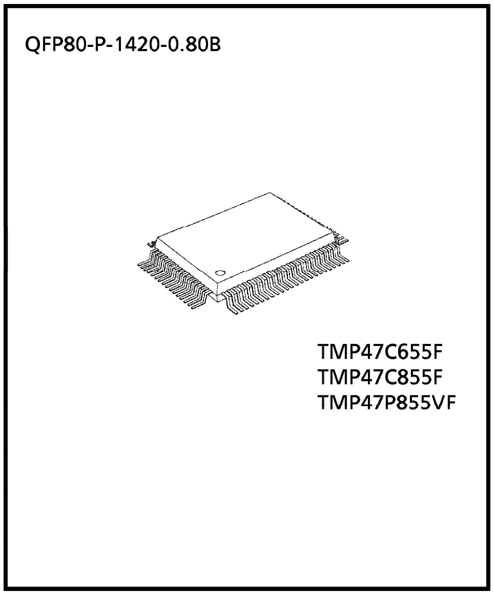
CMOS 4-BIT MICROCONTROLLER

TMP47C655F, TMP47C855F

The 47C655/855 are a high speed and high performance 4-bit single chip microcomputer based on the TLCS-470 CMOS series. The 47C655/855 have LCD driver, DTMF generator and large-capacity RAM for repertory dial, which are suitable for application in telephones. The 47C655/855 have two oscillation circuits. It is possible to switch the operating mode; high speed operation and low power consumption operation.

PART No.	ROM	RAM	PACKAGE	OTP	PIGGYBACK
TMP47C655F	6144 × 8-bit	896 × 4-bit	QFP80-P-1420-0.80B	TMP47P855VF	TMP47C055G
TMP47C855F	8192 × 8-bit	1024 × 4-bit			

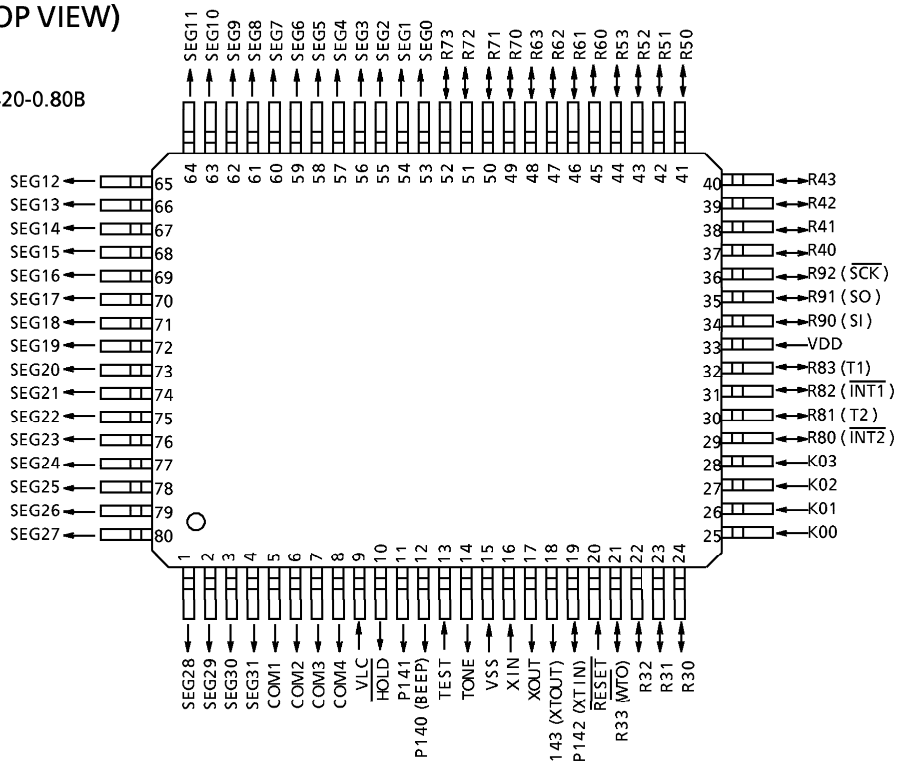
FEATURES



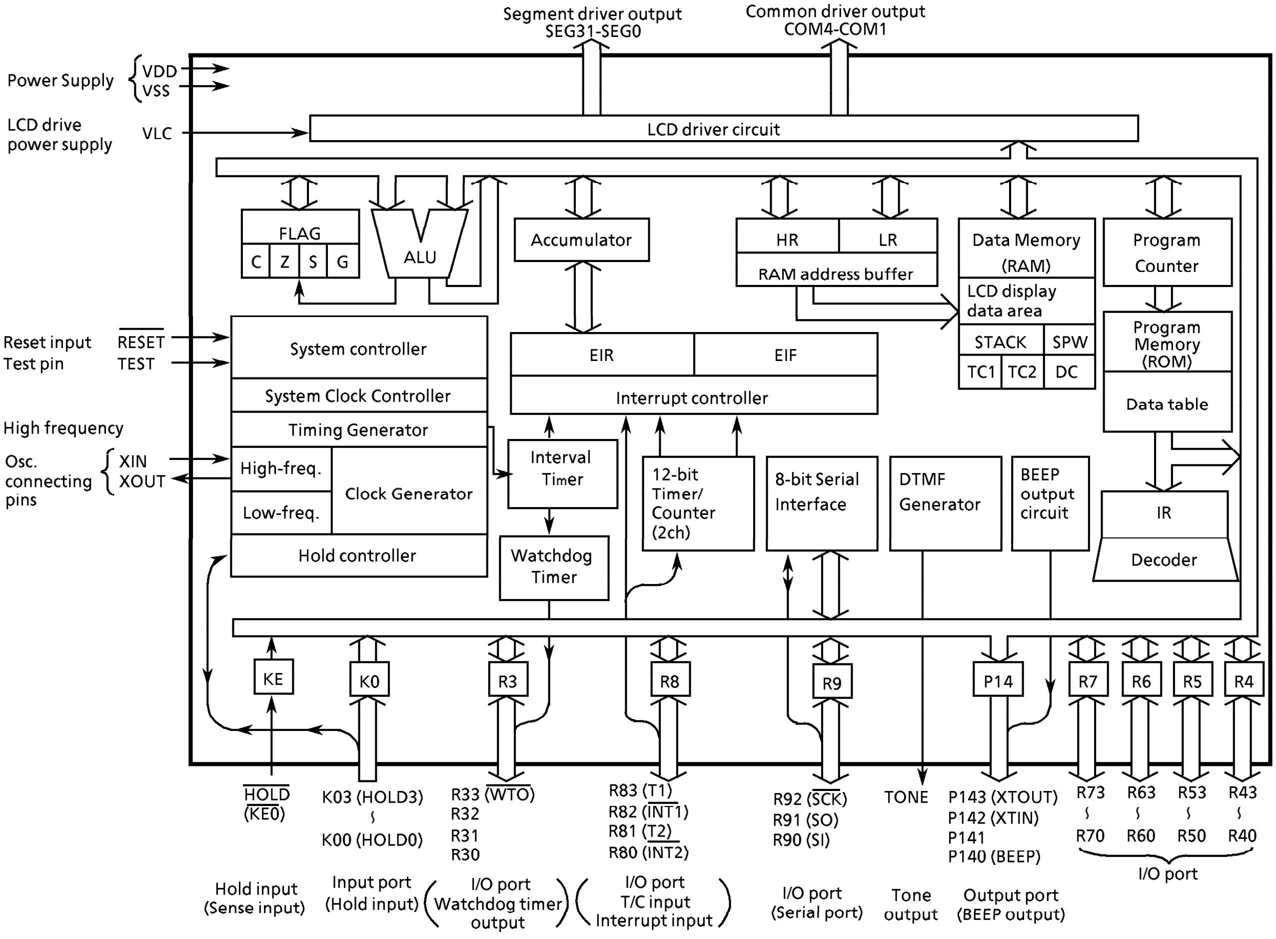
- ◆4-bit single chip microcomputer
- ◆Instruction execution time :
  8.3μs (at 960kHz), 244μs (at 32.8kHz)
- ◆Low voltage operation : 2.2V min.
- ◆92 basic instructions
- ◆Table look-up instructions
- ◆Subroutine nesting : 15 levels max.
- ◆6 interrupt sources (External :2, Internal : 4)
  All sources have independent latches each, and multiple interrupt control is available.
- ◆I/O port (36 pins)
  - Input 2 ports 5 pins
  - I/O 7 ports 27 pins
  - Output 1 port 4 pins
- ◆Interval Timer (15 stage)
- ◆Two 12-bit Timer/Counters
  Timer, event counter, and pulse width measurement mode
- ◆Watchdog Timer
- ◆Serial Interface with 8-bit buffer
  - Simultaneous transmission and reception capability
  - External/internal clock, and leading/trailing edge shift mode
- ◆LCD driver (automatic display)
  - LCD direct drive (Max. 16-digit display at 1/4 duty LCD)
  - 1/4, 1/3, 1/2 duties or static drive are programmably selectable.
- ◆DTMF (Dual Tone Multi Frequency) output
  - DTMF output with one instruction
  - Single tone output function
- ◆RAM for repartry dial : 1024 × 4-bit max.
- ◆BEEP output function
- ◆Dual-clock operation
  High-speed/Low-power-consumption operating mode
- ◆Hold function
  - Battery/Capacitor back-up
  - Hold function controlled by K0 port
- ◆Real-time Emulator : BM47C855

PIN ASSIGNMENT (TOP VIEW)

QFP80-P-1420-0.80B



BLOCK DIAGRAM



PIN FUNCTION

PIN NAME	Input/Output	FUNCTIONS	
K03 - K00	Input	4-bit input port	
R33 ($\overline{\text{WTO}}$)	I/O (Output)	4-bit I/O port with latch. When used as input port, the latch must be set to "1".	Watchdog timer output
R32 - R30	I/O		
R43 - R40	I/O	4-bit I/O port with latch. When used as input port, the latch must be set to one.	
R53 - R50			
R63 - R60			
R73 - R70			
R83 (T1)	I/O (Input)	4-bit I/O port with latch. When used as 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 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
P141	Output	2-bit I/O port with latch.	
P140 (BEEP)	Output (Output)		BEEP output
SEG31 - SEG0	Output	LCD Segment driver output	
COM4 - COM1		LCD Common driver output	
TONE	Output	Tone output	
XIN	Input	Resonator connecting pins (High-frequency).	
XOUT	Output		
XTIN (P142)	Input	Resonator connecting pins (Low-frequency).	
XTOUT (P143)	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 shipping test. Be opened or fixed to low level.	
VDD	Power Supply	+ 2.2V to 6.0V	
VSS		0V (GND)	
VLC		LCD drive power supply	

OPERATIONAL DESCRIPTION

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)
        a. Stack, b. Stack Pointer Word (SPW), c. Data Counter (DC)
    2.5 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	3.5 LCD Driver
3.2 Interval Timer	3.6 DTMF Generator
3.3 Timer/Counters (TC1, TC2)	3.7 BEEP Output Circuit
3.4 Watchdog Timer	3.8 Serial Interface

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

2. INTERNAL CPU FUNCTION

2.1 Program Counter (PC)

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

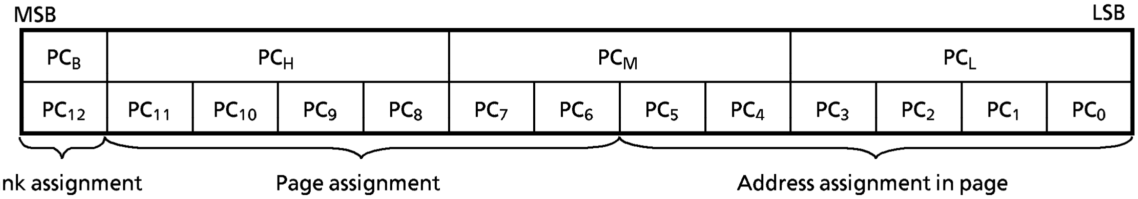


Figure 2-1. Configuration of Program Counter

The PC can directly address an 8192-byte address space. However, with the short/middle branch and subroutine call instructions, the following points must be considered:

(1) Short branch instruction [BSS a]
In [BSS a] instruction execution, when the branch condition is satisfied the status flag is "1", the value specified in the instruction is set to the lower 6 bits of the PC. That is, [BSS a] becomes the in-page branch instruction. When [BSS a] is stored at the last address of the page, the upper 7 bits of the PC point the next page, so that branch is made to the next page.

(2) Middle branch instruction [BS a]
In [BS a] instruction execution, when the branch condition is satisfied, the value specified in the instruction is set to the lower 12 bits of the PC. That is, [BS a] becomes the in-bank branch instruction.

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

Table 2-1. Status Change of Program Counter

When first byte or second byte of this instruction is stored at the last address of the bank, the most significant bit of the PC point the next bank, so that branch is made to the next bank.

(3) Subroutine call instruction [CALL a]

In [CALL a] instruction execution, the contents of the PC are saved to the stack then the value specified by the instruction is set to the PC. The address which can be specified by the instruction consists of 11 bits and the upper 2 bits of the PC is always "0". Therefore, the entry address of the subroutine should be within an address range of 0000_H through $07FF_H$.

2.2 Program Memory (ROM)

The 47C855 has 8192 × 8bits (addresses 0000 through $1FFF_H$)and the 47C655 has 6144 × 8bits (0000 through $17FF_H$), of the program memory (mask ROM).

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

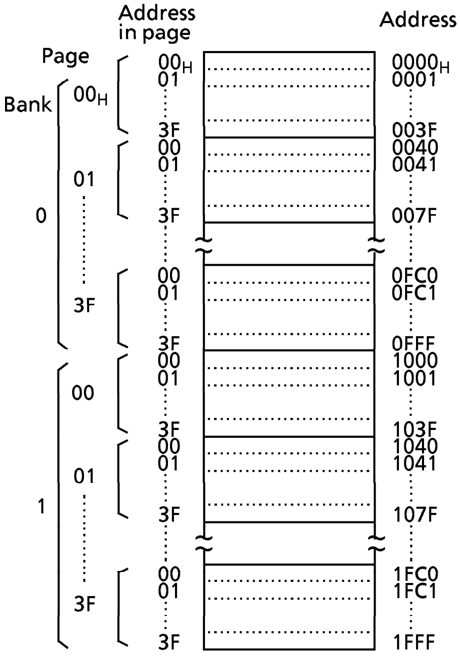


Figure 2-2. Configuration of Program Memory

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

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

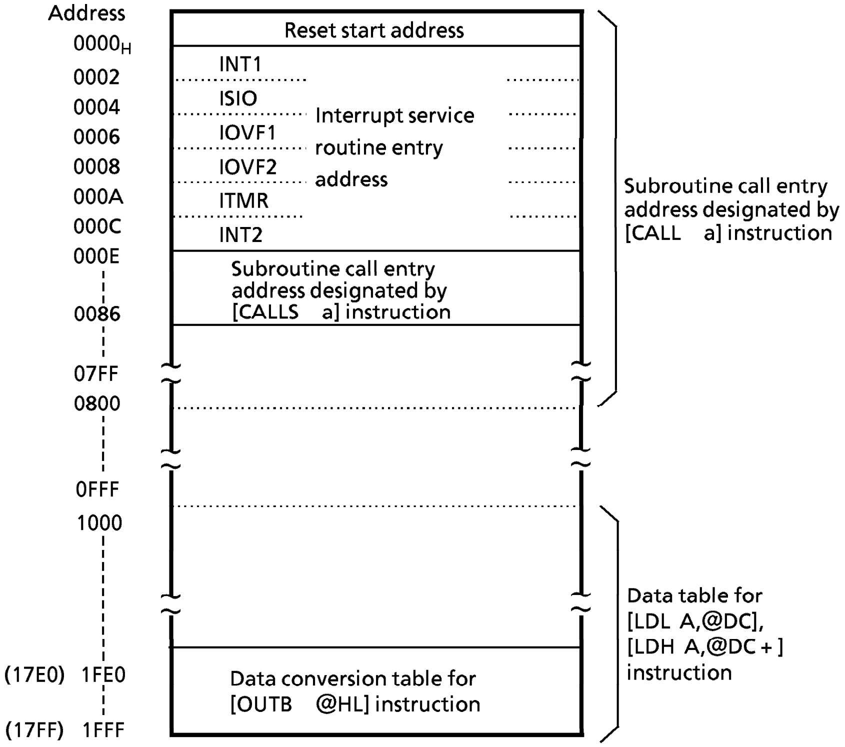
(2) 5-bit to 8-bit data conversion instruction [OUTB @HL]

The 5-bit to 8-bit data conversion instruction reads the fixed data (8 bits) from the data conversion table in the program memory to output the upper 4 bits to COLUMN register and the lower 4 bits to ROW register. The table is located in the last 32-byte space (addresses $1FE0_H$ through $1FFF_H$) in the program memory with the lower address consisting of the 5 bits obtained by linking the data memory contents specified by the HL register pair and the content of the carry flag.

This instruction is suitable for such applications as setting data of DTMF generator.

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 47C855, 0000 through 0086_H and 17E0 through $17FF_H$ for the 47C655, of the program memory are also used for special purposes.



※ ( ) : In case of the 47C655

Note . In case of the 47C655, 5-bit to 8-bit data conversion table is at address 17E0 through $1FFF_H$. When the piggyback is used in order to evaluate the 47C655, it is necessary to also place a conversion table at address 1FE0 through $1FFF_H$ and operation are exactly the same.

Figure 2-3. Program Memory Map

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

The H register and the L register are 4-bit general registers. They are also used as a register pair (HL) for the data memory (RAM) addressing pointer. The data memory consists of pages, each page being 16 words long (1word = 4bits). The H register specifies a page and the L register specifies an address in the page. The data memory consists of two banks (bank0 and bank1). The data memory bank selector (DMB) is a 1-bit register to specify a data memory bank. During reset, the DMB is initialized to "0". The DMB is set or cleared by the [CLR DMB] or [SET DMB] instructions. The currently selected data memory bank can be known by executing the [TEST DMB] or [TESTP DMB] instruction.

The L register has the automatic post-increment/decrement capability, implementing the execution of composite instructions. For example, [ST A,@HL +] instruction automatically increments the contents of the L register after data transfer. During the execution [SET @L], [CLR @L], or [TEST @L] instruction, the L register is also used to specify the bits corresponding to I/O port pins R73 through R40 (the indirect addressing of port bits by the L register).

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

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

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

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

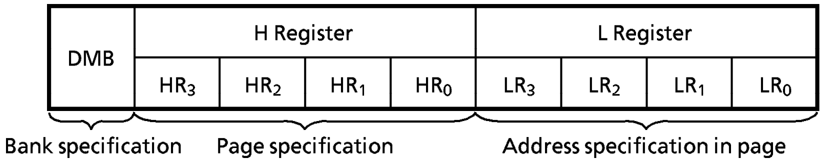


Figure 2-4. Configuration of H, L registers and DMB

2.4 Data Memory (RAM)

The 47C855 has 512 × 4 bits and extend 512 × 4 bits of data memory. The 47C655 has 384 × 4bits, and extend 512 × 4 bits of data memory.

2.4.1 RAM

The data memory is addressed in one of three ways :

(1) Register-indirect addressing mode
In this mode, a bank is specified by the DMB, a page by the H register and an address in the page by the L register.
Example: `LD A,@HL ; Acc ← RAM[HL]`

(2) Direct addressing mode
An address in the bank is directly specified by the 8 bits of the second byte (operand) in the instruction field. The bank is specified by the DMB.
Example: `LD A,2CH ;` Acc ← RAM[$2C_H$]

(3) Zero-page addressing mode
An address in zero-page of bank 0 (addresses 00_H through $0F_H$) by the lower 4 bits of the second byte (operand) in the instruction field.
Example: ST #3,05H ; RAM[05_H] ← 3

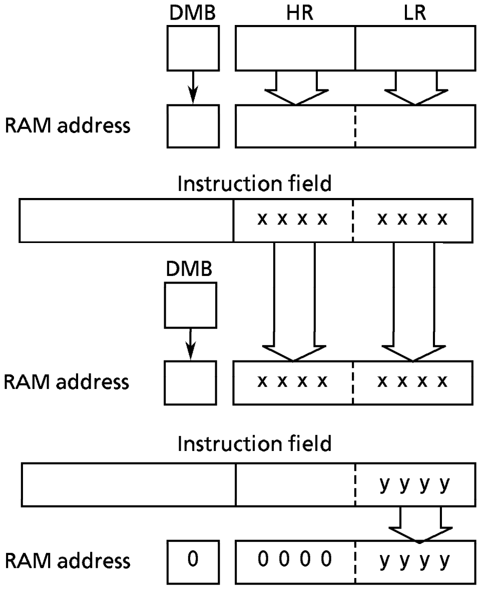


Figure 2-5. Addressing mode

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

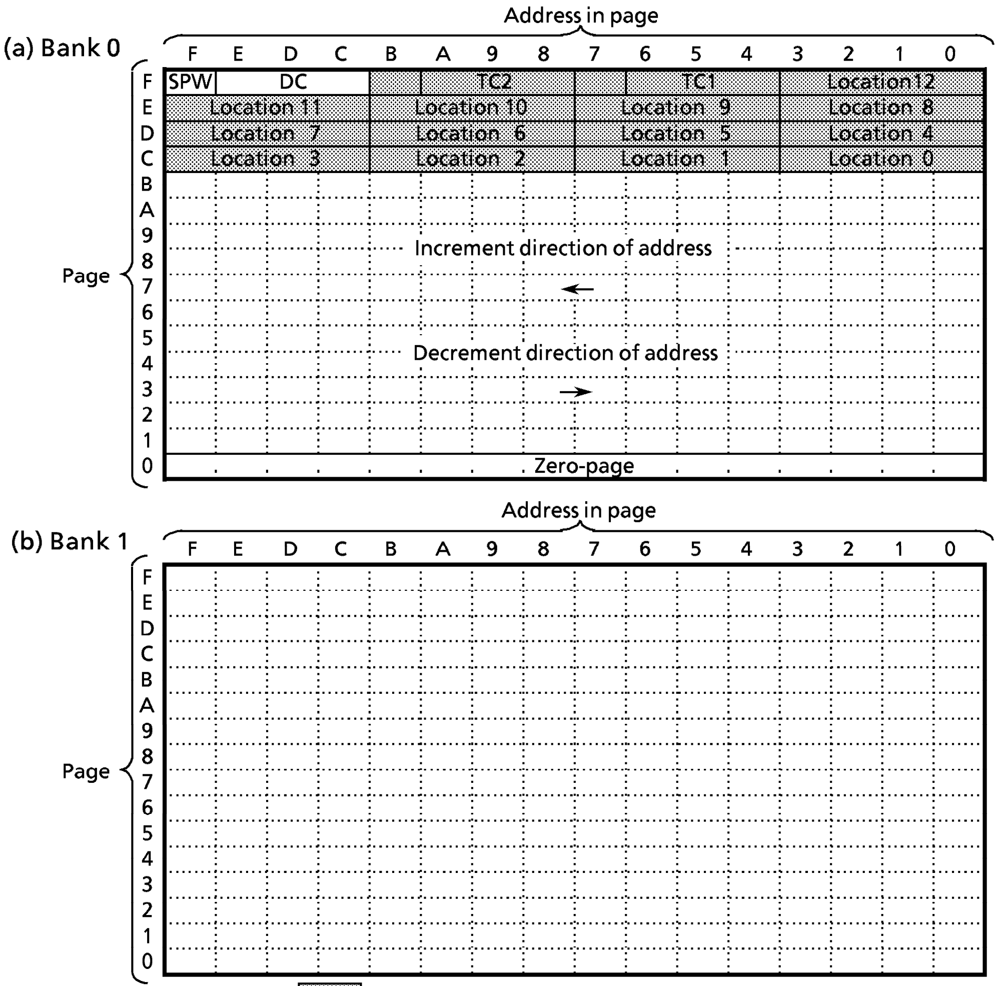
Example: To clear RAM

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

2.4.2 Data Memory Map of RAM

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

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



Note1. ▨ *denotes the stack area.*

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

Figure 2-6. Data Memory Map

(1) Stack

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

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

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

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

(2) Stack Pointer Word (SPW)

Address FF_H in the data memory (bank 0) is called the stack pointer word, which identifies the location in the stack to be accessed (save or restore).

Generally, location number 0 to 12 can be set to the SPW, providing up to 13 levels of stack nesting. Locations 13 and 14 are shared by the count registers of the timer/counters to be described later; therefore, when the timer/counters are not used, the stack area of up to 15 levels is available. Address FF_H is assigned with the SPW, so that the contents of the SPW cannot be set "15" in any case. The SPW is automatically updated when a subroutine call is executed or an interrupt is accepted. However, if it is used in excess of the stack area permitted by the data memory allocating configuration, the user-processed data may be lost. (For example, when the user-processed data area is in an address range 00_H through CF_H, up to location 4 of the stack are usable. If an interrupt is accepted with location 4 already used, the user-processed data stored in addresses CC_H through CF_H corresponding to the location 3 area is lost.)

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

(3) Data counter (DC)

The data counter is a 12-bit register to specify the address of the data table to be referenced in the program memory (ROM). Data table reference is performed by the table look-up instructions [LDL A,@DC] and [LDH A,@DC +]. The data tables are set in the program memory area between addresses 1000_H and $1FFF_H$. The DC is assigned with a RAM address in unit of 4 bits.

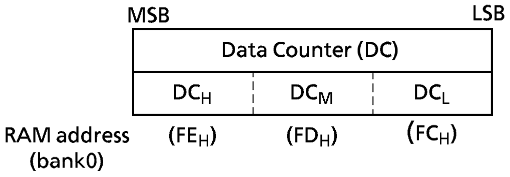
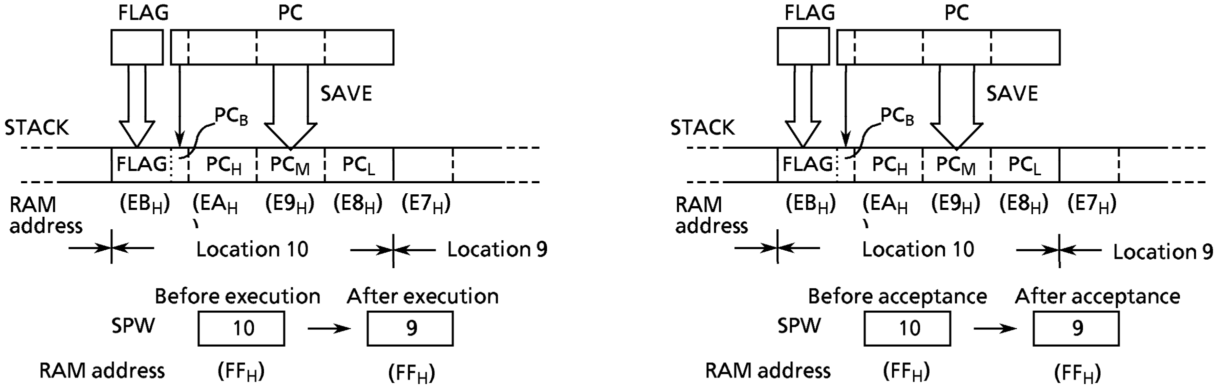


Figure 2-7. Data Counter

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

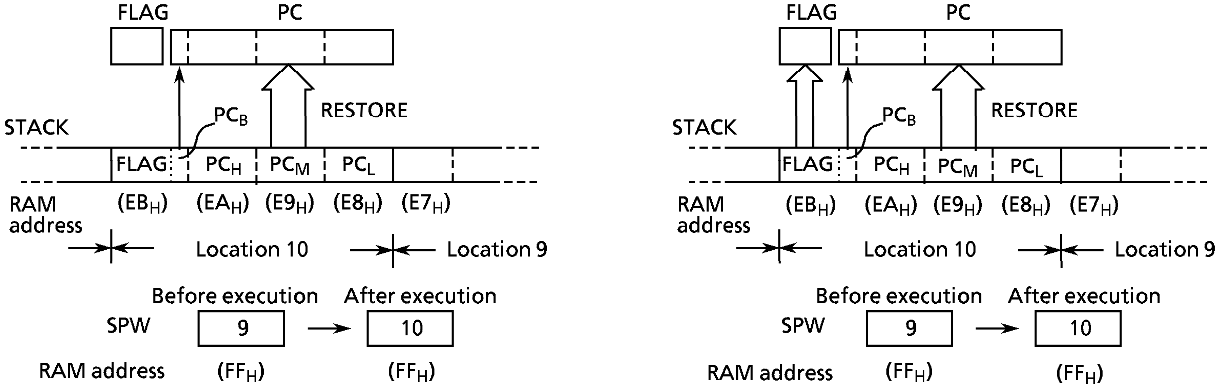
Example: To set the DC to 780_H

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



(a) At execution of a subroutine call instruction

(b) At acceptance of an interrupt



(c) At execution of a subroutine return instruction

(d) At execution of an interrupt return instruction

Figure 2-8. Accessing Stack (Save/Restore)

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

The 47C655/855 has two 12-bit timer/counters. The count register of the timer/counter is assigned with RAM addresses in unit of 4 bits, so that the initial value setting or contents reading is performed by using RAM manipulation instruction.

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

When both timer/counter 1 and timer/counter 2 are used, the data memory (bank 0) locations at addresses F7$_H$ and FB$_H$ can be used to store the user-processed data.

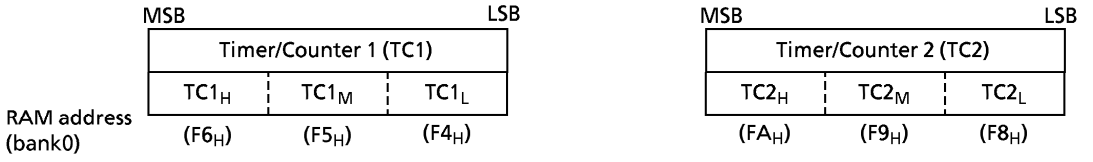


Figure 2-9. Count Registers of Timer/Counters (TC1,TC2)

(5) Zero - page

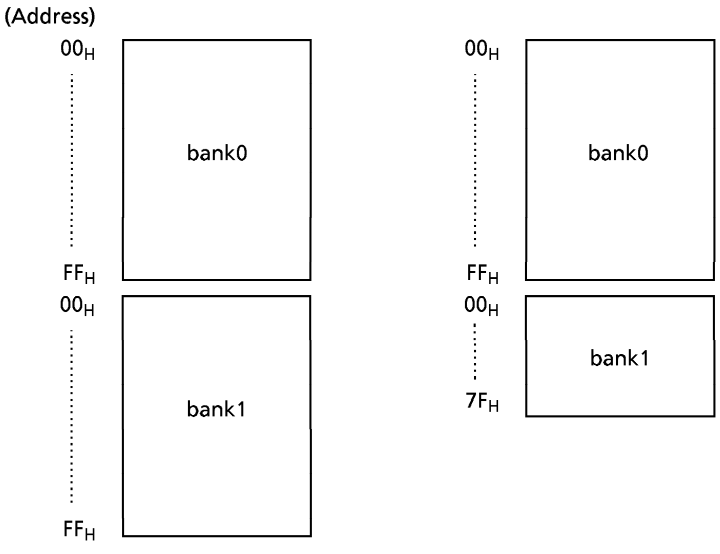
The 16 words (at addresses 00$_H$ through 0F$_H$) of the page zero of the data memory (bank 0) can be used as the user flag or pointer by using zero-page addressing mode instructions (comparison, addition, transfer, and bit manipulation),providing enhanced efficiency in programming.

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

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

2.4.3 Data Memory Capacity

The 47C855 contains two 256 × 4 bit data memory banks (bank0 and bank1). The 47C655 contains 256 × 4 bit bank0 and 128 × 4 bit bank1.



(a) 47C855 (b) 47C655

Figure 2-10. Data Memory (RAM)

2.4.4 Data Memory (RAM : 1024 × 4 bit)

The 47C655/855 data memory consists of a 1024 × 4-bit RAM. Extended 512 × 4-bit RAM is mainly used for storing repertory dialing data and is controlled by the RAM address register. RAM data buffer register and TONE/RAM command register.

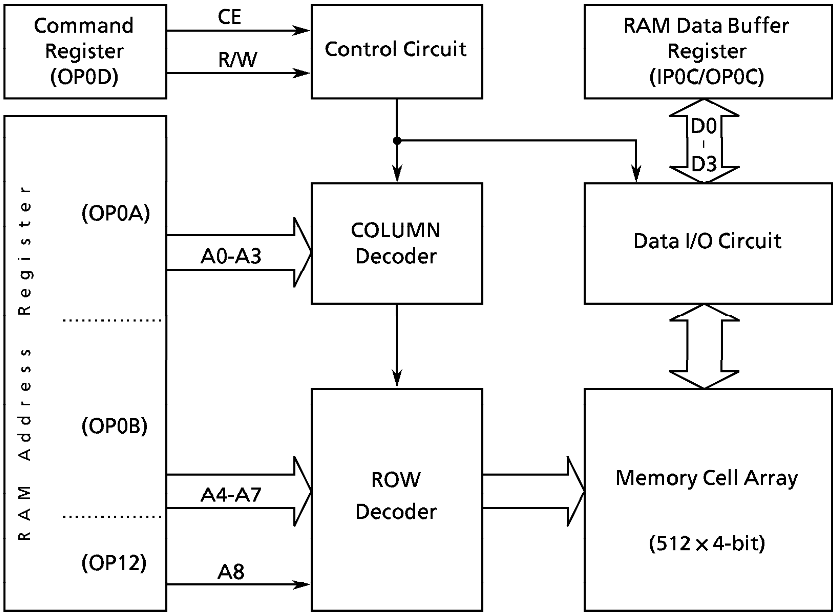


Figure 2-11. Block Diagram

(1) RAM (512 × 4bit) address register

The RAM address register is a 9-bit register to specify addresses for the RAM data memory. The upper 1 bit is accessed with port address OP12, the next 4 bits are accessed with the port address OP0B/IP0B and the lower 4 bits are accessed with port address OP0A/IP0A.

These registers are initialized to "0" during reset.

(Port address OP12)

3	2	1	0
			A8

(Initial value ***0)

(Port address OP0B/IP0B)

3	2	1	0
A7	A6	A5	A4

(Initial value 0000)

(Port address OP0A/IP0A)

3	2	1	0
A3	A2	A1	A0

(Initial value 0000)

Figure 2-12. RAM Address Register

(2) RAM (512 × 4bit) data buffer register

The RAM data buffer register is a 4-bit buffer register to read or write RAM data. When writing data to RAM, it is accessed as port address OP0C. Port address IP0C is used for access when reading data from RAM.

(Port address OP0C/IP0C)

3	2	1	0
D3	D2	D1	D0

Figure 2-13. Data Buffer Register

(3) RAM (512 × 4bit) command register

The RAM command register (OP0D/IP0D) controls the reading or writing data, and whether RAM is to be accessed or put in stand-by mode. This register is accessed as the port address OP0D/IP0D. The RAM command register is also used as the TONE command register.

(Port address OP0D/IP0D)

3	2	1	0
(STE)		R/W	CE

(Initial value 0*00)

R/W / CE	Controls RAM

*0 : Stand-by mode

01 : Read mode

11 : Write mode

*Note 1. * ; don't care*

Note 2. "1" is always read out when the STE bit is read.

Figure 2-14. RAM Command Register

2.4.5 Access for RAM (512 × 4bit)

To write data to RAM, load the address into the RAM address register and the data into the RAM data buffer register (OP0C), then put the RAM command register in the write mode. The data will be written to the specified RAM address by this operation.

The data are latched in the RAM data buffer register, therefore, RAM data buffer register operation is not necessary when the same data are written continuously.

To read data from RAM, set the RAM command register to the read mode and load the address into the RAM address register, then read the data via RAM data buffer register (IP0C). Data are not latched in the RAM data buffer register.

Example 1 : To write data "9" to address 182$_H$ and data "7" to address 15A$_H$ in RAM.

```
        LD     A, #1           ;  Sets data "182H" to RAM address register.
        OUT    A, %OP12
        OUT    #8, %OP0B
        OUT    #2, %OP0A
        OUT    #9, %OP0C       ;  Writes data "9" to RAM data buffer register.
        OUT    #0011B, %OP0D   ;  Sets RAM to write mode.
        OUT    #0010B, %OP0D   ;  Sets RAM to stand-by mode.
        OUT    #5, %OP0B       ;  Sets data "15AH" to RAM address register.
        OUT    #0AH, %OP0A
        OUT    #7, %OP0C       ;  Writes data "7" to RAM data buffer register.
        OUT    #0011B, %OP0D   ;  Sets RAM to write mode.
        OUT    #0010B, %OP0D   ;  Sets RAM to stand-by mode.
```

Example 2 : To write data "0" to address 120$_H$ through 127$_H$ in RAM.

```
          OUT    #0, %OP0C       ;  Writes data "0" to RAM data buffer register.
          LD     A, #0           ;  Sets data "120H" to RAM address register.
          OUT    #1, %OP12
          OUT    #2, %OP0B
          OUT    A, %OP0A
          OUT    #0011B, %OP0D   ;  Sets RAM to write mode.
SLOOP   : CMPR   A, #7           ;  Increases address register.
          TESTP  ZF
          B      SWEND
          INC    A
          OUT    A, %OP0A
          BR     SLOOP
SWEND   : OUT    #0010B, %OP0D   ;  Sets RAM to stand-by mode.
```

Example 3 : To read data from address 0B1$_H$ in RAM and store to Accumulator.

```
        OUT    #0001B, %OP0D   ;  Sets RAM to read mode.
        LD     A, #0           ;  Sets data "0B1H" to RAM address register.
        OUT    A, %OP12
        OUT    #0BH, %OP0B
        OUT    #1, %OP0A
        IN     %IP0C, A        ;  Reads data from RAM and stores to
                                  Accumulator.
```

2.5 ALU and Accumulator

2.5.1 Arithmetic / Logic Unit (ALU)

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

(1) Carry information (C)

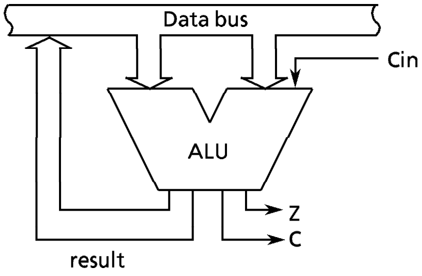
The carry information indicatets a carry-out from the most significant bit in an addtion. A subtraction is performed as addition of two's complement, so that, with a subtraction, the carry information indicates that there is no borrow to the most significant bit. With a rotate instruction, the information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

(2) Zero detect information (Z)

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

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

Operation	Result	C	Z
4 + 2 =	6	0	0
7 + 9 =	0	1	1
8 − 1 =	7	1	0
2 − 2 =	0	1	1
5 − 8 =	− 3 (1101_B)	0	0



Note. Cin indicates the carry input specified by instruction.

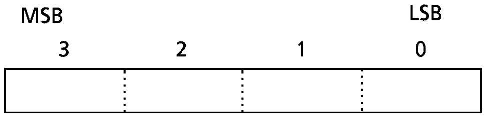
Figure 2-15. ALU

2.5.2 Accumulator (Acc)

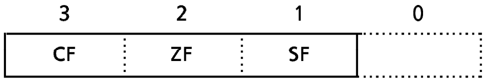
The accumulator is a 4-bit register used to hold source data or results of the operations and data manipulations.

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.



(a) Acc



(b) Flag

Figure 2-16. Acc, Flag

(1) Carry flag (CF)

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

① Addition/subtraction with carry instructions [ADDC A,@HL], [SUBRC A,@HL]
The CF becomes the input (Cin) to the ALU to hold the carry information.

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

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

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

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

(2) Zero flag (ZF)

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

(3) Status flag (SF)

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

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

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

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

2.7 Clock Generator, Timing Generator, and System Clock Controller

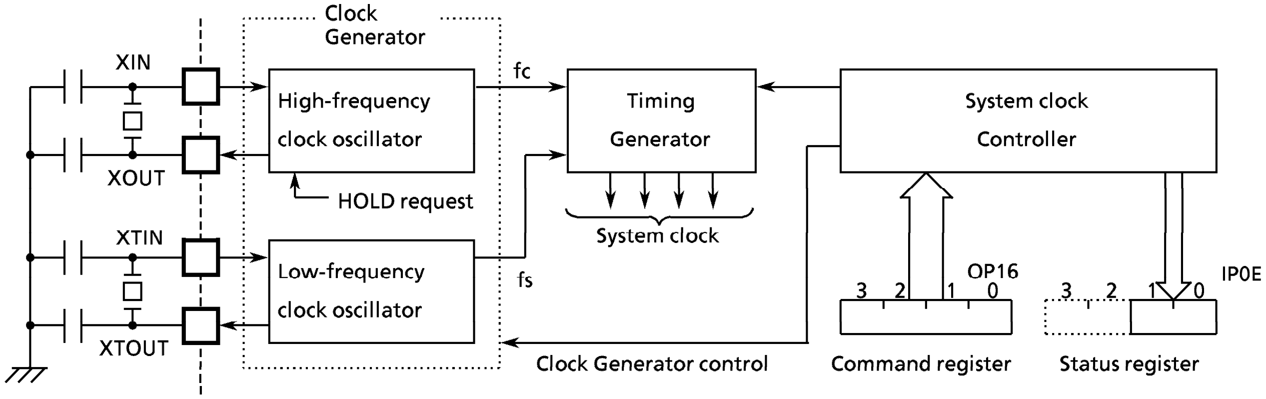
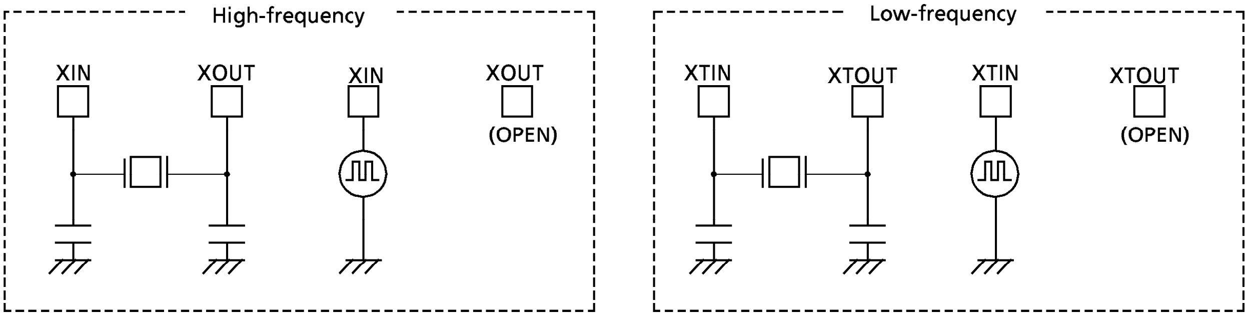


Figure 2-17. Clock Generator, Timing Generator and System Clock Controller

2.7.1 Clock Generator

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



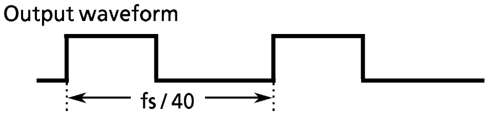
(a) Crystal or ceramic resonator (b) External oscillator (c) Crystal (d) External oscillator

Figure 2-18. Examples of Resonator Connection

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

Example : To output the low-frequency oscillation frequency adjusting monitor pulse to R70 pin.
(Under the condition that the low-frequency clock oscillation is stable in SLOW operation)

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



2.7.2 Timing Generator

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

2.7.3 System Clock Controller

The system clock controller starts or stops the high-frequency and low-frequency clock oscillator and switches between the basic clocks. The operating mode is generally divided into the single-clock mode and the dual-clock mode, which are controlled by command. Figure 2-19 shows the operating mode transition diagram. Figure 2-20 shows the command and status registers.

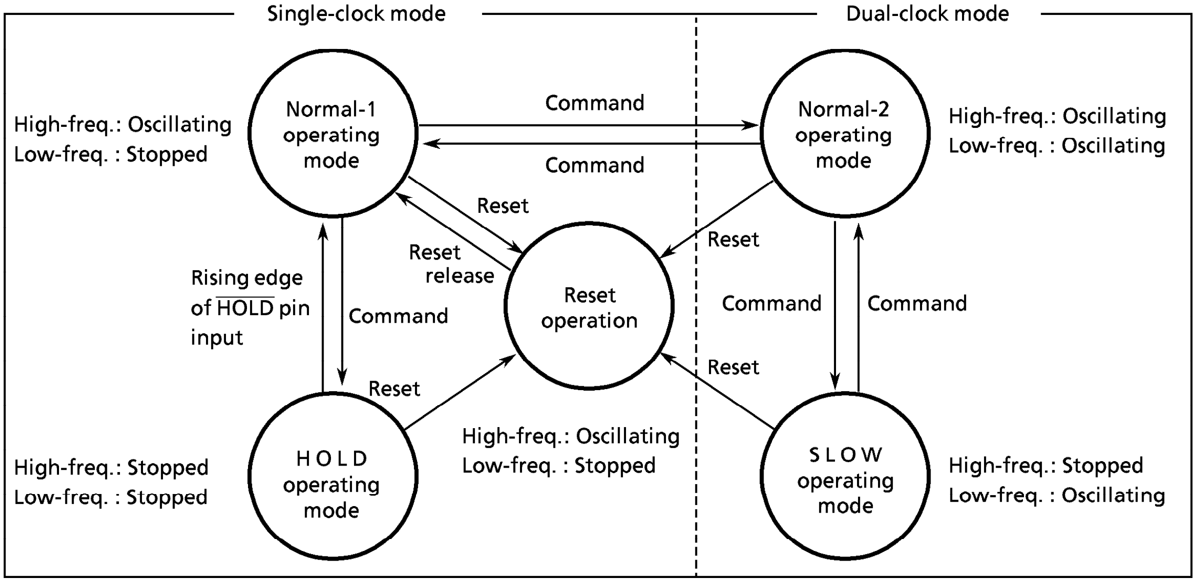


Figure 2-19. Operating Mode Transition Diagram

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.

b. Dual-clock mode

① Normal-2 operating mode

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

② SLOW operating mode

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

Notes.

1. *In the HOLD and SLOW operating modes, the power consumed by the oscillator and the internal hardware is reduced. However, the power for the pin interface (depending on the external circuitry and program) is not directly associated with the low-power consumption operation. This must be considered in system design as well as interface circuit design.*
2. *Normal-1 and Normal-2 operating modes are sometimes referred to as the Normal operating mode collectively.*

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

System clock control command register (Port address OP16)

3	2	1	0
DCLK3	SLCK	DCLK1	DWUT

(Initial value 0000)

DCLK3 / DCLK1	Selects operation mode
0 0 :	Single clock mode (Normal-1 operating mode)
0 1 :	Reserved
1 0 :	Dual clock mode (Normal-2 operating mode)
1 1 :	Dual clock mode (SLOW operating mode)

SLCK	Selects input clock for the interval timer
0 :	$fc / 2^7$ [Hz]
1 :	fs

(Note 2)

DWUT	Sets the warm-up time	Example : At fc = 4.19MHz fs = 32.8kHz
0 :	$2^8 / fs + 2^9 / fc$ [s]	7.9 [ms]
1 :	$2^{11} / fs + 2^9 / fc$	62.6

Note 1 . fc ; High-frequency clock [Hz]
fs ; Low-frequency clock [Hz]
Note 2. Only Normal-2 operating mode
Note 3. 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 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 2-20. System Clock Control Command Register/Status Register

(1) Single-clock mode

In this mode, only the high-frequency clock oscillator is used. Pins R72 (XTIN) and R73 (XTOUT) become the ordinary I/O port. The HOLD operating mode is available for reducing power consumption. It is controlled by the command register (OP10). In this mode, therefore, the system clock control command register (OP16) need not be manipulated. For the details of the HOLD operation, refer to Subsection "5.1 HOLD Operating Mode".

(2) Dual-clock mode

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

(3) System clock switching control

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

a. Switching from Normal-2 operation to SLOW operation

SMF (bit 1 of the status register) is monitored by program. First, Sets SLCK(bit2 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", bit1 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 2 of the command register is cleared to "0" and, at the same time,the warm-up time for return is set to DWUT. When the warm-up time has passed, the Normal-2 operation takes place. By monitoring SLS (bit 0 of the status register), the current operating mode can be known.

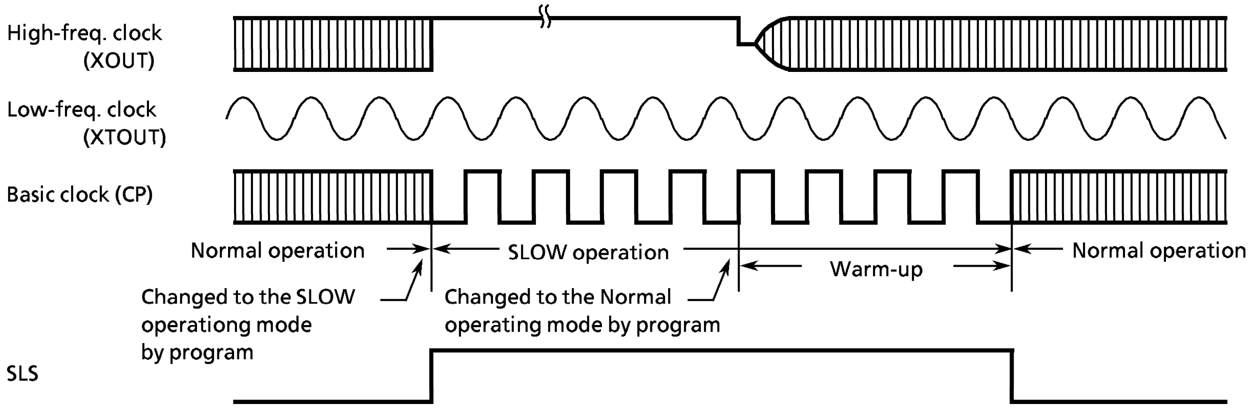


Figure 2-21. System Clock Switching Timing

2.7.4 Instruction Cycle

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

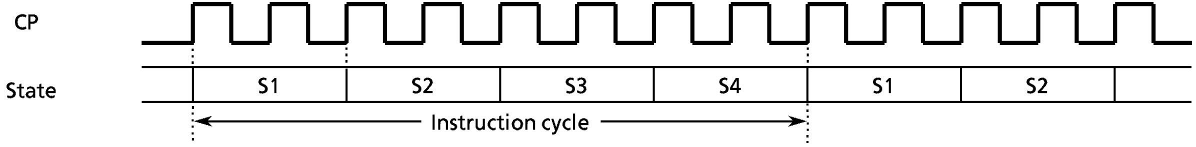


Figure 2-22. Instruction Cycle

2.7.5 Hold Operating Mode

The HOLD feature stops the system and holds the system's internal states active before stop with a low power. The HOLD operation is controlled by the command register (OP10) and the $\overline{\text{HOLD}}$ pin and K0 port inputs. The $\overline{\text{HOLD}}$ pin and K0 port inputs state can be known by the status registers (IP0E and IP10). The $\overline{\text{HOLD}}$ pin is shared by the $\overline{\text{KE0}}$ pin.

Configuration of HOLD operation circuit is shown in Figure 2-23.

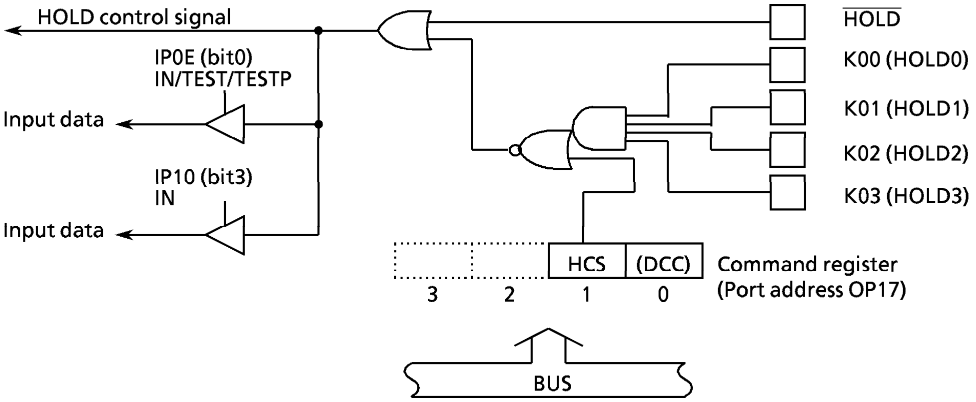


Figure 2-23. Configuration of Hold Control Circuit

The 47C655/855 have a $\overline{\text{HOLD}}$ pin and K0 port as HOLD control input. Therefore, in the case of using K0 port for key inputs, the HOLD operation can be released by key inputs. HOLD control by K0 port input can be inhibited by HOLD control input select command register. (bit 1 of OP17)

HOLD control input select command register (Port address OP17)

3	2	1	0	
		HCS	(DCC)	(Initial value **00)

HCS	Selects HOLD control input

0 : Controls HOLD operation by $\overline{\text{HOLD}}$ pin and K0 port inputs
1 : Controls HOLD operation by $\overline{\text{HOLD}}$ pin input

Figure 2-24. HOLD Control Input Select Command Register

(1) Starts the HOLD operating mode

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

① Oscillator stops and the system's internal operations are all held up.
② The interval timer is cleared to "0".
③ The states of the data memory, registers, and latches valid immediately before the system is put in the HOLD state are all held.
④ The program counter holds the address of the instruction to be executed after the instruction which starts the HOLD operating mode.

HOLD oprerating mode command register (Port address OP10)

3	2	1	0
HLDMS		HWUT	

(Initial value *0**)

HLDMS	Sets mode and starts HOLD operation

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

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

		Example : At fc = 960kHz
00:	2^{18}/ fc [s]	...... 27.3 [ms]
01:	2^{14}/ fc	...... 17.1
10:	Reserved	
11:	2^{6}/fc	...... 0.067

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

HOLD oprerating mode status register (Port address IP0E)

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

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

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

Figure 2-25. HOLD Operating Mode Command Register/Status Register

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

(1) Level-sensitive release mode

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

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

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

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

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

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

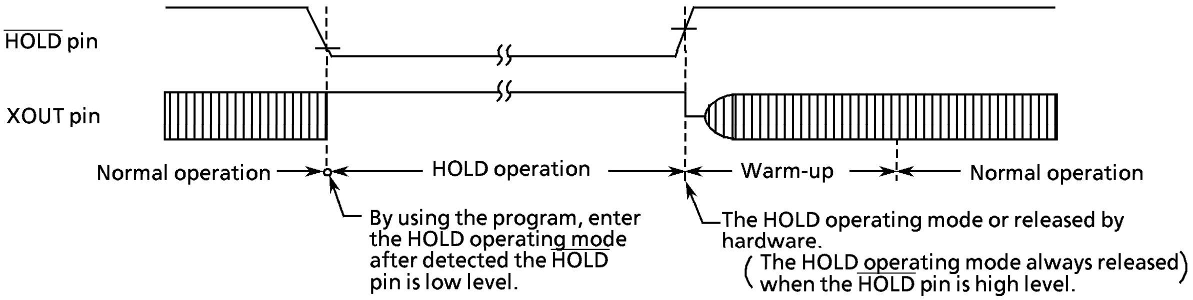


Figure 2-26. Level-sensitive Release Mode

(2) Edge-sensitive release mode

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

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

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
```

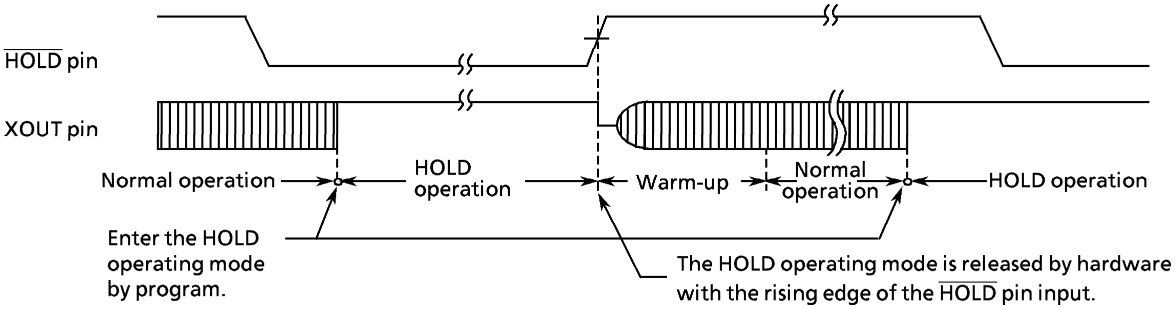


Figure 2-27. Edge-sensitive Release Mode

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 oscillator starts.

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

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

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

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

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

2.7.6 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 occurs simultaneously, the one with the highest priority determined by hardware is serviced first.

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

Table 2-2. Interrupt Sources

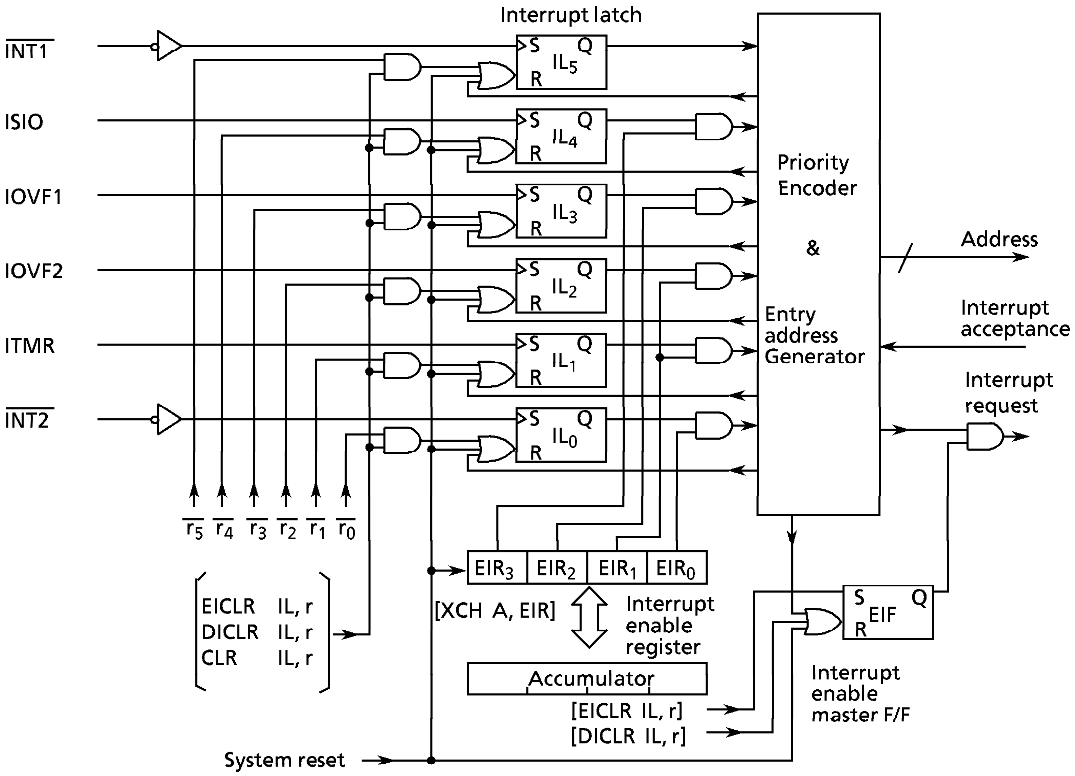


Figure 2-28. Interrupt Controller Block Diagram

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

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

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

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

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

(2) Interrupt enable register (EIR)

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

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

(3) Interrupt latches (IL_5 through IL_0)

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

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

Example 1 : To enable IOVF1, INT1, and INT2

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

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

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

(2) Interrupt Processing

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

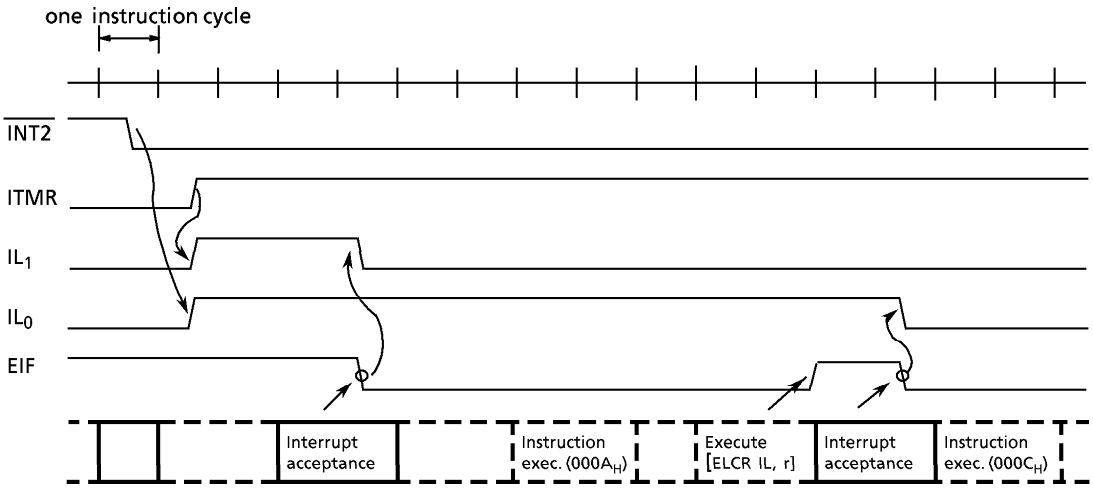
The interrupt acknowledge processing consists of the following sequence:

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

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

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

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



Note 1. *It is assumed that there is no other interrupt request and EIR = 0011_B.*
Note 2. *The value r in the [EICLR IL, r] instruction is assumed as 11111_B.*
Note 3. ┌ ─ ─ ┐ *denotes the execution of an instruction.*

Figure 2-29. Interrupt Timing Chart (Example)

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

① Restores the contents of the program counter and the flags from the stack.
② Sets the EIF to "1" to provide the interrupt enable state again.

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

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

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

```
XCH     HL, GSAV1     ;  RAM[GSAV1] ⇔ HL
XCH     A, GSAV1 + 2  ;  RAM[GSAV1 + 2] ⇔ Acc
```

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

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

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

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

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

(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 000_H.
The $\overline{\text{RESET}}$ pin is a hysteresis input with a pull-up resistor (220 kΩ typ.). Externally attaching a capacitor and a diode implement a simplified power-on-reset.

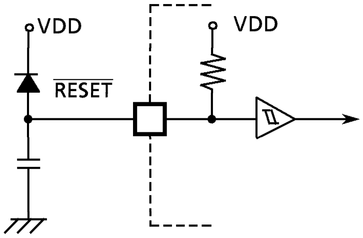


Figure 2-30. Simplified power-on-reset circuit

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

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

2.9.1 Warm-Start

The warm-start capability 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 2-31 shows Warm-start Circuit Example

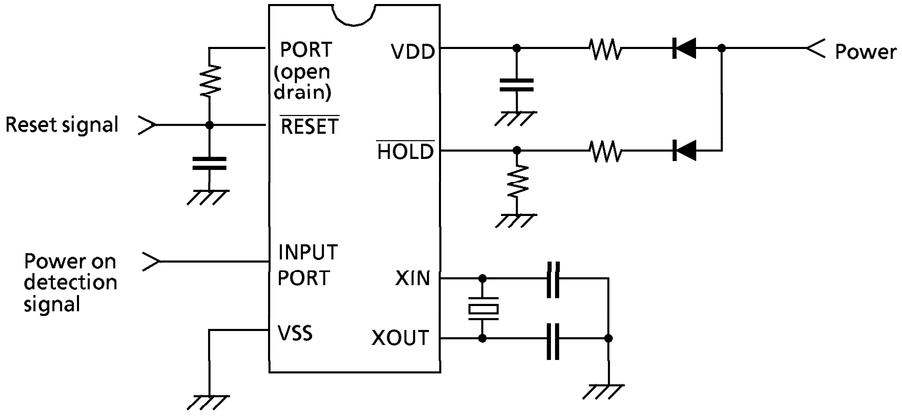


Figure 2-31. Warm-start Circuit Example

3. PERIPHERAL HARDWARE FUNCTION

3.1 Ports

The 47C655/855 have 10 ports (34pins) each as follows:

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

The 47C655/855 does not have the port P1 and P2.

3.1.1 I/O Timing

(1) Input timing

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

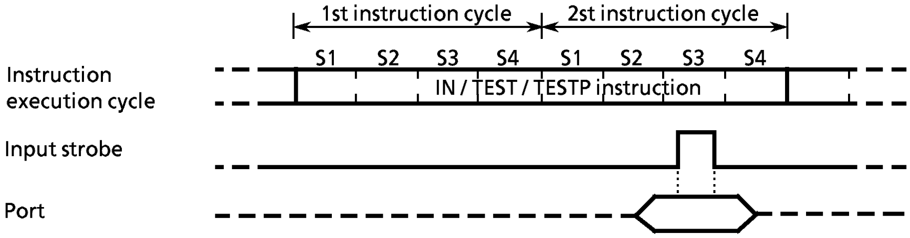


Figure 3-1. Input Timing

(2) Output timing

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

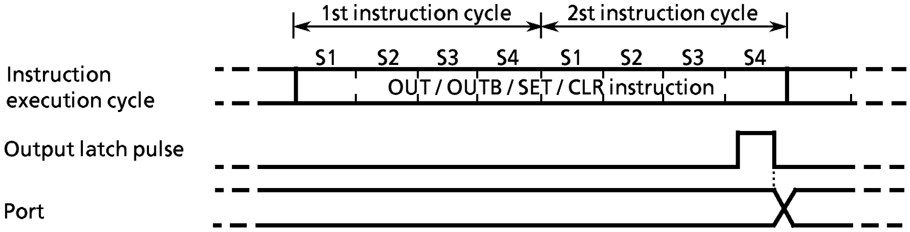


Figure 3-2. Output Timing

Port Address (**)	Port		Input/Output instruction						
	Input (IP**)	Output (OP**)	IN %p, A IN %p, @HL	OUT A, %p OUT @HL,%p	OUT #k, %p	OUTB @HL	SET %p, b CLR %p, b	TEST %p, b TESTP %p, b	SET @L CLR @L TEST @L
00_H	K0 input port	——	○	–	–	–	–	○	–
01	ROW register	ROW register	○	○	○	○ (Note 2)	○	○	–
02	COLUMN register	COLUMN register	○	○	○	○	○	○	–
03	R3 input port	R3 output port	○	○	○	–	○	○	–
04	R4 input port	R4 output port	○	○	○	–	○	○	○
05	R5 input port	R5 output port	○	○	○	–	○	○	○
06	R6 input port	R6 output port	○	○	○	–	○	○	○
07	R7 input port	R7 output port	○	○	○	–	○	○	○
08	R8 input port	R8 output port	○	○	○	–	○	○	–
09	R9 input port	R9 output port	○	○	○	–	○	○	–
0A	RAM2 address register	RAM2 address register	○	○	○	–	○	○	–
0B	RAM2 address register	RAM2 address register	○	○	○	–	○	○	–
0C	RAM2 data buffer register	RAM2 data buffer register	○	○	○	–	○	○	–
0D	RAM2 command register	RAM2 command register	○	○	○	–	–	○	–
0E	Status register	——	○	–	–	–	–	○	–
0F	Serial transfer buffer	Serial transfer buffer	○	○	○	–	–	–	–
10_H	Undefined	Hold operating mode control	–	○	–	–	–	–	–
11	Undefined	——	–	–	–	–	–	–	–
12	Undefined	RAM2 address register	–	○	–	–	–	–	–
13	Undefined	BEEP output control	–	○	–	–	–	–	–
14	Undefined	P14 output port	–	○	–	–	–	–	–
15	Undefined	Watchdog timer control	–	○	–	–	–	–	–
16	Undefined	System clock control	–	○	–	–	–	–	–
17	Undefined	HCS, DCC	–	○	–	–	–	–	–
18	Undefined	——	–	–	–	–	–	–	–
19	Undefined	Interval timer interrupt control	–	○	–	–	–	–	–
1A	Undefined	——	–	–	–	–	–	–	–
1B	Undefined	LCD driver control	–	○	–	–	–	–	–
1C	Undefined	Timer/counter 1 control	–	○	–	–	–	–	–
1D	Undefined	Timer/counter 2 control	–	○	–	–	–	–	–
1E	Undefined	Serial interface control	–	○	–	–	–	–	–
1F	Undefined	Serial interface control	–	○	–	–	–	–	–

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

Note 2. The 5-bit to 8-bit data conversion instruction [OUT @ HL], automatic access to ROW register and COLUMN register.

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

3.1.2 I/O Ports

(1) Port K0 (K03-K00)

The 4-bit input port with pull-up resistors, shared with hold request/release signal input.

Port K0 (Port address IP00)

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

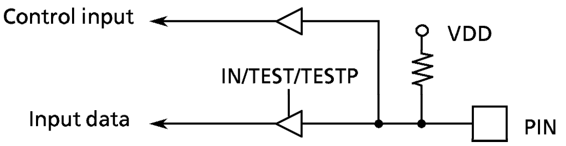


Figure 3-3. Port K0

(2) Port R3 (R33-R30)

The 4-bit I/O port with latch. When used as 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 ($\overline{\text{WTO}}$)	R32	R31	R30

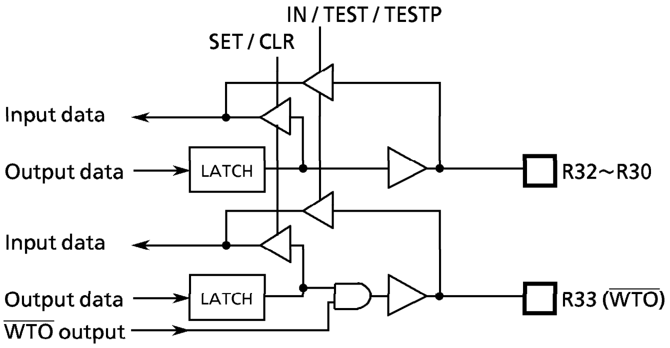


Figure 3-4. Port R3

(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				Pin
3	2	1	0	
0	0	0	0	R40
0	0	0	1	R41
0	0	1	0	R42
0	0	1	1	R43

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

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

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

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

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

Ports R4 and R5 are 4-bit I/O ports. When used as an input ports or analog inputs, the latch should be set to "1".

The latch is initialized to "1".

Port R5 (Port address OP05 / IP05)

3	2	1	0
R53	R52	R51	R50

Port R4 (Port address OP04 / IP04)

3	2	1	0
R43	R42	R41	R40

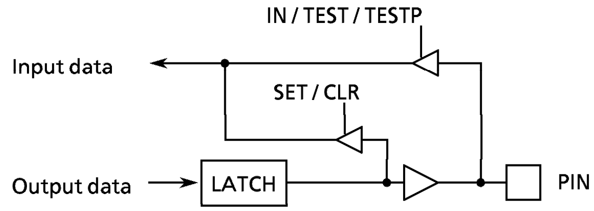


Figure 3-5. Ports R4, R5

b. Port R6 is 4-bit I/O port with a latch. When used as an input, the latch must be set to "1". The latch is initialized to "1" during reset.

Port R6 (Port address OP06 / IP06)

3	2	1	0
R63	R62	R61	R60

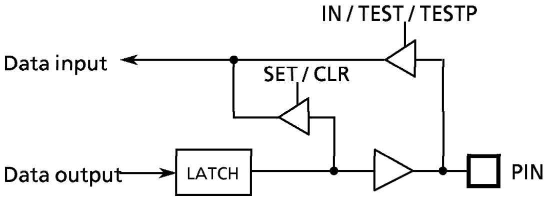


Figure 3-6. Port R6

c. Port R7 is shared by the low-frequency resonator connection pins (XTIN, XTOUT) and the watchdog timer output pin ($\overline{\text{WTO}}$). For the dual-clock mode operation, the low-frequency resonator (32.768kHz) is connected to R72 (XTIN) and R73 (XTOUT) pins. For the single-clock mode operation, R72 and R73 pins are used for the ordinary I/O ports. When the watchdog timer is used, R71 ($\overline{\text{WTO}}$) becomes the watchdog timer output pin. The watchdog timer output is the logical AND output with the port R71 output latch. To use the R71 pin for an ordinary I/O port, the watchdog timer must be disabled (with the watchdog timer output set to "1").

Port R7 (Port address OP07 / IP07)

3	2	1	0
R73	R72	R71	R70

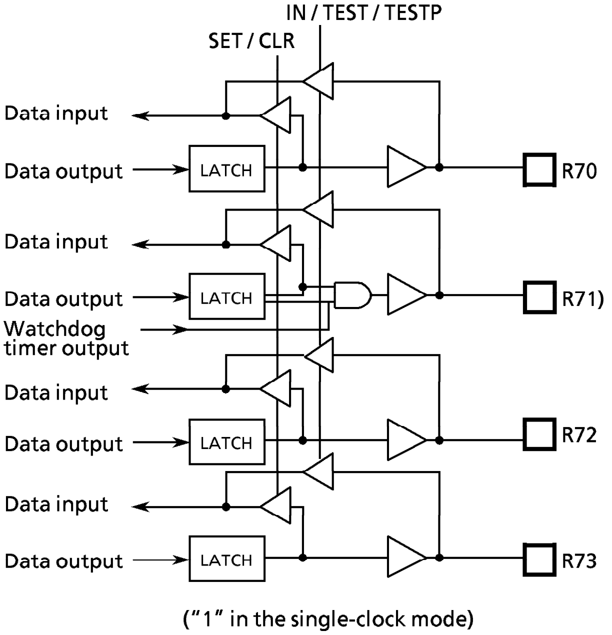


Figure 3-7. Port R7

(4) Port R8 (R83 - R80)

Port R8 is a 4-bit I/O port with a latch. When used as an input port, the latch must be set to "1". The latch is initialized to "1" during reset. Port R8 is shared by the external interrupt request input pin and the timer/counter input pin. To use this port for one of these functional pins, the latch should be set to "1". To use it for an ordinary I/O port, the acceptance of external interrupt must be disabled or the event counter/pulse width measurement modes of the timer/counter must be disabled.

Note. When R82 ($\overline{\text{INT1}}$) 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 ($\overline{\text{INT2}}$)

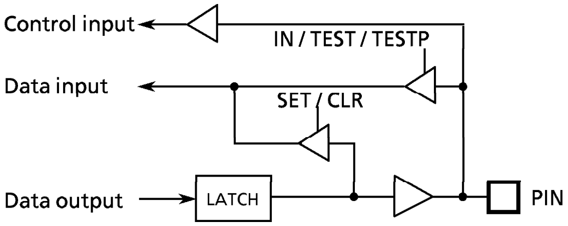


Figure 3-8. Port R8

(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.

Port R9 (Port address OP09 / IP09)

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

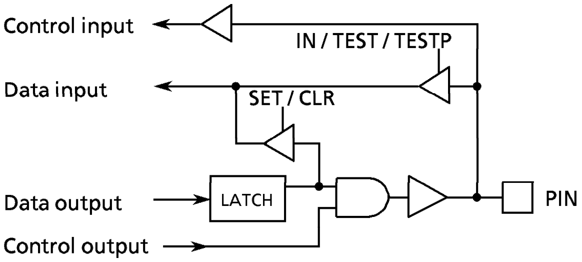


Figure 3-9. Port R9

(6) Port P14 (P141-P140)

The 4-bit output port with latch. The latch is initialized to "1" during reset. The pin P140 is shared with the BEEP output and the low-frequency resonator connection pins (XTIN, XTOUT).
When used as the BEEP output, the latch must be set to "1".

Port P14 (Port address OP14)

3	2	1	0
P143 (XTOUT)	P142 (XTIN)	P141	P140 (BEEP)

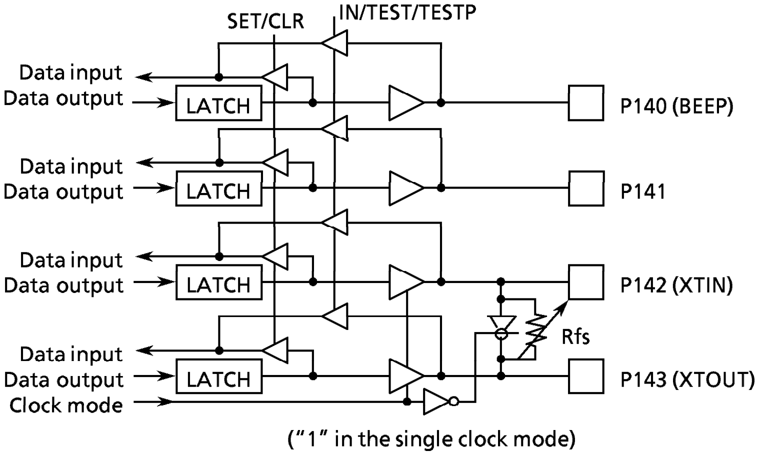


Figure 3-10. Port P14

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.

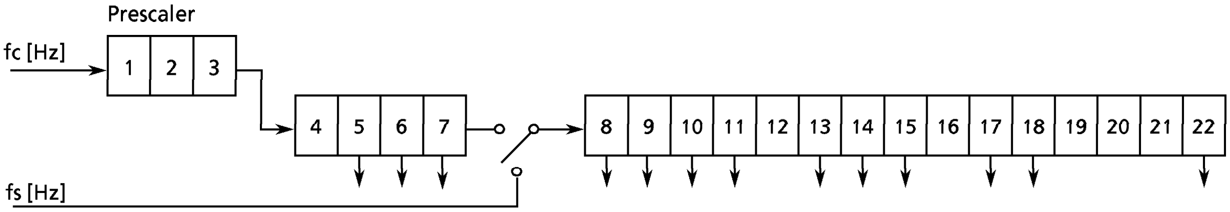


Figure 3-12. Configuration of Interval Timer

3.2.2 Functions of Interval Timer

The interval timer provides the following functions:

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

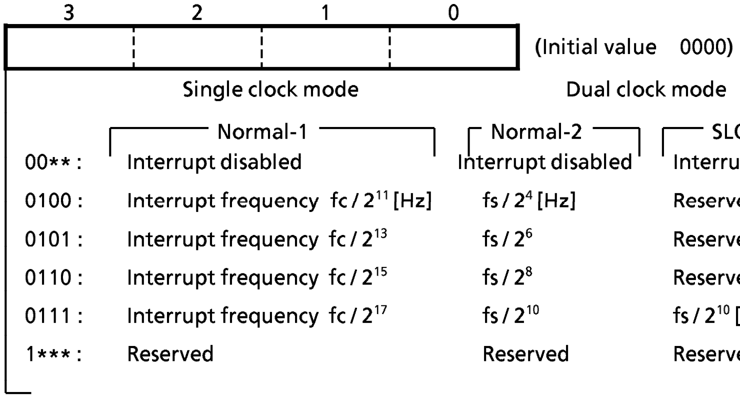
3.2.3 Interval Timer Interrupt (ITMR)

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

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

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

Interval timer interrupt control command register (Port address OP19)



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

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

(a) Command register

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 3-13. Interval Timer Interrupt Command Register

3.3 Timer/Counters (TC1, TC2)

The 47C655/855 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)	($F9_H$)	($F8_H$)

Figure3-14. 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 3-15. 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 3-16. Note that counting continues if there is a count request after overflow occurrence.

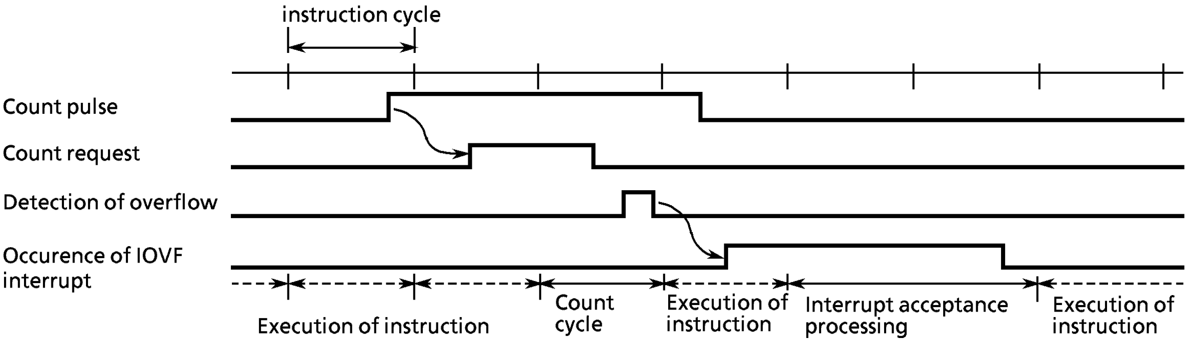


Figure 3-16. Timer/Counter Overflow Interrupt Timing

(1) Event counter mode
In the event counter mode, the timer/counter increments at each rising edge of the external pin (T1, T2) input. T1, T2 pins are shared by R83, R81 pins. Output latch of R83, R81, are set to "1" when used as timer/counter input. Also output latch is initialized "1" during reset. The maximum applied frequency of the external pin input is fc/32 for the 1-channel operation; for the 2-channel operation, the frequency is fc/32 for TC1 and fc/40 for TC2. The apparent instruction execution speed drops most to (9/11) × 100 = 82% when TC1 and TC2 are operated at the maximum applied frequency because the count operation is inserted once every 4 instruction cycles for TC1 and every 5 instruction cycles for TC2. For example, the instruction execution speed of 8.3μs drops to 15.1μs.

Example: To operate TC2 in the event counter mode.

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

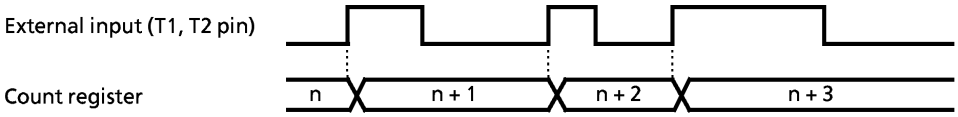


Figure 3-17. Event Counter Mode Timing Chart

(2) Timer mode
In the timer mode, the timer/counter increments at the rising edge of the internal pulse generated from the interval timer. One of 4 internal pulse rates can be selected by the command register. The selected rate can be initially set to the timer/counter to generate an overflow interrupt in order to crerate a desired time interval.
When an internal pulse rate of $fc/2^{10}$ is used, a count operation is inserted once every 128 instruction cycles, so that the apparent instruction execution speed drops by (1/127) × 100 =

0.8%. For example, the instruction execution speed of 8.3μs drops to 8.366μs. 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 ← FA2H (Setting of count register)
ST      #2,@HL+
ST      #0A,@HL+
ST      #0FH,@HL+
LD      A,#1010B    ;  OP1C ← 1010B (Timer mode rate fc/2^10)
OUT     A,%OP1C
LD      A,#0100B    ;  EIR ← 0100B (Enables interrupt)
XCH     A,EIR
EICLR   IL,110111B  ;  EIF ← 1, IL3 ← 0
```

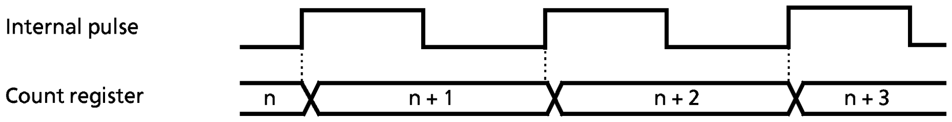


Figure 3-18. Timer Mode Timing Chart

※ Calculating the initial value of the count register

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

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

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

NORMAL 1 mode	NORMAL 2 mode (TG INPUT)		SLOW mode	MAX SETTING TIME [s] (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 3-20, the timer/counter increments only while the external pin input is high. The maximum applied frequency to the external pin input must be one that is enough for analyzing the count value by program. Normally, a frequency sufficifient slower than the internal pulse ratesetting is applied to the external pin.

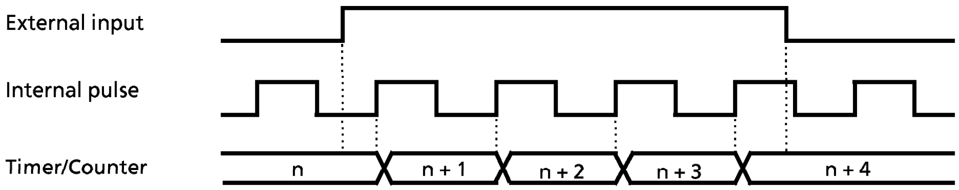


Figure 3-19. Pulse Width Measurement Mode Timing Chart

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 R33 ($\overline{\text{WTO}}$) pin as the malfunction detect signal. To use the WDT, the output latch of R33 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 3-10. 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.

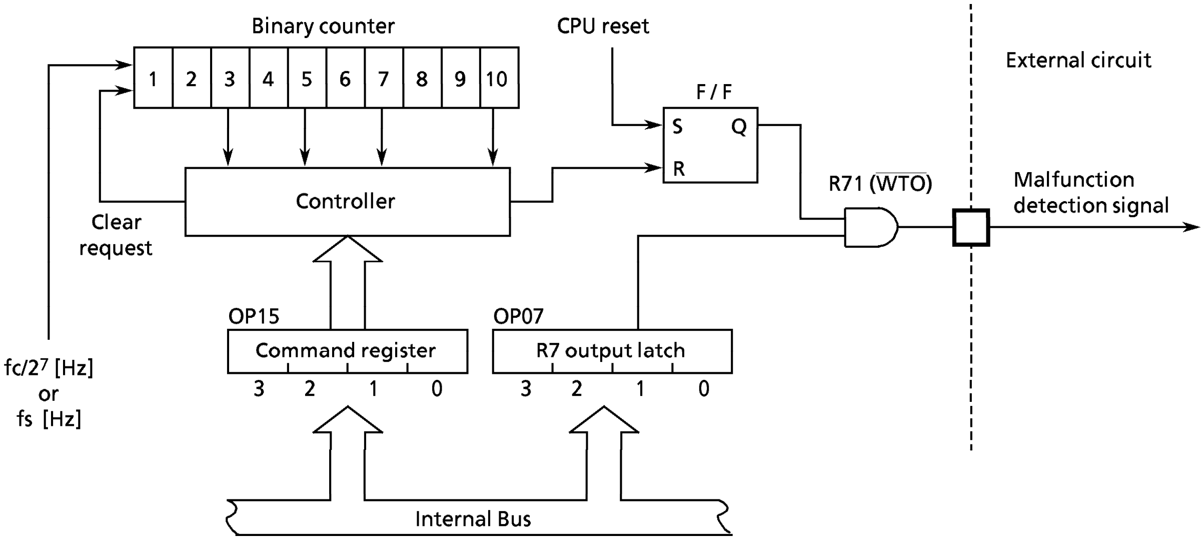


Figure 3-20. Configuration of Watchdog Timer

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 R33 is "1" at this time, the $\overline{\text{WTO}}$ output goes low.

Example : To enable the with detection time of 63 × 2^{15}/fc

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

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

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]

Figure3-21. Watchdog Timer Control Command Register

TWT	Normal 1 operation mode	Normal 2 operation mode (Interval timer input)		SLOW operation mode
		$fc / 2^7$ [Hz]	fs [Hz]	
00	$3 \times 2^{15} / fc$ [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 3-3. Watchdog Timing detection time

3.5 LCD Driver

The 47C655/855 have circuit that directly drives the Liquid Crystal Display (LCD) and its control circuit.

The 47C655/855 have the following connecting pins with:

① Segment output	32 pins	(SEG31-SEG1)
② Common output	4 pins	(COM4-COM1)

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

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

① 1/4 duty (1/3 bias) LCD	Max. 128 segments	(8 segments × 16 digits)
② 1/3 duty (1/3 bias) LCD	Max. 96 segments	(8 segments × 12 digits)
③ 1/2 duty (1/2 bias) LCD	Max. 64 segments	(8 segments × 8 digits)
④ Static LCD	Max. 32 segments	(8 segments × 4 digits)

3.5.1 Circuit Configuration

Figure 3-22 shows the configuration of the LCD driver.

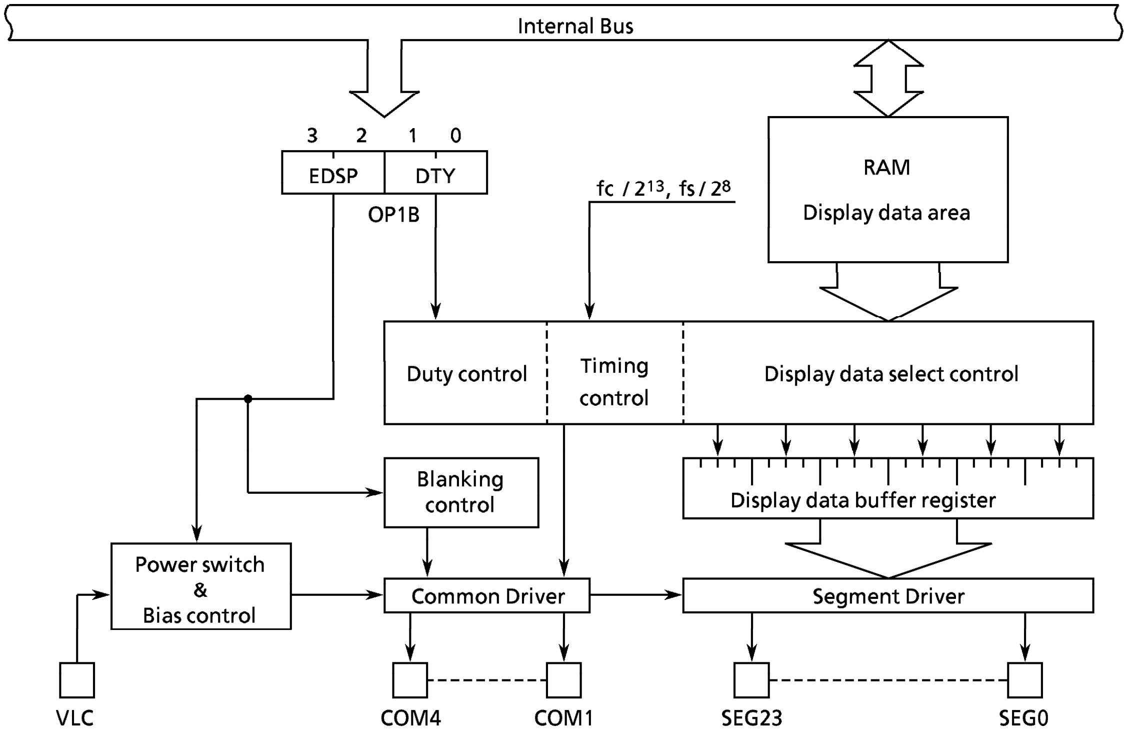


Figure 3-22. Configuration of LCD Driver

3.5.2 Control of LCD Driver

The LCD driver is controlled by the command register (OP1B).

LCD driver control command register (Port address OP1B)

3	2	1	0
EDSP		DTY	

(Initial value 0000)

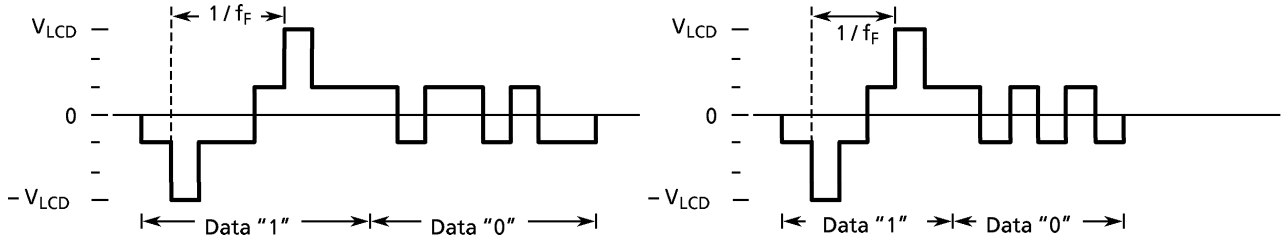
EDSP	LCD display control
0 0 :	Designation of driving methods
0 1 :	Blanking
1 0 :	Reserved
1 1 :	Enables display

DTY	Selection of driving methods
0 0 :	1/4 duty (1/3 bias)
0 1 :	1/3 duty (1/3 bias)
1 0 :	1/2 duty (1/2 bias)
1 1 :	Static

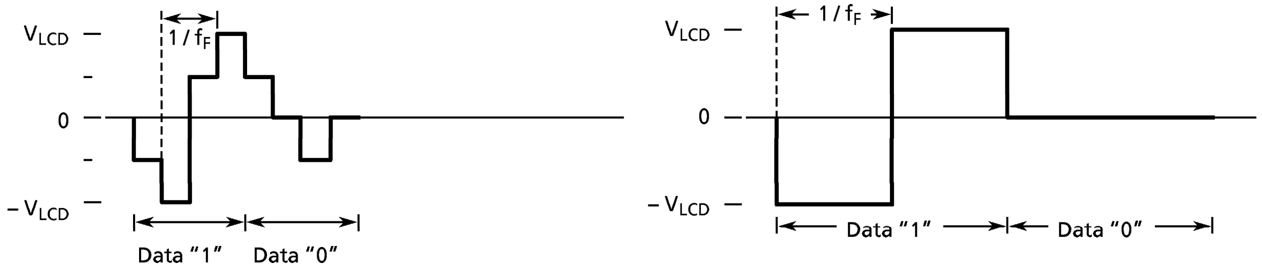
Figure 3-23. LCD Driver Control Command Register

(1) Driving methods of LCD

4 kinds of driving methods can be selected by DTY (bits 1 and 0 of command register). Figure 3-24 shows driving waveforms for LCD.



(a) 1 / 4 duty (1 / 3bias) drive (b) 1 / 3 duty (1 / 3 bias) drive



(c) 1 / 2 duty (1 / 2bias) drive (d) Static drive

Note. : f_F ; Frame frequency V_{LCD} ; LCD drive voltage (= V_{DD}–V_{LC})

Figure 3-24. Driving Waveform for LCD (Voltage between COM-SEG)

(2) Frame frequency

The frame frequency is set according to the driving method and base frequency as shown in Table 3-4. The base frequency is given by the Interval Timer.

Base Frequency[Hz] \ Driving Method	Frame Frequency [Hz]			
	1 / 4duty	1 / 3duty	1 / 2duty	static
$\frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$	$\frac{4}{3} \bullet \frac{fc}{2^{13}}$	$\frac{4}{2} \bullet \frac{fc}{2^{13}}$	$\frac{fc}{2^{13}}$
At fc = 960kHz	117	156	234	117
At fc = 480kHz	59	78	117	59
$\frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$	$\frac{4}{3} \bullet \frac{fs}{2^{8}}$	$\frac{4}{2} \bullet \frac{fs}{2^{8}}$	$\frac{fs}{2^{8}}$
At fs = 32.768kHz	128	170	256	128

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

Table 3-4. Frame Frequency Setting

(3) LCD drive voltage

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

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

During reset, the power switch of the LCD driver is turned off automatically, shutting off the VLC voltage. Both the segment output and common output become V_{DD} level at this time and the LCD turn off.

The power switch is turned on to supply VLC voltage to the LCD driver by setting EDSP (bits 2 and 3 of the command register) to "11_B". After that, the power switch will not turn off even during blanking (setting EDSP to "01_B") and the VLC voltage continues to flow.

3.5.3 LCD Display Operation

(1) Display data setting

Display data are stored to the display data area (Max. 32 words) in the data memory.

The display data stored to the display data area are read automatically and sent to the LCD driver by the hardware.

The LCD driver generates the segment signals and common signals in accordance with the display data and drive method. Thus, display patterns can be changed by merely overwriting the contents of the display data area with a program. The table look-up instruction is mainly used for this overwriting.

Figure 3-25 shows the correspondence between the display data area and the SEG/COM pins. The LCD light when the display data is "1" and turn off when "0".

The number of segment which can be driven differs depending on the LCD drive method; therefore, the number of display data area bits used to store the data also differs (Refer to Table 3-5). Consequently, data memory not used to store display data and data memory for which the addresses are not connected to LCD can be used to store ordinary user's processing data.

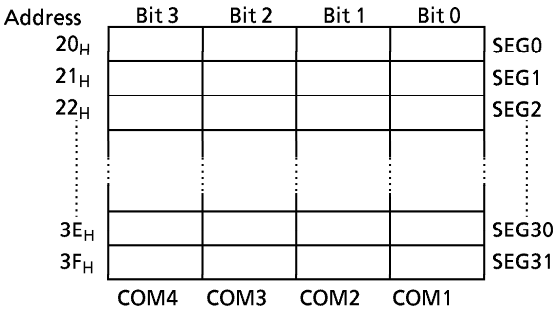


Figure 3-25. Display Data Area and SEG/COM

Driving Method	Bit 3	Bit 2	Bit 1	Bit 0
1 / 4 duty	COM4	COM3	COM2	COM1
1 / 3 duty	–	COM3	COM2	COM1
1 / 2 duty	–	–	COM2	COM1
Static	–	–	–	COM1

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

Table 3-5. Driving Method and Bit for Display Data

(2) Blanking

Blanking is applied by setting EDSP to "01_B" and turns off the LCD by outputting the non light operation level to the COM pin. The SEG pin continuously outputs the signal level in accordance with the display data and drive method.

At static drive, no voltage is applied between the COM and SEG pins when the LCD is turned off by data (display data cleared to "0"), but the COM pin output becomes constant at the $V_{LCD}/2$ level when turning off the LCD by blanking, so the COM and SEG pins are then driven by $V_{LCD}/2$.

3.5.4 Control Method of LCD Driver

(1) Initial Setting

Flow chart of initial setting is shown in Figure 3-26.

Example : Driving of 1 / 4duty LCD

```
LD      A, #0000B   ; Sets 1/4 duty drive
OUT     A, %OP1B
 ⋮                  ; Initializes display data area
 ⋮
LD      A, #1100B   ; Enable display (Release of blanking)
OUT     A, %OP1B
 ⋮
```

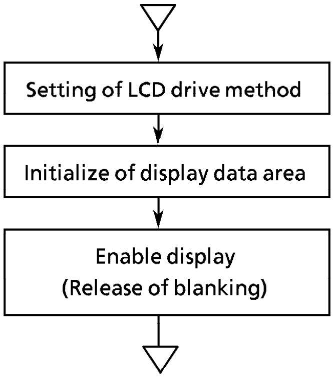


Figure 3-26. Initial Setting of LCD Driver

(2) Display Data

Normally, display data are kept permanently in the program memory and are then stored to the display data area by the table look-up instruction. This can be explained using numerical display with 1/4 duty LCD as an example. The COM and SEG connections to the LCD are the same as those shown in Figure 3-15 and the display data are as shown in Table 3-6.

Programming example for displaying numerals corresponding to BCD data stored at address 10_H in the data memory is shown below. The display data area is at addresses 20_H and 21_H.

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

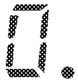
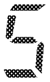
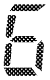
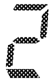
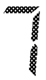
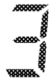
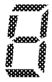
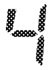
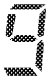
Numeral	Display	Display data		Numeral	Display	Display data	
		Upper	Lower			Upper	Lower
0		1101	1111	5		1011	0101
1		0000	0110	6		1111	0101
2		1110	0011	7		0001	0111
3		1010	0111	8		1111	0111
4		0011	0110	9		1011	0111

Table 3-6. Example of Display Data (1/4 Duty)

Table 3-7 shows the same numerical display used in Table 3-6, but using 1/2 duty LCD. The connections of the COM and SEG pins to the LCD are the same as those shown in Figure 3-18.

Programming example for displaying numerals corresponding to BCD data stored at address 10_H in the data memory is shown below. The display data area is at addresses 20 through 23_H.

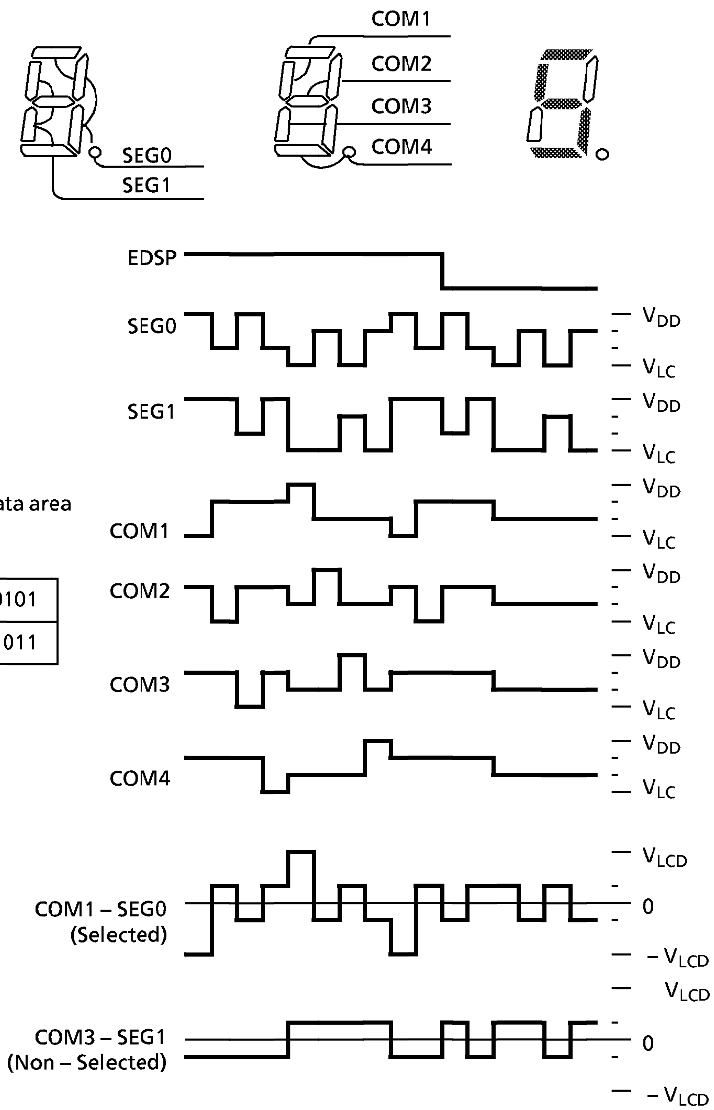
```
        LD      HL, 0FCH          ; Sets the data counter
        LD      A, 10H
        ST      A, @HL+
        ST      #DTBL/16, @HL+
        ST      #DTBL/256, @HL+
        LD      HL, #20H          ; Stores display data
        LDL     A, @DC
        ST      A, @HL+
        RORC    A
        RORC    A
        ST      A, @HL+
        LDH     A, @DC+
        ST      A, @HL+
        RORC    A
        RORC    A
        ST      A, @HL+
         ⋮
DTBL  : DATA    01110111B, 00100010B, 10010111B, 10100111B, 11100010B,
                11100101B, 11110101B, 01100011B, 11110111B, 11100111B
```

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

*Note. * ; don't care*

Table 3-7. Example of Display Data (1/2 Duty)

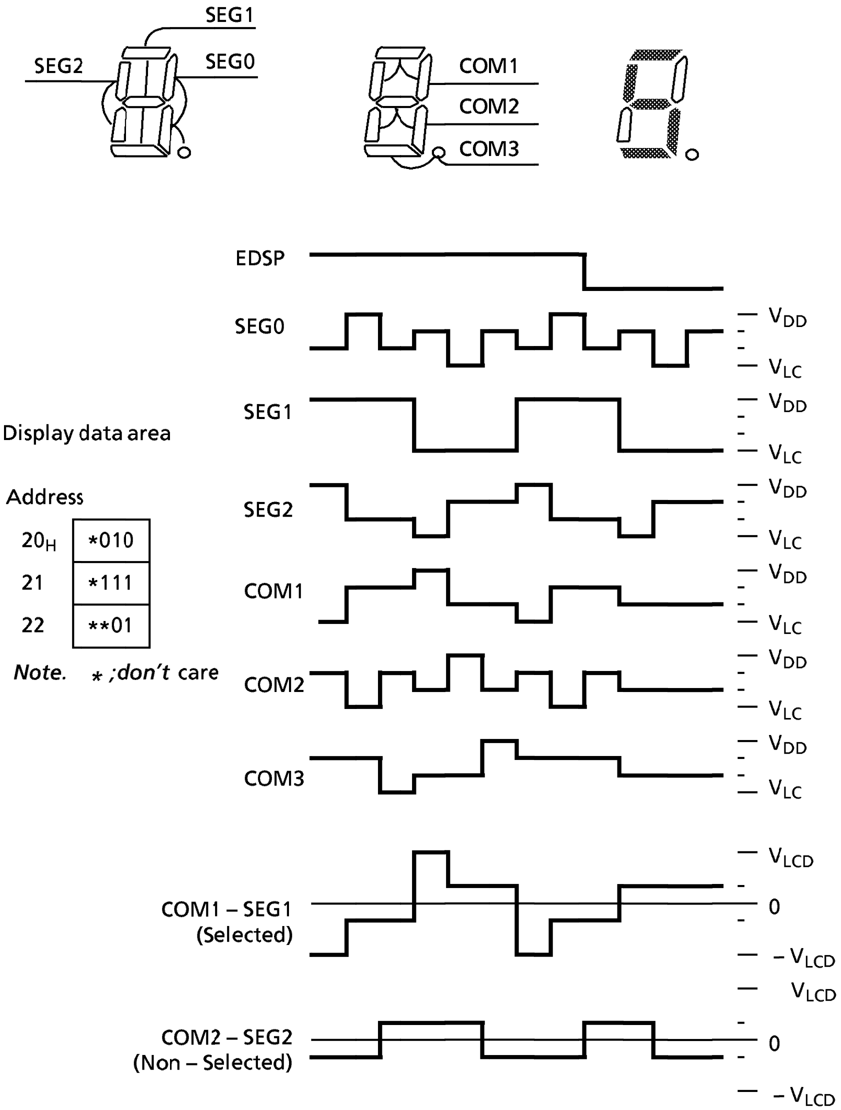
(3) Driving Example



Display data area

Address	
20_H	0101
21	1011

Figure 3-27. 1/4 Duty (1/3 Bias) Drive

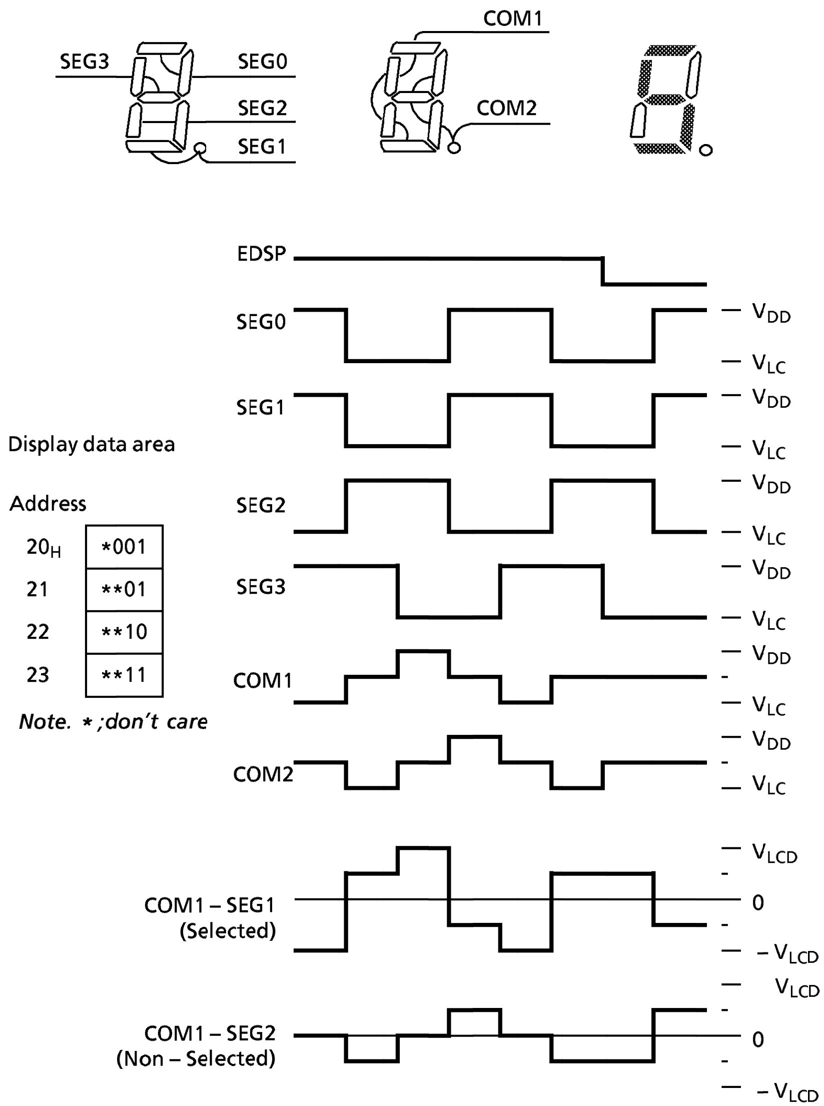


Display data area

Address	
20_H	*010
21	*111
22	**01

Note. * ;*don't* care

Figure 3-28. 1/3 Duty (1/3 Bias) Drive

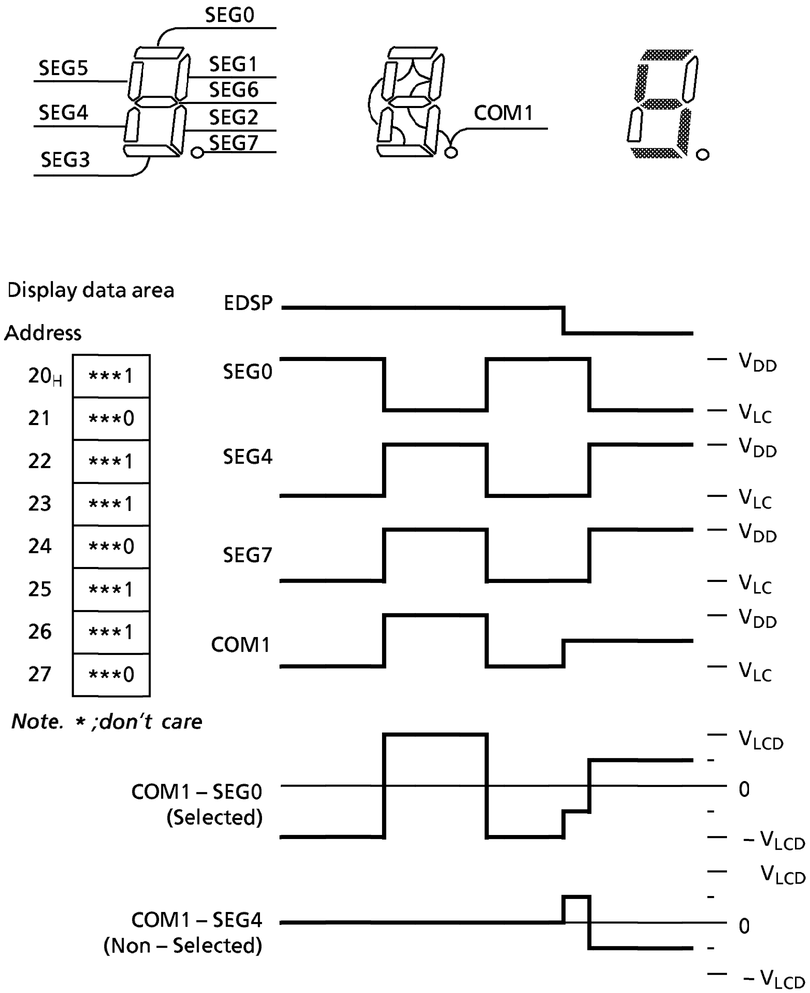


Display data area

Address	
20_H	*001
21	**01
22	**10
23	**11

*Note. * ;don't care*

Figure 3-29 . 1/2 Duty (1/2 Bias) Drive



Display data area

Address	
20_H	***1
21	***0
22	***1
23	***1
24	***0
25	***1
26	***1
27	***0

*Note. * ;don't care*

Figure 3-30. Static Drive

3.6 DTMF Generator

The 47C655/855 have a DTMF (Dual Tone Multi Frequency) 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.

3.6.1 Selection of Input Clock for DTMF Generator

DTMF generator of the 47C655/855 are based on 480kHz clock. Therefore, when high-frequency clock is 960kHz, fc / 2 must be applied to DTMF generator. And, when fc is 480kHz, fc must fc applied to one.
DTMF input clock control command register controls switching of input clock for DTMF generator. This command register must be set at the beginning of program. This command register is initialized to "0" during reset.

DTMF input clock control command register (Port address OP17)

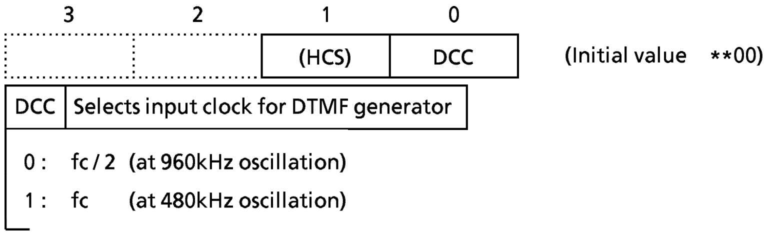


Figure 3-31. DTMF Input Clock Control Command Register

3.6.2 Configuration of DTMF Generator

Figure 3-32 shows configuration of the DTMF generator. The 47C655/855 generate two stepped, quasi sine waves for tone dial signals which can be combined and output. The high or low group of frequencies is selected by setting frequency selection codes into the ROW and COLUMN registers.

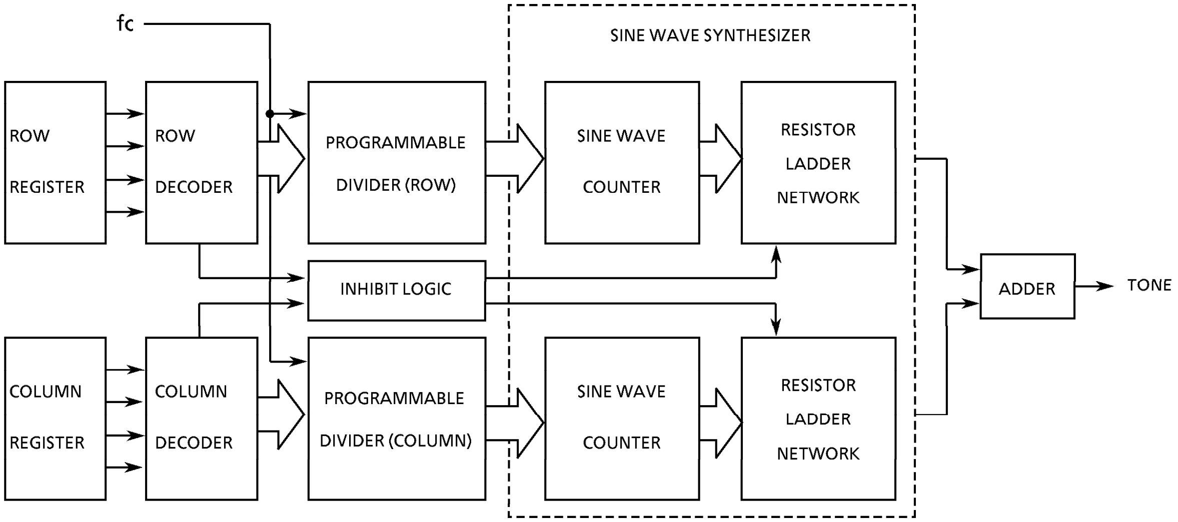
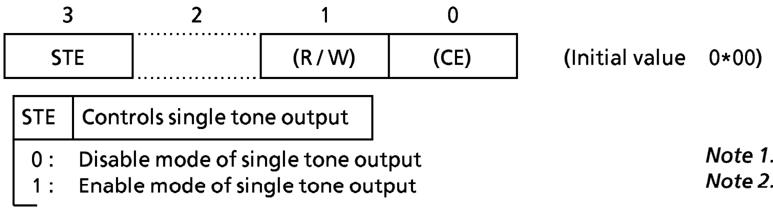


Figure 3-32. Configuration of DTMF Generator

3.6.3 Control of DTMF Generator

Tone output is controlled by ROW register (OP01/IP01) and COLUMN register (OP02/IP02). And single tone is controlled by TONE command register (OP0D/IP0D). ROW register, COLUMN register and TONE command register are initialized to "0" during reset.

TONE command register (Port address OP0D/IP0D)



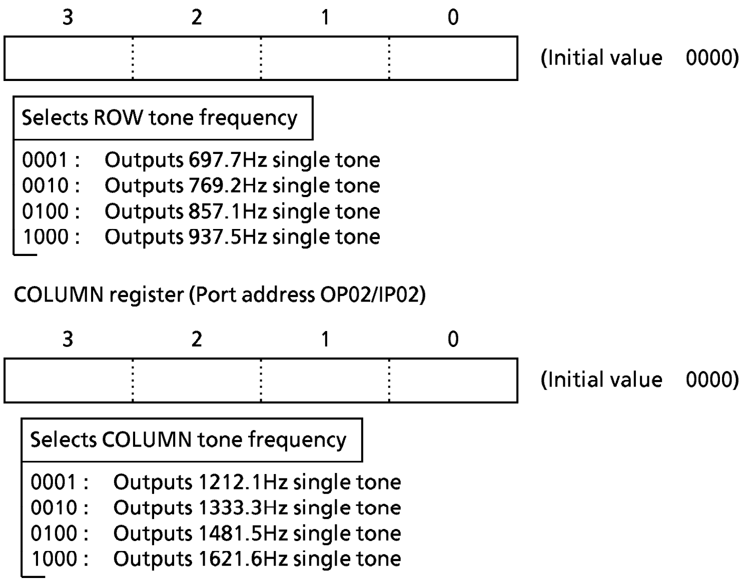
STE Controls single tone output

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

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

Figure 3-33. TONE Command Register

ROW register (Port address OP01/IP01)



Selects ROW tone frequency

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)

Selects COLUMN tone frequency

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 3-34. ROW, COLUMN Register

Tones are outputted by loading the frequency selection codes shown in Figure 3-34 into the ROW and COLUMN registers. In the enable mode of single tone output and either ROW or COLUMN register is disabled, another register remains to be enabled, and so single tone can be outputted, by loading an ineffective code into the register. When both the registers are enabled, dual tone can be outputted. In the disable mode of single tone output, effective codes are loaded into both ROW and COLUMN registers and then dual tone can be outputted. At this time, an ineffective code is loaded into ROW or COLUMN register and then the 47C655/855 have no tone output signal.
The [OUTB @HL] instruction can set 8-bit data into both registers (the upper 4 bits of the ROM data go to the COLUMN register and the lower 4 bits go to the ROW register) at the same time, and DTMF signal is outputted without single tone output.

Example 1: To output 1481. 5Hz single tone

```
OUT     #8, %OP0D     ; Sets the enable mode of single tone output
OUT     #0, %OP01     ; Sets an ineffective code into ROW register
OUT     #4, %OP02     ; Sets data "4" into COLUMN register
```

Example 2: 8 bits data corresponding to the 5 bits of data linking the content of carry flag and the contents of data memory RAM1 address 90_H are read from the ROM, frequency selection codes are loaded into ROW and COLUMN registers, and dual tone is outputted.

```
LD      HL, #90H      ; HL←90H (Sets the address of the data memory)
OUTB    @HL           ; Sets the ROM data into the ROW and COLUMN
                        register
```

Table 3-8 shows the corresponding frequency selection codes of the ROW and COLUMN registers for the telephone dial keys. Table 3-9 shows the deviation between the 47C655/855 tone output frequency and standard frequency.

		COLUMN register (OP02 / IP02)		
	Frequency selection code	0001 (1209)	0010 (1336)	0100 (1477)
ROW register (OP01 / IP01)	0001 (697)	1	2	3
	0010 (770)	4	5	6
	0100 (852)	7	8	9
	1000 (941)	*	0	#
		Standard telephone dial key		

Contents of ( ) are standard frequencies, unit: Hz

Table 3-8. Corresponding Frequency Selection Codes of the ROW and COLUMN Registers for the Telephone Dial Keys

ROW Tone						
Frequency selection code				Tone output frequency [Hz]	Standard frequency [Hz]	Deviation [%]
3	2	1	0			
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						
Frequency selection code				Tone output frequency [Hz]	Standard frequency [Hz]	Deviation [%]
3	2	1	0			
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 3-9. Tone Output Frequencies and Deviation from Standard

3.6.4 Test Mode for Tone Output

The 47C655/855 include a test mode for checking tone output waveforms. Tones can be outputted by the circuit shown in Figure 3-35. ROW data are inputted from the port R6 and COLUMN data are inputted from the port R3, and any desired single or dual tones can be outputted by setting the frequency selection codes shown in Figure 3-34. Figure 3-36 shows a single tone waveform and Figure 3-37 shows a dual tone waveform.

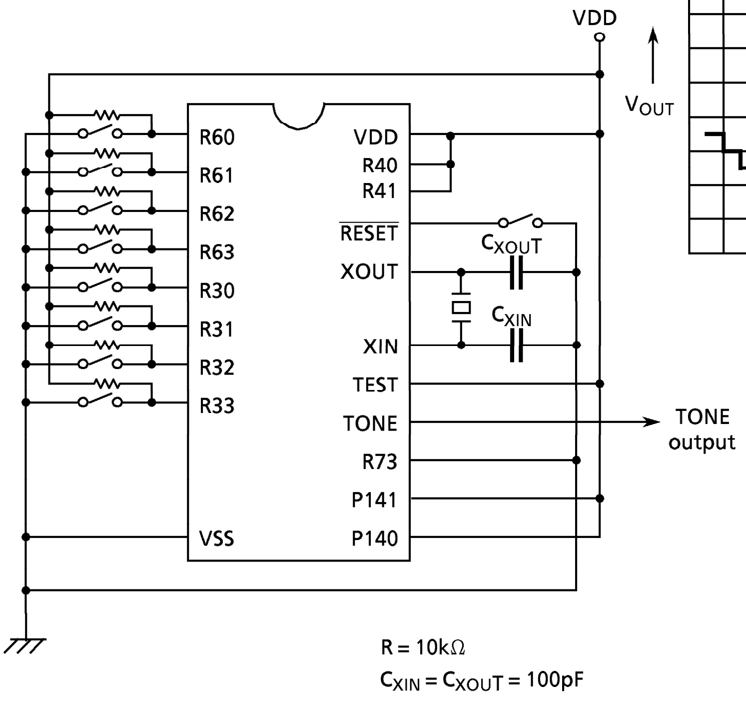


Figure 3-35. Tone Test Circuit

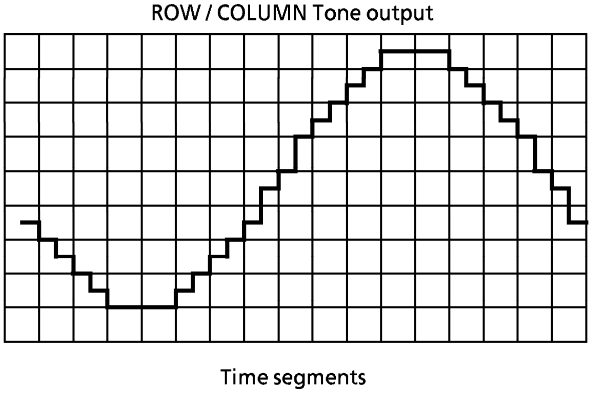


Figure 3-36. Single Tone Waveform

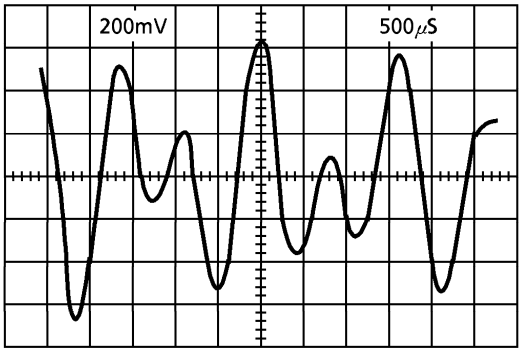


Figure 3-37. Dual Tone Waveform

3.7 BEEP Output Circuit

BEEP output circuit generates square wave in the audible frequency range. This circuit can drive the key input confirmation tone generator circuit for telephone applications.

BEEP output is from the P140 (BEEP) pin. This pin is for both P140 output and BEEP output. Set the P140 output latch to "1" for BEEP output. BEEP function can be used in Normal-1 and Normal-2 mode.

3.7.1 BEEP Output Circuit Configuration

Figure 3-38 shows the BEEP output circuit configuration. The clock pulse of BEEP output circuit is supplied by an interval timer. BEEP output is controlled by frequency selection and output enable/disable setting.

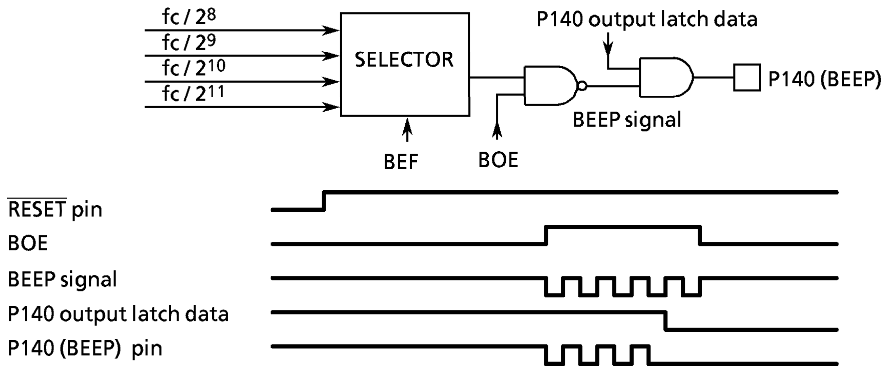


Figure 3-38. BEEP Output Circuit Configuration and Timing Chart

3.7.2 Control of BEEP Output

BEEP output is controlled by the command register (OP13) .

BEEP Output Control command register (Port address OP13)

3	2	1	0
BOE		BEF	

(Initial value 0*00)

BOE	Controls BEEP output
0 :	Disable output
1 :	Enable output

BEF	Selects BEEP output frequency
00 :	$fc / 2^8$ [Hz]
01 :	$fc / 2^9$
10 :	$fc / 2^{10}$
11 :	$fc / 2^{11}$

Figure 3-39. BEEP Output Control Command Register

3.8 Serial Interface (SIO)

The 47C655/855 have a serial interface with an 8-bit buffer. 4-bit/8-bit tramsfer mode can be selected. In the 8-bit transfer mode, data may be transmitted and received simultaneously. The serial interface is connected to the exterenal device via 3 pins (the serial port): R92 ($\overline{\text{SCK}}$), R91 (SO), and R90 (SI). The serial port is shared by port R9. For the serial port, the output latch of port R9 must be set to "1". In the transmit mode, R90 pin provides the I/O port; in the receive mode, R91 pin provides the I/O port.

3.8.1 Configuration of Serial Interface

Figure 3-40. shows configuration of serial interface.

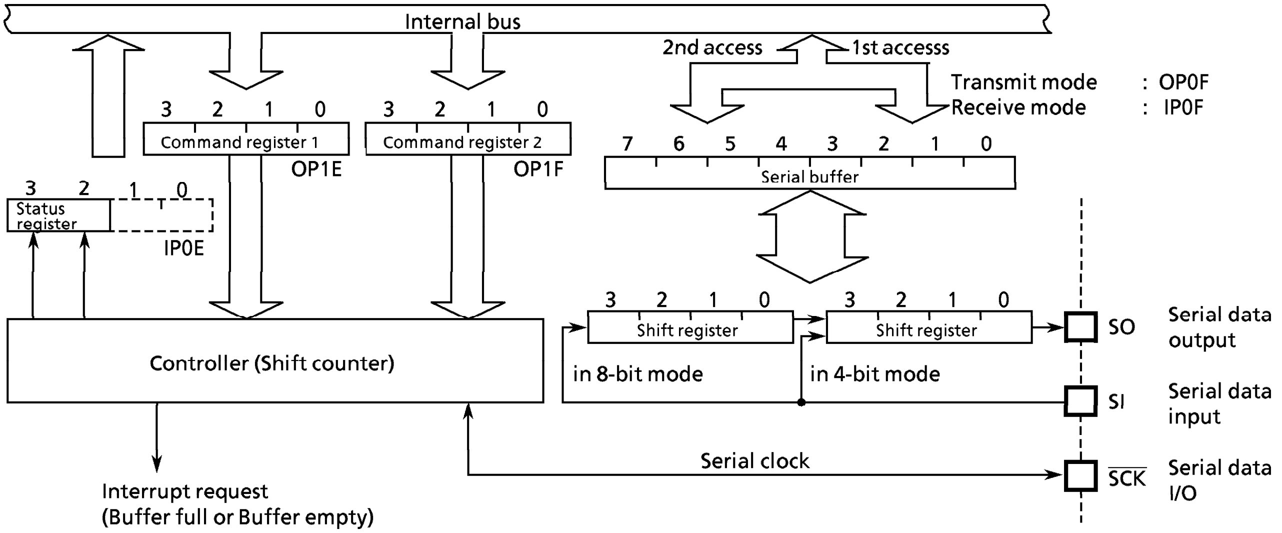


Figure 3-40. Configuration of Serial Interface

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 3-41. 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 $\overline{\text{SCK}}$ pin)
1	: External clock (input from $\overline{\text{SCK}}$ pin)

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

Figure 3-42. 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 $\overline{\text{SCK}}$ pin. Note that the start of transfer, the $\overline{\text{SCK}}$ 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/second (at fc = 960kHz).

② External clock

The signal obtained by the clock supplied to the $\overline{\text{SCK}}$ pin from the outside is used for the serial clock. In this case, the output latch of R92 ($\overline{\text{SCK}}$) 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 $\overline{\text{SCK}}$ pin input) of the serial clock.

② Trailing edge

Data is shifted at the trailing edge (the rising edge of $\overline{\text{SCK}}$ 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

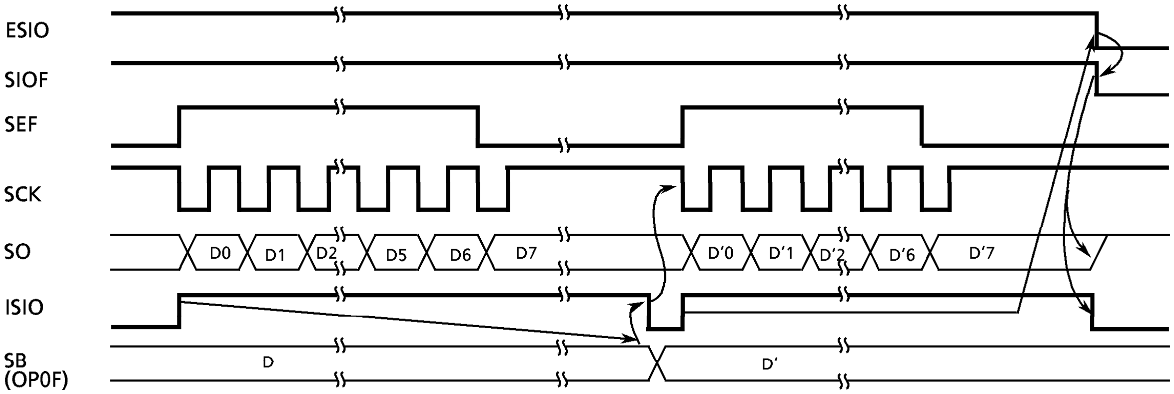
The transmit mode is set to the command register than writes the first transmit data (4 bits or 8 bits) is written to the buffer register (OP0F). (If the transmit mode is not set, the data is not written to the buffer register). In the 8-bit transfer mode, the 8-bit data is wirtten by accessing the buffer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

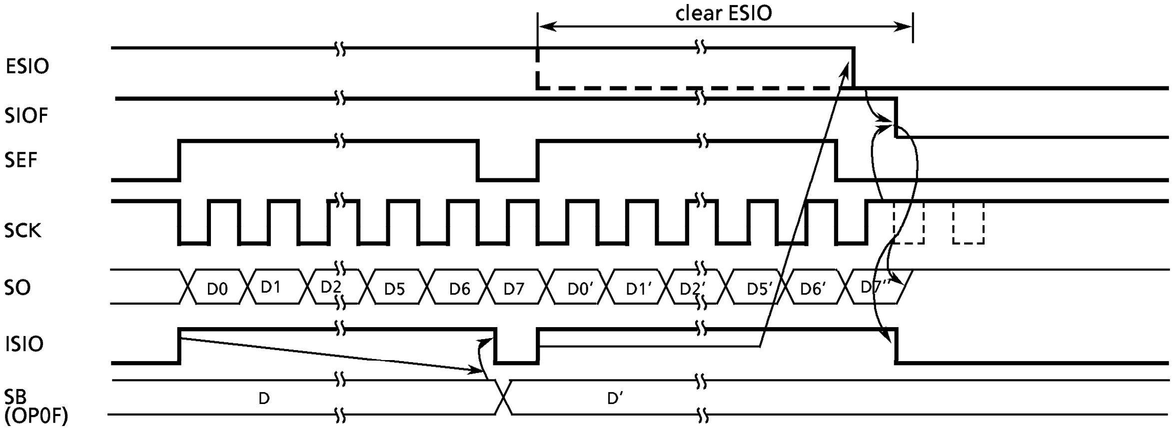
To end transmission, ESIO is cleared to "0" instead of writing the next transmit data by the buffer empty interrupt service program. When ESIO is cleared,transmission stops upon termination of the currently shifted-out data. The transmission end can be known by the SIOF state (SIOF goes "0" upon transmission 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 ← 0101B   (Sets the 8-bit serial transfer)
OUT     A,   %OP1E
LD      A,   #0010B   ; OP1F ← 0010B   (Sets the transmit mode)
OUT     A,   %OP1F
OUT     @HL,  %OP0F   ; OP0F ← RAM [HL] (Writes the lower 4-bit data)
INC     L
OUT     @HL,  %OP0F   ; OP0F ← RAM [HL] (Writes the upper 4-bit data)
LD      A,   #1010B   ; ESIO ← 1 (Instructs transmission start)
OUT     A,   %OP1F
```



(a) Internal-clock-based operation with wait



(b) External-clock-based operation

Figure 3-43. 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).

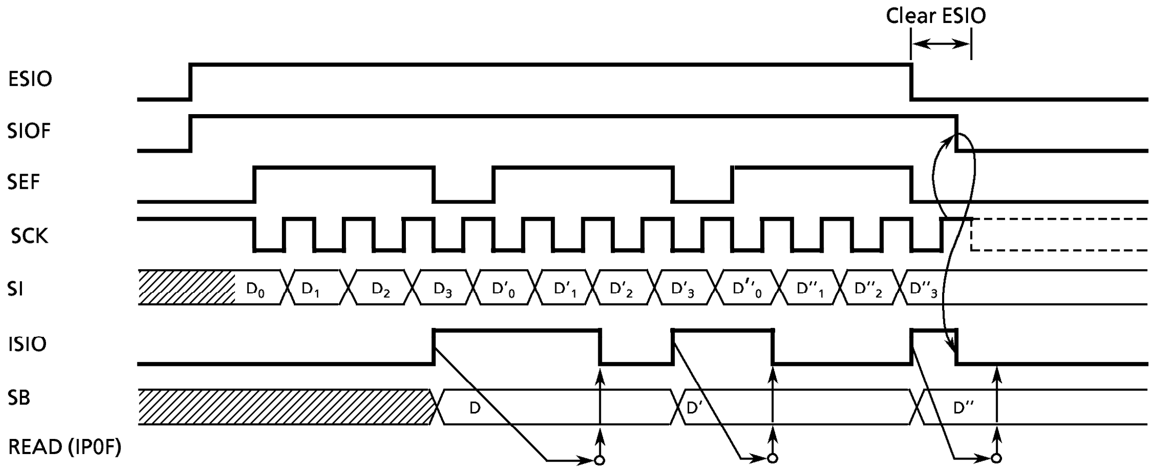
```
SSEF0 :   TEST    %IP0E,  2      ; Waits until SEF = "1"
          B       SSEF0
          LD      A,  #0110B     ; ESIO ← 0
          OUT     A,  %OP1F
          IN      %IP0F,  A      ; Acc ← IP0F (Reads received data)
```

③ One-word reception

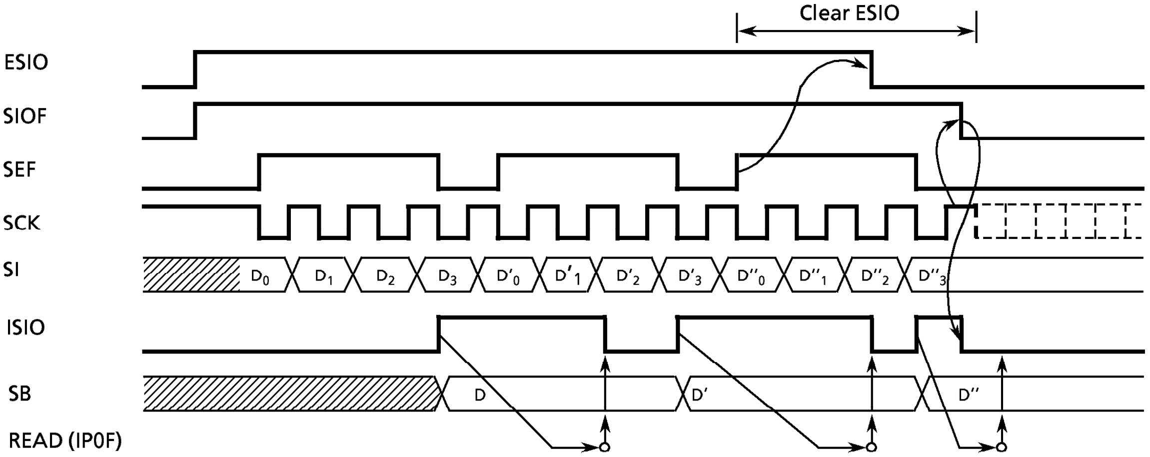
When receiving only 1 word, ESIO is set to "1" then it is cleared to "0" after confirming that SEF has gone "1". In this case, buffer full interrupt is caused only once, so that the received data is read by the interrupt service program.

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)

Figure3-44. 4-bit Receive Mode

- 8-bit Transmit/Receive Mode

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

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

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

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

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

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

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

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

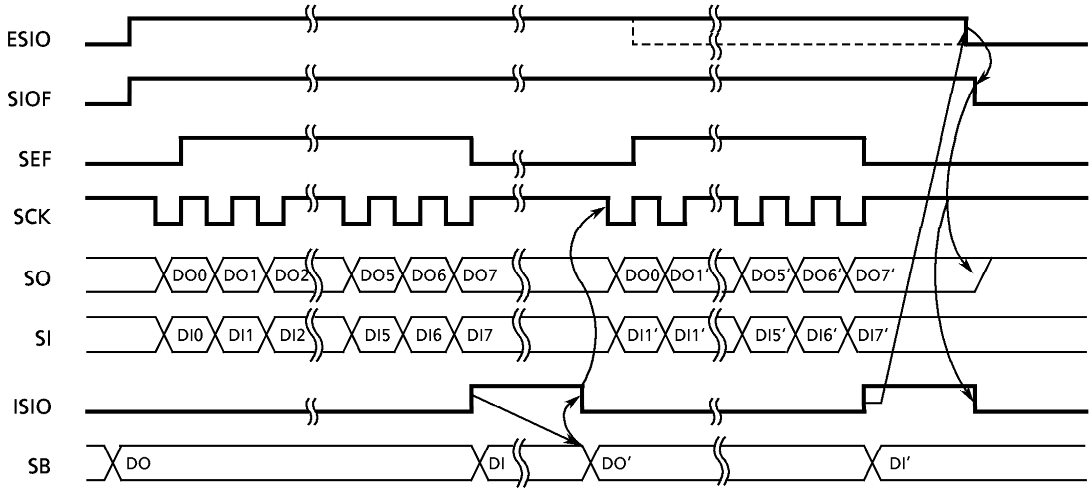
Example 2 : To read data received and to write next data to be transmitted.

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

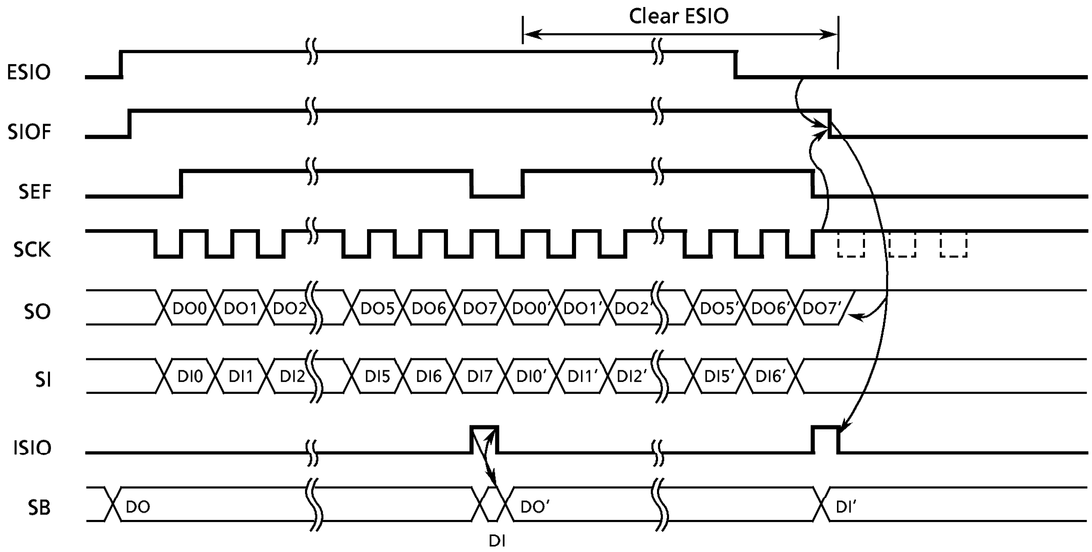
(4) Stopping serial transfer

A serial transfer operation can be stopped forcibly.

It is stopped by setting INH (bit 3 of command register 1) to "1", clearing the shift counter. When the serial transfer is over, INH is automatically cleared to "0" with no other bits of command register affected. In the transmit mode of this case, $\overline{\text{SCK}}$ 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



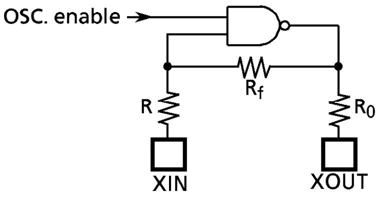
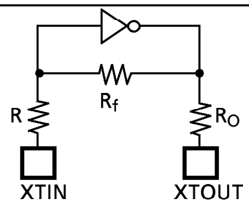
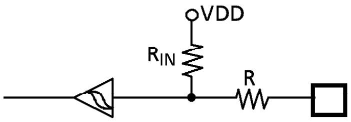
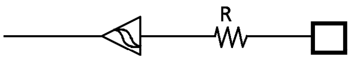
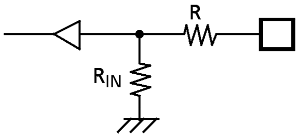
(b) External clock based operation

Figure3-45. 8-bit Transmit/Receive Mode

INPUT/OUTPUT CIRCUITRY

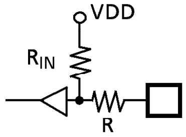
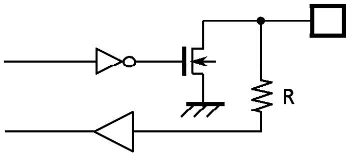
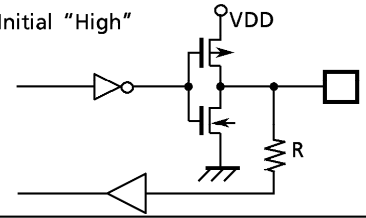
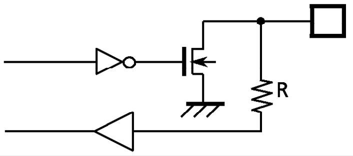
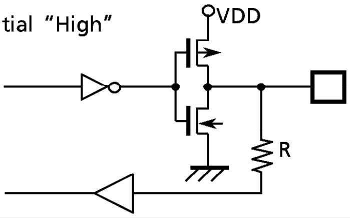
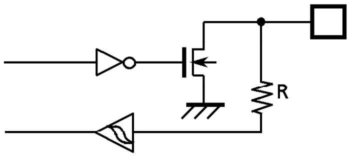
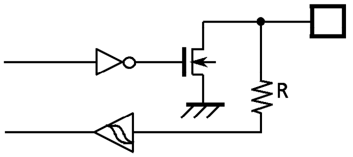
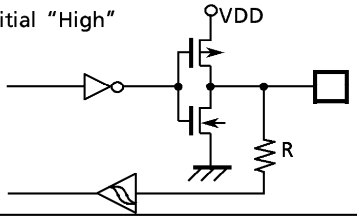
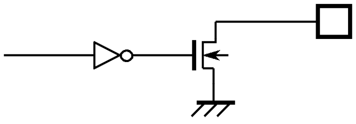
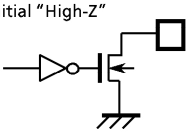
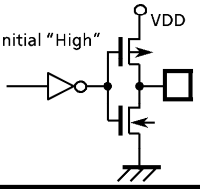
(1) Control pins

The input/output circuitries of the 47C655/855 control pins are shown below.

CONTROL PIN	I/O	CIRCUITRY	REMARKS
XIN XOUT	Input Output		Resonator connecting pins R = 1kΩ (typ.) R_f = 1.5MΩ (typ.) R_0 = 2kΩ (typ.)
XTIN XTOUT	INPUT OUTPUT		Resonator connecting pins (Low frequency) R = 1kΩ (typ.) R_f = 6MΩ (typ.) R_O = 200kΩ (typ.)
RESET	Input		Hysteresis input Pull-up resistor R_{IN} = 220kΩ (typ.) R = 1kΩ (typ.)
$\overline{\text{HOLD}}$ ($\overline{\text{KE0}}$)	Input (Input)		Hysterisis input (Sense input) R = 1kΩ (typ.)
TEST	Input		Pull-down resistor R_{IN} = 70kΩ (typ.) R = 1kΩ (typ.)

(2) I/O Ports

The input/output circuitries of the 47C655/855 I/O ports are shown below, any one of the circuitries can be chosen by a code (WB, WE, WH) as a mask option.

PORT	I/O	INPUT/OUTPUT CIRCUITRY and CODE		REMARKS
K0	Input			Pull-up resistor R_{IN} = 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 Initial "High" 	Sink open drain or push-pull output R = 1kΩ (typ.)
R8	I/O	Initial "Hi-Z" 		Sink open drain 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 P142, P143 Initial "High-Z"  P140, P141 Initial "High" 	Sink open drain or push-pull output

ELECTRICAL CHARACTERISTICS

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

PARAMETER	SYMBOL	PINS	RATING	UNIT
Supply Voltage	V_{DD}		– 0.3 to 7	V
Supply Voltage (LCD drive)	V_{LC}		– 0.3 to V_{DD} + 0.3	V
Input Voltage	V_{IN}		– 0.3 to V_{DD} + 0.3	V
Output Voltage	V_{OUT1}	Except sink open drain pin, but include R7, P142, P143	– 0.3 to V_{DD} + 0.3	V
	V_{OUT2}	Sink open drain pin except R7, P142, P143	– 0.3 to 10	
Output Current (per 1 pin)	I_{OUT}		3.2	mA
Power Dissipation [T_{opr} = 60℃]	PD		600	mW
Soldering Temperature (time)	T_{sld}		260 (10 s)	℃
Storage Temperature	T_{stg}		– 55 to 125	℃
Operating Temperature	T_{opr}		– 30 ~ 60	℃

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0V, T_{opr} = – 30 to 60℃)

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

D.C. CHARACTERISTICS (V_{SS} = 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 $\overline{\text{RESET}}$	V_{DD} = 5.5V, V_{IN} = 5.5V / 0V	–	–	± 2	μA
	I_{IN2}	Ports R (open drain)					
Low Level Input Current	I_{IL}	Ports R (push-pull)	V_{DD} = 5.5V, V_{IN} = 0.4V	–	–	– 2	mA
Input Resistance	R_{IN1}	Port K0 with pull-up/pull-down		30	70	150	kΩ
	R_{IN2}	$\overline{\text{RESET}}$		100	220	450	
Output Leakage Current	I_{LO}	Ports R (open drain)	V_{DD} = 5.5V, V_{OUT} = 5.5V	–	–	2	μA
Output Level High Voltage	V_{OH}	Ports R (push-pull)	V_{DD} = 4.5V, I_{OH} = – 200μA	2.4	–	–	V
Output Level Low Voltage	V_{OL2}	Except XOUT	V_{DD} = 4.5V, I_{OL} = 1.6mA	–	–	0.4	V
Segment Output Resistance	R_{OS}	SEG pin	V_{DD} = 5V, $V_{DD} - V_{LC}$ = 3V	–	20	–	kΩ
Common Output Resistance	R_{OC}	COM pin					
Segment/Common Output Voltage	$V_{O2/3}$	SEG / COM pin		3.8	4.0	4.2	V
	$V_{O1/2}$			3.3	3.5	3.7	
	$V_{O1/3}$			2.8	3.0	3.2	
Supply Current (in the Nomal mode)	I_{DD}		V_{DD} = 5.5V, V_{LC} = V_{SS} fc = 960kHz	–	0.8	1.5	mA
	I_{DDT}		V_{DD} = 5.5V, V_{LC} = V_{SS} fc = 960kHz When tone is oscillating	–	2.5	4.0	
Supply Current (in the SLOW mode)	I_{DDS}		V_{DD} = 3V, V_{LC} = V_{SS} 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 shows those at T_{opr} = 25°C, V_{DD} = 5V.

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

Note 3. Output Resistance Ros, Roc : Shows on-resistance at the level switching.

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

Note 5. Supply Current I_{DD} : V_{IN} = 5.3V/0.2V
The Port K0 is open when the input resistor is contained.
The voltage applied to the Port R is within the valid range.

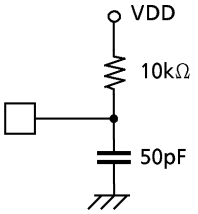
Note 6. Supply Current I_{DDS} : V_{IN} = 2.8V/0.2V. Only low frequency clock is only osillated (connecting XTIN, XTOUT).

A. C. CHARACTERISTICS

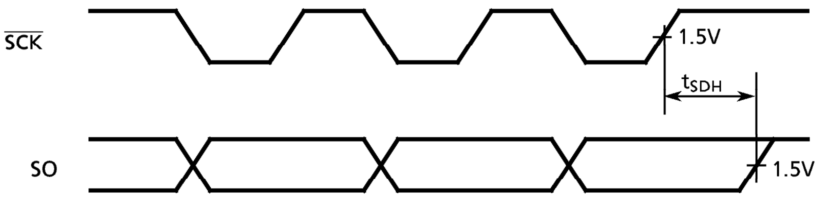
($V_{SS} = 0V$, $V_{DD} = 2.2$ to $6.0V$, $T_{opr} = -30$ to $60°C$)

PARAMETER	SYMBOL	CONDITIONS	Min.	Typ.	Max.	UNIT
Instruction Cycle Time	t_{cy}	In the Normal mode	8.3 / 16.7			μs
		In the SLOW mode	235	–	267	μs
High level Clock pulse Width	t_{WCH}	External clock mode	80	–	–	ns
Low level Clock pulse Width	t_{WCL}					
Shift data Hold Time	t_{SDH}		$0.5t_{cy} - 300$	–	–	ns

Note. Shift data Hold Time :
External circuit for $\overline{SCK}$ pin and SO pin.



Serial port (completion of transmission)



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≧10kΩ, $V_{DD} = 2.2V$	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	Δf	Except error of osc. frequency	–	–	0.7	%

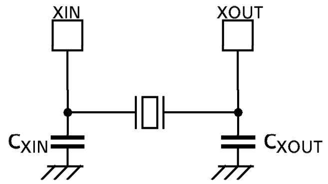
RECOMMENDED OSCILLATING CONDITIONS

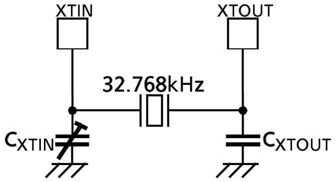
($V_{SS} = 0V$, $V_{DD} = 2.2$ to $6.0V$, $T_{opr} = -30$ to $60°C$)

(1) 960kHz
Ceramic Resonator
CSB960J916 (MURATA) $C_{XIN} = C_{XOUT} = 100pF$

(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.