

A serial transfer operation can be stopped forcibly.

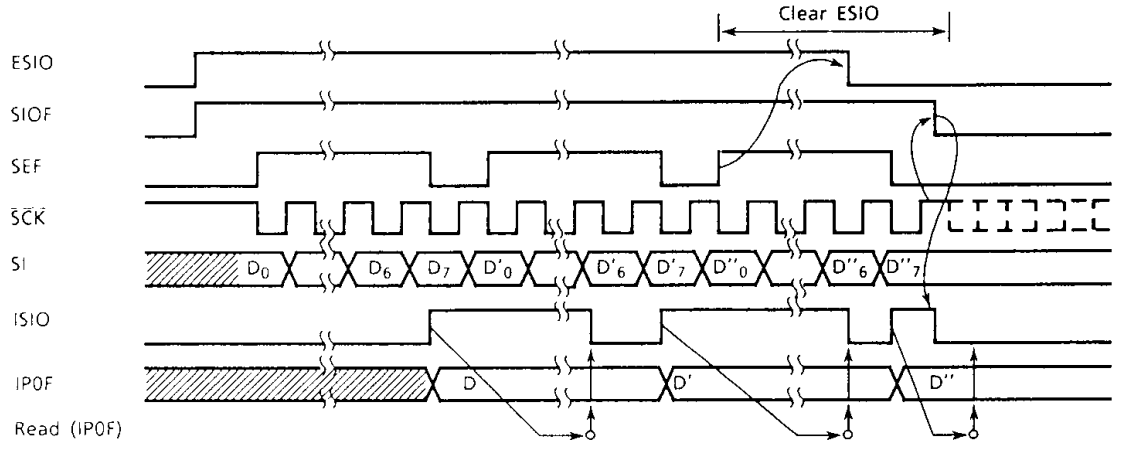
It is stopped by setting INH (bit 3 of command register 1) to "1", clearing the shift counter. When the serial transfer is over, $\overline{SCK}$ and SO terminal are set to "H"level, INH is automatically cleared to "0" with no other bits of command register affected. Further, in the transmit mode of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



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

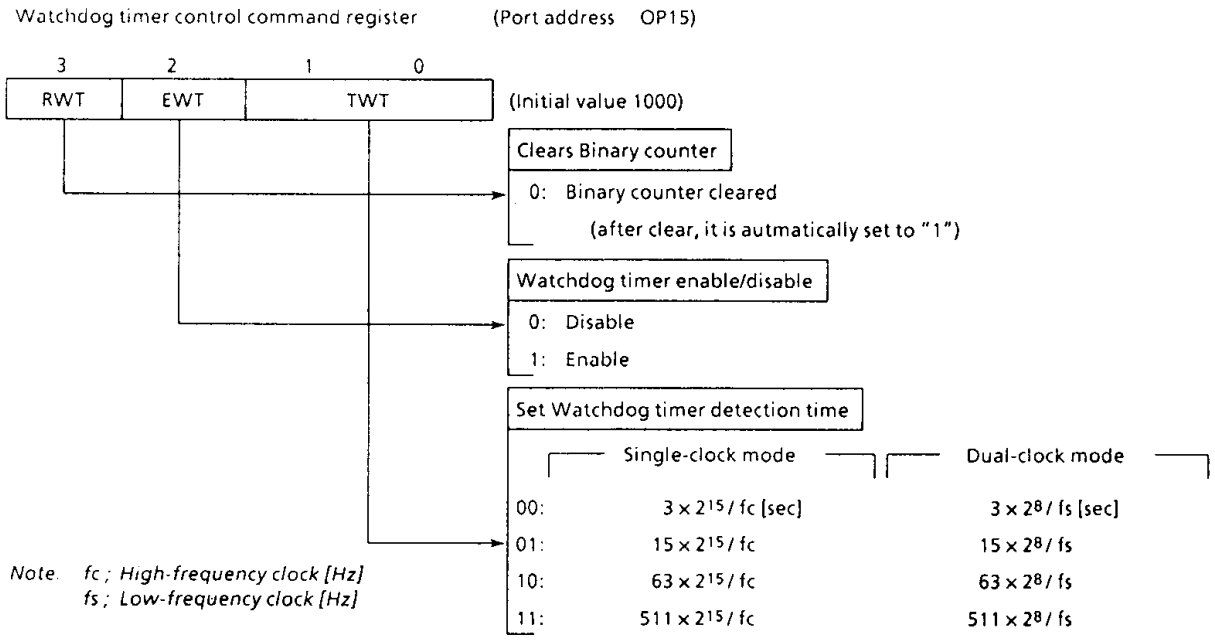


(b) External-clock-based operation, leading-edge shift (when transfer rate is low)



