十速

TM8720

4-Bit Microcontroller with LCD Driver

Rev 1.4

AMENDMENT HISTORY

Version	Date	Description
V1.0	May, 2005	New release
V1.1	Aug, 2005	Modify IOA, B pull low resistor SPEC range to 100 ~ 1000 Kohm.
V1.2	Sept, 2005	Modify the voltage and operating condition in Segment Driver Output Characteristics Section.
V1.3	Dec, 2011	Add Ordering Information table
V1.4	Apr, 2022	Modify bit6 of SCC machine code to “0”

CONTENTS

Chapter 1 General Description

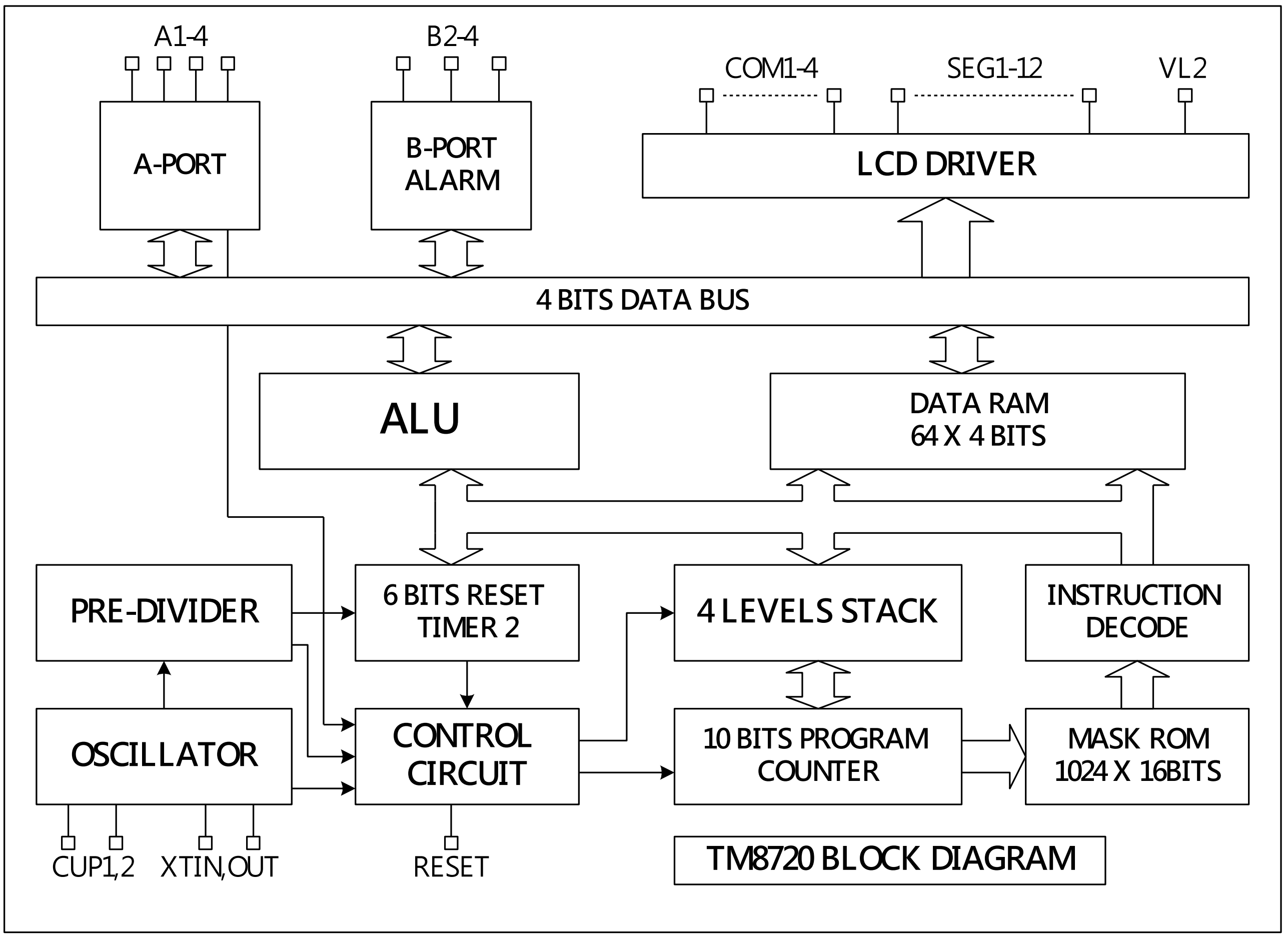
1-1. GENERAL DESCRIPTION

The TM8720 is an embedded high-performance 4-bit microcomputer with LCD driver. It contains all the necessary functions, such as 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation and LCD driver in a single chip.

1-2. FEATURES

1. 1.5V operation only and with low power dissipation.
2. Powerful instruction set (86 instructions).
   - Binary additions, subtraction, BCD conversion, logical operation in direct addressing mode.
   - Single-bit manipulation (set, reset, decision for branch).
   - Various conditional branches.
   - 16 working registers and manipulation.
   - LCD driver data transfer.
3. Memory capacity.
   - ROM capacity 1024 x 16 bits.
   - RAM capacity 64 x 4 bits. (Only 40h~7Fh are addressable)
4. LCD driver output.
   - 4 common outputs and 12 segment outputs (48 LCD pixels).
   - 1/4 Duty and 1/2 Bias for LCD display.
   - Single instruction to turn off all segments.
5. Input/output ports.
   - Port IOA 4 pins (with internal pull-low), IOA port had a built-in signal chattering prevention circuitry.
   - Port IOB 3 pins (with internal pull-low) & mask option with IOB3 (BZB) and IOB4 (BZ).
6. 4 level subroutine nesting.
7. Interrupt function.
   - External factors 1 (Port IOA).
   - Internal factors 2 (Pre-Divider, Timer2).
8. Built-in Alarm, clock or single tone melody generator.
   - BZB, BZ (Muxed with IOB3, 4).
9. One 6-bit programmable timer with programmable clock source.
10. Built-in Voltage doubler, charge pump circuit.
11. HALT function.
12. STOP function.

1-3. BLOCK DIAGRAM



1-4. PIN DESCRIPTIONS

Name	I/O	Description
VBAT	P	Positive power supply. Connect a 0.1uF capacitor to GND.
VL2	P	LCD supply voltage. Connect a 0.1uF capacitor to GND.
RESET	I	Input pin for chip reset request signal, with internal pull-down resistor.
TEST	I	Test signal input pin.
CUP1,2	O	Switching pins for supply the LCD driving voltage. Connect the CUP1 and CUP2 pins with a 0.1uf non-polarized electrolytic capacitor for LCD mode.
COM1~4	O	Output pins for driving the common pins of the LCD.
SEG1~12	O	Output pins for driving the LCD panel segment or output.
IOA1~4	I/O	Input/Output port A.
IOB2~4	I/O	Input/Output port B.
BZB, BZ	O	Output port for alarm (MUXed with IOB3, 4).
XIN XOUT	I O	System clock oscillation. Connected with 32KHz crystal oscillator or internal R or external R by mask option.
GND	P	Negative supply voltage.

1-5. CHARACTERISTIC

ABSOLOUTE MAXIMUM RATINGS

at Ta=0 to 70℃,GND= 0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	V_{BAT}	-0.3 to 2.0	V
	VL2	-0.3 to 4.0	V
Maximum Input Voltage	V_{in}	-0.3 to V_{BAT}+0.3	V
Maximum output Voltage	V_{out1}	-0.3 to V_{BAT}+0.3	V
	V_{out2}	-0.3 to VL2 +0.3	V
Maximum Operating Temperature	T_{opg}	0 to +70	℃
Maximum Storage Temperature	T_{stg}	-25 to +125	℃

ALLOWABLE OPERATING CONDITIONS

at Ta=0 to 70℃, GND= 0V

Name	Symbol	Condition	Min.	Typ.	Max.	Unit
Supply Voltage	V_{BAT}		1.2	1.5	1.8	V
	VL2		2 x V_{BAT} x 0.9		2 x V_{BAT} + 0.1	V
Stand-by current	I_{sb}	STOP mode	-	-	1	uA
Input "H" Voltage	V_{ih1}	IOA and IOB port in input mode	V_{BAT} - 0.7	-	V_{BAT} + 0.7	V
Input "L" Voltage	V_{il1}		-0.7	-	0.7	V

ELECTRICAL CHARACTERISTICS INTERNAL RC FREQUENCY RANGE

(V_{BAT}=1.5V)

Option Mode	Min.	Typical	Max.
600KHz	500KHz	600KHz	700KHz

Input Resistance

(V_{BAT}=1.5V)

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
IOA1~4, IOB2~4 Pull-Down Tr.	R_{mad1}	Vi = V_{BAT}	100	300	1000	Kohm
RES Pull-Down R	R_{res1}	Vi = GND or V_{BAT}	10	50	100	Kohm

DC Output Characteristics

(V_{BAT}=1.2V)

Name	Symbol	Condition	Port	Min.	Typ.	Max.	Unit
Output ”H” Voltage	V_{oh2c}	I_{oh} = -200uA	IOA1~4, IOB2, IOB3,4/ BZB, BZ	0.8	0.9	1.0	V
Output ”L” Voltage	V_{ol2c}	I_{ol} = 400uA		0.2	0.3	0.4	V

Segment Driver Output Characteristics

(V_{BAT} =1.2V)

Name	Symbol	Condition	For	Min.	Typ.	Max.	Unit.
1/2 Bias Display Mode							
Output ”H” Voltage	V_{oh12f}	I_{oh} = -1uA	SEG-n	2.2	2.4	---	V
Output ”L” Voltage	V_{ol12f}	I_{ol} = 1uA		---	0.0	0.2	V
Output ”H” Voltage	V_{oh12g}	I_{oh} = -10uA	COM-n	2.2	2.4	---	V
Output ”M” Voltage	V_{om12g}	$I_{ol/h}$ = +/-10uA		1.0	1.2	---	V
Output ”L” Voltage	V_{ol12g}	I_{ol} = 10uA		---	0.0	0.2	V

Analog Circuit Characteristics

(V_{BAT} =1.5V GND=0V, X’tal fosc= 32.768KHz, Ta=25℃, Always in operation mode.)

Name	Symbol	Condition	Min.	Typ.	Max.	Unit
Internal Voltage	VL2	Connects a 1MΩ load resistance between GND and VL2 (No panel load)	2x V_{BAT} x0.9	3.0	2x V_{BAT} +0.1	V

POWER CONSUMPTION

at Ta=-20℃ to 70℃,GND= 0V, V_{BAT}=1.5V

Name	Symbol	Condition	Min.	Typ.	Max.	Unit
HALT mode	I_{HALT}	Only 32.768KHz Crystal oscillator operating, without loading.		2		uA
STOP mode	I_{STOP}				1	uA

Note : When RC oscillator function is operating, the current consumption will depend on the frequency of oscillation.

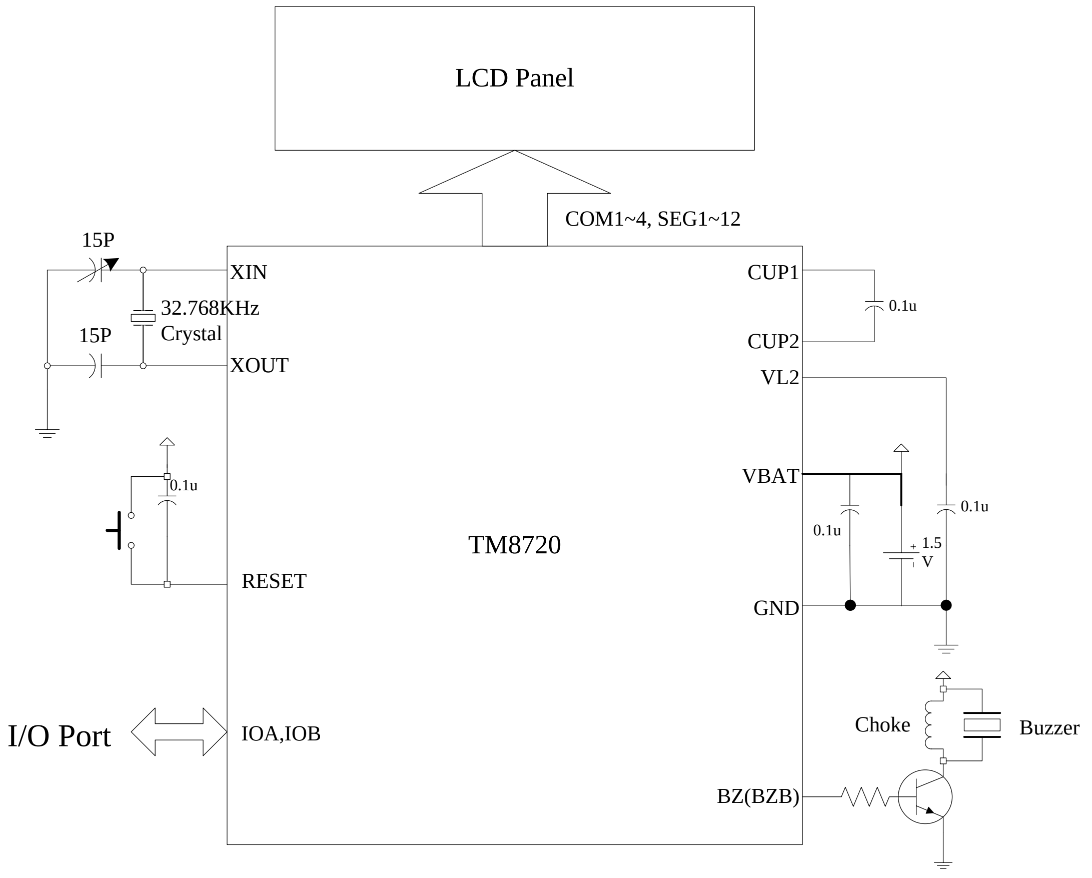
ALLOWABLE OPERATING FREQUENCY

at Ta=-20℃ to 70℃,GND= 0V

Condition	Max, Operating Frequency
V_{BAT} =1.5V	700KHz

1-6. Typical Application Circuit

This application circuit is simply an example, and is not guaranteed to work.



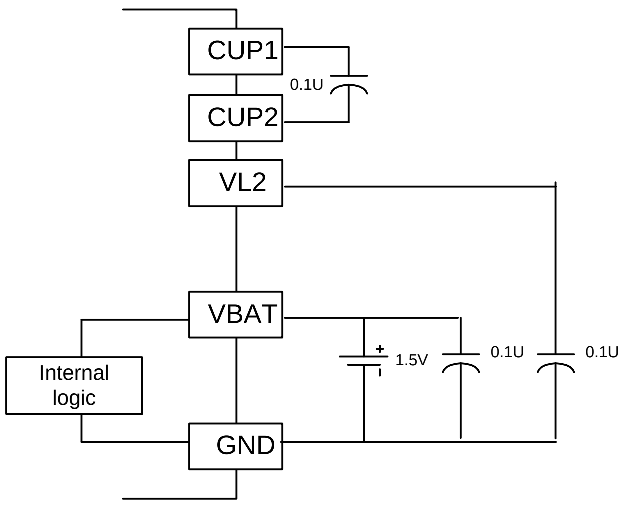
Ag power mode, 1/2 Bias, 1/4 Duty

Chapter 2 TM8720 Internal System Architecture

2-1. Power Supply

TM8720 could operate at Ag type (1.5V) voltage supply. The power supply circuitry also generates the necessary voltage level to drive the LCD panel with different bias. Shown below are the connection diagrams for 1/2 bias application.

2-1-1. 1/2 BIAS & STATIC AT AG BATTERY POWER SUPPLY



Note 1: The input/output ports operate between GND and VBAT.
Note 2: At the initial clear mode the backup flag (BCF) is set. When the backup flag is set, the oscillator circuit becomes large in inverter size and the oscillation conditions are improved, but the operating current is also increased. Therefore, the backup flag must be reset unless otherwise required.

2-2. SYSTEM CLOCK

XT clock (slow clock oscillator) and CF clock (fast clock oscillator) compose the clock oscillation circuitry.

The system clock generator provides the necessary clocks for execution of instruction. The pre-divider generates several clocks with different frequencies for the usage of LCD driver, frequency generator … etc.

The following table shows the clock sources of system clock generator and pre-divider in different conditions.

	PH0	BCLK
Slow clock only option	XT clock	XT clock
fast clock only option	CF clock	CF clock

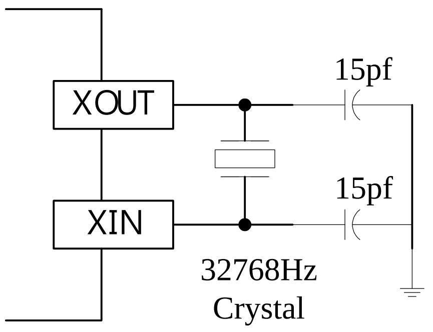
2-2-1 CONNECTION DIAGRAM OF SLOW CLOCK OSCILLATOR (XT CLOCK)

This clock oscillation circuitry provides the lower speed clock to the system clock generator, pre-divider, timer, chattering prevention of IO port and LCD circuitry. This oscillator will be disabled when the fast clock only option is selected by mask option, or it will be active all the time after the initial reset. In stop mode, this oscillator will be stopped.

2-2-1-1. External 32.768KHz Crystal oscillator (XT CLOCK)

MASK OPTION table :

Mask Option name	Selected item
CLOCK SOURCE	(3) SLOW



(1) X'tal

When backup flag (BCF) is set to 1, the oscillator operates with an extra buffer in parallel in order to shorten the oscillator start-up time but this will increase the power consumption. Therefore, the backup flag should be reset unless required otherwise.

The following table shows the power consumption of Crystal oscillator in different conditions:

	Ag power option
BCF=1	Increased
BCF=0	Normal
Initial reset	Increased
After reset	Increased

2-2-2. CONNECTION DIAGRAM OF FAST CLOCK OSCILLATOR (CF CLOCK)

The CF clock is a multiple type oscillator (mask option) which provides a faster clock source to system. In fast clock operation, this oscillator will provide the clock to the system clock generator, pre-divider, timer, I/O port chattering prevention clock and LCD circuitry.

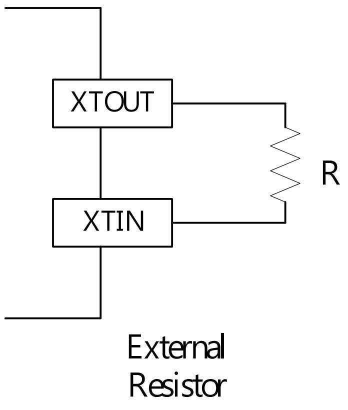
There are 2 type oscillators can be used in fast clock oscillator, selected by mask option:

2-2-2-1. RC OSCILLATOR WITH EXTERNAL RESISTOR (CF CLOCK)

This kind of oscillator could only be used in “FAST” option. When this oscillator is used, the frequency option of the RC oscillator with internal RC is not cared.

MASK OPTION table :

Mask Option name	Selected item
CLOCK SOURCE	(2) FAST & USE EXTERNAL RESISTOR

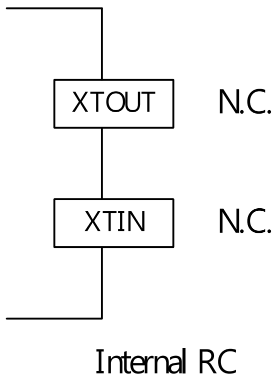


2-2-3-1. RC OSCILLATOR WITH INTERNAL RESISTOR (CF CLOCK)

When this oscillator is used, leave XOUT and XIN two pins opened. This kind of oscillator could be used in “FAST only”.

MASK OPTION table :

Mask Option name	Selected item
CLOCK SOURCE	(1) FAST ONLY & USE INTERNAL RESISTOR



2-2-3-2. SINGLE CLOCK

MASK OPTION table :

For Fast clock oscillator

Mask Option name	Selected item
CLOCK SOURCE	(1) FAST & USE INTERANL RESISTOR or (2) FAST & USE EXTERANL RESISTOR

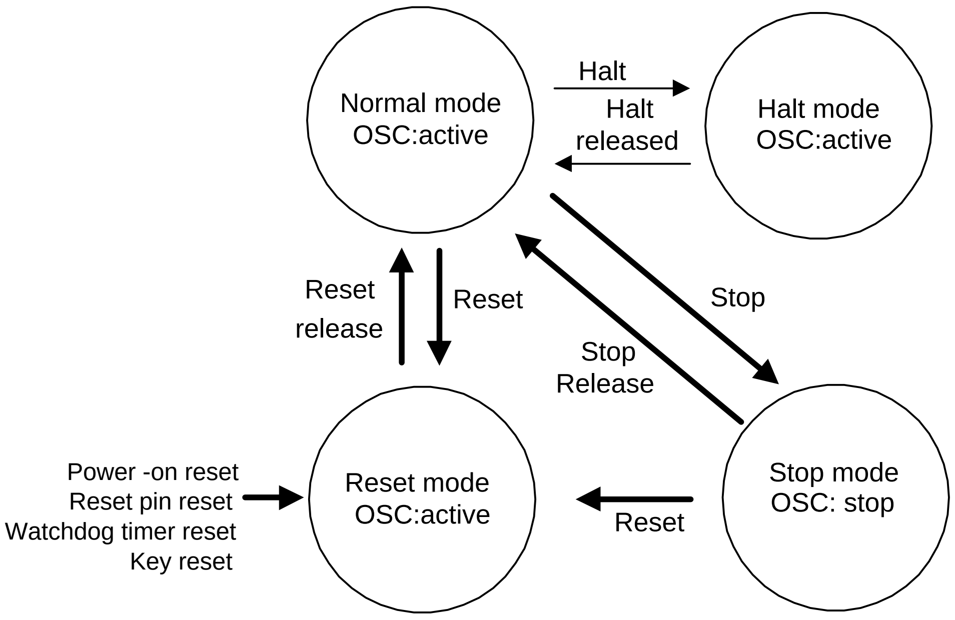
For slow clock oscillator only

Mask Option name	Selected item
CLOCK SOURCE	(3) SLOW ONLY

The operation of the single clock option is shown in the following figure.

Either XT or CF clock may be selected by mask option in this mode.

The backup flag (BCF) will be set to 1 automatically before the program enters the stop mode. This could ensure the Crystal oscillator would start up in a better condition.

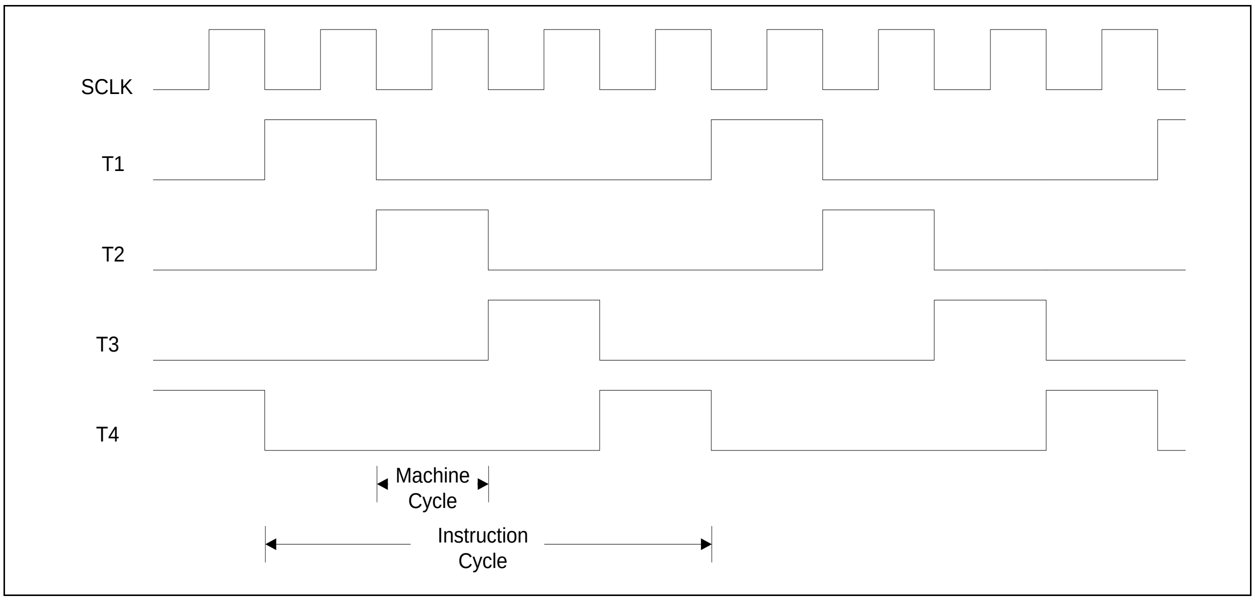


This figure shows the State Diagram of Single Clock Option

2-2-4 SYSTEM CLOCK GENERATOR

For the system clock, the clock switch mask option permits the different clock input from XTOSC and CFOSC to be selected.

The basic system clock is shown below:



2-3. PROGRAM COUNTER (PC)

This is a 10-bit counter, which addresses the program memory (ROM) up to 1024 addresses.

- The program counter (PC) is normally increased by one (+1) with every instruction execution.

  PC 🡐 PC + 1

- When executing JMP instruction, subroutine call instruction (CALL), interrupt service routine or reset occurs, the program counter (PC) loads the specified address corresponding to table 2-1.

  PC 🡐 specified address shows in

  Table 2- 1

- Return instruction (RTS)

  PC 🡐 content of stack specified by the stack pointer

  Stack pointer 🡐 stack pointer - 1

Table 2- 1

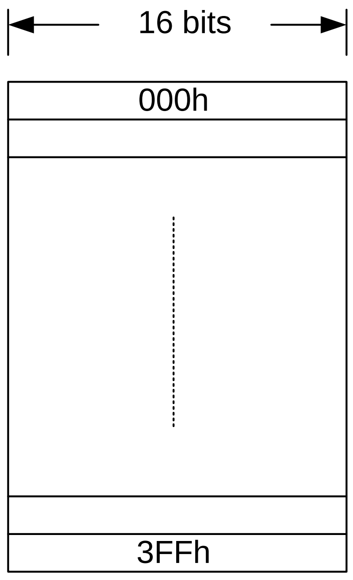
	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	0	0	0	0	0	0	0	0	0	0
Interrupt 0 (input port A)	0	0	0	0	0	1	0	1	0	0
Interrupt 4 (timer 2 interrupt)	0	0	0	0	1	0	0	0	0	0
Interrupt 3 (pre-divider interrupt)	0	0	0	0	0	1	1	1	0	0
Jump instruction	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Subroutine call	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

P9 to P0: Low-order 10 bits of instruction operand.

When executing the subroutine call instruction or interrupt service routine, the contents of the program counter (PC) are automatically saved to the stack register (STACK).

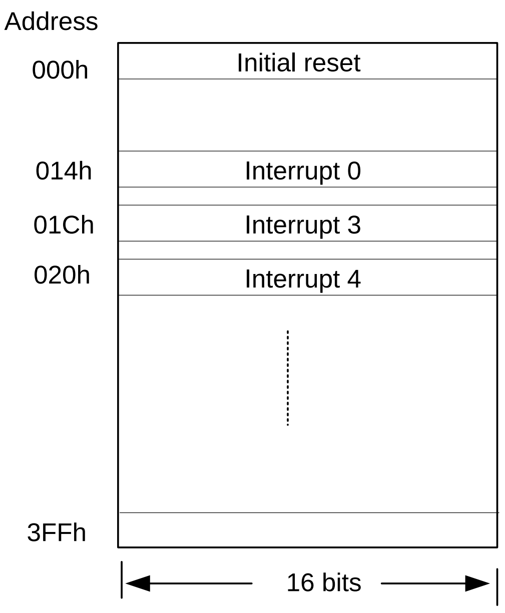
2-4. PROGRAM (ROM)

The built-in mask ROM is organized with 1024 x 16 bits.



2-4-1. INSTRUCTION ROM (PROM)

There are some special locations that serve as the interrupt service routines, such as reset address (000H), interrupt 0 address (014H), interrupt 3 address (01CH), in the program memory.



Instruction ROM (PROM) organization

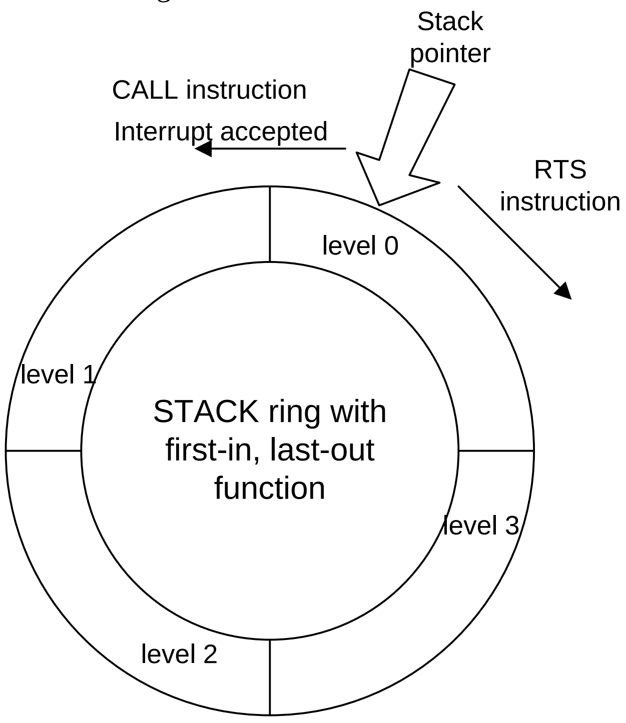
This figure shows the Organization of ROM

2-5. STACK REGISTER (STACK)

Stack is a special design register following the first-in-last-out rule. It is used to save the contents of the program counter sequentially during subroutine call or execution of the interrupt service routine.
The contents of stack register are returned sequentially to the program counter (PC) while executing return instructions (RTS).

The stack register is organized using 10 bits by 4 levels but with no overflow flag; hence only 4 levels of subroutine call or interrupt are allowed (If the stacks are full, and either interrupt occurs or subroutine call executes, the first level will be overwritten).

Once the subroutine call or interrupt causes the stack register (STACK) overflow, the stack pointer will return to 0 and the content of the level 0 stack will be overwritten by the PC value.
The contents of the stack register (STACK) are returned sequentially to the program counter (PC) during execution of the RTS instruction.
Once the RTS instruction causes the stack register (STACK) underflow, the stack pointer will return to level 3 and the content of the level 3 stack will be restored to the program counter.
The following figure shows the diagram of the stack.



2-6. DATA MEMORY (RAM)

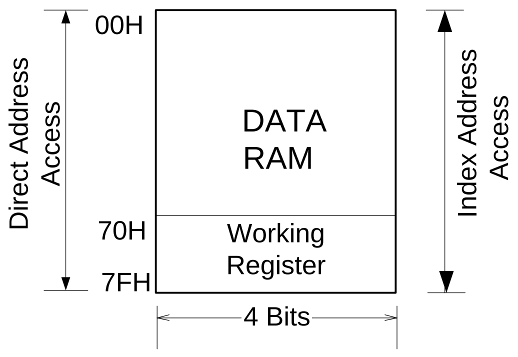
The static RAM is organized with 64 addresses x 4 bits and is used to store data. The address range of data memory is from 00h to 7Fh, but addresses between 00h to 3Fh are not reachable.

The data memory may be accessed by direct addressing:

Direct addressing mode

The address of the data memory is specified by the instruction and the addressing range is from 00H to 7FH. (Addresses between 00h to 3Fh are not reachable)

The 16 specified addresses (70H to 7FH) in the direct addressing memory are also used as 16 working registers. The function of working register will be described in detail later.



This figure shows the Data Memory (RAM) and Working Register Organization

2-7. WORKING REGISTER (WR)

The locations 70H to 7FH of the data memory (RAM) are not only used as general-purpose data memory but also as the working register (WR). The following will introduce the general usage of working registers:

1. Be used to perform operations on the contents of the working register and immediate data. Such as: ADCI, ADCI*, SBCI, SBCI*, ADDI, ADDI*, SUBI, SUBI*, ADNI, ADNI*, ANDI, ANDI*, EORI, EORI*, ORI, ORI*

2. Be transferred the data between the working register and any address in the direct addressing data memory (RAM). Such as:

MWR Rx, Ry; MRW Ry, Rx

3. Decode (or directly transfer) the contents of the working register and output to the LCD PLA circuit. Such as:

LCP

2-8. ACCUMULATOR (AC)

The accumulator (AC) is a register that plays the most important role in operations and controls. By using it in conjunction with the ALU (Arithmetic and Logic Unit), data transfer between the accumulator and other registers or data memory can be performed.

2-9. ALU (Arithmetic and Logic Unit)

This is a circuitry that performs arithmetic and logic operation. The ALU provides the following functions:

Binary addition/subtraction	ADC, SBC, ADD, SUB, ADN, ADCI, SUBI, ADNI)
Logic operation	(AND, EOR, OR, ANDI, EORI, ORI)
Shift	(SR0, SR1, SL0, SL1)
Decision	(JB0, JB1, JB2, JB3, JC, JNC, JZ, and JNZ)
BCD operation	(DAA, DAS)

2-10. HEXADECIMAL CONVERT TO DECIMAL (HCD)

Decimal format is another number format for TM8720. When the content of the data memory has been assigned as decimal format, it is necessary to convert hexadecimal number to decimal format after the execution of BCD instructions.

Instructions DAA, DAA* can convert the data from hexadecimal to decimal format after any addition operation. The conversion rules are shown in the following table and illustrated in example 1.

AC data before DAA execution	CF data before DAA execution	AC data after DAA execution	CF data after DAA execution
0 ≤ AC ≤ 9	CF = 0	no change	no change
A ≤ AC ≤ F	CF = 0	AC= AC+ 6	CF = 1
0 ≤ AC ≤ 3	CF = 1	AC= AC+ 6	no change

Example 1:

```
LDS     40h, 9          ; Load immediate data”9”to data memory address 40H.
LDS     41h, 1          ; Load immediate data”1”to data memory address 41H
                        ; and AC.
RF      1h              ; Reset CF to 0.
ADD*    40h             ; