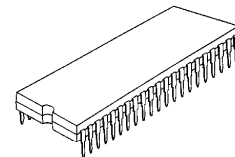## CMOS 4-BIT MICROCONTROLLER

TMP47C853N
TMP47C853F

The 47C853 is a high performance, 4-bit single-chip microcomputer based on the TLCS-470A CMOS series. The 47C853 has a build-in large-capacity RAM for repertory dial and a DTMF generator, making it ideal for application in telephones.

PART No.	ROM	RAM	PACKAGE	OTP
TMP47C853N	8192 × 8-bit	1024 × 4-bit	SDIP42-P-600-1.78	TMP47P853VN
TMP47C853F			QFP44-P-1414-0.80D	TMP47P853VF

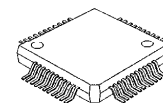## FEATURES

◆4-bit single-chip microcomputer
◆Instruction execution time : 16.7 / 8.3μs (at 480 / 960kHz)
    244μs (at 32.8kHz)
◆Low voltage operation :   2.2V min.
◆101 basic instructions
◆Table look-up instructions
◆Sub-routine nesting   :   15 levels max.
◆6 interruption sources (External : 2, Internal : 4)
    All sources have independent latches, and multiple
    interruption control is available.
◆I/O port (35 pins)
   ● Input   2ports   5pins
   ● I/O   7ports   27pins
   ● Output   1port   3pins
◆Two 12-bit Timer/Counters
    Timer, event counter, and pulse-width measurement mode
◆Interval Timer
◆Watchdog Timer
◆Serial Interface with 8-bit buffer
   ● Simultaneous transmission and reception
   ● External/internal clock, leading/trailing edge shift mode,
     4-bit / 8-bit
◆DTMF (Dual-Tone Multifrequency) output
   ● DTMF output with one instruction
   ● Single-tone output function
◆BEEP output function
◆Warmstart function

SDIP42-P-600-1.78

TMP47C853N
TMP47P853VN

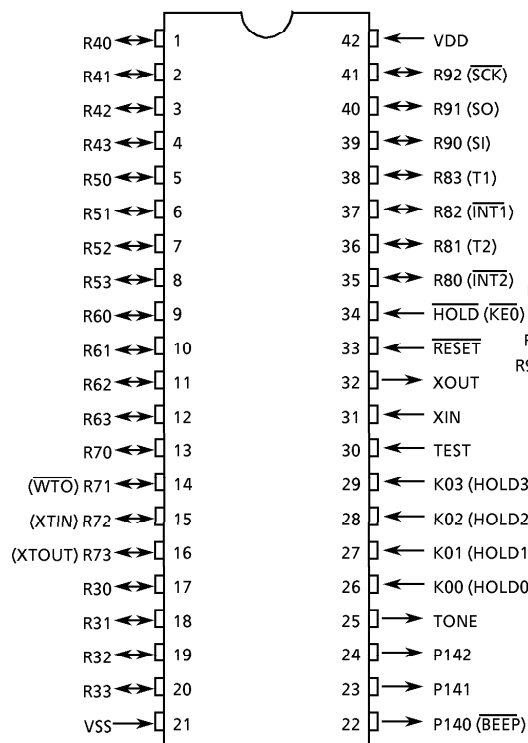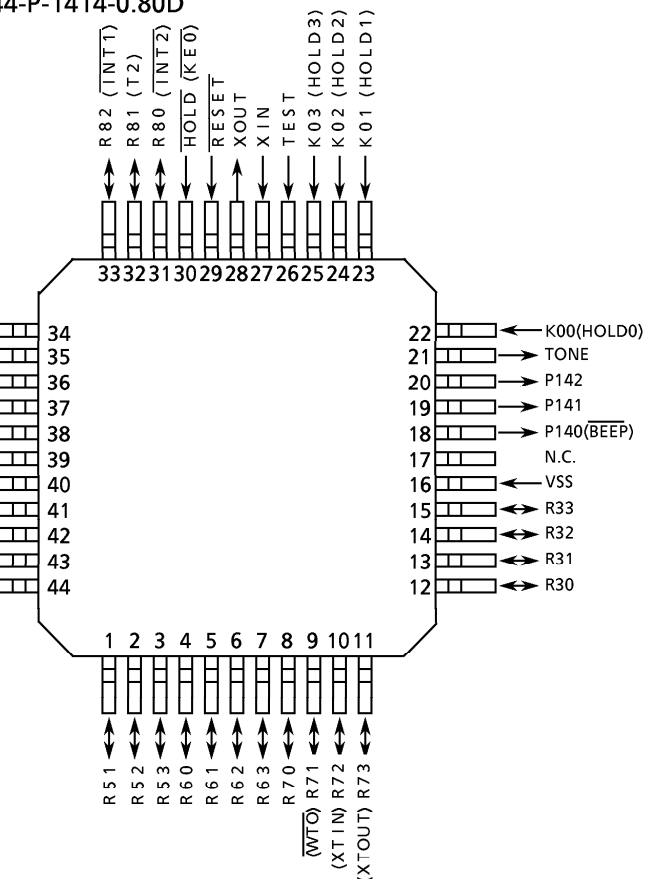QFP44-P-1414-0.80D

TMP47C853F
TMP47P853VF

980901EBP1

◆Dual clock operations
   High-speed operation/low-electricity operation
◆Hold function
   ● Battery/Capacitor backup
   ● Hold function controlled by port K0
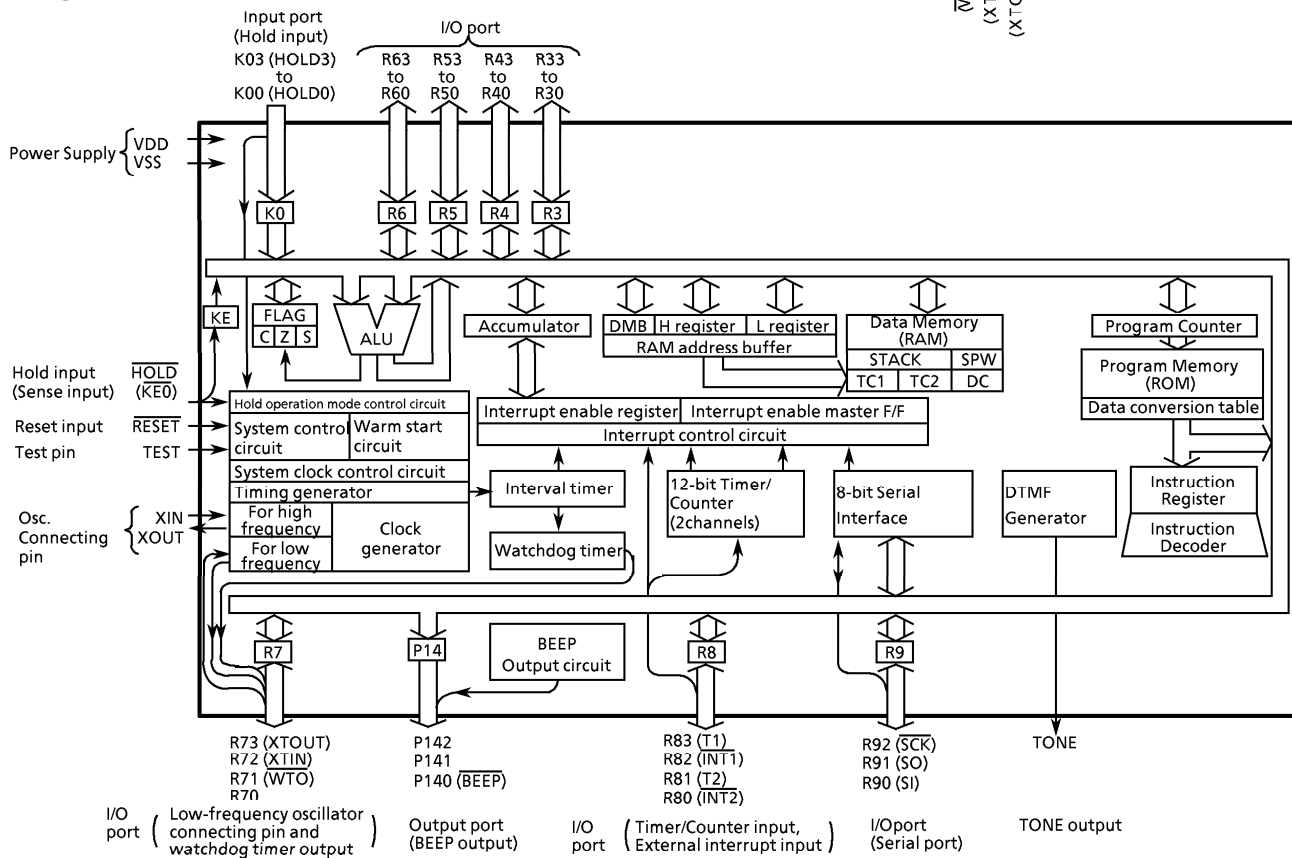◆Real-time Emulator   : BM47C853

PIN ASSIGNMENTS (TOP VIEW)

(1) SDIP42-P-600-1.78

R40	1	42	VDD
R41	2	41	R92 ($\overline{SCK}$)
R42	3	40	R91 (SO)
R43	4	39	R90 (SI)
R50	5	38	R83 (T1)
R51	6	37	R82 ($\overline{INT1}$)
R52	7	36	R81 (T2)
R53	8	35	R80 ($\overline{INT2}$)
R60	9	34	$\overline{HOLD}$ ($\overline{KE0}$)
R61	10	33	$\overline{RESET}$
R62	11	32	XOUT
R63	12	31	XIN
R70	13	30	TEST
($\overline{WTO}$) R71	14	29	K03 (HOLD3)
(XTIN) R72	15	28	K02 (HOLD2)
(XTOUT) R73	16	27	K01 (HOLD1)
R30	17	26	K00 (HOLD0)
R31	18	25	TONE
R32	19	24	P142
R33	20	23	P141
VSS	21	22	P140 ($\overline{BEEP}$)

(2) QFP44-P-1414-0.80D

Top pins (33 32 31 30 29 28 27 26 25 24 23):
R82 ($\overline{INT1}$), R81 (T2), R80 ($\overline{INT2}$), $\overline{HOLD}$ ($\overline{KE0}$), $\overline{RESET}$, XOUT, XIN, TEST, K03 (HOLD3), K02 (HOLD2), K01 (HOLD1)

Left pins:
R83 (T1)	34	
R90 (SI)	35	
R91 (SO)	36	
R92 ($\overline{SCK}$)	37	
N.C.	38	
VDD	39	
R40	40	
R41	41	
R42	42	
R43	43	
R50	44	

Right pins:
22		K00 (HOLD0)
21		TONE
20		P142
19		P141
18		P140 ($\overline{BEEP}$)
17		N.C.
16		VSS
15		R33
14		R32
13		R31
12		R30

Bottom pins (1 2 3 4 5 6 7 8 9 10 11):
R51, R52, R53, R60, R61, R62, R63, R70, ($\overline{WTO}$) R71, (XTIN) R72, (XTOUT) R73

BLOCK DIAGRAM

Input port (Hold input) K03 (HOLD3) to K00 (HOLD0)

I/O port: R63 to R60, R53 to R50, R43 to R40, R33 to R30

Power Supply: VDD, VSS

K0, R6, R5, R4, R3

KE, FLAG (C Z S), ALU, Accumulator, DMB, H register, L register, RAM address buffer, Data Memory (RAM), STACK, SPW, TC1, TC2, DC, Program Counter, Program Memory (ROM), Data conversion table

Hold input (Sense input): $\overline{HOLD}$ ($\overline{KE0}$)
Reset input: $\overline{RESET}$
Test pin: TEST

Hold operation mode control circuit
System control circuit, Warm start circuit
System clock control circuit
Timing generator

Interrupt enable register, Interrupt enable master F/F
Interrupt control circuit

Osc. Connecting pin: XIN, XOUT
For high frequency, For low frequency, Clock generator

Interval timer, Watchdog timer

12-bit Timer/Counter (2 channels)
8-bit Serial Interface
DTMF Generator
Instruction Register
Instruction Decoder

R7, P14, BEEP Output circuit, R8, R9

R73 (XTOUT), R72 (XTIN), R71 ($\overline{WTO}$), R70
P142, P141, P140 ($\overline{BEEP}$)
R83 (T1), R82 ($\overline{INT1}$), R81 (T2), R80 ($\overline{INT2}$)
R92 ($\overline{SCK}$), R91 (SO), R90 (SI)
TONE

I/O port — Low-frequency oscillator connecting pin and watchdog timer output
Output port (BEEP output)
I/O port — Timer/Counter input, External interrupt input
I/O port (Serial port)
TONE output

PIN FUNCTION

PIN NAME	Input/Output	FUNCTIONS	
K03 (HOLD3) to K00 (HOLD0)	Input (Input)	4-bit input port	Hold request/release signal input (Active "H")
R33 to R30	I/O	4-bit I/O port with latch. When used as the input port, the latch must be set to "1".	
R43 to R40			
R53 to R50			
R63 to R60			
R73 (XTOUT)	I/O (Output)	When R73 or R72 is used as the oscillator, it must be set to the dual colck mode. Setting, clearance and testing of 1-bit unit is possible by the bit processing instruction of the L-register indirect addressing.	Osc. connecting pin (Low-frequency). When of external clock input, input is made to XTIN and releases XTOUT.
R72 (XTIN)	I/O (Input)		
R71 ($\overline{\text{WTO}}$)	I/O (Output)		Watchdog timer output
R70	I/O		
R83 (T1)	I/O (Input)	4-bit I/O port with latch. When 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 ($\overline{\text{INT1}}$)			External interrupt 1 input
R81 (T2)			Timer/Counter 2 external input
R80 ($\overline{\text{INT2}}$)			External interrupt 2 input
R92 ($\overline{\text{SCK}}$)	I/O (I/O)	3-bit I/O port with latch. When used as an 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
P142, P141	Output	3-bit port with latch	
P140 ($\overline{\text{BEEP}}$)	Output (Output)		BEEP Output
TONE	Output	Tone output	
XIN	Input	Osc. connecting pin	
XOUT	Output		
$\overline{\text{RESET}}$	Input	Reset signal input	
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input	Hold request/release signal input	Sense input
TEST	Input	Test pin for outgoing test. Either opened, or fixed to low level.	
VDD	Power supply	+ 2.2V to 6.0V	
VSS		0 V (GND)	

OPERATIONAL DESCRIPTION

In addition, an operation of the 47C853 is possible at 960kHz oscillation when applications necessitate high-speed processing, and at 480kHz when applications necessitate an operation with low-electricity consumption. When operating the 47C853 at 480kHz oscillation, it is necessary to set the DTMF input clock control register at the beginning of the program. For details, see "3.5.1 Selection of Input Clock to DTMF Generator".

1.   SYSTEM CONFIGURATION

◆INTERNAL CPU FUNCTION
  2.1   Program Counter (PC)
  2.2   Program Memory (ROM)
  2.3   H Register, L Register, and Data Memory Bank Selector (DMB)
  2.4   Data Memory (RAM)
      ● Stack
      ● Stack Pointer Word (SPW)
      ● Data Counter (DC)
  2.5   ALU, Accumulator
  2.6   Flags
  2.7   Clock Generator, Timing Generator and System Clock Controller
  2.8   Interrupt Controller
  2.9   Reset Circuit

◆PERIPHERAL HARDWARE FUNCTION
  3.1   Input/Output Ports
  3.2   Interval Timer
  3.3   Timer/Counters (TC1, TC2)
  3.4   Watchdog Timer
  3.5   DTMF Genarator
  3.6   BEEP Output Circuit
  3.7   Warm Start Circuit
  3.8   Serial Interface

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

2.   INTERNAL CPU FUNCTION

2.1   Program Counter (PC)

The program counter is a 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 acknowledged, the specified values listed in Table 2-1 are set to the PC. The PC is initialized to "0" during reset.
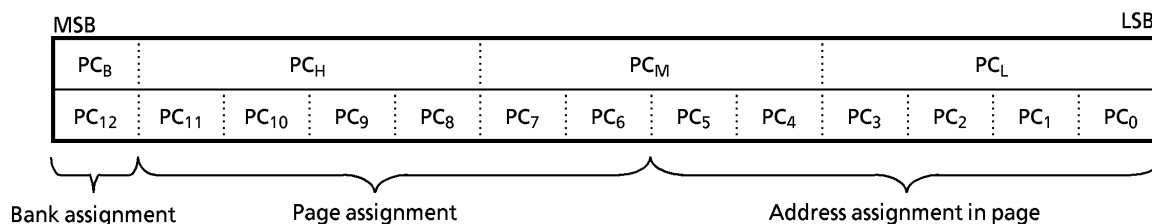
```
MSB                                                                                        LSB
┌──────┬──────────────────────────┬──────────────────────────┬──────────────────────────┐
│ PCB  │          PCH             │          PCM             │          PCL             │
├──────┼──────┬──────┬──────┬─────┼──────┬──────┬──────┬─────┼──────┬──────┬──────┬─────┤
│ PC12 │ PC11 │ PC10 │ PC9  │ PC8 │ PC7  │ PC6  │ PC5  │ PC4 │ PC3  │ PC2  │ PC1  │ PC0 │
└──────┴──────┴──────┴──────┴─────┴──────┴──────┴──────┴─────┴──────┴──────┴──────┴─────┘
   Bank assignment      Page assignment              Address assignment in page
```

Figure 2-1.   Configuration of Program Counter

The program counter can make a direct address assignment to the 8192-byte address space.

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 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.When first byte or second byte of this instruction is stored at the last address of the bank, the upper 1 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 in 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 $07FF_H$.

Instruction /Operation	Condition		PC12	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
BSL a	SF = 1 (Branch condition is satisfied)		Immediate data specified by the instruction												
	SF = 0 (Branch condition is not satisfied)		+ 3												
BS a	SF = 1	Lower 12-bit address ≠ FFE, FFF$_H$	Hold	Immediate data specified by the instruction											
		Lower 12-bit address = FFE, FFF$_H$ (The bank end address)	+ 1	Immediate data specified by the instruction											
	SF = 0		+ 2												
BSS a	SF = 1	Lower 6-bit address ≠ 3F$_H$	Hold								Immediate data specified by the instruction				
		Lower 6-bit address = 3F$_H$ (The page end address)	+ 1								Immediate data specified by the instruction				
	SF = 0		+ 1												
CALL a			0	0	Immediate data specified by the instruction										
CALLS a			0	0	0	0	0	The data generated by the immediate data specified by the instruction					1	1	0
RET			The return address restored from stack												
RETI			The return address restored from stack												
Others			Incremented by the number of bytes of the instruction												
Interrupt acceptance			0	0	0	0	0	0	0	0	0	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

2.2　Program Memory (ROM)

The 47C853 has a built-in program memory (mask ROM) of 8192 × 8-bit (0000-1FFF$_H$ address). The 5-8 bit data conversion table is placed in the last 32-byte space (1FE0-1FFF$_H$) of the program memory. The fixed data which can be read out by the table look-up instructions are placed at the 1000-1FFF$_H$ address of the program memory, and address assignment is made possible by the 12-bit length data counter.

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

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

Instruction /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$_0$
BSL a	SF = 1 (Branch condition is satisfied)		Immediate data specified by the instruction
	SF = 0 (Branch condition is not satisfied)		+ 3
BS a	SF = 1	Lower 12-bit address ≠ FFE, FFF$_H$	PC$_{12}$: Hold; remaining: Immediate data specified by the instruction
		Lower 12-bit address = FFE, FFF$_H$ (The bank end address)	PC$_{12}$: + 1; remaining: Immediate data specified by the instruction
	SF = 0		+ 2
BSS a	SF = 1	Lower 6-bit address ≠ 3F$_H$	PC$_{12}$–PC$_6$: Hold; PC$_5$–PC$_0$: Immediate data specified by the instruction
		Lower 6-bit address = 3F$_H$ (The page end address)	PC$_{12}$–PC$_6$: + 1; PC$_5$–PC$_0$: Immediate data specified by the instruction
	SF = 0		+ 1
CALL a			PC$_{12}$ = 0, PC$_{11}$ = 0; remaining: Immediate data specified by the instruction
CALLS a			PC$_{12}$ = 0, PC$_{11}$ = 0, PC$_{10}$ = 0, PC$_9$ = 0, PC$_8$ = 0; The data generated by the immediate data specified by the instruction; PC$_2$ = 1, PC$_1$ = 1, PC$_0$ = 0
RET			The return address restored from stack
RETI			The return address restored from stack
Others			Incremented by the number of bytes of the instruction
Interrupt acceptance			PC$_{12}$ = 0, PC$_{11}$ = 0, PC$_{10}$ = 0, PC$_9$ = 0, PC$_8$ = 0, PC$_7$ = 0, PC$_6$ = 0, PC$_5$ = 0, PC$_4$ = 0; Interrupt vector; PC$_0$ = 0
RESET			PC$_{12}$ = 0, PC$_{11}$ = 0, PC$_{10}$ = 0, PC$_9$ = 0, PC$_8$ = 0, PC$_7$ = 0, PC$_6$ = 0, PC$_5$ = 0, PC$_4$ = 0, PC$_3$ = 0, PC$_2$ = 0, PC$_1$ = 0, PC$_0$ = 0

Table 2-1. Status Change of Program Counter

2.2　Program Memory (ROM)

The 47C853 has a built-in program memory (mask ROM) of 8192 × 8-bit (0000-1FFF$_H$ address). The 5-8 bit data conversion table is placed in the last 32-byte space (1FE0-1FFF$_H$) of the program memory. The fixed data which can be read out by the table look-up instructions are placed at the 1000-1FFF$_H$ address of the program memory, and address assignment is made possible by the 12-bit length data counter.

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

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

(2) 5-bit to 8-bit data conversion instruction

[OUTB @HL]

The 5-bit to 8-bit data conversion instruction reads the fixed data (8 bits) from the data conversion table in the program memory to send the upper 4 bits to COLUMN register and the lower 4 bits to ROW resister. The data conversion table is located in the last 32-byte space (addresses 1FE0 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 setting data of DTMF generator.
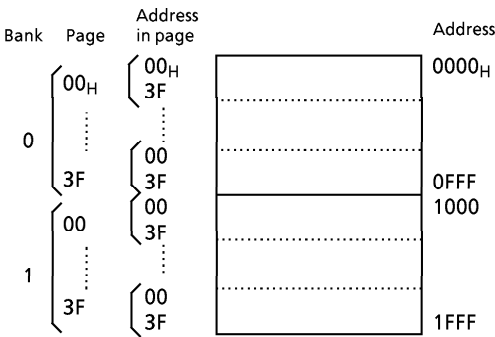
Figure 2-2. Configulation of Program Memory

2.2.1 Program Memory Map

Figure 2-3 shows the program memory map. Address 0000 through 0086$_H$ and 1FE0 through 1FFF$_H$ for the 47C853, of the program memory are also used for special purposes.

2.2.2 Program Memory Capacity

The 47C853 contains a program memory (masked ROM) of 8192 × 8 bit (addresses 0000 through 1FFF$_H$).
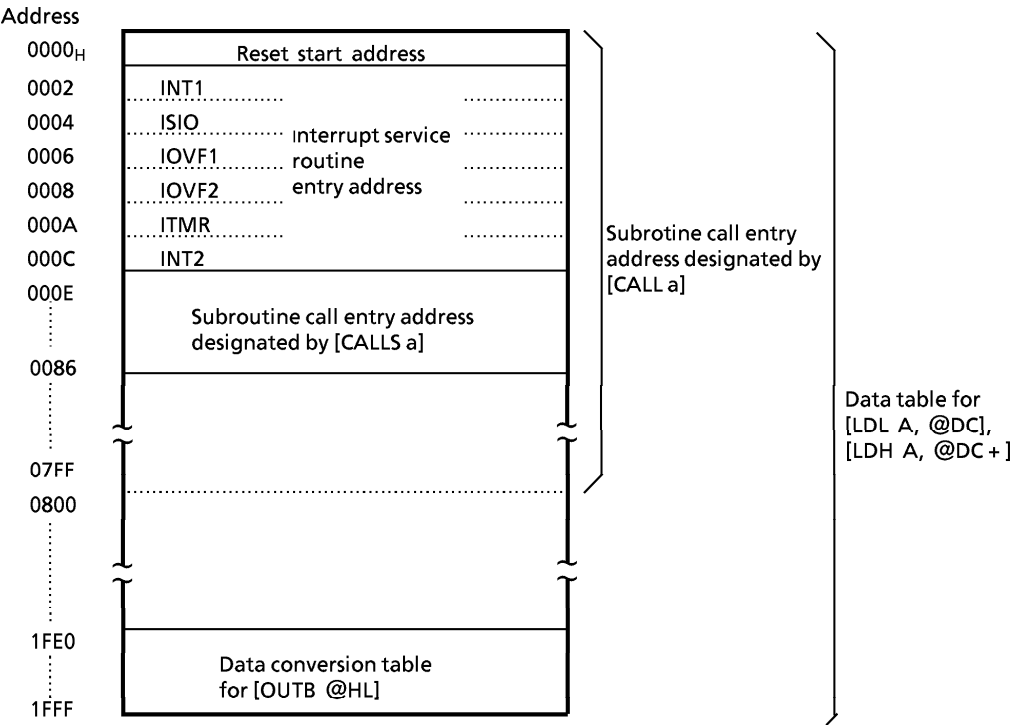
Figure 2-3. Program Memory Map

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 four banks. Bank 0, 1, 2 and 3. The data memory bank selector (DMB) is a 2-bit register to specify a data memory bank. During reset, the DMB is initialized to "0". The DMB is set by the [CLR DMB0] , [SET DMB0] [CLR DMB1] , [SET DMB1] , [LD DMB, #K] , [MOV A, DMB] or [LD DMB, @ HL] instruction. The currently selected data memory bank can be known by executing the [TEST DMB0] or, [TESTP DMB0], [TEST DMB1] or [TESTP DMB1] instruction. The L register has the automatic post-increment/decrement capability, implementing the execution of composite instructions. For example, [ST A,@HL +] instruction automatically increments the contents of the L register after data transfer. During the execution [SET @L], [CLR @L], or [TEST @L] instruction, the L register is also used to specify the bits corresponding to I/O port pins R73 through R40 (the indirect addressing of port bits by the L register).

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

```
LD      DMB,  #0        ; DMB ← 0
LD      HL,   #10H      ; HL ← 10 H
ST      #5,   @HL+      ; RAM [10 H] ← 5,   LR ← LR+1
ST      #0FH,  @HL+     ; RAM [11 H] ← F H, 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
```
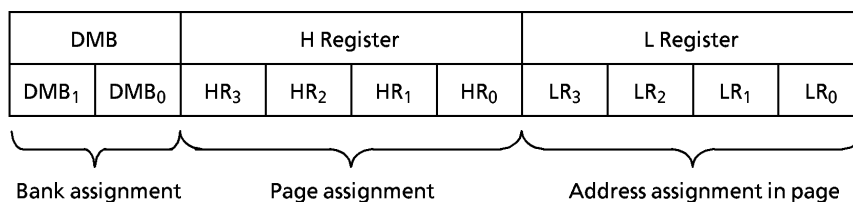
DMB		H Register				L Register			
DMB_1	DMB_0	HR_3	HR_2	HR_1	HR_0	LR_3	LR_2	LR_1	LR_0

Bank assignment / Page assignment / Address assignment in page

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

2.4 Data Memory (RAM)

The 47C853 data memory (RAM) has a total of 1024 × 4-bit, which consists of four data memory banks (banks 0, 1, 2, and 3) of 256 × 4-bit (00-FF_H address). A private stack (STK13) and stack pointer word (SPW13) for PC_{13} are not components of the 47C853.

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 four banks. Bank 0, 1, 2 and 3. The data memory bank selector (DMB) is a 2-bit register to specify a data memory bank. During reset, the DMB is initialized to "0". The DMB is set by the [CLR DMB0] , [SET DMB0] [CLR DMB1] , [SET DMB1] , [LD DMB, #K] , [MOV A, DMB] or [LD DMB, @ HL] instruction. The currently selected data memory bank can be known by executing the [TEST DMB0] or, [TESTP DMB0], [TEST DMB1] or [TESTP DMB1] instruction. The L register has the automatic post-increment/decrement capability, implementing the execution of composite instructions. For example, [ST A,@HL +] instruction automatically increments the contents of the L register after data transfer. During the execution [SET @L], [CLR @L], or [TEST @L] instruction, the L register is also used to specify the bits corresponding to I/O port pins R73 through R40 (the indirect addressing of port bits by the L register).

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

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

DMB		H Register				L Register			
DMB_1	DMB_0	HR_3	HR_2	HR_1	HR_0	LR_3	LR_2	LR_1	LR_0

Bank assignment   Page assignment   Address assignment in page

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

2.4 Data Memory (RAM)

The 47C853 data memory (RAM) has a total of 1024 × 4-bit, which consists of four data memory banks (banks 0, 1, 2, and 3) of 256 × 4-bit (00-FF_H address). A private stack (STK13) and stack pointer word (SPW13) for PC_{13} are not components of the 47C853.

(1) Register-indirect addressing mode
In this mode, a bank is specified by the DMB, a page by the H register and an address in the page by the L register.

Example: LD A, @HL   ;   Acc ← RAM [HL]

(2) Direct addressing mode
An address in the bank is directly specified by the 8 bits of the second byte (operand) in the instruction field. The bank is specified by the DMB.

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

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

Example: ST #3, 05H   ;   RAM [05$_H$] ← 3

Figure 2-5. Addressing mode

2.4.1 Data Memory Map

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

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

Figure 2-6. Data Memory Map

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

(1) Register-indirect addressing mode

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

Example: LD A, @HL   ; Acc ← RAM [HL]

(2) Direct addressing mode

An address in the bank is directly specified by the 8 bits of the second byte (operand) in the instruction field. The bank is specified by the DMB.

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

(3) Zero-page addressing mode

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

Example: ST #3, 05H   ; RAM [05$_H$] ← 3

Figure 2-5. Addressing mode

2.4.1   Data Memory Map

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

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

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

Figure 2-6. Data Memory Map

(1) Stack

The stack provides the area in which the return address is saved before a jump is performed to the processing routine at the execution of a subroutine call instruction or the acceptance of an interrupt. When a subroutine call instruction is executed, the contents (the return address) of the program counter are saved; when an interrupt is accepted, the contents of the program counter and flags are saved. When returning from the processing routine, executing the subroutine return instruction [RET] restores the contents of the program counter from the stack; executing the interrupt return instruction [RETI] restores the contents of the program counter and flags. Stacks have a maximum of 15 levels (locations 0 -14). The lower 13 bits (PC_0-PC_{12}) of the program counter and the flag area are in bank 0 (C0-FB_H) of the data memory. The area for the highest order bit (PC_{13}) of the program counter is provided in a Stack 13 (STK13). One stack level consists of a 4-word data memory and 1 bit from STK13. Locations 13 and 14 in the data memory are also used as count registers (TC1, TC2) for the Timer/Counter described later.

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

(2) Stack Pointer Word (SPW)

Address FF_H in the data memory (bank 0) is called the stack pointer word, which identifies the location in the stack to be accessed (save or restore).Generally, location number 0 to 12 can be set to the SPW, providing up to 13 levels of stack nesting. Locations 13 and 14 are shared by the count registers of the Timer/Counters to be described later; therefore, when the Timer/Counters are not used, the stack area of up to 15 levels is available. Address FF_H is assigned with the SPW, so that the contents of the SPW cannot be set "15" in any case. The SPW is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost. (For example, when the user-processed data arera is in an address range 00_H through CF_H, up to location 4 of the stack are usable. If an interrupt is accepted with location 4 already used, the user-processed data stored in addresses CC_H through CFH corresponding to the location 3 area is lost.)The SPW is not initialized in terms of hardware, requiring to write the initial value (the location with which the use of the stack starts) by using the initialization routine. Normally, the initial value of "12" is used.

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

```
        LD       A,   #12
        LD       DMB,  #0H        ; DMB ← 0 H
        ST       A,   0FFH        ; RAM [FF] ← Acc
```

(a) At execution of a subroutine call instruction

(b) At acceptance of an interrupt

(c) At execution of a subroutine return
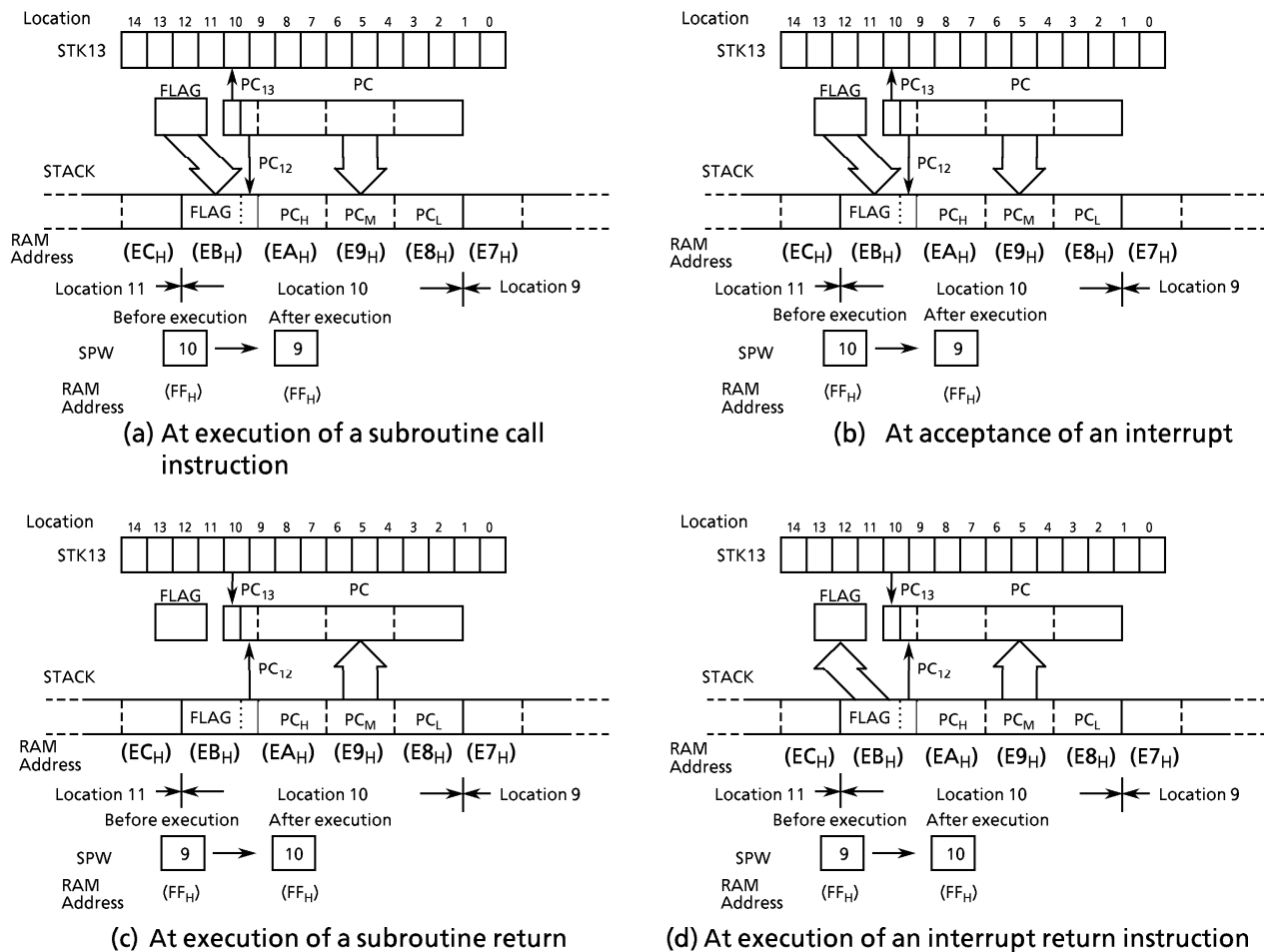
(d) At execution of an interrupt return instruction

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

(3) Data counter (DC)

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

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

Figure 2-8. Data Counter

Example: To set the DC to 780_H

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

(4) Count registers of the Timer/Counter (TC1, TC2)
The 47C853 have two 12-bit Timer/Counters. The count register of the Timer/Counter is assigned with RAM addresses in unit of 4 bits, so that the initial value setting or contents reading is performed by using RAM instruction.The count registers are shared by the stack area (locations 13 and 14) described earlier, so that the stack is usable from location 13 when the Timer/Counter 1 is not used. When none of Timer/Counter 1 and Timer/Counter 2 are used,the stack is usable from location 14. When both Timer/Counter 1 and Timer/Counter 2 are used, the data memory (bank 0) locations at addresses $F7_H$ and FB_H can be used to store the user-processed data.

MSB		LSB
Timer/Counter1 (TC1)		
$TC1_H$	$TC1_M$	$TC1_L$

RAM address $(F6_H)$ $(F5_H)$ $(F4_H)$

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

(FA_H) $(F9_H)$ $(F8_H)$

Figure 2-9. 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 data memory
                              is "0".

     B   SKIP
     ST  #8,09H             ; Writes "8" to address 09H in the data memory.
SKIP :
```

2.4.2 Data Memory Capacity

The 47C853 have 1024 × 4 bit of data memory (RAM) , 256 × 4 bits (addresses 00_H through FF_H) on each of banks (bank0, bank1, bank2 and bank3). When power-on is performed, the contents of the data memory become unpredictable, so that they must be initialized by the initialize routine.

```
Example:     To clear RAM
             LD     HL,    #00H     ; HL ← 00H
SCLR0 :      LD     DMB,   #0       ; DMB ← 0
SCLR1 :      ST     #0,    @HL+     ; RAM [HL] ← 0,   LR ← LR+1
             B      SCLR1
             LD     DMB,   #1       ; DMB ← 1
SCLR2 :      ST     #0,    @HL+     ; RAM [HL] ← 0,   LR ← LR+1
             B      SCLR2
             LD     DMB,   #2       ; DMB ← 2
SCLR3 :      ST     #0,    @HL+     ; RAM [HL] ← 0,   LR ← LR+1
             B      SCLR3
             ADD    H,     #1       ; HR ← HR+1
             B      SCLR0
```
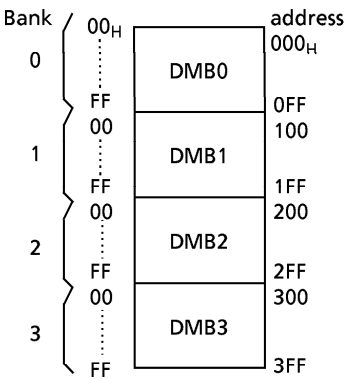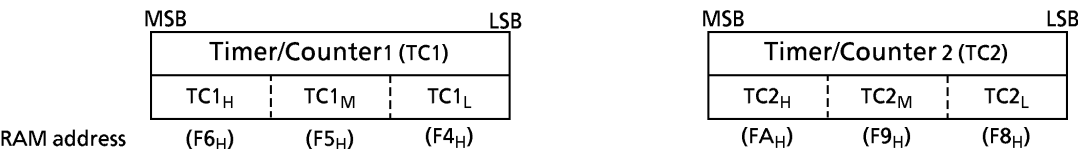
Bank			address
0	00_H ⋮ FF	DMB0	000_H 0FF
1	00 ⋮ FF	DMB1	100 1FF
2	00 ⋮ FF	DMB2	200 2FF
3	00 ⋮ FF	DMB3	300 3FF

Figure 2-10. Data Memory (RAM)

(4) Count registers of the Timer/Counter (TC1, TC2)
The 47C853 have two 12-bit Timer/Counters. The count register of the Timer/Counter is assigned with RAM addresses in unit of 4 bits, so that the initial value setting or contents reading is performed by using RAM instruction.The count registers are shared by the stack area (locations 13 and 14) described earlier, so that the stack is usable from location 13 when the Timer/Counter 1 is not used. When none of Timer/Counter 1 and Timer/Counter 2 are used,the stack is usable from location 14. When both Timer/Counter 1 and Timer/Counter 2 are used, the data memory (bank 0) locations at addresses $F7_H$ and FB_H can be used to store the user-processed data.

MSB		LSB
Timer/Counter1 (TC1)		
$TC1_H$	$TC1_M$	$TC1_L$

RAM address $(F6_H)$ $(F5_H)$ $(F4_H)$

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

(FA_H) $(F9_H)$ $(F8_H)$

Figure 2-9. 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 data memory
                                is "0".
       B   SKIP
       ST  #8,09H             ; Writes "8" to address 09H in the data memory.
SKIP :
```

2.4.2 Data Memory Capacity

The 47C853 have 1024 × 4 bit of data memory (RAM) , 256 × 4 bits (addresses 00_H through FF_H) on each of banks (bank0, bank1, bank2 and bank3). When power-on is performed, the contents of the data memory become unpredictable, so that they must be initialized by the initialize routine.

```
Example:       To clear RAM
               LD     HL,   #00H      ; HL ← 00H
SCLR0 :        LD     DMB,  #0        ; DMB ← 0
SCLR1 :        ST     #0,   @HL+      ; RAM [HL] ← 0,  LR ← LR+1
               B      SCLR1
               LD     DMB,  #1        ; DMB ← 1
SCLR2 :        ST     #0,   @HL+      ; RAM [HL] ← 0,  LR ← LR+1
               B      SCLR2
               LD     DMB,  #2        ; DMB ← 2
SCLR3 :        ST     #0,   @HL+      ; RAM [HL] ← 0,  LR ← LR+1
               B      SCLR3
               ADD    H,    #1        ; HR ← HR+1
               B      SCLR0
```
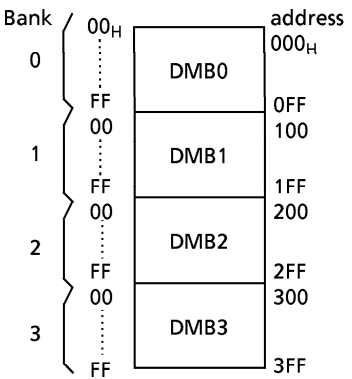
Bank		address	
0	00_H ⋮ FF	DMB0	000_H $0FF_H$
1	00 ⋮ FF	DMB1	100 1FF
2	00 ⋮ FF	DMB2	200 2FF
3	00 ⋮ FF	DMB3	300 3FF

Figure 2-10. Data Memory (RAM)

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).
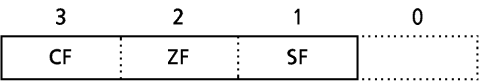

Figure 2-11. ALU

Note. Cin indicates the carry input specified by instruction

(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

2.5.2   Accumulator (Acc)

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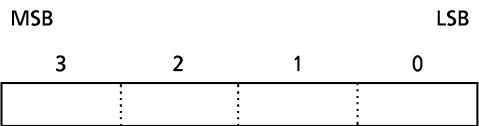

Figure 2-12. Accumulator

2.6   Flags

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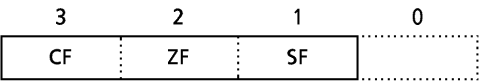

Figure 2-13. Flags

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
```


Figure 2-11. ALU

Note. Cin indicates the carry input specified by instruction

2.5.2 Accumulator (Acc)

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

Figure 2-12. Accumulator

2.6 Flags

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

Figure 2-13. Flags

(1) Carry flag (CF)

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

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

The CF becomes the input (Cin) to the ALU to hold the carry information.

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

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

③ Rotate instructions [ROLC A],[RORC A]

The CF is shifted into the accumulator to hold the carry information (the data shifted out from the accumulator).

④ Carry flag test instructions [TESTP CF], [TEST CF]

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

(2) Zero flag (ZF)

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

(3) Status flag (SF)

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

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

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

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

2.7   Clock Generator, Timing Generator and System Clock Controller
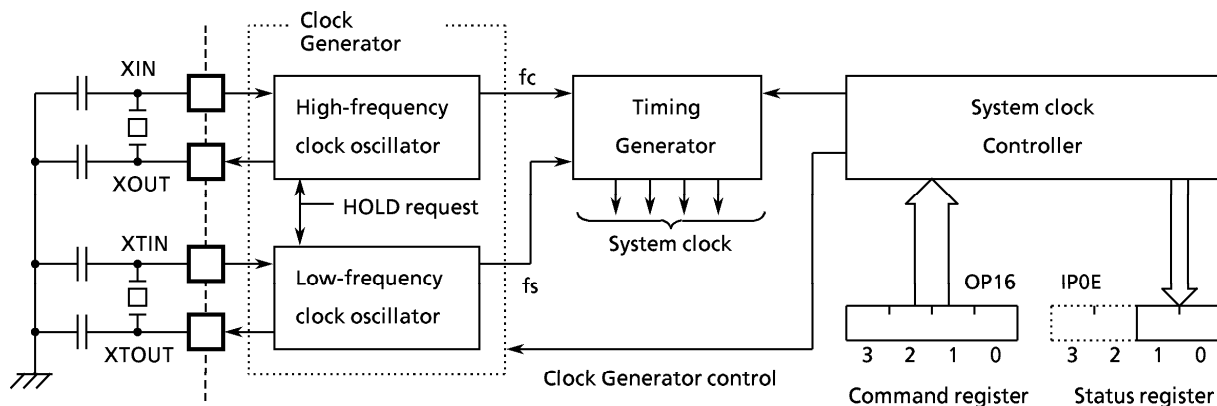

Figure 2-14. Clock Generator, Timing Generator and System Clock Controller

2.7.1   Clock Generator (CG)

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

(a) Crystal or ceramic resonator   (b) External oscillator   (c) Crystal   (d) External
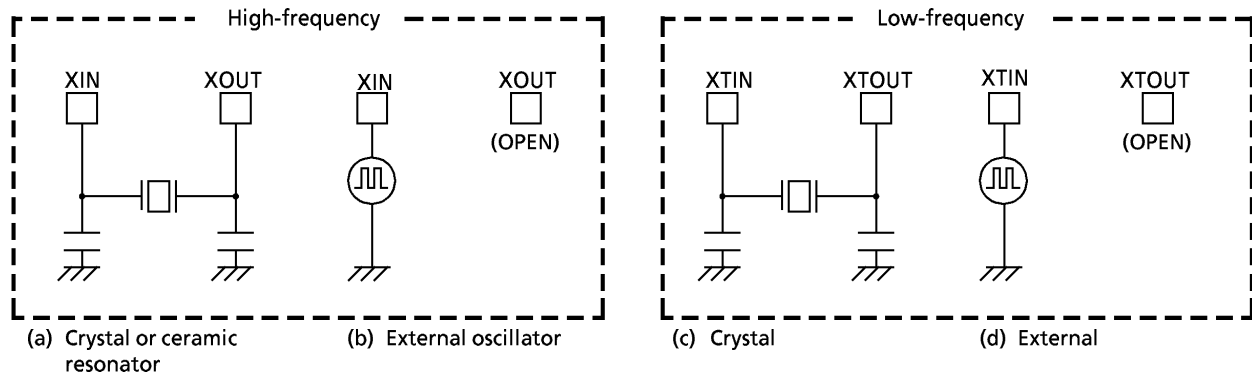
Figure 2-15. Examples of Resonator Connection

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

Example :   To adjust the high-frequency oscillation frequency by outputting monitor pulse to R70 pin.

(Before executing the program, make sure that MCU outputs the stable low-frequency clock oscillation in SLOW mode.)

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

Output waveform

$fc / 40$

2.7.2 Timing Generator (TG)

The timing generator produces the system clocks from clock pulse which are supplied to the CPU and peripheral hardweare.
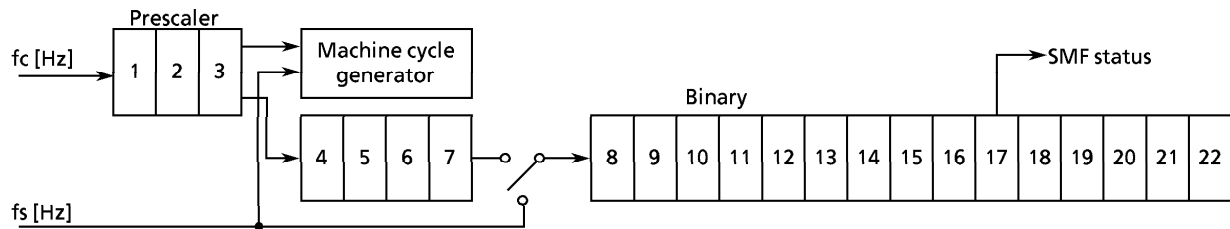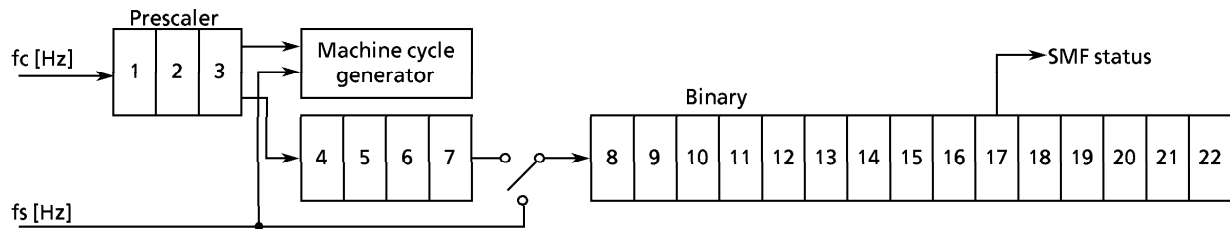

Figure 2-16. Configuration of Timing Generator

2.7.3 System Clock Controller

The system clock controller starts or stops the high-frequency and low-frequency clock oscillator and switches between the basic clocks. The operating mode is generally divided into the single-clock mode and the dual-clock mode, which are controlled by command. Figure 2-16 shows the operating mode transition diagram. Figure 2-17 shows the command register. Figure 2-18 shows the status register.

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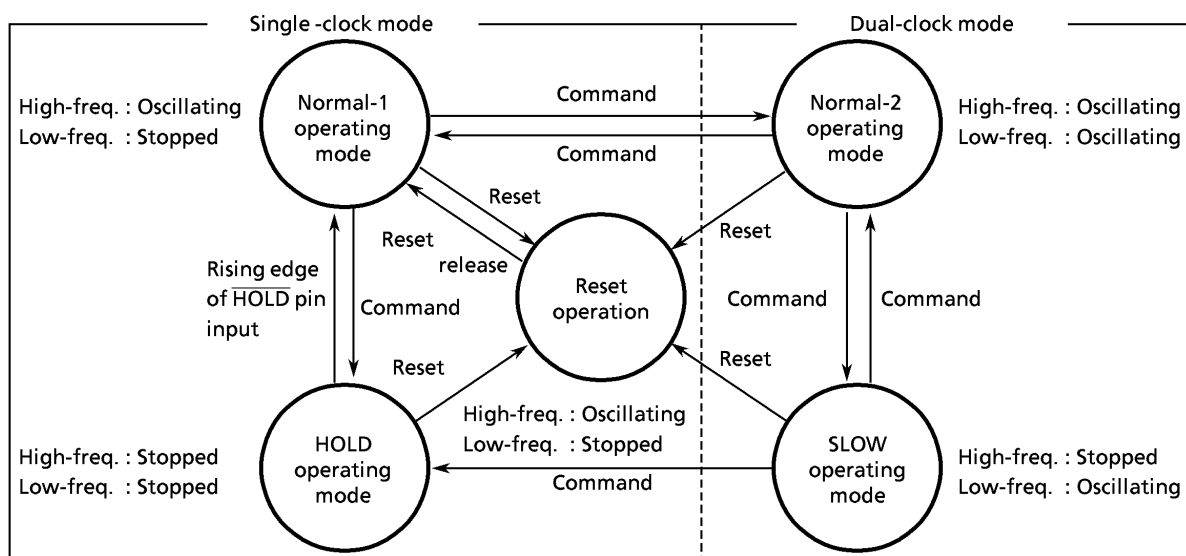
2.7.2   Timing Generator (TG)

The timing generator produces the system clocks from clock pulse which are supplied to the CPU and peripheral hardweare.

Figure 2-16. Configuration of Timing Generator

2.7.3   System Clock Controller

The system clock controller starts or stops the high-frequency and low-frequency clock oscillator and switches between the basic clocks. The operating mode is generally divided into the single-clock mode and the dual-clock mode, which are controlled by command. Figure 2-16 shows the operating mode transition diagram. Figure 2-17 shows the command register. Figure 2-18 shows the status register.

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

Figure 2-17. Operating Mode Transition Diagram

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

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

(1)   System clock control

System clock control is performed by the command register (OP16). During reset, this register is initialized to "0" and the single-clock mode is selected. Each state at operating mode switching can be read from the status register (IP0E).

Note   The access to command register (OP16) may cause the outputs over the eighth stage of timing generator to precede that to be expected by maximum $2^7/fc$ or $1/fs$ [s].

System clock control command register   (Port address   OP16)

3	2	1	0
DCLK3	SLCK	DCLK1	DWUT

(Initial value   0000)

DCLK1 / DCLK3	Selects operation mode

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

SLCK	Selects input clock for the interval timer	(Note2)

0   : $fc/2^7$ [Hz]   Example   At fc = 960kHz
1   : fs

DWUT	Sets the warm-up time

Example   At fc = 960kHz
                       fs = 32.8kHz

0 : $2^8/fs + 2^9/fc$ [s]   ······   8.3 [ms]
1 : $2^{11}/fs + 2^9/fc$ ······   63.0

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

Figure 2-18. System Clock Control Command Register

System clock control status register (Port address IP0E)

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

SMF	Low-frequency clock oscillating state

0 : Not oscillating or unstable oscillation
1 : Stable oscillation

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

	┌──Single clock mode──┐	┌── Dual clock mode──┐
0 :	$\overline{\text{HOLD}}$ pin at "H" level	In normal operation
1 :	$\overline{\text{HOLD}}$ pin at "L" level	In SLOW operation

Figure 2-19. System Clock Control Status Register

(1) Single-clock mode

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

(2) Dual-clock mode

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

(3) System clock switching control

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

a. Switching from Normal-2 operation to SLOW operation

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

b. Returning from SLOW operation to Normal-2 operation

Bit 1 of the command register is cleared to "0" and, at the same time,the warm-up time for return is set to DWUT. When the warm-up time has passed, the Normal-2 operation takes place. By monitoring SLS (bit 0 of the status register), the current operating mode can be known.
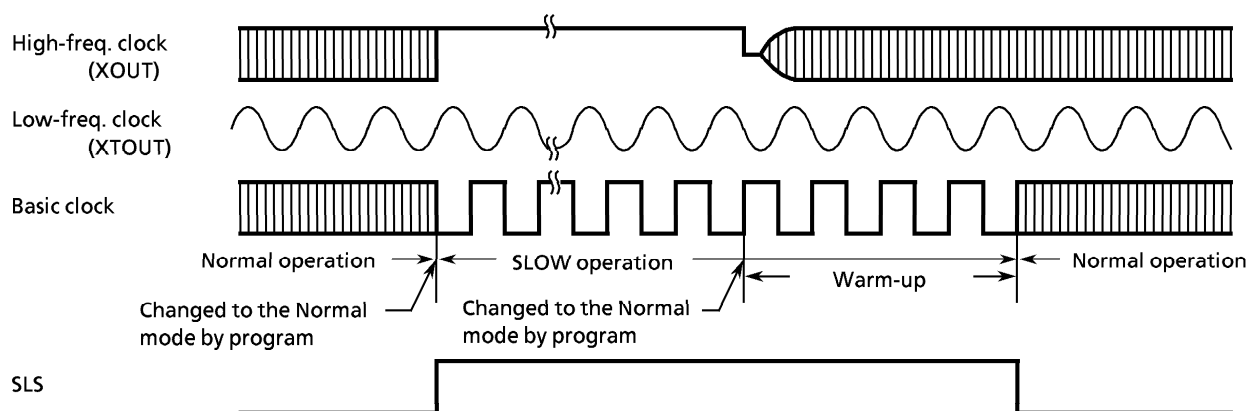

Figure 2-20. System Clock Switching Timing

2.7.4 Instruction Cycle

The instruction execution and on-chip peripheral hardware operations are performed in synchronization with the basic clock. The smallest unit of instruction execution is called the "instruction cycle". The TLCS-470A series instruction set has 3 kinds of instructions, 1-cycle instruction to 3-cycle instruction. Each instruction cycle consists of 4 states (S1 through S4). Each state consists of 2 basic clock pulses.
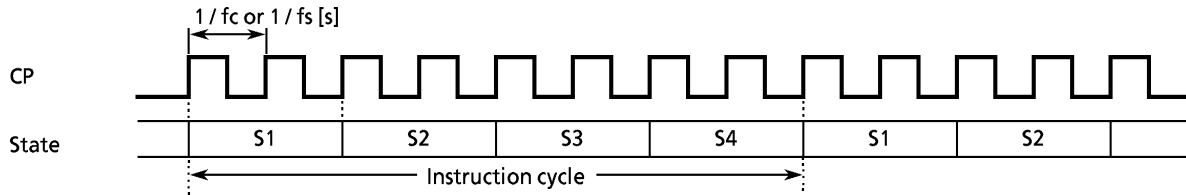

Figure 2-21. Instruction Cycle

2.7.5 HOLD Operational Mode

The hold control is also available in the 47C853 from the K0 port in addition to the control by the $\overline{\text{HOLD}}$ pin. Figure 2-21 shows the KE0 input, K103 input and hold control signal.

The KE port has been allocated to the least significant bit of the port address IP0E and the most significant bit of IP10; as shown in the figure, it is read as the logical operational result of the $\overline{\text{KE0}}$ pin input and K0 pin input.

For example, when the $\overline{\text{KE0}}$ pin is "L" and all four K0 pins are "H", then "1" is read for the KE0 input data; in other cases, "0" is read for the KE0 input data.

Figure 2-22. Relationship of KE0 Input, K103 Input and HOLD Control

The HOLD control by the K0 port input can be inhibited. The selection is made by the command register (OP17).

When monitoring the HOLD control input status, the bit 0 of IP0E can be used in the single clock mode. However, since the bit is used in operational status monitoring (SLS) and cannot be used when in the dual clock mode, use bit 3 of IP10.

HOLD Control Input Selection
(Port address : OP17   Initial Value : 1001)

Figure 2-23. Command Register

The HOLD operational mode is released when the HOLD control signal increases from "0" to "1". Therefore, the HOLD operation can be released by either shutting off the hook switch or pressing the key by connecting the hook switch to the $\overline{\text{HOLD}}$ pin and the key switch to the K0 pin; this gives access to the on-hook dialing function. In this case, HCS must be cleared to "0".

2.7.4 Instruction Cycle

The instruction execution and on-chip peripheral hardware operations are performed in synchronization with the basic clock. The smallest unit of instruction execution is called the "instruction cycle". The TLCS-470A series instruction set has 3 kinds of instructions, 1-cycle instruction to 3-cycle instruction. Each instruction cycle consists of 4 states (S1 through S4). Each state consists of 2 basic clock pulses.
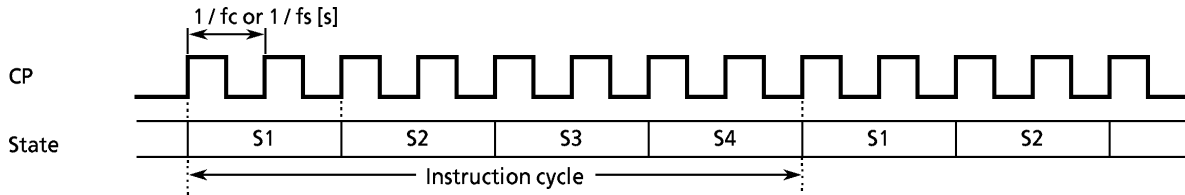

Figure 2-21. Instruction Cycle

2.7.5 HOLD Operational Mode

The hold control is also available in the 47C853 from the K0 port in addition to the control by the $\overline{\text{HOLD}}$ pin. Figure 2-21 shows the KE0 input, K103 input and hold control signal.

The KE port has been allocated to the least significant bit of the port address IP0E and the most significant bit of IP10; as shown in the figure, it is read as the logical operational result of the $\overline{\text{KE0}}$ pin input and K0 pin input.

For example, when the $\overline{\text{KE0}}$ pin is "L" and all four K0 pins are "H", then "1" is read for the KE0 input data; in other cases, "0" is read for the KE0 input data.
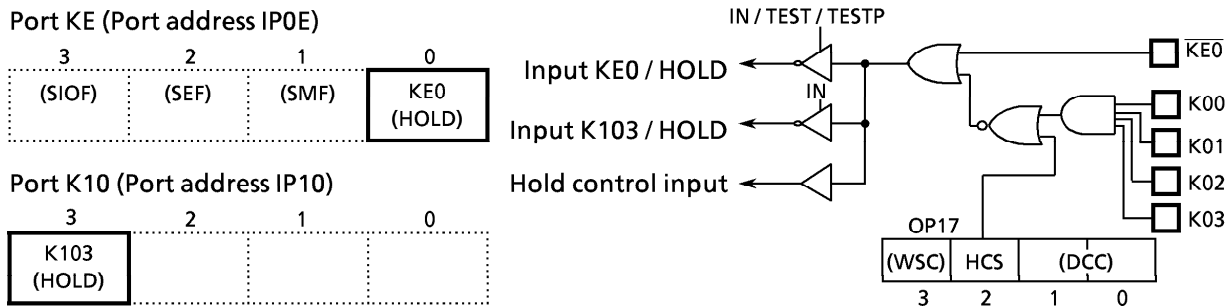

Figure 2-22. Relationship of KE0 Input, K103 Input and HOLD Control

The HOLD control by the K0 port input can be inhibited. The selection is made by the command register (OP17).

When monitoring the HOLD control input status, the bit 0 of IP0E can be used in the single clock mode. However, since the bit is used in operational status monitoring (SLS) and cannot be used when in the dual clock mode, use bit 3 of IP10.

HOLD Control Input Selection
(Port address : OP17   Initial Value : 1001)

Figure 2-23. Command Register

The HOLD operational mode is released when the HOLD control signal increases from "0" to "1". Therefore, the HOLD operation can be released by either shutting off the hook switch or pressing the key by connecting the hook switch to the $\overline{\text{HOLD}}$ pin and the key switch to the K0 pin; this gives access to the on-hook dialing function. In this case, HCS must be cleared to "0".
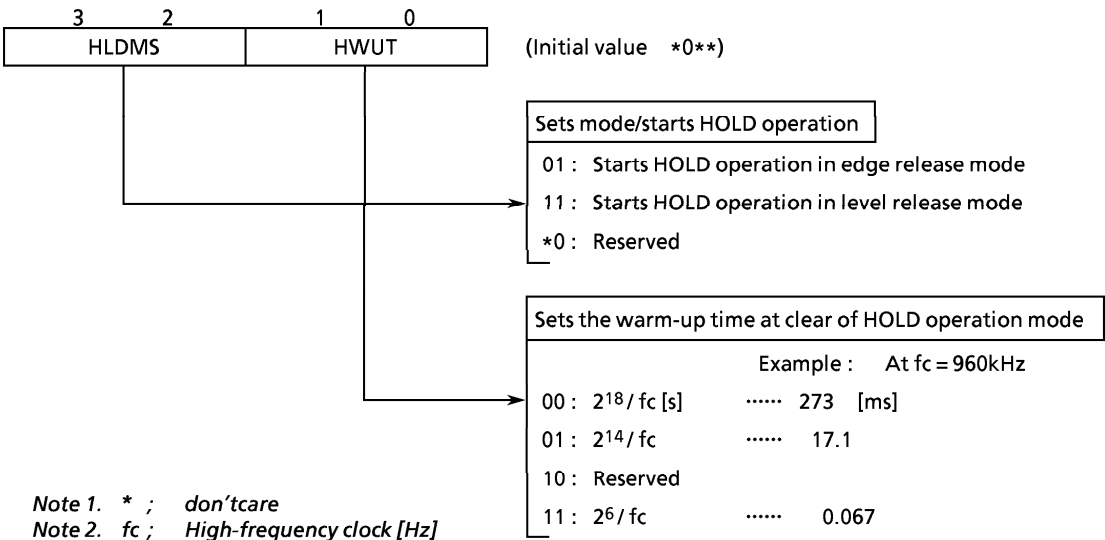
(1)   Starts the HOLD operating mode

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

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

② The interval timer is cleared to "0".

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

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

HOLD operation mode command register   (Port address   OP10)

3	2	1	0
HLDMS		HWUT	

(Initial value   *0**)

Sets mode/starts HOLD operation

01 :   Starts HOLD operation in edge release mode

11 :   Starts HOLD operation in level release mode

*0 :   Reserved

Sets the warm-up time at clear of HOLD operation mode

Example :   At fc = 960kHz

00 : 2^{18} / fc [s]   ······   273   [ms]

01 : 2^{14} / fc   ······   17.1

10 :   Reserved

11 : 2^{6} / fc   ······   0.067

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

HOLD operation mode status register

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

(Port address IP0E)

Note.   Usable in single clock mode only.

$\overline{HOLD}$ pin input state

0 : $\overline{HOLD}$ pin is high

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

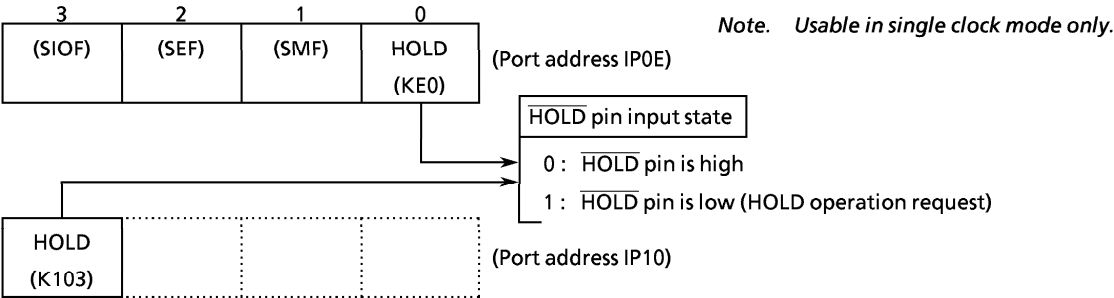| HOLD (K103)			

(Port address IP10)

Figure 2-24.   HOLD Operating Mode Command Register/Status Register

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

a.   Level-sensitive release mode
In this mode, the $\overline{\text{HOLD}}$ operating is released by setting the $\overline{\text{HOLD}}$ pin to the high level. This mode is used for the capacitor backup with the main power off or for the battery backup for long hours.   If the instruction to start the HOLD operation is executed with the $\overline{\text{HOLD}}$ pin input being high, the HOLD operation does not start but the clear sequence (warm-up) sets in immediately. Therefore, to start the HOLD operation in the level-sensitive release mode, that the $\overline{\text{HOLD}}$ pin input is low (the HOLD operation request) must be recognized in program. This recognition is one of the two ways below:
① Testing HOLD
② Applying the $\overline{\text{HOLD}}$ pin input also to the $\overline{\text{INT1}}$ pin to generate the external interrupt 1 request.
In single clock mode.

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

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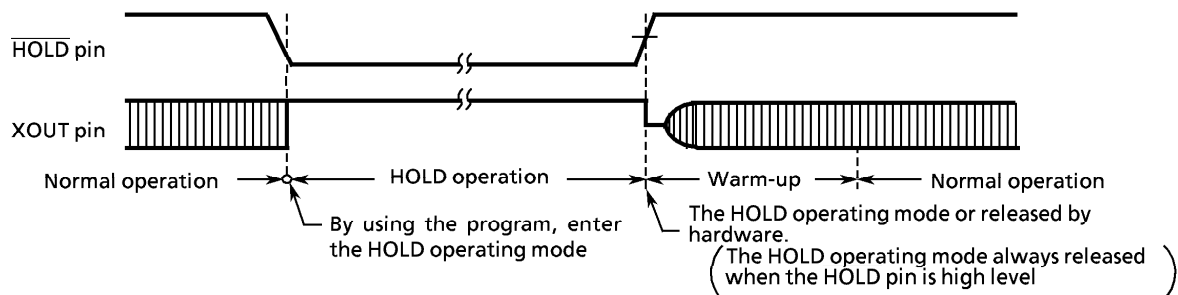

Figure 2-25. Level-sensitive Release Mode

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

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

```
                      LD       A, #0101B       ; OP10 ← 0101B
                      OUT      A, %OP10
```
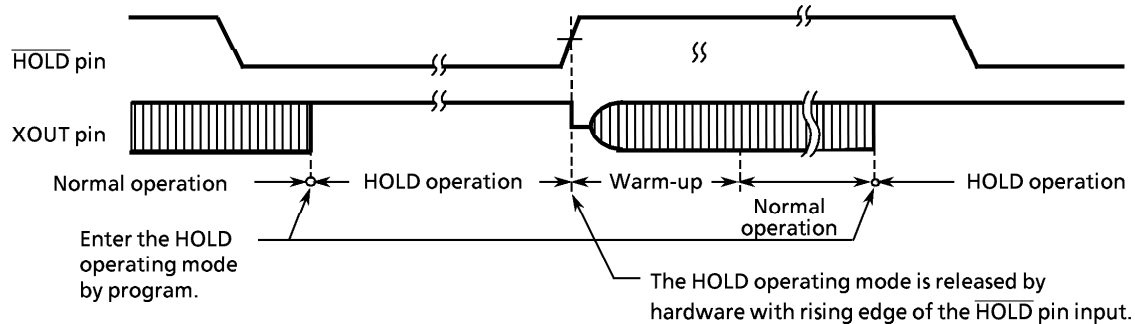
Figure 2-26. Edge-sensitive Release Mode

Note ;   In the HOLD operation, the dissipation of the power associated with the oscillator and the internal hardware is lowered; however, the power dissipation associated with the pin interface (depending on the external circuitry and program) is not directly determined by the hardware operation of the HOLD feature. This point should be considered in the system design and the interface circuit design. In the CMOS circuitry, little current flows when the input level is stable at the power voltage level (V_{DD}/V_{SS}); however, when the input level gets higher than the power voltage level (by approximately 0.3 to 0.5V), a current begins to flow. Therefore, if cutting off the output transistor at an I/O port (the open drain output pin with an input transistor connected) puts the pin signal into the high-impedance state, a current flow across the ports input transistor, requiring to fix the level by pull-up or other means.

(2)   Releases the HOLD operating mode
   The HOLD operating mode is released in the following sequence:
      ① The high-freg. oscillator starts.
      ② Warm-up is performed to acquire the time for stabilizing oscillation. During the warm-up, the internal operations are all stopped. One of three warm-up times can be selected by program depending on the characteristics of the oscillator used.
      ③ When the warm-up time has passed, an ordinary operation restarts in single clock mode from the instruction next to the instruction which starts the HOLD operation. At this time, the interval timer starts from the reset state "0".
      ※ *The warm-up time is obtained by dividing the basic clock by the interval timer, so that, if the frequency at clearing the HOLD operation is unstable, the warm-up time shown in Figure 2-24 includes an error. Therefore, the warm-up time must be handled as an approximate value.*

   The HOLD operation is also released by setting the $\overline{RESET}$ pin to the low level. In this case, the normal reset operation follows immediately.

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

(3)   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".

2.8 INTERRUPT FUNCTION

(1) Interrupt Controller

There are 6 interrupt sources (2 external and 4 internal). The prioritized multiple interrupt capability is supported. The interrupt latches (IL_5 through IL_0) to hold interrupt requests are provided for the interrupt sources. Each interrupt latch is set to "1" when an interrupt request is made, asking the CPU to accept the interrupt. The acceptance of interrupt can be permitted or prohibited by program through the interrupt enable master flip-flop (EIF) and interrupt enable register (EIR). When two or more interrupts occur simultaneously, the one with the highest priority determined by hardware is serviced first.

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

Table 2-2. Interrupt Sources
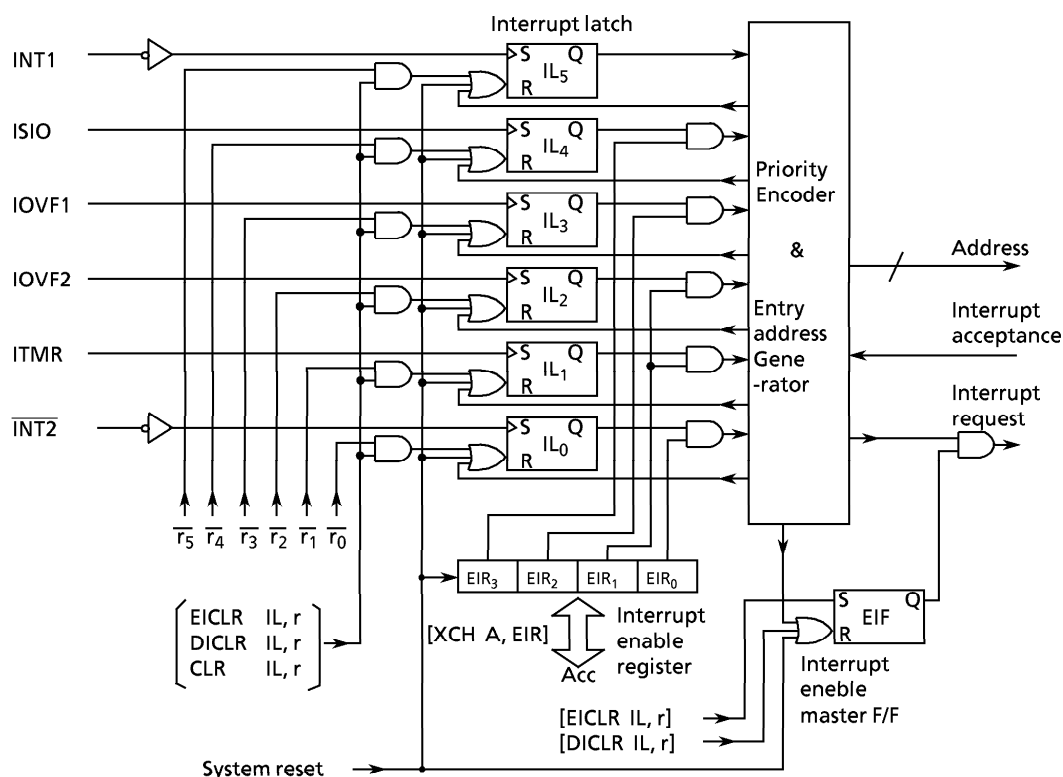
Figure 2-27. Interrupt Controller Block Diagram

a. Interrupt enable master flip-flop (EIF)

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

accepted, the EIF is cleared to "0", temporarily disabling the acceptance of subsequent interrupts. When the interrupt service program has been executed, the EIF is set to "1" by the execution of the interrupt return instruction [RETI], being put in the enabled state again. Set or clear of the EIF in program is performed by instructions [EICLR IL,r] and [DICLR IL,r], respectively. The EIF is initialized to "0" during reset.

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

c. Interrupt latches (IL$_5$ to IL$_0$)
An interrupt latch (IL) is provided for each interrupt source. It is set to "1" when an interrupt request is made to ask the CPU for accepting the interrupt. Each latch is cleared to "0" upon acceptance of the interrupt. It is initialized to "0" during the reset. The interrupt latches can be cleared independently by interrupt latch operation instructions ([EICLR IL,r], [DICLR IL,r], and [CLR IL,r] to make them cancel interrupt requests or initialize by program. When the value of instruction field(r) is "0", the IL is cleared; when the value is "1", the IL is held. Note that the interrupt latches cannot be set by instruction.

Example 1 : To enable IOVF1, INT1, and INT2 interrupts.
    LD   A, #0101B   ; EIR ← 0101$_B$
    XCH   A, EIR
    EICLR   IL, #111111B   ; EIF ← 1
Example 2 : To set the EIF to "1" and to clear the interrupt latches except ISIO to "0".
    EICLR   IL, #010000B   ; EIF ← 1, IL5 ← 0, IL3~IL0 ← 0

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

The interrupt acknowledge processing consists of the following sequence:
① The contents of the program counter and the flags are saved on the stack.
② The interrupt entry address corresponding to the interrupt source is set to the program counter.
③ The status flag is set to "1".
④ The EIF is cleared to "0", temporarily disabling the acceptance of subsequent interrupts.
⑤ The IL for the accepted interrupt source is cleared to "0".
⑥ The instruction stored at the interrupt entry address is executed. (Generally, in the program memory space at the interrupt entry address, the branch instruction to each interrupt processing program is stored. Note that the interrupt entry address is assigned every 2-byte, so that the long branch instruction can not be stored in the program normally. The interrupt service program is assigned to the memory locations 0000 to 0FFF$_H$.)

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

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

PINT1 : TEST 05H, 0 ; RAM[05_H]0 = 1, returns without the interrupt service

B SINT1

RETI

SINT1 : ⋮

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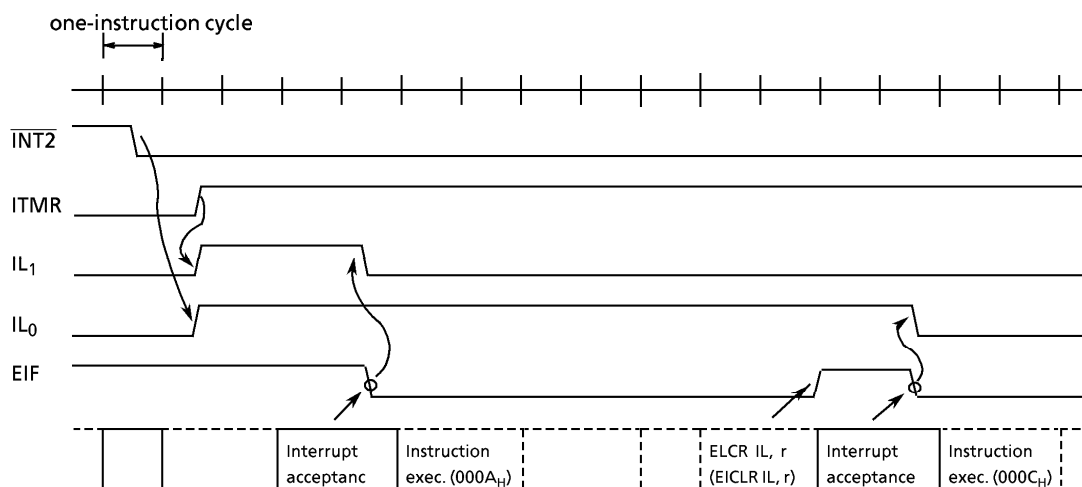
Note1. It is assumed that there is no other interrupt request and EIR = 0011_B.
Note2. The value r in the [EICLR IL, r] instruction is assumed as 11111_B.
Note3. ⌐‾‾⌐ denotes the execution of an instruction.

Figure 2-28. Interrupt Timing Chart (Example)

The interrupt return instruction [RETI] performs the following operations:
① Restores the contents of the program counter and the flags from the stack.
② Sets the EIF to "1" to provide the interrupt enable state again. That is the next interrupt request is accepted immediately after execution of the interrupt return instruction.

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

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

Example 1 :   To save/restore accumlator and HL register pair.

                XCH   HL, GSAV1   ; RAM [GSAV1] ↔ HL

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

    Note   *The lower 2 bits of GSAV1 should be "0".*

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

                CLR   GSAV3, 0   ; RAM [GSAV3] 0 ← 0

                CLR   GSAV3, 1   ; RAM [GSAV3] 1 ← 0

                TEST   DMB0   ; DMB0 = 0, skip

                B   SKIPS0

                SET   GSAV3, 0   ; RAM [GSAV3] 0 ← 1

    SKIPS0 :   TEST   DMB1   ; DMB1 = 0, skip

                B   SKIPS1

                SET   GSAV3, 1   ; RAM [GSAV3]1 ← 1

    SKIPS1 :

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

                LD   DMB, #0000B   ; DMB ← 0

                TEST   GSAV3, 0   ; RAM [GSAV3] 0 = 0, skip

                B   SKIPR0

                SET   DMB0   ; DMB0 ← 1

    SKIPR0 :   TEST   GSAV3, 1   ; RAM [GSAV3]1 = 0, skip

                B   SKIPR1

                SET   DMB1   ; DMB1 ← 1

    SKIPR1 :

(3) External Interrupt

When an external interrupt (INT1 or INT2) occurs, the interrupt latch is set at the falling edge of the corresponding pin input ($\overline{\text{INT1}}$ or $\overline{\text{INT2}}$) . The INT1 interrupt cannot be disabled by the EIR, so that it is always accepted in the interrupt enable state (EIF = "1") . Therefore, INT1 is used for an interrupt with high priority such an emergency interrupt. When R82 ($\overline{\text{INT1}}$) pin is used for the I/O port, the INT1 interrupt occurs at the falling edge of the pin input, so that the interrupt return [RET1] instruction must be stored at the interrupt entry address to perform dummy interrupt processing.

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

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

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

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

   ※   The external interrupt through the rising edge.

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

2.9 RESET FUNCTION

When the $\overline{\text{RESET}}$ pin is held to the low level for at 3 or more instruction cycles when the power voltage is within the operating voltage range and the oscillation is stable, reset is performed to initialize the internal states.

When the $\overline{\text{RESET}}$ pin input goes high, the reset is cleared and program execution starts from address 0000_H. The $\overline{\text{RESET}}$ pin is a hysteresis input with a pull-up resistor (220 kΩ typ.). Externally attaching a capacitor and a diode implement a simplified power-on-reset.
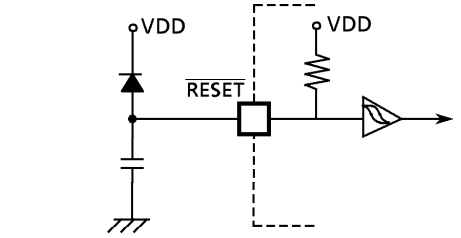
Figure 2-29. Simplified power-on-reset circuit

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

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

2.9.1 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 $\overline{\text{HOLD}}$ pin the waveform synchronized with the power voltage variation.
  ③ Set the HOLD operating mode during the power is off.
  ④ Perform reset by using the output port of sink open drain (initial "Hi-Z") after relesing HOLD operation.
  ⑤ Apply to an input port the power-on detect signal, and skip the initialize routines such as clearing RAM.
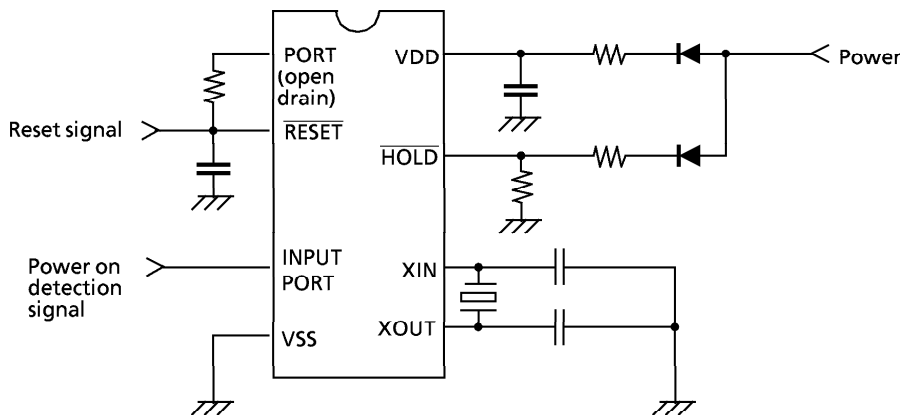
Figure 2-30 shows Warm-start Circuit Example

Figure 2-30. Warm-start Circuit Example

3.   PERIPHERAL HARDWARE FUNCTION

3.1   Ports

[Data transfer with external circuit] and [Command status data with internal circuitry] are done using input/output instruction.

There are four kinds to port
 (1) Input/Output ports
 (2) Command register
 (3) Status register
 (4) Data register
, and it is divided, and port address is touched.

Individual port is selected by designating port address by input/output instruction.
Show the input/output instruction that can access arrangement of port address and the port in table3.

3.1.1   I/O Timing

(1) Input timing
External data is read from an input port or an I/O port in the S3 state of the second instruction cycle during the input instruction (2-cycle instruction) execution. This timing cannot be recognized from the outside, so that the transient input such as chattering must be processed by program.
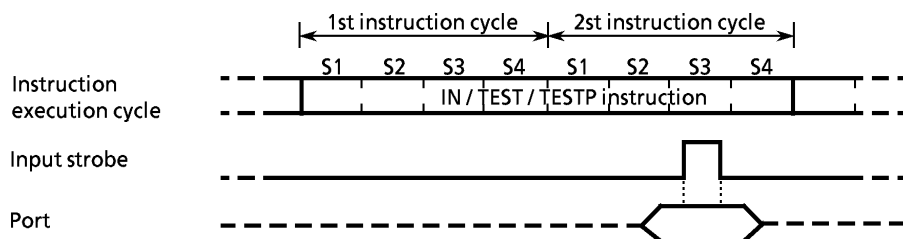
Figure 3-1. Input Timing

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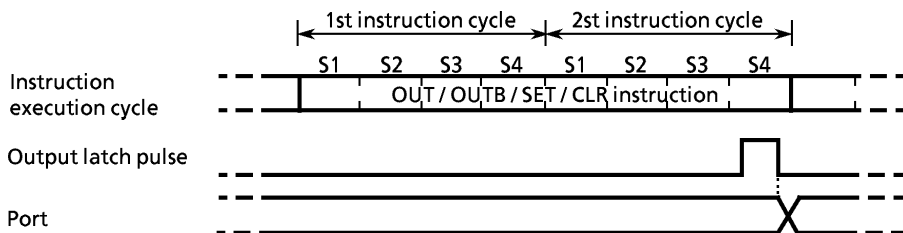

Figure 3-2. Output Timing

3.1.2   I/O Ports

The 47C853 has 10 I/O ports (35 pins) as follows.

①	Port K0	;	4-bit input (shared by hold request/release signal input)
②	Port R3	;	4-bit input/output
③	Port R4, R5, R6	;	4-bit input/output
④	Port R7	;	4-bit input/output (shared by low-frequency oscillator connecting pin output and watchdog timer output)
⑤	Port R8	;	4-bit input/output (shared by external interrupt input and timer/counter input)
⑥	Port R9	;	3-bit input/output (shared by serial port)
⑦	Port P14	;	3-bit output (P140 is shared by BEEP output)
⑧	Port KE	;	1-bit sense input (shared by hold request/release signal input)

Each output port contains a latch, which holds the output data. The input ports have no latch, therefore, it is desired to hold data externally until it is read or read twice or more before processing
Table 3-1. shows the port address assignments and the I/O instructions for accessing the ports.

Port address (**)	Port — Input (IP**)	Port — 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
00H	K0 Input port	ROW register	○	—	—	—	—	○	—
01	ROW register	COLUMN register	○	○	○	○ (Note2)	—	○	—
02	COLUMN register	R3 Output port	○	○	○	○	○	○	—
03	R3 Input port	R4 Output port	○	○	○	—	○	○	○
04	R4 Input port	R5 Output port	○	○	○	—	○	○	○
05	R5 Input port	R6 Output port	○	○	○	—	○	○	○
06	R6 Irput port	R7 Output port	○	○	○	—	○	○	○
07	R7 Input port	R8 Output port	○	○	○	—	○	○	—
08	R8 Input port	R9 Output port	○	○	○	—	○	○	—
09	R9 Input port	——	○	○	○	—	○	○	—
0A	——	——	—	—	—	—	—	—	—
0B	——	——	—	—	—	—	—	—	—
0C	——	TONE command register	—	○	○	—	—	—	—
0D	Status input (Note 3)	——	○	—	—	—	—	○	—
0E	——	——	—	—	—	—	—	—	—
0F	Serial receive buffer	Serial receive buffer	○	○	○	—	—	○	—
10H	HOLD pin status	HOLD operation mode control	○	○	—	—	—	—	—
11	——	——	—	—	—	—	—	—	—
12	——	——	—	—	—	—	—	—	—
13	——	BEEP output control	—	○ (Note5)	—	—	—	—	—
14	——	P14 Output port (BEEP output)	—	○	—	—	—	—	—
15	——	Watchdog timer control	—	○	—	—	—	—	—
16	——	System clock control	—	○	—	—	—	—	—
17	——	Command register (Note 4)	—	○	—	—	—	—	—
18	——	——	—	—	—	—	—	—	—
19	——	Interval Timer interrupt control	—	○	—	—	—	—	—
1A	——	——	—	○	—	—	—	—	—
1B	——	——	—	—	—	—	—	—	—
1C	——	Timer/Counter 1 control	—	○	—	—	—	—	—
1D	——	Timer/Counter 2 control	—	○	—	—	—	—	—
1E	——	Serial interface 1 control	—	○	—	—	—	—	—
1F	——	Serial interface 2 control	—	○	—	—	—	—	—

Note 1. "——" means the reserved state. Unavailable for the user program.
Note 2. Automatic access to ROW and COLUMN registers by the [OUTB @HL] instruction.
Note 3. Serial interface, system clock control circuit status, and HOLD (KE0) pin input.
Note 4. HOLD control input selection, warm start control and DTMF input clock selection.
Note 5. To output data to P14 output port, use "MOV A, SPW13".

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

(1) Port K0 (K03~K00)
The 4-bit input port with pull-up resistors is shared by the hold request/release signal input.

Port K0 (Port address   IP00)

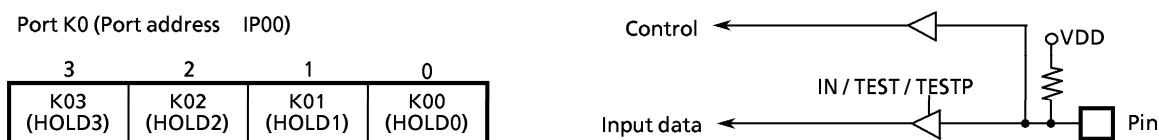| 3	2	1	0
K03 (HOLD3)	K02 (HOLD2)	K01 (HOLD1)	K00 (HOLD0)

Figure 3-3. Port K0

(2) Port R3 (R33~R30)
This 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.
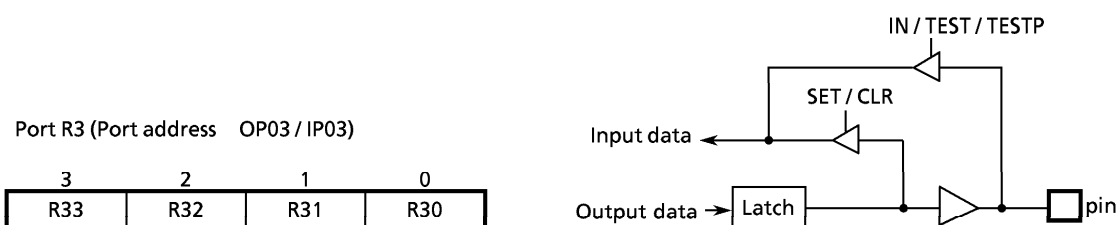
Port R3 (Port address   OP03 / IP03)

3	2	1	0
R33	R32	R31	R30

Figure 3-4. Port R3

(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 3 2 1 0	Pin	L register 3 2 1 0	Pin	L register 3 2 1 0	Pin	L register 3 2 1 0	Pin
0 0 0 0	R40	0 1 0 0	R50	1 0 0 0	R60	1 1 0 0	R70
0 0 0 1	R41	0 1 0 1	R51	1 0 0 1	R61	1 1 0 1	R71
0 0 1 0	R42	0 1 1 0	R52	1 0 1 0	R62	1 1 1 0	R72
0 0 1 1	R43	0 1 1 1	R53	1 0 1 1	R63	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∗)

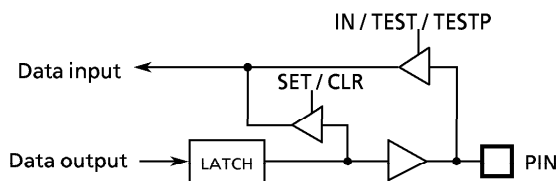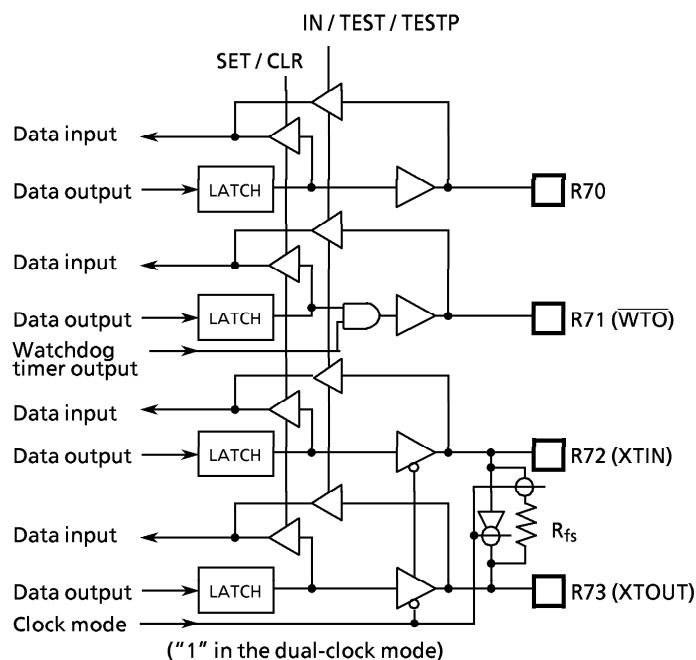| 3	2	1	0
R∗3	R∗2	R∗1	R∗0

∗ ; 4 to 6

Figure 3-5. Ports R4, R5, R6

Figure 3-6. Port R7

Port R7 (Port address   OP07 / IP07)

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

(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 ($\overline{\text{INT1}}$) pin is used for an I/O port, external interrupt 1 occurs upon detection of the falling edge of pin input, and if the interrupt enable master flip-flop is enabled, the interrupt request is always accepted, so that a dummy interrupt processing must be performed (only the interrupt return instruction [RETI] is executed).
With R80 ($\overline{\text{INT2}}$) pin, external interrupt 2 occurs like R82 but bit 0 of the interrupt enable register is only kept at "0", not accepting the interrupt request.

Port R8 (Port address   OP08 / IP08)

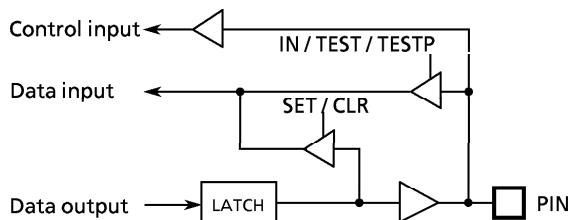| 3	2	1	0
R83 (T1)	R82 ($\overline{\text{INT1}}$)	R81 (T2)	R80 ($\overline{\text{INT2}}$)

Figure 3-7. Port R8

(5) Port R9 (R92 - R90)

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

To use port R9 for the serial port, the latch should be set to "1". To use port R9 for an ordinary I/O port, the serial port must be disabled. Although R93 pin does not exist actually, execution of the set or clear instruction for R93 ([SET %OP09,3] or [CLR %OP09,3]) affects the operation of the internal CPU. Therefore, these instructions should not be execution on R93. However, other instructions may be used, in which an uncertain value is read upon execution of an input instruction.

Figure 3-8. Port R9

(6) Port P14 (P142~P140)

This is a 3-bit output port with a latch. The latch is initialized to "1" during reset. In addition, pin 140 is shared by the BEEP output. When used as the BEEP output, the latch must be set to "1". According to the system structure, the instruction "OUT A, % op 14" is not accepted. To output data to P14 port, use the instruction "MOV A, SPW13".

Example : To output "5" to port P14:

        LD   A, #5   ; Acc   ← 5
        MOV   A, SPW13   ; P14 port← Acc

Pins 140 and 141 are also shared by the test input pin during the test mode for tone output. See "3.2.4 Test Mode for Tone Output" on the testing mode for tone.
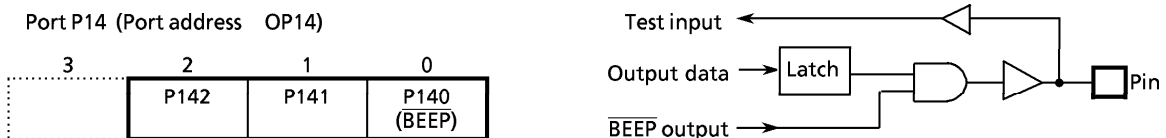

Figure 3-9. Port P14

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

The logic operation result of the $\overline{\text{KE0}}$ pin input and port K0 input is read as the least significant bit of the port address IP0E in port KE of the 47C853. For details, see "2.7.5 HOLD Operational Mode".

3.2 Interval Timer

3.2.1 Configuration of Interval Timer

The interval timer consists of a 19-stage binary counter with a divide-by-3 prescaler. The source clock to the interval timer and its input stage depend on the operating mode as shown below.

During reset, the binary counter is cleared to "0". However, the prescaler is not cleared.
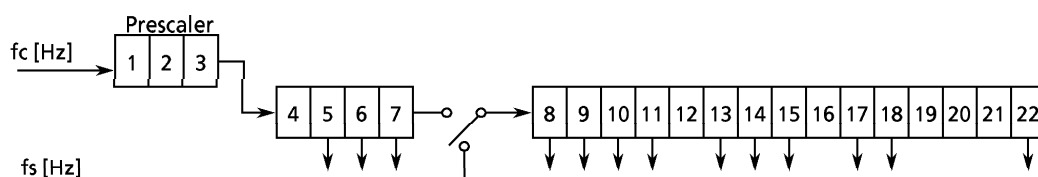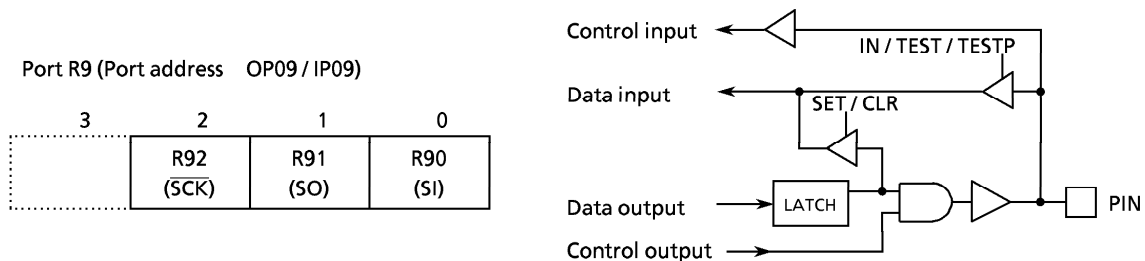

Figure 3-10. Configuration of Interval Timer

(5) Port R9 (R92 - R90)

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

To use port R9 for the serial port, the latch should be set to "1". To use port R9 for an ordinary I/O port, the serial port must be disabled. Although R93 pin does not exist actually, execution of the set or clear instruction for R93 ([SET %OP09,3] or [CLR %OP09,3]) affects the operation of the internal CPU. Therefore, these instructions should not be execution on R93. However, other instructions may be used, in which an uncertain value is read upon execution of an input instruction.

Port R9 (Port address   OP09 / IP09)

3	2	1	0
	R92 (SCK̄)	R91 (SO)	R90 (SI)

Figure 3-8. Port R9

(6) Port P14 (P142~P140)

This is a 3-bit output port with a latch. The latch is initialized to "1" during reset. In addition, pin 140 is shared by the BEEP output. When used as the BEEP output, the latch must be set to "1". According to the system structure, the instruction "OUT A, % op 14" is not accepted. To output data to P14 port, use the instruction "MOV A, SPW13".

Example :   To output "5" to port P14:

```
LD     A, #5        ;  Acc      ← 5
MOV   A, SPW13     ;  P14 port← Acc
```

Pins 140 and 141 are also shared by the test input pin during the test mode for tone output. See "3.2.4 Test Mode for Tone Output" on the testing mode for tone.
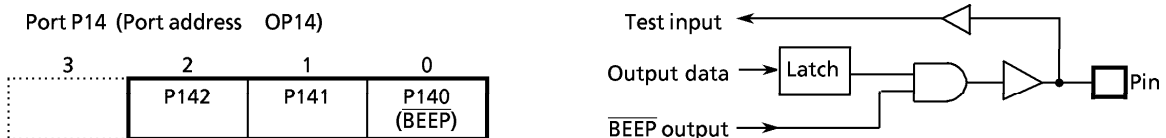
Port P14 (Port address   OP14)

3	2	1	0
	P142	P141	P140 (BEEP̄)

Figure 3-9. Port P14

(7) Port KE (KE̅0̅)

The logic operation result of the KE̅0̅ pin input and port K0 input is read as the least significant bit of the port address IP0E in port KE of the 47C853. For details, see "2.7.5 HOLD Operational Mode".

3.2   Interval Timer

3.2.1   Configuration of Interval Timer

The interval timer consists of a 19-stage binary counter with a divide-by-3 prescaler. The source clock to the interval timer and its input stage depend on the operating mode as shown below.

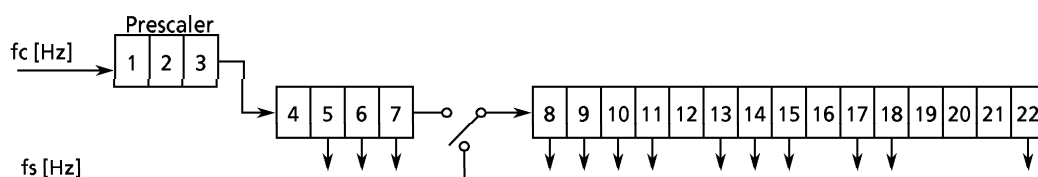During reset, the binary counter is cleared to "0". However, the prescaler is not cleared.

Figure 3-10. Configuration of Interval Timer

3.2.2   Functions of Interval Timer

The interval timer provides the following functions:

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

3.2.3   Interval Timer Interrupt (ITMR)

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

    Example: To set the interval timer interrupt frequency to $fc/2^{15}$[Hz](Single clock mode)

                LD   A,#0110B   ;   OP19 ← 0110$_B$
                OUT   A,%OP19

Interval timer interrupt control command register (Port address OP19)

3	2	1	0
TMRE		TMRF	

TMRE	Interrupt Enabled / Disabled

00 : Stopped
01 : Enabled
1* : Reserved

TMRF	Interrupt frequency

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

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

(a) Command register

TMRF	Normal 1 mode	Normal 2 mode (TG INPUT)		SLOW mode	At fs = 32.768kHz
		SLCK = 0 ($fc/2^7$)	SLCK = 1 (fs)		
00		$fc/2^{11}$ [Hz]	$fs/2^4$ [Hz]	reserved	2048 [Hz]
01		$fc/2^{13}$	$fs/2^6$	〃	512
10		$fc/2^{15}$	$fs/2^8$	〃	128
11		$fc/2^{17}$	$fs/2^{10}$	$fs/2^{10}$	32

(b) Example of interrupt frequency

Figure 3-11. Interval Timer Interrupt Command Register

3.3   Timer/Counters (TC1, TC2)

The 47C853 contains two 12-bit timer/counters. RAM addresses are assigned to the count register in unit of 4 bits, permitting the initial value setting and counter reading through the RAM mauipulation instruction.

When the timer/counter is not used, the mode selection may be set to "stopped" to use the RAM at the address corresponding to the timer/counter for storing the ordinary use-processed data.
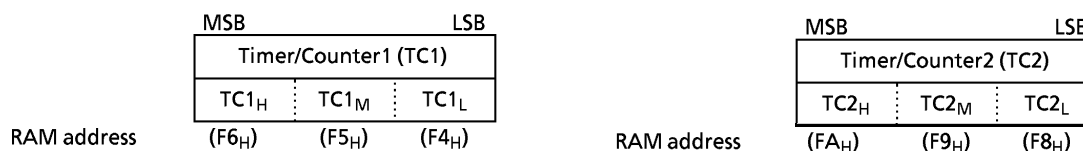
MSB		LSB
Timer/Counter1 (TC1)		
TC1$_H$	TC1$_M$	TC1$_L$

RAM address   (F6$_H$)   (F5$_H$)   (F4$_H$)

MSB		LSB
Timer/Counter2 (TC2)		
TC2$_H$	TC2$_M$	TC2$_L$

RAM address   (FA$_H$)   (F9$_H$)   (F8$_H$)

Figure3-12. The Count Registers of the Timer/Counters (TC1, TC2)

3.3.1   Functions of Timer/Counters

The timer/counters provide the following functions:

① Event counter
② Programmable timer
③ Pulse width measurement

3.3.2   Control of Timer/Counters

The timer/counters are controlled by the command registers. The command register is accessed as port address OP1C for timer/counter 1, and port address OP1D for timer/counter 2. These registers are initialized to "0" during reset.

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

3 2	1 0	
TC1MS	IPR1	(Initial value 0000)

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

IPR1	Internal pulse rate (interval timer output) select

	Single clock mode	Dual clock mode	
	Normal 1	Normal 2	SLOW
00 :	$fc / 2^6$ [Hz]	$fc / 2^6$ [Hz]	Reserved
01 :	$fc / 2^8$	$fs / 2$	Reserved
10 :	$fc / 2^{10}$	$fs / 2^3$	Reserved
11 :	$fc / 2^{14}$	$fs / 2^7$	$fs / 2^7$ [Hz]

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

3 2	1 0	
TC2MS	IPR2	(Initial value 0000)

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

IPR2	Internal pulse rate (interval timer output) select

	Single clock mode	Dual clock mode	
	Normal 1	Normal 2	SLOW
00 :	$fc / 2^{10}$ [Hz]	$fs / 2^3$ [Hz]	Reserved
01 :	$fc / 2^{14}$	$fs / 2^7$	$fs / 2^7$ [Hz]
10 :	$fc / 2^{18}$	$fs / 2^{11}$	$fs / 2^{11}$
11 :	$fc / 2^{22}$	$fs / 2^{15}$	$fs / 2^{15}$

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

Figure 3-13. Timer/Counter Control Command Register

3.3.1 Functions of Timer/Counters

The timer/counters provide the following functions:

① Event counter

② Programmable timer

③ Pulse width measurement

3.3.2 Control of Timer/Counters

The timer/counters are controlled by the command registers. The command register is accessed as port address OP1C for timer/counter 1, and port address OP1D for timer/counter 2. These registers are initialized to "0" during reset.

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

3	2	1	0	
TC1MS		IPR1		(Initial value 0000)

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

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

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

3	2	1	0	
TC2MS		IPR2		(Initial value 0000)

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

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

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

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 1 instruction cycle after the current instruction execution, during which the execution of a next instruction and the acceptance of an interrupt are delayed. If counting is requested by both TC1 and TC2 simultaneously, the request by TC1 is preferred. The request from TC2 is accepted in the next instruction cycle. Therefore, during a count operation, the apparent instruction execution speed drops as counting occurs more frequently. The timer/counter causes an interrupt upon occurrence of an overflow (a transition of the count value from FFF_H to 000_H). If the timer/counter is during the interrupt enabled state and the overflow interrupt is accepted immediately after its occurrence, the interrupt is processed in the sequence shown in Figure 3-14. Note that counting continues if there is a count request after overflow occurrence.
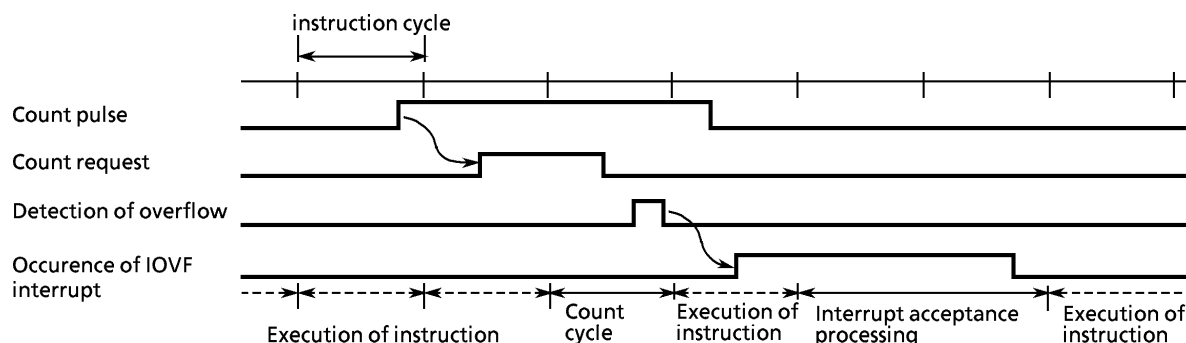
Figure 3-14. Timer/Counter Overflow Interrupt Timing

(1) Event counter mode

In the event counter mode, the timer/counter increments at each rising edge of the external pin (T1, T2) input. T1, T2 pins are shared by R83, R81 pins. Output latch of R83, R81, are set to "1" when used as timer/counter input. Also output latch is initialized "1" during reset. The maximum applied frequency of the external pin input is fc/32 for the 1-channel operation; for the 2-channel operation, the frequency is fc/32 for TC1 and fc/40 for TC2. The apparent instruction execution speed drops most to $(9/11) \times 100 = 82\%$ 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 $8.3\mu s$ drops to $15.1\mu s$.

Example: To operate TC2 in the event counter mode.
```
        LD      A,#0100B        ; OP1D ← 01**B
        OUT     A,%OP1D
```
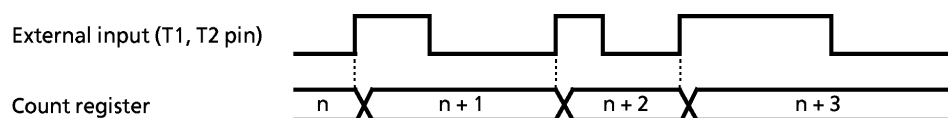
Figure 3-15. Event Counter Mode Timing Chart

(2) Timer mode

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

When an internal pulse rate of $fc/2^{10}$ is used, a count operation is inserted once every 128 instruction cycles, so that the apparent instruction execution speed drops by $(1/127) \times 100 = 0.8\%$. For example, the instruction execution speed of $8.3\mu s$ drops to $8.366\mu s$. In the timer mode, R83 (T1) and R81 (T2) pins provide the ordinary I/O ports.

Example: To generate an overflow interrupt (at fc = 960kHz) by TC1 after 100 ms.

```
            LD      HL,#0F4H          ;   TC1 ← FA2_H (Setting of count register)
            ST      #2,@HL+
            ST      #0A,@HL+
            ST      #0FH,@HL+
            LD      A,#1010B          ;   OP1C ← 1010_B (Timer mode rate fc/2^10)
            OUT     A,%0P1C
            LD      A,#0100B          ;   EIR ← 0100_B (Enables interrupt)
            XCH     A,EIR
            EICLR   IL,110111B        ;   EIF ← 1, IL_3 ← 0
```

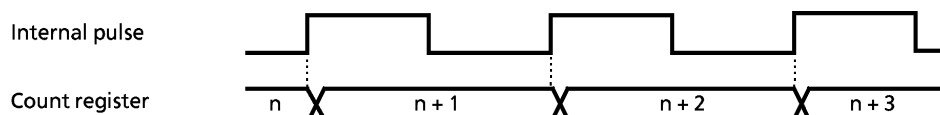Internal pulse

Count register   n   n + 1   n + 2   n + 3

Figure 3-16. Timer Mode Timing Chart

※ The drop of the apparent instruction execution speed can be calculated by the following way

$$1 \div \left\{ \frac{(\text{basical clock freq.}) / 8}{(\text{internal pulse rate})} - 1 \right\} \times 100 \, [\%]$$

※ Calculating the initial value of the count register

$2^{12} - (\text{interrupt setting time}) \times (\text{internal pulse rate})$

For example, to generate an overflow interrupt after 100 ms at fc = 960kHz with the internal pulse rate of $fc/2^{10}$, set the following value to the count register as the initial value:

$2^{12} - (100 \times 10^{-3}) \times (960 \times 10^3 / 2^{10}) = 4002$ (FA2_H)

NORMAL1 mode	NOMAL2 mode (TG INPUT)		SLOW mode	MAX SETTING TIME (At fs = 32.768kHz)		
	SLCK = 0 (fc / 2^7)	SLCK = 1 (fs)				
$fc/2^6$ [Hz]	$fc/2^6$ [Hz]	reserved	$2^{18}/fc$			
$fc/2^8$	fs / 2	"	$2^{20}/fc$	or	$2^{13}/fs$	(0.25)
$fc/2^{10}$	fs / 2^3	"	$2^{22}/fc$	or	$2^{15}/fs$	(1)
$fc/2^{14}$	fs / 2^7	fs / 2^7	$2^{26}/fc$	or	$2^{19}/fs$	(16)
$fc/2^{18}$	fs / 2^{11}	fs / 2^{11}	$2^{30}/fc$	or	$2^{23}/fs$	(256)
$fc/2^{22}$	fs / 2^{15}	fs / 2^{15}	$2^{34}/fc$	or	$2^{27}/fs$	(4096)

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 by program. Normally, a frequency sufficifient slower than the internal pulse ratesetting is applied to the external pin.

Figure 3-17. Pulse Width Measurement Mode Timing Chart

3.4   Watchdog Timer (WDT)

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

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

3.4.1   Configuration of Watchdog Timer

The WDT consists of 10 binary counters, a flip-flop, and a controller. Source input clock of binary counters is fc/2^7[Hz] or fs[Hz]. Table 3-4. shows watchdog timing detection time and operating mode (detection time of watchdog timer) .

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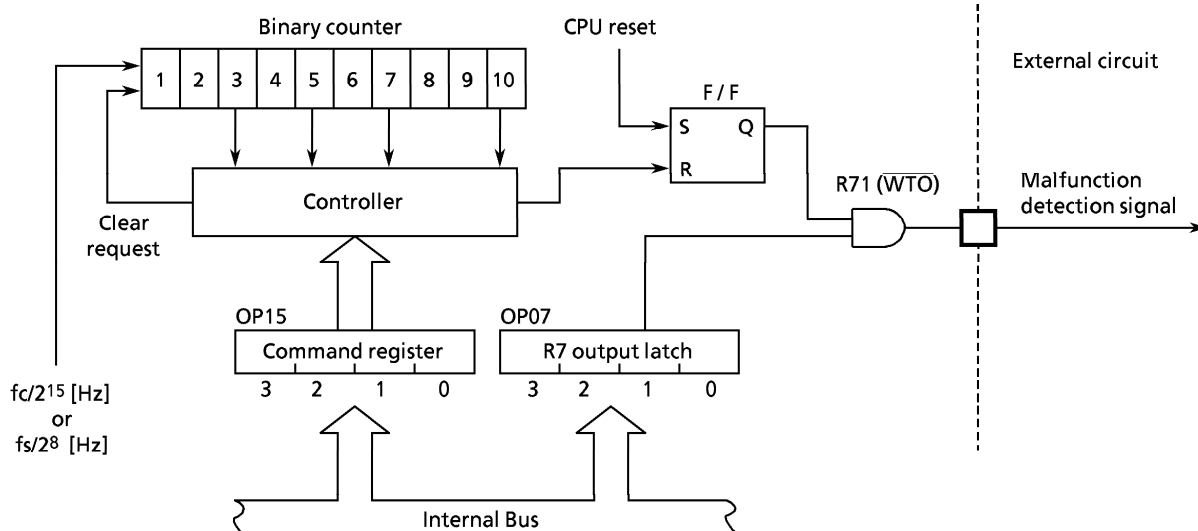

Figure 3-18.   Configuration of Watchdog Timer

3.4.2   Control of Watchdog Timer

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

To detect the CPU malfunction by the WDT:

(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 by program. Normally, a frequency sufficifient slower than the internal pulse ratesetting is applied to the external pin.
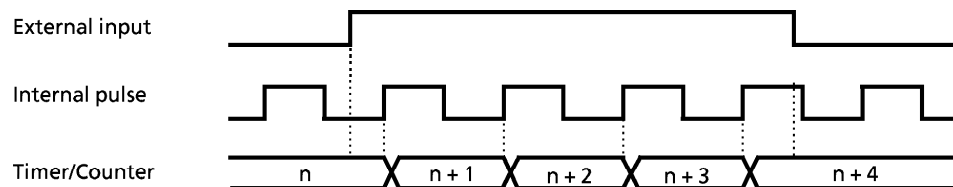

Figure 3-17.   Pulse Width Measurement Mode Timing Chart

3.4   Watchdog Timer (WDT)

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

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

3.4.1   Configuration of Watchdog Timer

The WDT consists of 10 binary counters, a flip-flop, and a controller. Source input clock of binary counters is $fc/2^7$[Hz] or fs[Hz]. Table 3-4. shows watchdog timing detection time and operating mode (detection time of watchdog timer) .

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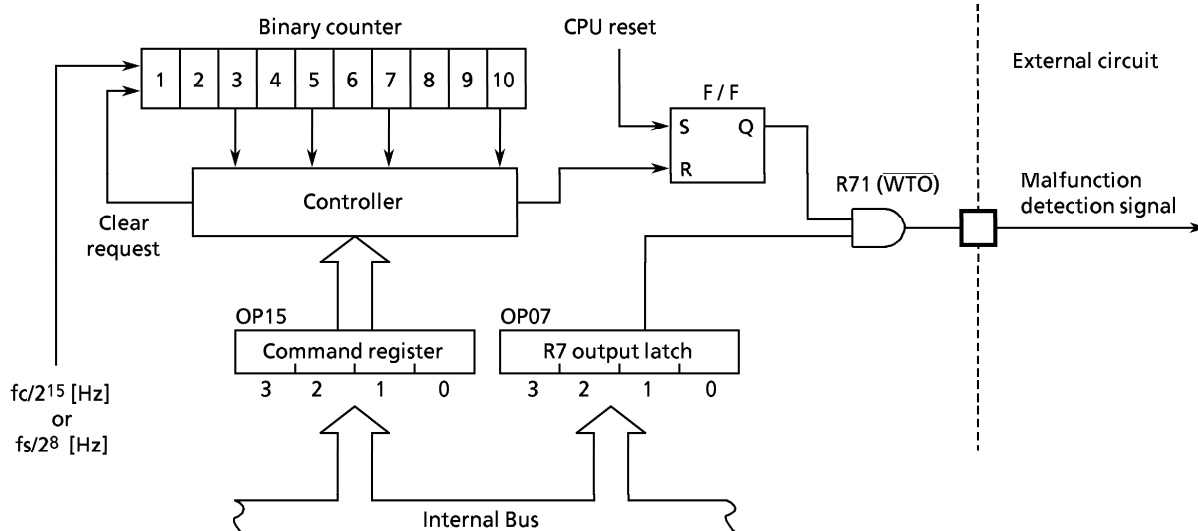

Figure 3-18.   Configuration of Watchdog Timer

3.4.2   Control of Watchdog Timer

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

To detect the CPU malfunction by the WDT:

(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 by program. Normally, a frequency sufficifient slower than the internal pulse ratesetting is applied to the external pin.
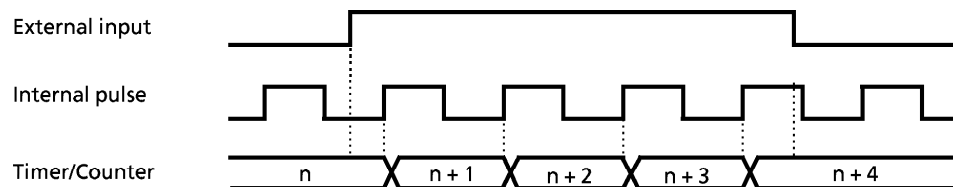

Figure 3-17. Pulse Width Measurement Mode Timing Chart

3.4   Watchdog Timer (WDT)

The watchdog timer capability is provided to quickly detect the CPU malfunction such as endless looping caused by noises or the like, and restore the CPU to the normal state.
The watchdog timer output appears on R71 ($\overline{\text{WTO}}$) pin as the malfunction detect signal. To use the WDT, the output latch of R71 must be set to "1" (during reset, it is set to "1"). Note that, the WDT is disabled during reset. Connecting the $\overline{\text{WTO}}$ pin and the $\overline{\text{RESET}}$ pin resets malfunction.

3.4.1   Configuration of Watchdog Timer

The WDT consists of 10 binary counters, a flip-flop, and a controller. Source input clock of binary counters is $fc/2^7$[Hz] or fs[Hz]. Table 3-4. shows watchdog timing detection time and operating mode (detection time of watchdog timer) .
The flip-flop is set to "1" during reset, and cleared to "0" on the rising edge of the binary counter output.
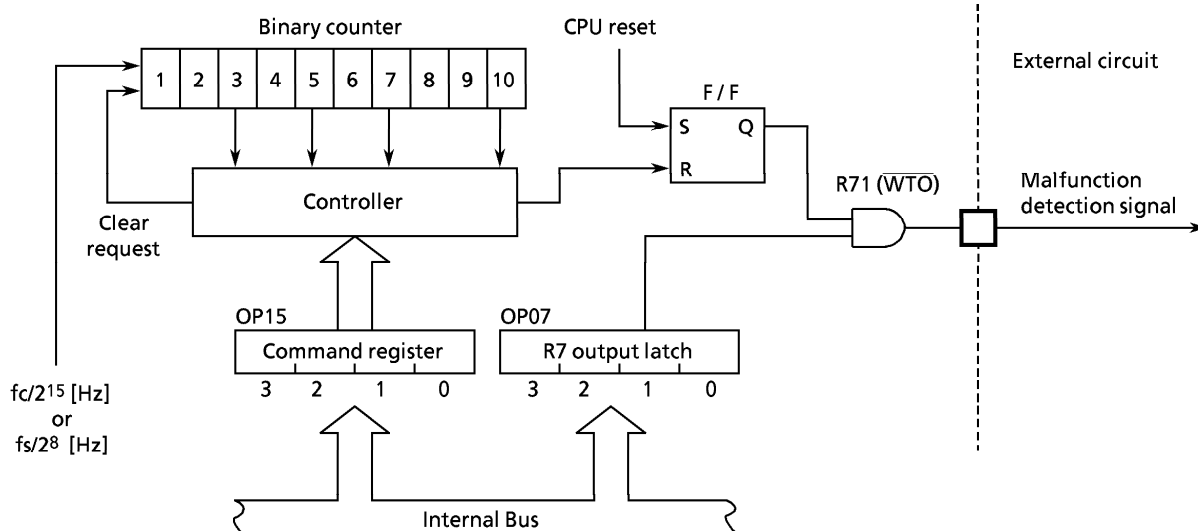

Figure 3-18.   Configuration of Watchdog Timer

3.4.2   Control of Watchdog Timer

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

To detect the CPU malfunction by the WDT:

① Set the WDT detection time, and clear the binary counters.
② Enable the WDT.
③ Clear the binary counters within WDT detection time that was set in ①. If a CPU malfunction occurs, preventing the binary counters from being cleared, the flip-flop is clearred to "0" on the rising edge of the binary counter output, making the malfunction detection signal active.
If the output latch of R71 is "1" at this time, the $\overline{WTO}$ output goes low.

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

	LD	A, #0010B	; OP15 ← 0010_B (Set WDT detection time, clear
	OUT	A, %OP15	binary counters)
	LD	A, #1110B	; OP15 ← 1110_B (Enable WDT)
	OUT	A, %OP15	
Within WDT detection timer
	LD	A, #0110B	; OP15 ← 0110_B (Clear binary counters)
	OUT	A, %OP15	

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

Watchdog timer control command register   (port address   OP15)

3	2	1	0
RWT	EWT	TWT	

(Initial value   1000)

RWT	Clears Binary counter

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

EWT	Watchdog timer enable/disable

0 : Disable
1 : Enable

TWT	Set Watchdog timer detection time

Example   At fc = 960kHz

00 :	$3 \times 2^{15}/fc$ [s]	⋯⋯	102 [ms]
01 :	$15 \times 2^{15}/fc$	⋯⋯	512
10 :	$63 \times 2^{15}/fc$	⋯⋯	2150
11 :	$511 \times 2^{15}/fc$	⋯⋯	17442

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

Figure3-19. Watchdog Timer Control Command Register

| TWT | Normal 1 | Normal 2 operation mode (Interval timer input) | | SLOW operation mode |
		SLCK = 0 (fc / 2^7)	SLOW mode fs [Hz]	
00	$3 \times 2^{15}/fc$ [s]		$3 \times 2^8/fs$ [s]	
01	$15 \times 2^{15}/fc$		$15 \times 2^8/fs$	
10	$63 \times 2^{15}/fc$		$63 \times 2^8/fs$	
11	$511 \times 2^{15}/fc$		$511 \times 2^8/fs$	

Table 3-4. Watchdog Timing detection time

3.5   DTMF Generator

The 47C853 has a built-in DTMF generator which generates dialing signals for tone dialing-type telephones. There are two groups of tone dial signals, one group of 4 sine wave low frequencies and another group of 4 sine wave high frequencies. All of these frequencies can be selected individually and combined with a frequency from the other group for a total of 16 different DTMF composite waves. (DTMF : Dual-tone Multifrequency)

3.5.1 Selection of Input Clock for DTMF Generator

Since the DTMF generator of the 47C853 is composed of a 480kHz clock as the base, it is necessary to supply the DTMF generator with fc when fc = 480kHz, and fc/2 when fc = 960kHz.
The input clock is selected by the DTMF input clock control register. The command register must always be set at the beginnig of the program.
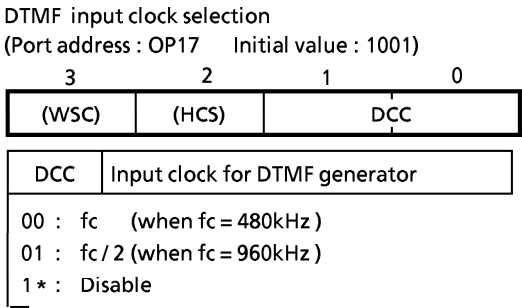
DTMF input clock selection
(Port address : OP17   Initial value : 1001)

3	2	1	0
(WSC)	(HCS)	DCC	

DCC	Input clock for DTMF generator
00 :	fc (when fc = 480kHz)
01 :	fc / 2 (when fc = 960kHz)
1 * :	Disable

Figure 3-20.   DTMF Input Clock Control Register

3.5.2 Configuration of DTMF Generator

Figure 3-21 shows the configuration of the DTMF generator.
The 47C853 generates two-stepped, quasi-sine waves, and outputs tone dial signals by combining the quasi-sine waves.
The high and low groups of frequencies are selected by setting the frequency selection codes in the ROW and COLUMN registers.
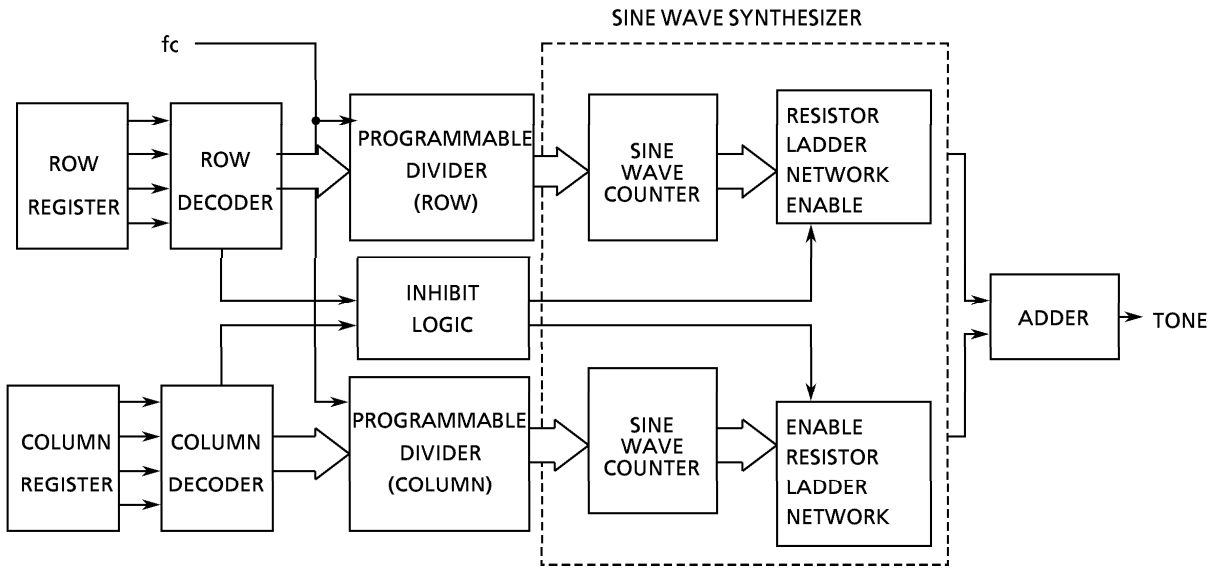

Figure 3-21.   Configuration of DTMF Generator

3.5.3 Control of DTMF Generator

Selection of frequency in tone output is controlled by the ROW register (OP01/IP01) and Column register (OP02/IP02). Enable /disable of single tone output is controlled by the TONE command register (OP0D).

Note 1.   *;   don't care
Note 2.   When reading STE bit, "1"
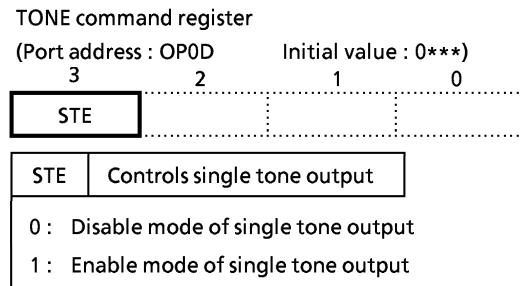   is always read.

TONE command register
(Port address : OP0D   Initial value : 0***)

3	2	1	0
STE			

STE	Controls single tone output

0 :   Disable mode of single tone output
1 :   Enable mode of single tone output

Figure 3-22. TONE Command Register

ROW register
(Port address : OP01 / IP01   Initial value : 0000)

3	2	1	0

ROW tone frequency selection

0001 :   Outputs 697.7Hz single tone
0010 :   Outputs 769.2Hz single tone
0100 :   Outputs 857.1Hz single tone
1000 :   Outputs 937.5Hz single tone

COLUMN register
(Port address : OP02 / IP02   Initial value : 0000)

3	2	1	0

COLUMN tone frequency selection

0001 :   Outputs 1212.1Hz single tone
0010 :   Outputs 1333.3Hz single tone
0100 :   Outputs 1481.5Hz single tone
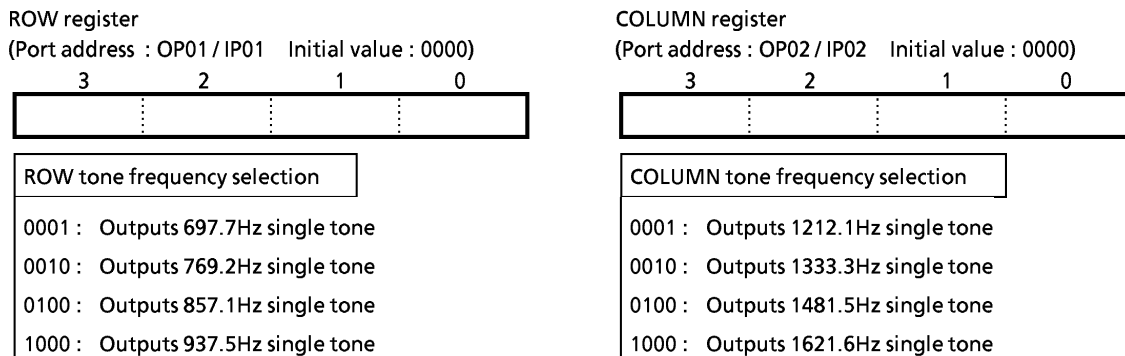1000 :   Outputs 1621.6Hz single tone

Figure 3-23. ROW, COLUMN Registers

Tones are outputted by entering the frequency selection codes shown in Figure 3-23 into the ROW and COLUMN registers.

When either the ROW or COLUMN register is set with an ineffective code when the single-tone output is in the enable mode, the register's tone output enters the disable mode and only registers in which an effective code has been set will output a tone. When both registers are enabled, dual tones can be output. When the single-tone output is in the disable mode, dual tones are output only when effective codes are loaded into both the ROW and COLUMN registers. No tones are output in cases other than the preceding ones.

The [OUTB @HL] instruction can simultaneously set 8-bit data into both registers (the upper 4 bits of the ROM data in the COLUMN register, and the lower 4 bits in the ROW register).

The operation and data table for the [OUTB @HL] instructions are the same as those of the 5-8 bit conversion instructions.

Example 1 : Output 1481.5Hz single tone (COLUMN)
            OUT   #8,%0P0D ;   Sets the enable mode of single tone output.
            OUT   #0,%0P01 ;   Sets an ineffective code into ROW register.
            OUT   #4,%0P02 ;   Sets data "4" into COLUMN register.

Example 2 : 8-bit data corresponding to the 5 bits of data which link the contents of the carry flag and the data memory RAM address 90_H, are read from the ROM, and frequency selection codes are loaded into the ROW and COLUMN registers.
            LD   HL,#90H   ;   HL←90_H (Sets the address of the data memory)
            OUTB   @HL   ;   Sets the ROM data into the ROW and COLUMN registers.

Table 3-5 shows the deviation between the 47C853 tone output and standard frequencies. Table 3-6 shows the corresponding frequency selection codes of the ROW and COLUMN registers for the telephone dial keys.

ROW Tone (Low Group)					
Frequency selection code 3 2 1 0				Tone output frequency [Hz]	Standard frequency [Hz]	Deviation [%]
0	0	0	1	697.7	697	+ 0.10
0	0	1	0	769.2	770	− 0.10
0	1	0	0	857.1	852	+ 0.60
1	0	0	0	937.5	941	− 0.37

COLUMN Tone (High Group)					
Frequency selection code 3 2 1 0				Tone output frequency [Hz]	Standard frequency [Hz]	Deviation [%]
0	0	0	1	1212.1	1209	+ 0.26
0	0	1	0	1333.3	1336	− 0.20
0	1	0	0	1481.5	1477	+ 0.30
1	0	0	0	1621.6	1633	− 0.70

Table 3-5. Tone output frequencies and Deviation from standard

	COLUMN register (OP02 / IP02)		
Frequency selection code	0001 (1209)	0010 (1336)	0100 (1477)
0001 (697)	1	2	3
0010 (770)	4	5	6
0100 (852)	7	8	9
1000 (941)	*	0	#

ROW register (OP01/IP01)

Standard telephone dial key

Contents of (   ) are standard frequencies, unit : Hz

Table 3-6. Corresponding frequency selection codes of the ROW and COLUMN registers for the telephone dial keys

3.5.4 Test Mode for Tone Output

The 47C853 includes a testing mode for checking tone output waveforms. Tones are output by the relevant circuit shown in Figure 3-24. Since ROW data are input from port R6 and COLUMN data are input from R3 by using the frequency selection codes shown in Figure 3-23, an optional single tone or dual tone can be output. Figure 3-25 shows a single-tone waveform and Figure 3-26 shows a dual-tone waveform.
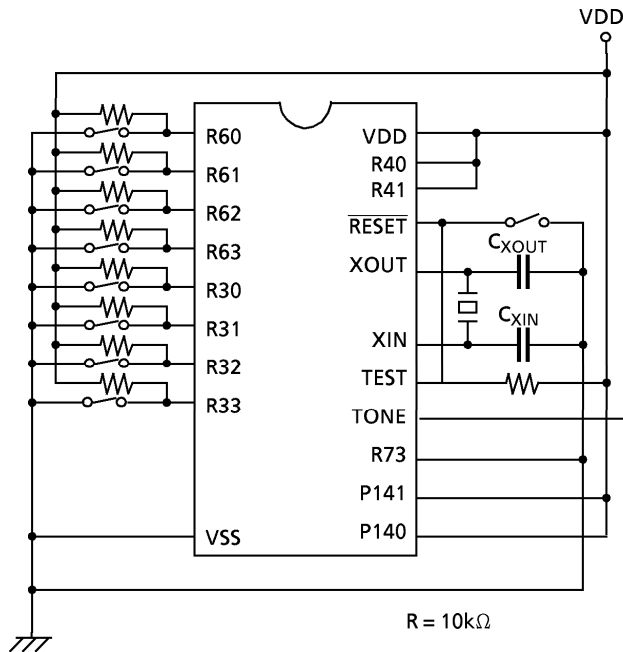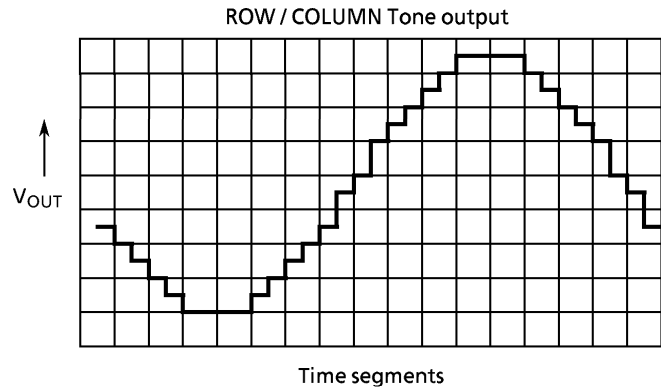

Figure 3-24. Tone test circuit

ROW / COLUMN Tone output
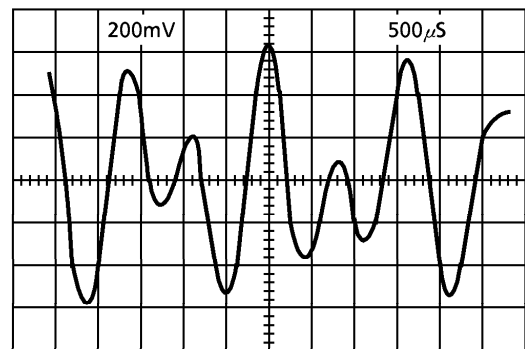
Figure 3-25. Single tone waveform

Figure 3-26. Dual tone waveform

3.6 BEEP Output Circuit

The BEEP output circuit generates square waves in the audible frequency range. This circuit drives the key input confirmation tone generator circuit for telephone applications. The BEEP output pin is shared by the P140 output. This pin controls both P140 and BEEP output.

3.6.1 Configuration of BEEP Output Circuit

Figure 3-27 shows the configuration of the BEEP output circuit. The clock pulse of the BEEP output circuit is supplied from a timing generator and the BEEP output is controlled by the frequency selection and output enable/disable setting.
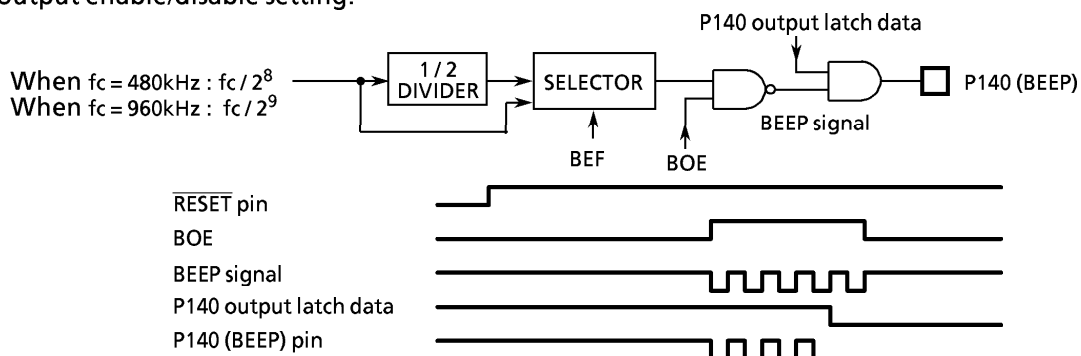
When $fc = 480kHz : fc / 2^8$
When $fc = 960kHz : fc / 2^9$

Figure 3-27. BEEP Output Circuit Configuration and Timing Chart

3.6.2 Control of BEEP Output Circuit

The BEEP output is controlled by the BEEP output control command register (OP13).
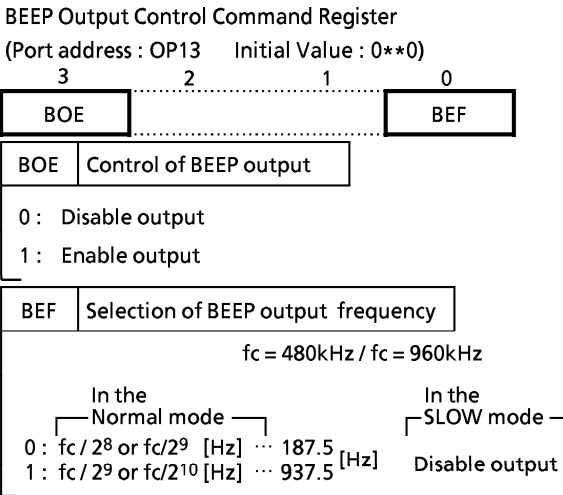
BEEP Output Control Command Register
(Port address : OP13   Initial Value : 0**0)

3	2	1	0
BOE			BEF

BOE	Control of BEEP output

0 : Disable output

1 : Enable output

BEF	Selection of BEEP output frequency

fc = 480kHz / fc = 960kHz

In the
Normal mode

In the
SLOW mode

0 : $fc / 2^8$ or $fc/2^9$ [Hz] ··· 187.5 [Hz]
1 : $fc / 2^9$ or $fc/2^{10}$ [Hz] ··· 937.5 [Hz]   Disable output

Figure 3-28. BEEP Output Control Command Register

The BEEP output is disable in the SLOW mode.

3.7 Warm Start Function

The reset operation is executed by adjusting the warm start control command register (Bit 3 of OP17) to "0".

Warm Start Control Command Register
(Port address : OP17   Initial value : 1001)

3	2	1	0
WSC	(HCS)	(DCC)	

WSC	Warm start control

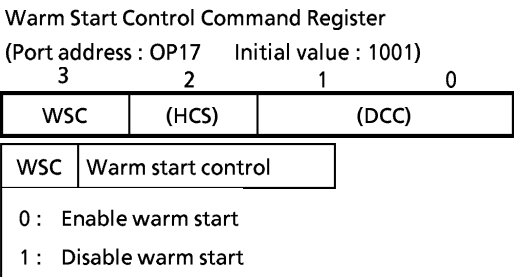0 : Enable warm start

1 : Disable warm start

Figure 3-29. Warm Start Control Command Register

By setting WSC to "0" , "L" level signal is output to a reset terminal and reset operation is executed.

3.8 Serial Interface (SIO)

The 47C853 have a serial interface with an 8-bit buffer. 4-bit/8-bit tramsfer mode can be selected. In the 8-bit transfer mode, data may be transmitted and received simultaneously. The serial interface is connected to the exterenal device via 3 pins (the serial port): R92 ($\overline{SCK}$), R91 (SO), and R90 (SI). The serial port is shared by port R9. For the serial port, the output latch of port R9 must be set to "1". In the transmit mode, R90 pin provides the I/O port; in the receive mode, R91 pin provides the I/O port.

3.8.1 Configuration of Serial Interface

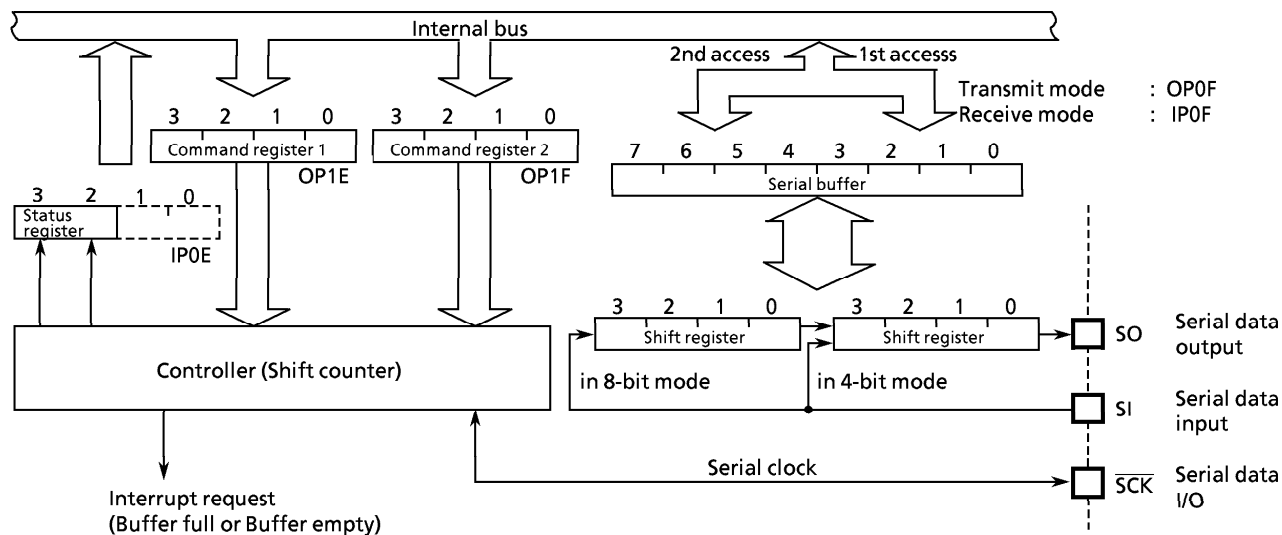Figure 3-30. shows configuration of serial interface.

Figure 3-30.   Configuration of Serial Interface

3.8.2 Control of Serial Interface

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

Serial interface status register   (Port address   IP0E).

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

SIOF	Monitor serial transfer operation state

0 : Transfer is terminated
1 : Transfer is in progress

SEF	Monitors shift operation status

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

Figure 3-31.   Serial Interface Status Register

3.8 Serial Interface (SIO)

The 47C853 have a serial interface with an 8-bit buffer. 4-bit/8-bit tramsfer mode can be selected. In the 8-bit transfer mode, data may be transmitted and received simultaneously. The serial interface is connected to the exterenal device via 3 pins (the serial port): R92 ($\overline{SCK}$), R91 (SO), and R90 (SI). The serial port is shared by port R9. For the serial port, the output latch of port R9 must be set to "1". In the transmit mode, R90 pin provides the I/O port; in the receive mode, R91 pin provides the I/O port.

3.8.1 Configuration of Serial Interface

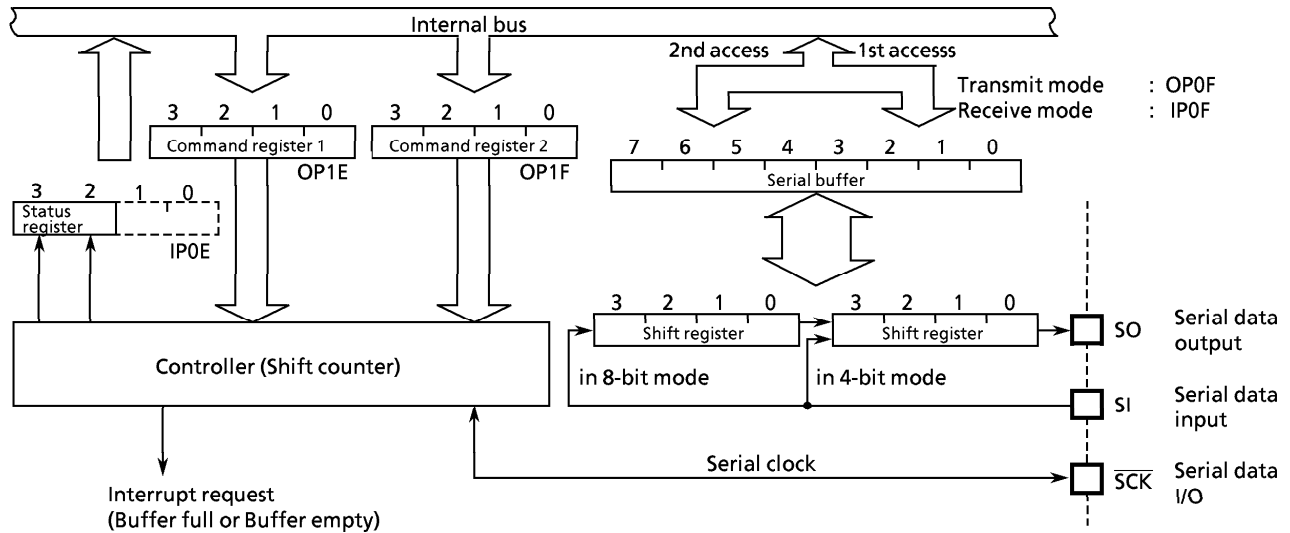Figure 3-30. shows configuration of serial interface.

Figure 3-30. Configuration of Serial Interface

3.8.2 Control of Serial Interface

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

Serial interface status register (Port address IP0E).

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

SIOF	Monitor serial transfer operation state

0 : Transfer is terminated
1 : Transfer is in progress

SEF	Monitors shift operation status

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

Figure 3-31. Serial Interface Status Register

Serial interface control command register1 (port address OP1E)

3	2	1	0
INH	SBIT	CKR	

(Initial value 0001)

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

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

CKR	Select serial clock frequency

	Single clock mode	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]

Serial interface control command register 2 (port address OP1F)

3	2	1	0
ESIO	RM	LM	ECKM

(Initial value 0000)

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

RM	Select transfer mode

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

LM	Select shift edge
0	: Shift at the trailing edge of serial clock
1	: Shift at the leading edge of serial clock

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

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

Figure 3-32. Serial Interface Control Command Register

(1) Serial clock
For the serial clock, one of the following can be selected according to the contents of the command registers:
   a. Clock source selection
      ① Internal clock
         The serial clock frequency is selected by command register.
         The serial clock is output on the $\overline{SCK}$ pin. Note that the start of transfer, the $\overline{SCK}$ pin output goes high. This device provides the wait function in which the shift is not occurred until these processings are completed.
         The highest transfer rate based on the internal clock is 15000 bits/s (at fc = 960kHz).
      ② External clock
         The signal obtained by the clock supplied to the $\overline{SCK}$ pin from the outside is used for the serial clock. In this case, the output latch of R92 ($\overline{SCK}$) must be set to "1" beforehand. For the shift operation to be performed correctly, each of the serial clock high and low levels needs 2 instruction cycles or more to be completed.
   b. Shift edge selection
      ① Leading edge
         Date is shifted at the leading edge (the falling edge of $\overline{SCK}$ pin input) of the serial clock.
      ② Trailing edge
      Data is shifted at the trailing edge (the rising edge of $\overline{SCK}$ pin input) of the serial clock. However, in the transmit mode, the trailing-edge shift is not supported.

(2) Transfer bit number
   SBIT (bit 2 of the command register 1) can select 4-bit/8-bit serial transfer.
      a. 4-bit serial transfer
         In this mode, transmission/reception is performed on 4-bit basis. ISIO interrupt is generated every 4-bit transfer. Transmit/receive data is written/read by accessing the buffer register (OP0F/IP0F) respectively.
      b. 8-bit serial transfer
         In this mode, transmission/reception is performed on 8-bit basis. ISIO interrupt is generated every 8-bit transfer. Transmit /receive data is written / read by accessing the buffer register (OPP0F / 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 register2).

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

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

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

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

(a) Internal-clock-based operation with wait
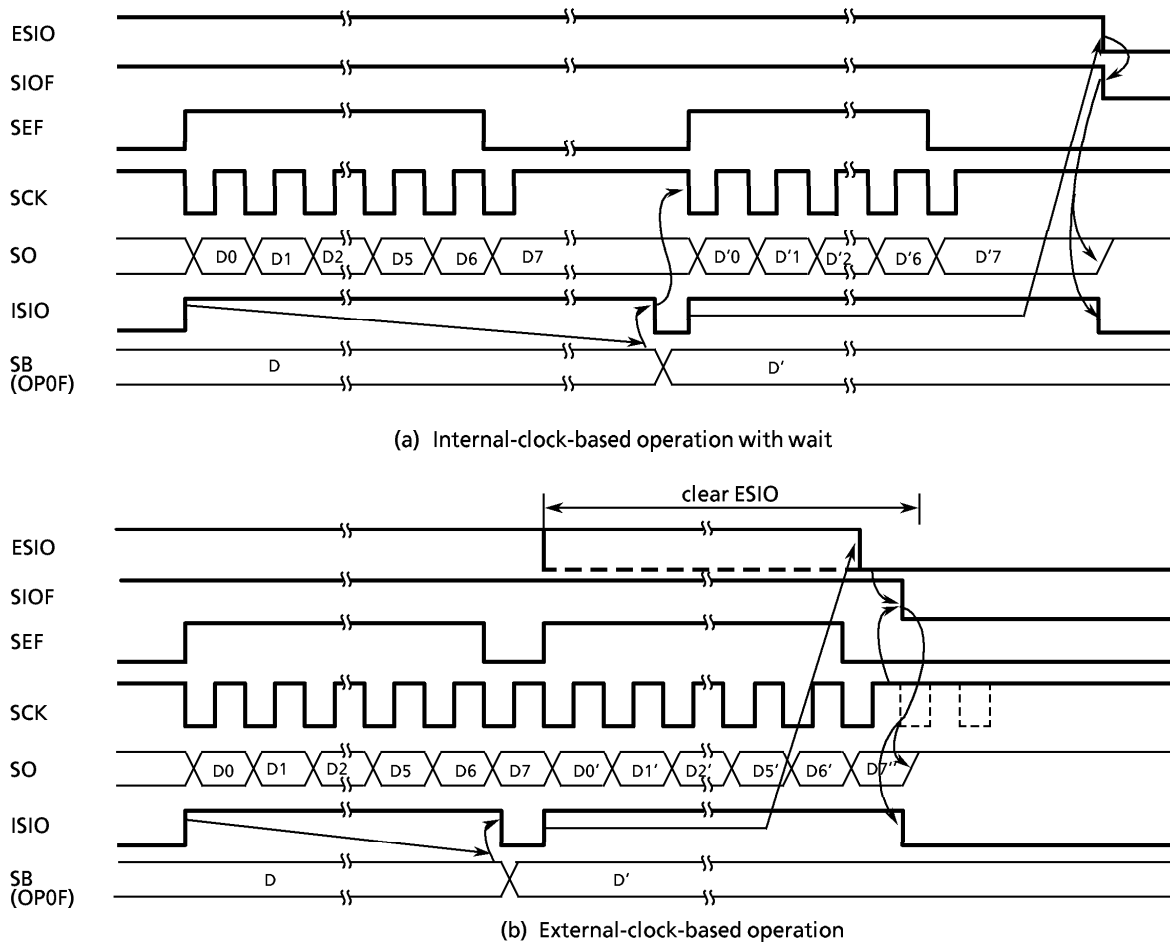
(b) External-clock-based operation

Figure 3-33. Transmit Mode

b. Receive mode

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

● 4-bit receive mode

Data can be received when ESIO is set to "1" after setting the receive mode to the command register. The data is put from the SI pin to the shift register in synchronization with the serial clock. Then the 4/8-bit data is transferred from the shift register to the buffer register (IP0F),upon which the (buffer full)interrupt (ISIO) to reguest for readingreceived data is generated. The receive data is read from the buffer register by the interrupt service program. When the data has been read, the interrupt request is reset and the next data is put in the shift register to be transferred to the buffer register. In the operation based on the internal clock, if the previous receive data has not been read from the buffer register at the end of capturing the next data, the serial clock is stopped and the wait operation is performed until the data has been read. In the operation based on the external clock, the shift operation is performed in synchronization with the externally-suppliedclock, so that the data must be read from the buffer register before the next receive data is transferred to it. The maximum transfer rate in the external-clock-based operation is determined by the maximum delay time between the generation of interrupt request and the reading of receive data. In the receive mode, the shift operation may be performed at either the leading edge or the trasiling edge. In the leadingedge shift operation,data is captured at the leading edge of the serial clock, so that the first shift data must be put in the SI pin before the first serial clock is applied at the start of transfer.

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

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

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

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

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

① When the transfer rate is sufficiently low (the external-clock-based operation):
If ESIO can be cleared to "0" before the next serial clock is applied upon occurrence of buffer full interrupt in the external-clock-based operation, ESIO is cleared to "0" by the interrupt service program, then the last received data is read.

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

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

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

③ One-word reception

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

Example: To instruct the start/end of 1-word reception (the internal clock, the trailing edge shift).
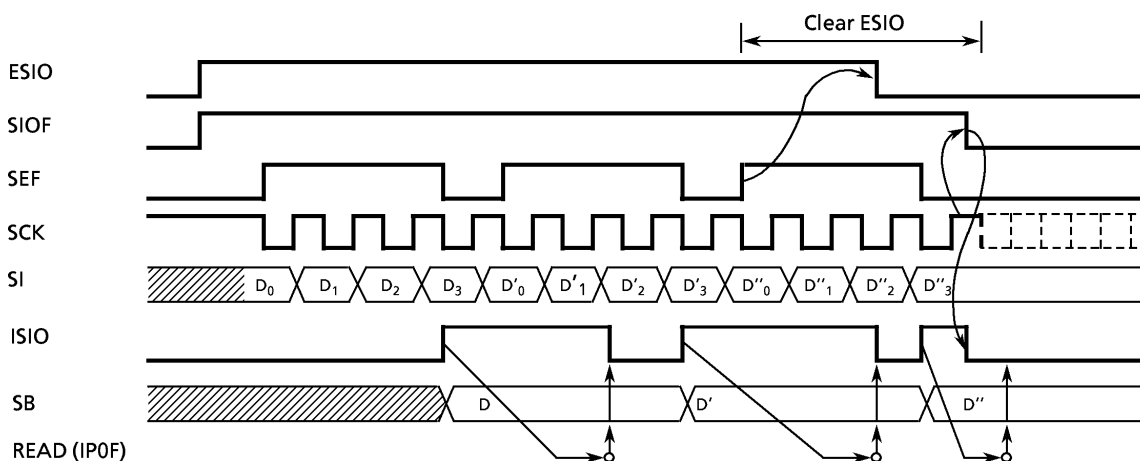
```
          LD     A,   #0100B     ; OP1F ←0100B (Sets in the receive mode)
          OUT    A,   %OP1F
          EI                     ; EIF ← 1 (Enables interrupt)
          LD     A,   #1110B     ; ESIO ← 1 (Instructs reception start)5
          OUT    A,   %OP1F
  SSEF0 : TEST   %IP0E,  2       ; Confirms that SEF = "1"
          B      SSEF0
          LD     A,   #0110B     ; ESIO ← 0 (Instructs reception end)
          OUT    A,   %OP1F
```
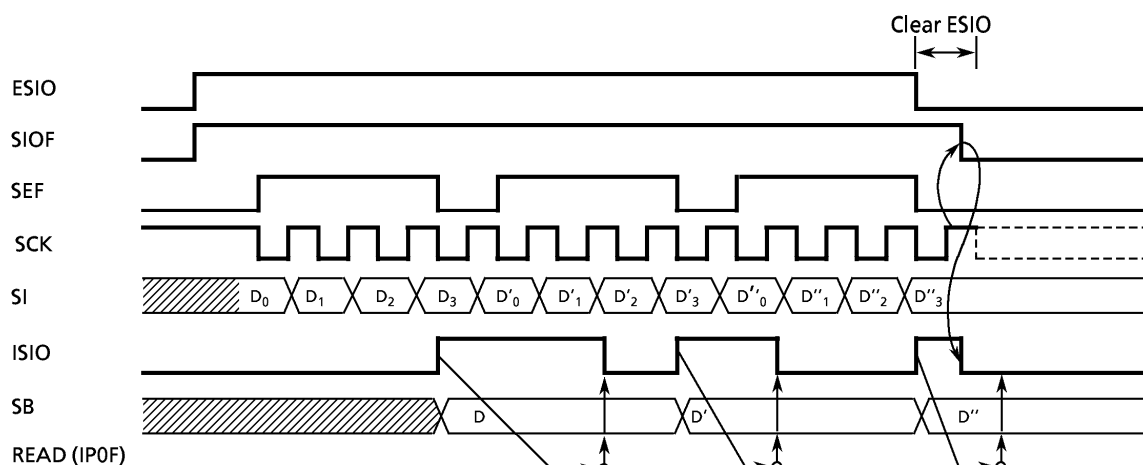

(a)　External-clock-based operation, leading-edge shift (when transfer rate is low)

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

Figure3-34.　4-bit Receive Mode

● 8-bit Transmit/Receive Mode

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

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

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

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

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

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

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

```
        LD      A, #0110B       ; Sets the 8-bit transfer and serial clock frequency.
        OUT     A, %0P1E
        LD      A, #0110B       ; Sets the transmit/receive mode of internal clock
                                  operation
        OUT     A, %0P1F
        LD      HL, #20H        ; OP0F←RAM[20H] (Writes lower 4-bit data to be
                                  transmitted)
        OUT     @HL, %0P0F
        INC     L               ; OP0F←RAM[21H] (Writes upper 4-bit data to be
                                  transmitted)
        OUT     @HL, %0P0F
        LD      A, #1110B       ; ESIO ← 1 (Instructs serial transfer start)
        OUT     A, %0P1F
        ⋮                       ; Data transfer
```

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

```
        LD      HL, #30H        ; Stores lower 4-bit data received in RAM[30H].
        IN      %IP0F, @HL
        INC     L               ; Stores upper 4-bit received in RAM[31H].
        IN      %IP0F, @HL
        LD      HL, #22H        ; Writes next lower 4-bit data to be transmitted.
        OUT     @HL, %0P0F
        INC     L               ; Writes next upper 4-bit data to be transmitted.
        OUT     @HL, %0P0F
```

(4) Stopping serial transfer

A serial transfer operation can be stopped forcibly.

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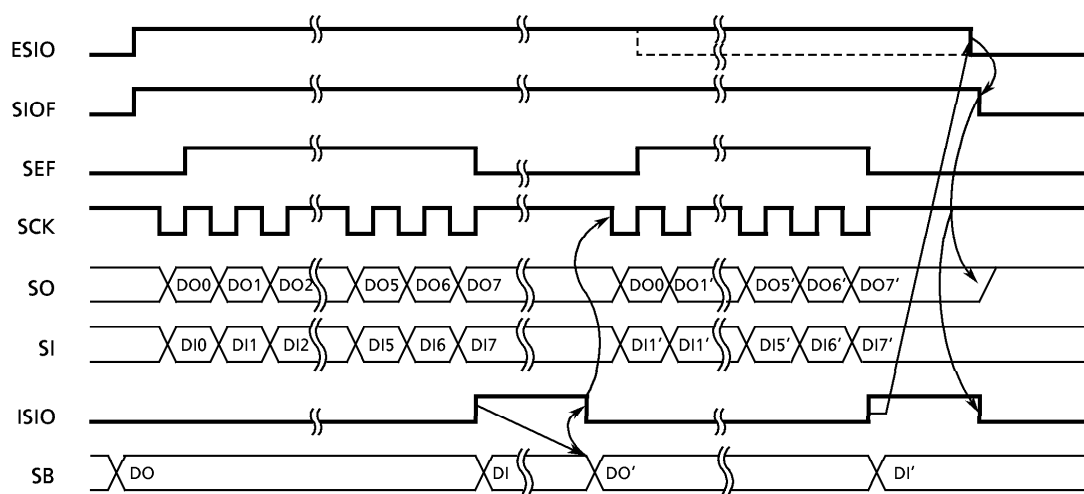
(a) Internal clock based operation with Iwait
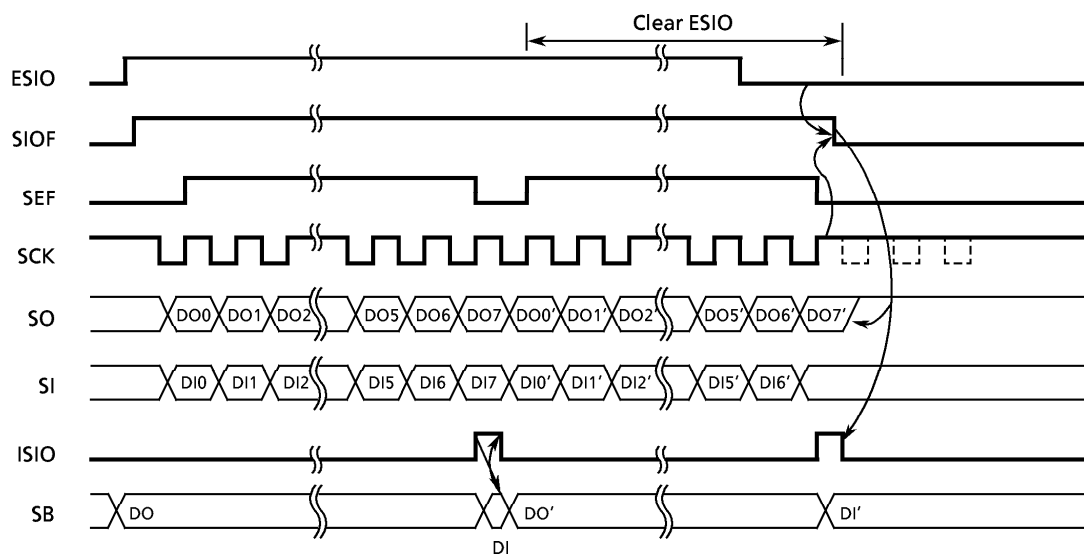
(b) External clock based operation

Figure3-35. 8-bit Transmit/Receive Mode

Port Condition by RESET Operation

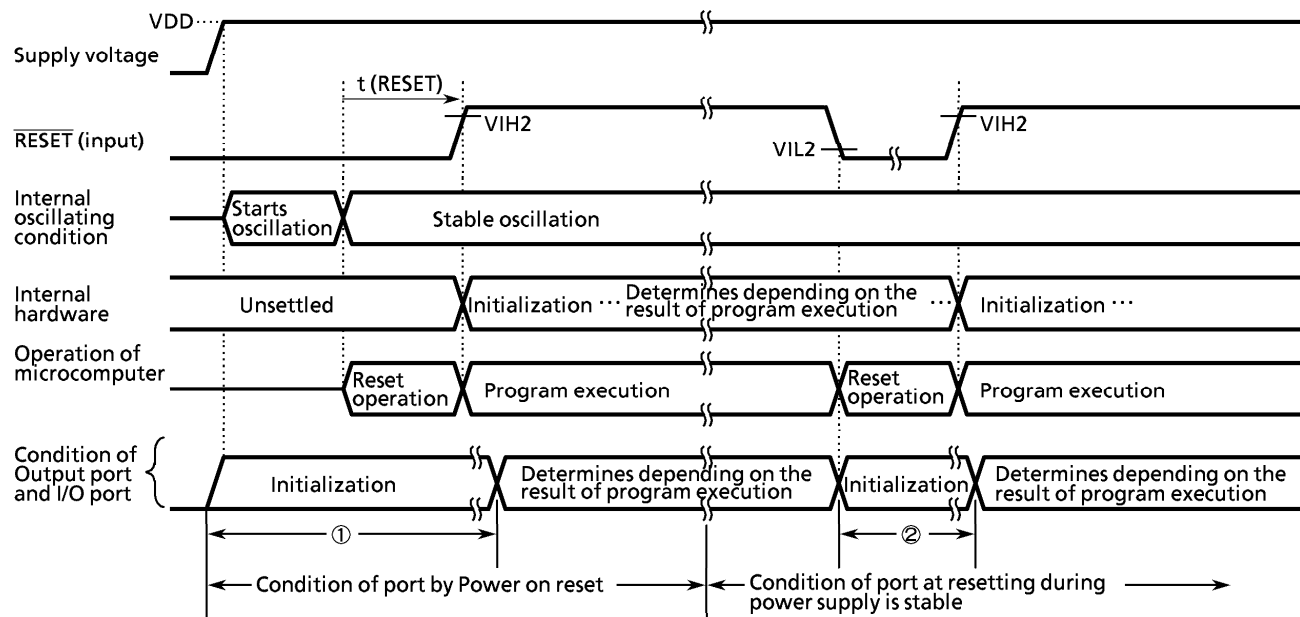The transition of Port condition by RESET operation is shown as below.

Figure 3-36.   Port condition by Reset operation

Note 1:   *t (RESET) > 24/fc*
Note 2:   *VIL2 : Stands for low level input voltage of RESET pin.*
            VIH2 : Stands for high level input voltage of RESET pin.
Note 3:   *The term from power on reset to the time program is executed (above ①) and also the term*
            starting from reset operation during power supply is stable to the program is executed (above
            ②), the port is on the initial condition. The initial condition of Port differs from I/O circuit by
            each port, refer to the section of "INPUT/OUTPUT CIRCUITRY". Thus, when using Port as an
            output pin, in the term of the above ① and ②, the voltage level on the signal that connects
            with the output pin of Port to the input pin of external application circuit should be
            determined by the external circuitry such as pull-up resistor and / or pull-down resistor.
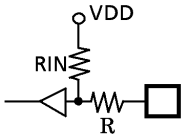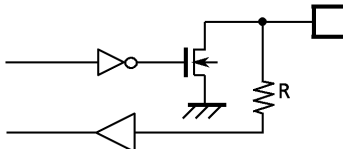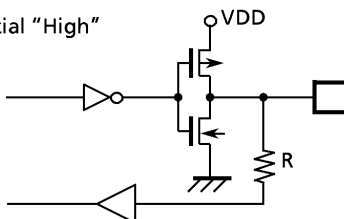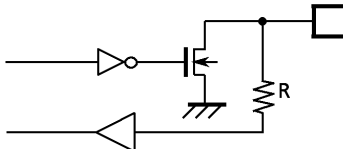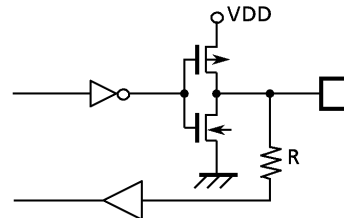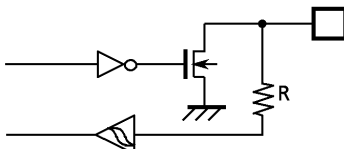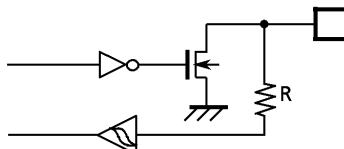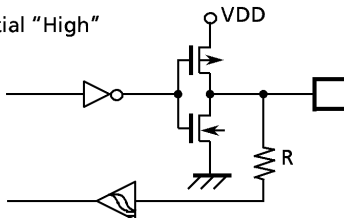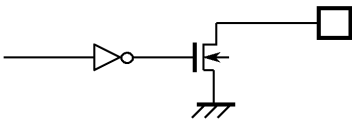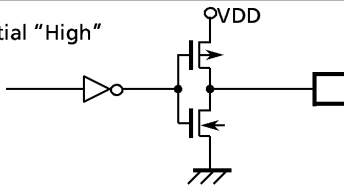
INPUT/OUTPUT CIRCUITRY

(1)   Control pins
      The input/output circuitries of the 47C853 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_O = 2kΩ (typ.)
XTIN XTOUT	Input Output		Resonator connecting pins (Low frequency) R = 1kΩ (typ.) R_f = 6MΩ (typ.) R_O = 220kΩ (typ.)
$\overline{\text{RESET}}$	Input		Hysteresis input Pull-up resistor R_{IN} = 220kΩ (typ.) R = 1kΩ (typ.)
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input (Input)		Hysteresis 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 47C853 I/O ports are shown in the following chart, any of which can be chosen by a code (WB, WE, or WH) as a mask option.

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE	REMARKS
K0	Input		Pull-up resistor RIN = 70kΩ (typ.) R = 1kΩ (typ.)
R3 R4 R5 R6	I/O	WB — Initial "Hi-Z" / WE, WH — Initial "High" 	Sink open drain or push-pull output R = 1kΩ (typ.)
R7	I/O	WB, WE — Initial "Hi-Z" / WH 	Sink open drain output Initial "Hi-Z" R = 1kΩ (typ.)
R8	I/O	Initial "Hi-Z" 	Sink open drain Initial "Hi-Z" Hysteresis input R = 1kΩ (typ.)
R9	I/O	WB, WE — Initial "Hi-Z" / WH — Initial "High" 	Sink open drain or push-pull output Hysteresis input R = 1kΩ (typ.)
P14	Output	WB — Initial "Hi-Z" / WE, WH — Initial "High" 	Sink open drain or push-pull output

ELECTRICAL CHARACTERISTICS

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

PARAMETER	SYMBOL	PINS	RATINGS	UNIIT
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 sink open drain pin	-0.3 to $V_{DD} + 0.3$	V
	V_{OUT2}	Sink open drain pin Except R7 Port	-0.3 to 10	
Output Current (per 1 pin)	I_{OUT}		3.2	mA
Power Dissipation [$T_{opr} = 60°C$]	PD		600	mW
Soldering Temperature (time)	T_{sld}		260 (10s)	°C
Storage Temperature	T_{stg}		-55 to 125	°C
Operating Temperature	T_{opr}		-30 to 60	°C

RECOMMENDED OPERATING CONDITIONS ($V_{SS} = 0V$, $T_{opr} = -30$ to 60°C)

PARAMETER	SYMBOL	PINS	CONDITIONS	Min.	Max.	UNIT
Supply Voltage	VDD		Normal mode	2.2	6.0	V
			SLOW mode	2.7		
			HOLD mode	2.0		
Input High Voltage	V_{IH1}	Except hysteresis input	$V_{DD} \geqq 4.5V$	$V_{DD} \times 0.7$	V_{DD}	V
	V_{IH2}	Hysteresis input		$V_{DD} \times 0.75$		
	V_{IH3}		$V_{DD} < 4.5V$	$V_{DD} \times 0.9$		
Input Low Voltage	V_{IL1}	Except hysteresis input	$V_{DD} \geqq 4.5V$	0	$V_{DD} \times 0.3$	V
	V_{IL2}	Hysteresis input			$V_{DD} \times 0.25$	
	V_{IL3}		$V_{DD} < 4.5V$		$V_{DD} \times 0.1$	
Clock Frequency (High)	fc	XIN, XOUT		960/480		kHz
Clock Frequency (Low)	fs	XIN, XOUT		30.0	34.0	kHz

D.C. CHARACTERISTICS ($V_{SS} = 0V$, $V_{DD} = 2.2$ to $6.0V$, $T_{opr} = -30$ to $60°C$)

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}	Port R (open drain)					
Input Low Current	I_{IL}	Port R (push-pull)	$V_{DD} = 5.5V$, $V_{IN} = 0.4V$	—	—	− 2	mA
Input Resistance	R_{IN1}	Port K0		30	70	150	$k\Omega$
	R_{IN2}	$\overline{RESET}$		100	220	450	
Output Leakage Current	I_{LO}	Ports P, R (open drain)	$V_{DD} = 5.5V$, $V_{OUT} = 5.5V$	—	—	2	μA
Output High Voltage	V_{OH}	Port R (push-pull)	$V_{DD} = 4.5V$, $I_{OH} = -200\mu A$	2.4	—	—	V
Output Low Voltage	V_{OL2}	Except XOUT	$V_{DD} = 4.5V$, $I_{OL} = 1.6mA$	—	—	0.4	V
Supply Current (in the Normal mode)	I_{DD}		Except TONE generating $V_{DD} = 2.2V$ $fc = 960kHz$	—	0.3	0.5	mA
	I_{DDT}		TONE generating $V_{DD} = 2.2V$ $fc = 960kHz$	—	0.6	1.2	
Supply Current (in the SLOW mode)	I_{DDS}		$V_{DD} = 3.0V$, $fs = 32.768kHz$	—	30	60	μA
Supply Current (in the HOLD mode)	I_{DDH}		$V_{DD} = 5.5V$	—	0.5	10	μA

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

Note 2. Input Current I_{IN1} : The current through resistor is not included, when containing the pull-up/pull-down resistor.

Note 3. Supply Current : $V_{IN} = 2.0V / 0.2V$
The K0 port is opened when containing the pull-up/pull-down resistor.
The Voltage applied to the R port is within the valid range V_{IL} or V_{IH}.

TONE OUTPUT CHARACTERISTICS ($V_{SS} = 0V$, $V_{DD} = 2.2$ to $6.0V$, $T_{opr} = -30$ to $60°C$)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Tone Output Voltage (ROW)	V_{TONE}	$RL \geqq 10k\Omega$, $V_{DD} = 2.2V$	125	185	250	mVrms
Pre-Emphasis High Band (COL / ROW)	PEHB	PEHB = 20log (COL /ROW)	1	2	3	dB
Output Distortion	DIS		—	—	10	%
Frequency Stability	f	Except error of osc. frequency	—	—	0.7	%

| | | A.C. CHARACTERISTICS | | (V_{SS} = 0V, V_{DD} = 2.2 to 6.0V, T_{opr} = − 30 to 60°C) | | | |

| PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Instruction Cycle Time	t_{cy}		8.3 / 16.7			μs
Shift Data Hold Time	t_{SDH}		0.5t_{cy}-300	−	−	ns

Note. Shift Data Hold Time :

External circuit for $\overline{SCK}$ pin and SO pin   *Serial port (completion of transmisson)*
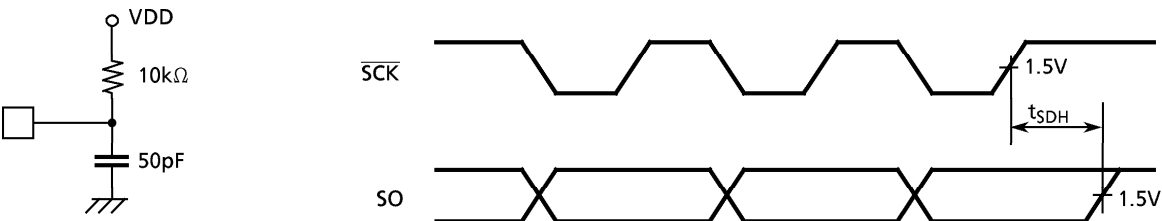
| | | RECOMMENDED OSCILLATING CONDITIONS | | (V_{SS} = 0V, V_{DD} = 2.2 to 6.0V, T_{opr} = − 30 to 60°C) |

(1) 960kHz
      Ceramic Resonator
            CSB960J916 (MURATA) $C_{XIN} = C_{XOUT}$ = 100pF
            KBR-960F6 (KYOCERA) $C_{XIN} = C_{XOUT}$ = 30pF
            KBR-960F7 (KYOCERA) $C_{XIN} = C_{XOUT}$ = 100pF

(2) 480kHz
      Ceramic Resonator
            CSB480E16 (MURATA) $C_{XIN} = C_{XOUT}$ = 100pF

(3) 32.768kHz
      Crystal Oscillator C_{XTIN}, C_{XTOUT} ; 10 to 33pF
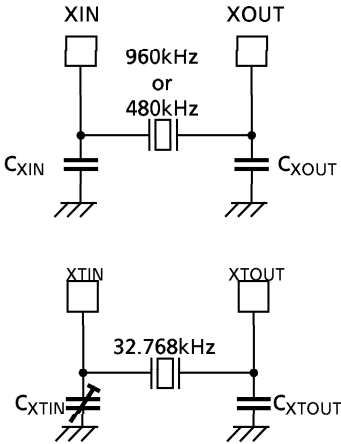
Note : In order to get the accurate oscillation frequency, the adjustment of capacitors must be required.

TYPICAL CHARACTERISTICS

R – Ta K0 port
$V_{DD} = 5.5V$

R – Ta $\overline{RESET}$ pin
$V_{DD} = 5.5V$

$I_{OH} - V_{OH}$ CMOS R port
$V_{DD} = 4.5V$
$Ta = 25°C$

$I_{IL} - V_{IN}$ CMOS R port
$V_{DD} = 5.5V$
$Ta = 25°C$

$I_{OL} - V_{OL}$ R port
$V_{DD} = 4.5V$
$Ta = 25°C$

TONE $V_{PP} - V_{DD}$
$Ta = 25°C$

$I_{DD} - V_{DD}$
$Ta = 25°C$

$I_{DDT} - V_{DD}$
$Ta = 25°C$