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

Figure 3-22. Receive Mode



(a) Command register

Single-clock mode	Dual-clock mode	At fc = 4.194304MHz, fs = 32.768KHz
3×2^{15} / fc [sec]	3×2^{8} / fs [sec]	23.4 ms
15×2^{15} / fc	15×2^{8} / fs	117.2
63×2^{15} / fc	63×2^{8} / fs	492.2
511×2^{15} / fc	511×2^{8} / fs	3992.2

(b) Watchdog timer detection time example

Figure 3-24. Watchdog Timer Control Command Register

Example : To enable the WDT with a detection time of 63×2^{15}/fc[sec].

```
        LD      A,  #0010B          ; OP15←0010B (Sets WDT detection time ,
                                      clears binary counter)
        OUT     A,  %OP15
        LD      A,  #1110B          ; OP15←1110B (Enables WDT)
        OUT     A,  %OP15
          :
          :                                  Within WDT
          :                                  detection
        LD      A,  #0110B          ; OP15←0110B (Clears binary counters)   time
        OUT     A,  %OP15
          :
          :
```

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			Priority	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

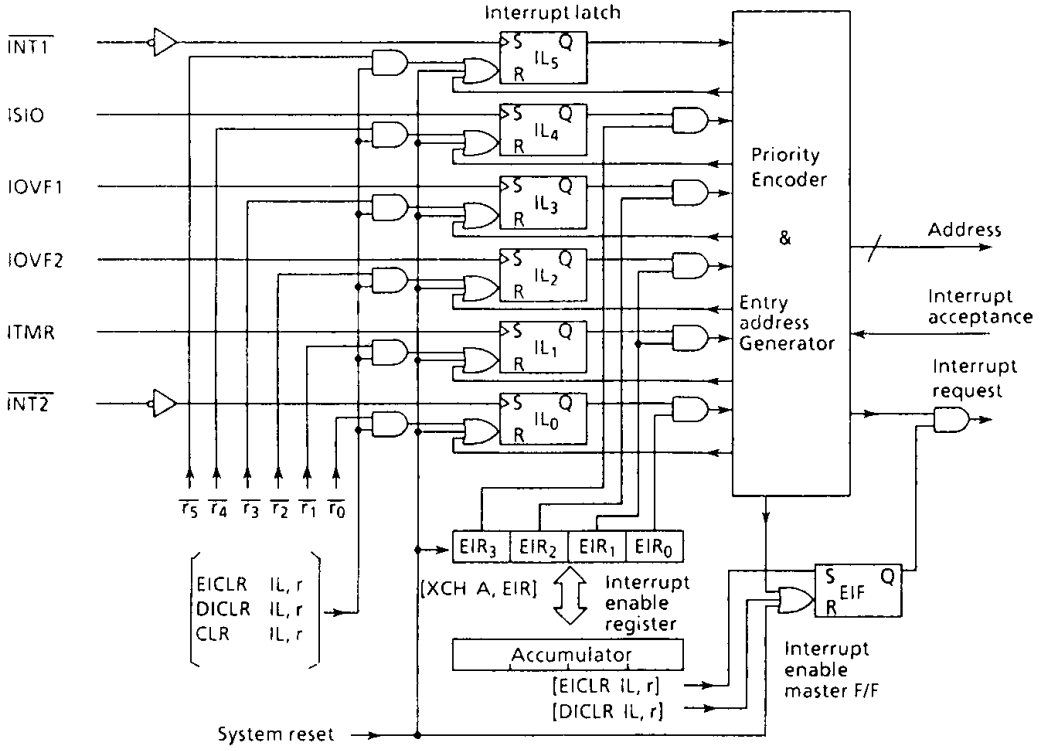


Figure 4-1. Interrupt Controller Block Diagram

(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 through IL_0)

An interrupt latch 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 interrupt latch 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

```
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].

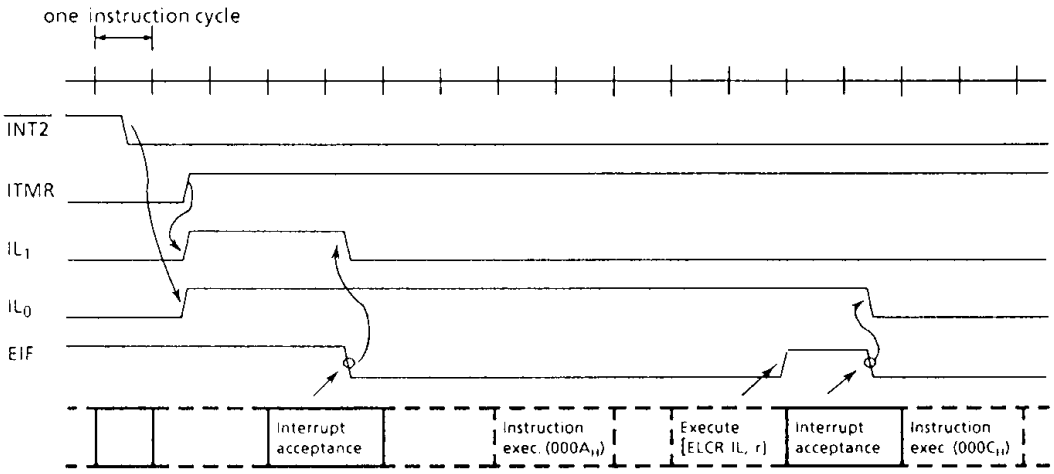
The interrupt acknowledge processing consists of the following sequence:

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_H through $0FFF_H$.)

To perform the multi-interrupt, 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      ; If RAM [05H]0 = 1,returns without the interrupt service
         B      SINT1
         RETI
SINT1 :   :
```



Note 1. *It is assumed that there is no other interrupt request and EIR = 0011_B.*
Note 2. *The value r in the [EICLR IL, r] instruction is assumed as 11111_B.*
Note 3. ⌐ ¬ *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.

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's".

Example 2 : To save DMB to bit 0 of address GSAV2 in the RAM.

```
         CLR    GSAV2, 0     ; RAM[GSAV2] 0←0
         TEST   DMB          ; If DMB = 0 then skip
         B      SKIPS
         SET    GSAV2, 0     ; RAM[GSAV2] 0←1
SKIPS:
```

Example 3 : Restore DMB from bit 0 of address GSAV3 in the RAM.

```
        CLR     DMB         ; DMB←0
        TEST    GSAV3, 0    ; If RAM [GSAV3]0 = 0 then skip
        B       SKIPR
        SET     DMB         ; DMB←1
SKIPR:
```

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 ($\overline{INT1}$ or $\overline{INT2}$).

Because the exterenal interrupt input is the hysteresis type, each of high and low level time requires two or more instruction cycles for a correct interreupt operation. The INT1 interrupt cannot be disabled by the EIR, so that it is always accepted in the interrupt enable state (EIF = "1").

Therefore, INT1 is used for an interrupt with high priority such as an emergency interrupt.

When R82($\overline{INT1}$) pin is used for the I/O port,the INT1 interrupt occurs at the falling edge of the pin input, so that the [RETI] instruction must be stored at the interrupt entry address to perform dummy interrupt processing.

5. POWER SAVING FUNCTION

The 47C800 provides 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 system's internal states active before stop with a low power. The HOLD operation is controlled by the command register (OP10) and the $\overline{\text{HOLD}}$ pin input. The $\overline{\text{HOLD}}$ pin input state can be known by the status register (IP0E). The $\overline{\text{HOLD}}$ pin is shared with the $\overline{\text{KE0}}$ pin.

5.1.1 Starts the HOLD operating mode

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

① Oscillator stops and the system's 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 which starts the HOLD operating mode.

HOLD oprerating mode command register (Port address OP10)

3	2	1	0
HLDMS		HWUT	

(initial value *0**)

Sets mode and starts HOLD operation
- 01: Starts HOLD operation in edge sensitive release mode
- 11: Starts HOLD operation in level sensitive release mode
- *0: Reserved

Sets the warm-up time at release of HOLD operating 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

Note 1. * ; *don't care*
Note 2. fc ; *High-frequency clock [Hz]*

HOLD opeating mode status register (Port address IP0E)

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

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

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 HOLD operating is released by setting the $\overline{HOLD}$ pin to the high level. This mode is used for the capacitor backup with the main power off or for the battery backup for long hours.

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

① Testing $\overline{HOLD}$ (bit 0 of the status register)

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

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

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

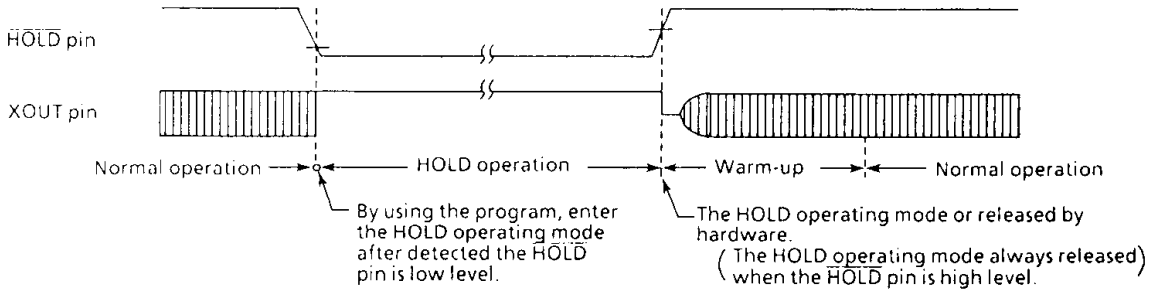


Figure 5-2. Level-sensitive Release Mode

(2) Edge-sensitive release mode

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

In the edge-sensitive release mode, even if the $\overline{HOLD}$ pin input is high, the HOLD operation is performed.

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

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

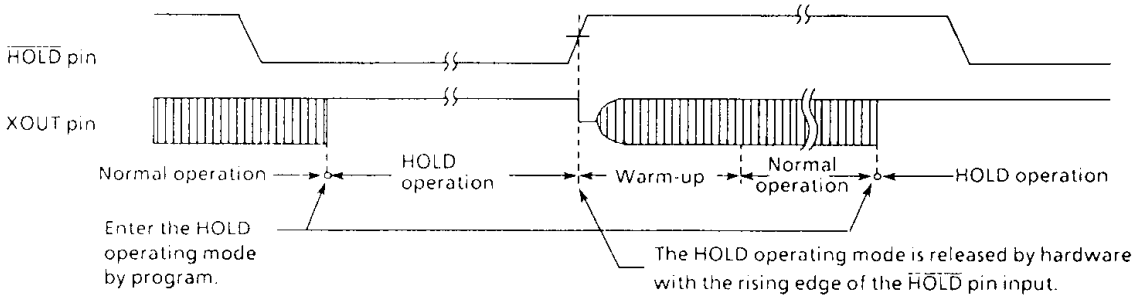


Figure 5-3. Edge-sensitive Release Mode

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 port's 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 oscillator starts.
(2) Warm-up is performed to acquire the time for stabilizing oscillation. During the warm-up, the internal operations are all stopped. One of three warm-up times can be selected by program depending on the characteristics of the oscillator used.
(3) When the warm-up time has passed, an ordinary operation restarts from the instruction next to the instruction which starts the HOLD operation. At this time, the interval timer starts from the reset state "0".

Note. 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 $\overline{\text{RESET}}$ pin to the low level. In this case, the normal reset operation follows immediately.

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

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 $\overline{RESET}$ 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 $\overline{RESET}$ pin input goes high, the reset is cleared and program execution starts from address 0000_H.

The $\overline{RESET}$ 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.

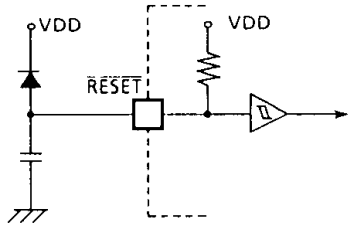


Figure 6-1. Simplified power-on-reset circuit

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)	Refe to "I/O circuitry"
Data memory bank selector	(DMB)	0		
Interrupt enable master flip-flop	(EIF)	0	Command register	Refer to the description of each relative command register.
Interrupt enable register	(EIR)	0_H		
Interrupt latch	(IL)	"0"		

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

6.1 Warm-Start

The warm-start capacibility to hold the data memory contents in the reset operation is not supported by hardware. However, it can be implemented by the following measures:

① Back up the voltage to be supplied to VDD pin.
② Apply to the $\overline{HOLD}$ pin the waveform synchronized with the power voltage variation.
③ Set the HOLD operating mode during the power is off.
④ Reset by program using the output port of sink open drain (initial "Hi-Z") after releasing HOLD operation.
⑤ Apply to an input port the power-on detect signal, and skip the initialize routines such as clearing RAM.

Figure 6-2 shows Warm-start Circuit Example

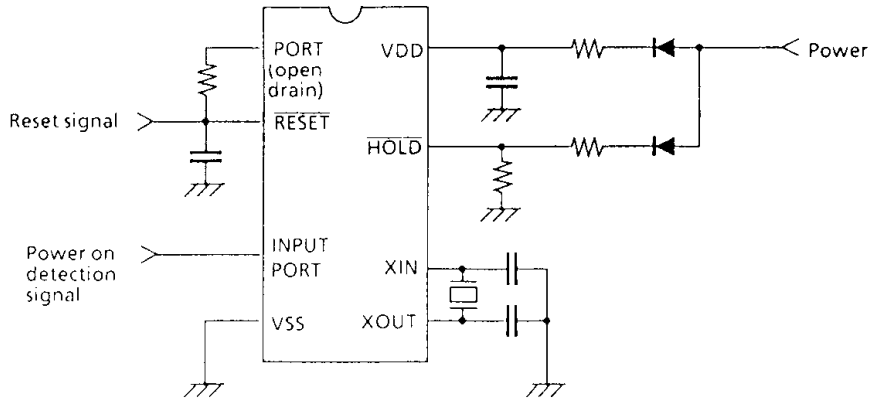


Figure 6-2. Warm-start Circuit Example

ELECTRICAL CHARACTERISTICS

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

PARAMETER	SYMBOL	PINS	RATINGS	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}	Except the sink open drain pin, but include R7	− 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	The sink open drain pin except R7	− 0.3 to 10	
Output Current (Per 1 pin)	I_{OUT2}	Ports R4-R9	3.2	mA
Output Current (Total)	ΣI_{OUT1}	Ports P1, P2	120	mA
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 operating mode	4.5	6.0	V
			in the SLOW operating mode	2.7		
			in the HOLD operating mode	2.0		
High Input 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}		$VDD < 4.5V$	$V_{DD} \times 0.9$		
Low Input 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		30.0	34.0	KHz

Note 1. Input voltage V_{IH3}, V_{IL3} : in the SLOW or HOLD operating 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)					
Input Low Current	I_{IL}	Ports R (push-pull)	V_{DD} = 5.5V, V_{IN} = 0.4V	–	–	–2	mA
Input Resistance	R_{IN1}	Part K0 with pull-up or pull-down		30	70	150	KΩ
	R_{IN2}	$\overline{RESET}$		100	220	450	
Output Leakage Current	I_{LO}	Ports P1, P2, R (open drain)	V_{DD} = 5.5V, V_{OUT} = 5.5V	–	–	2	μA
Output High Voltage	V_{OH}	Ports R (push-pull)	V_{DD} = 4.5V, I_{OH} = – 200μA	2.4	–	–	V
Output Low Voltage	V_{OL2}	Except XOUT, XTOUT, and Ports P1, P2	V_{DD} = 4.5V, I_{OL} = 1.6mA	–	–	0.4	V
Output Low Current	I_{OL1}	Ports P1, P2	V_{DD} = 4.5V, V_{OL} = 1.0V	–	20	–	mA
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	–	μA
Supply Current (in the HOLD mode)	I_{DDS}		V_{DD} = 5.5V	–	0.5	10	μA

Note 1. Typ. value 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 port K0 is opened when the pull-up / pull-down resistor is contained.
The voltage applied to the port R is within the valid range.

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

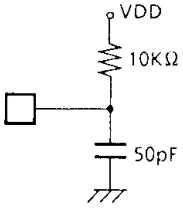
A.C. CHARACTERISTICS

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

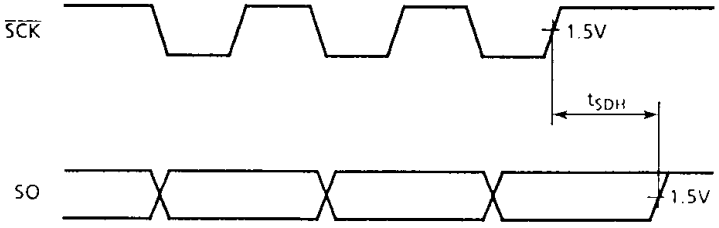
PARAMETER	SYMBOL	CONDITION	Min.	Typ.	Max.	UNIT
Instruction Cycle Time	t_{cy}	In the Normal operating mode	1.33	–	20	μs
		In the SLOW operating mode	235	–	267	
High level Clock pulse Width	t_{WCH}	For external clock operation	80	–	–	ns
Low level Clock pulse Width	t_{WCL}					
Shift Data Hold Time	t_{SDH}		$0.5t_{cy}$ – 300	–	–	ns

Note. Shift Data Hold Time;

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



Serial port (Completion of transmission)



RECOMMENDED OSCILLATING CONDITION

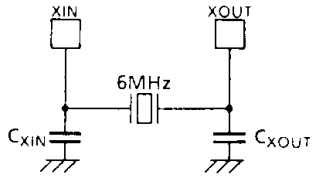
(V_{SS} = 0V, V_{DD} = 4.5 to 6.0V, T_{opr} = – 40 to 70°C)

(1) 6MHz

Ceramic Resonator

CSA6.00MGU (MURATA) $C_{XIN} = C_{XOUT}$ = 30pF

KBR-6.00MS (KYOCERA) $C_{XIN} = C_{XOUT}$ = 30pF



(2) 4MHz

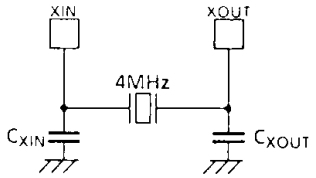
Ceramic Resonator

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

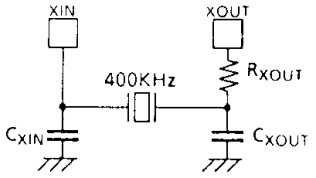


(3) 400KHz

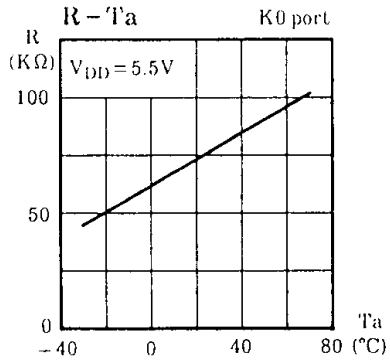
Ceramic Resonator

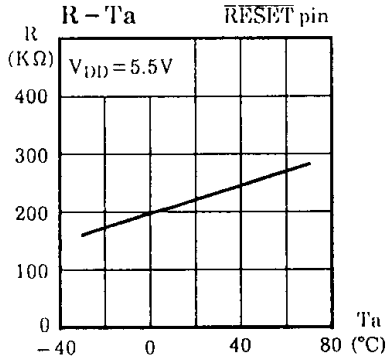
CSB400B (MURATA) $C_{XIN} = C_{XOUT}$ = 220pF, R_{XOUT} = 6.8KΩ

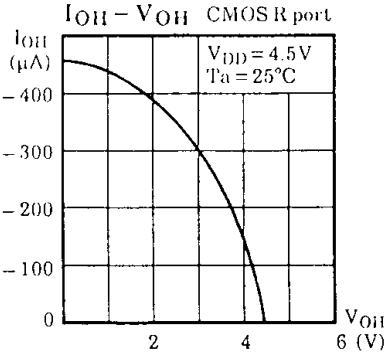
KBR-400B (KYOCERA) $C_{XIN} = C_{XOUT}$ = 100pF, R_{XOUT} = 10KΩ

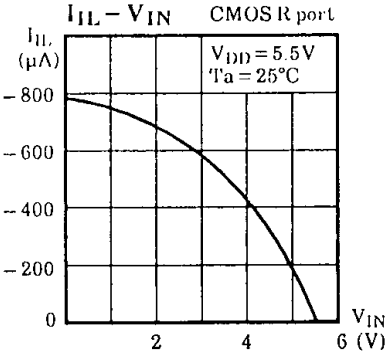


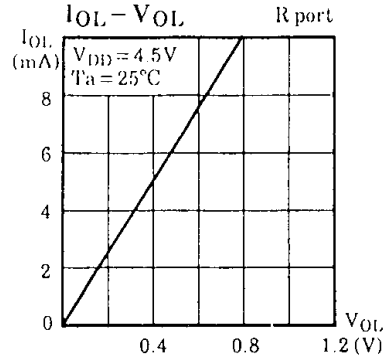
TYPICAL CHARACTERISTICS

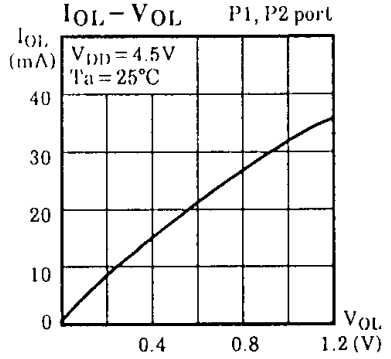


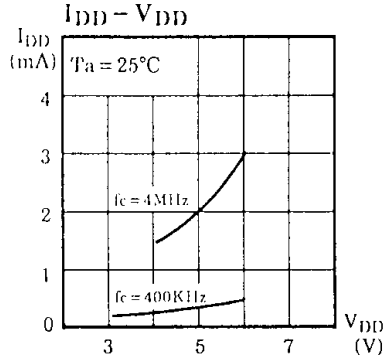


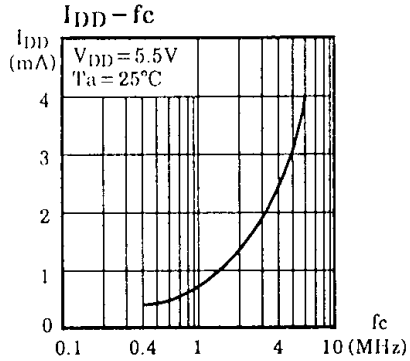








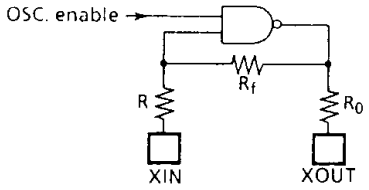
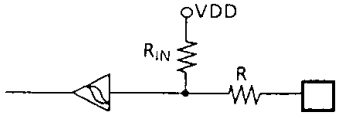
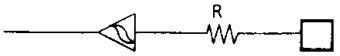
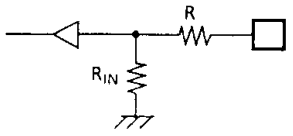




INPUT/OUTPUT CIRCUITRY

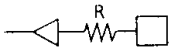
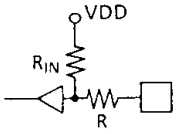
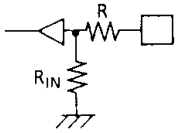
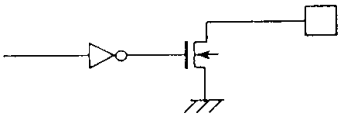
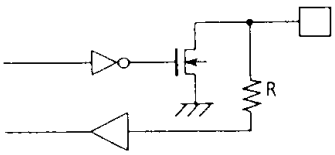
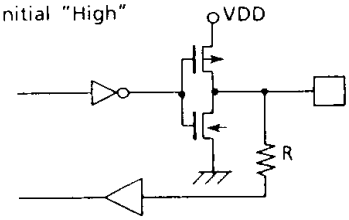
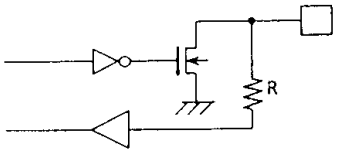
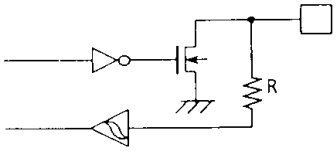
(1) Control pins

The input/output circuitries of the 47C800 control pins are shown below.

CONTROL PIN	I/O	CIRCUITRY	REMARKS
XIN XOUT	Input Output		Resonator connecting pins R = 1KΩ (typ.) R_f = 1.5MΩ (typ.) R_0 = 2KΩ (typ.)
RESET	Input		Hysteresis input Pull-up resistor R_{IN} = 220KΩ (typ.) R = 1KΩ (typ.)
$\overline{HOLD}$ ($\overline{KE0}$)	Input (Input)		Hysterisis input (Sense input) R = 1KΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70KΩ (typ.) R = 1KΩ (typ.)

(2) I/O ports

The input/output circuitries of the 47C800 I/O ports are shown below, any one of the circuitries can be chosen by a code (RA-RF) as a mask option.

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE			REMARKS
		RA, RD	RB, RE	RC, RF	
K0	Input				Pull-up/ Pull-down resistor R_{IN} = 70KΩ (typ.) R = 1KΩ (typ.)
P1 P2	Output				Sink opren dain output Initial "Hi-Z" High current I_{OL} = 20mA (typ.)
R4 R5 R6	I/O	RA, RB, RC Initial "Hi-Z" 	RD, RE, RF Initial "High" 		Sink open dain or Push-pull output R = 1KΩ (typ.)
R7	I/O				Sink open dain output Initial "Hi-Z" R = 1KΩ (typ.)
R8 R9	I/O				Sink open dain output Initial "Hi-Z" Hysteresis input R = 1KΩ (typ.)

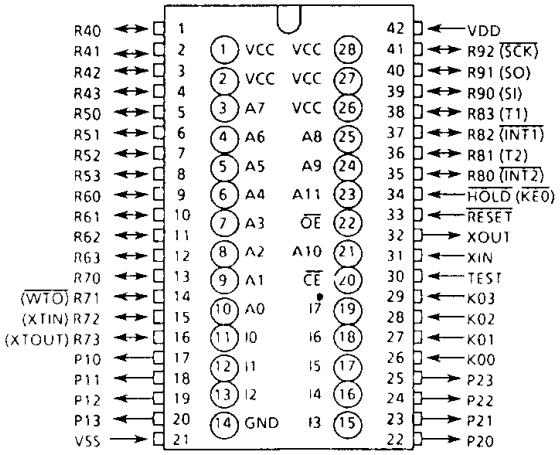
CMOS 4-BIT MICROCONTROLLER

TMP47C980E

The 47C980, which is equipped with an EPROM as program memory, is a piggyback type evaluator chip used for development and operational confirmation of the 47C800 application systems (programs).
The 47C980 is pin compatible with the 47C800 which is mask-programmed ROM device.

PIN ASSIGNMENT (TOP VIEW)

SDIC42



PIN FUNCTION (Top of the package)

PIN NAME	Input / Output	FUNCTIONS
A12 - A0	Output	Program memory address output
I7 - I0	Input	Program memory data input
$\overline{CE}$	Output	Chip enable signal output
$\overline{OE}$		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 to 6.0V	–	–	150	ns
Data Setup Time	t_{IS}	C_L = 100pf	150	–	–	ns
Data Hold Time	t_{IH}	T_{opr} = – 40 to 70°C	50	–	–	ns

NOTES FOR USE

(1) Program memory

The program area depends on the capacity of EPROM. See Figure 1.

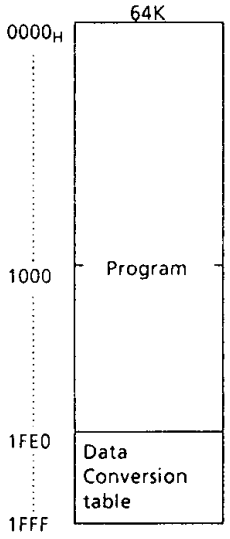


Figure 1. Program area

(2) Data memory

The 47C980 contains two 256 x 4-bit data memory banks (bank0, bank1).

(3) I/O ports

Input/Output circuitries of the 47C980 I/O ports are similar to the code RA of the 47C800. When this chip is used as evaluator with other I/O code (RB-RF) . It is necessary to provide the external resistors (See Figure 2).

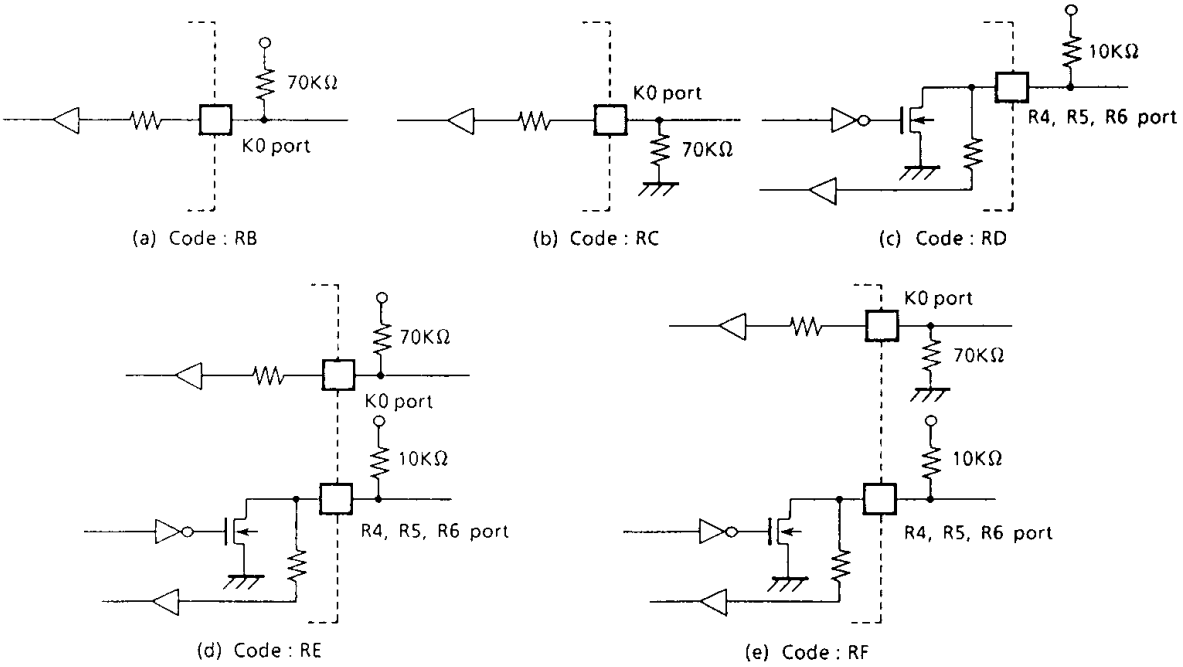


Figure 2. I/O code and External circuitry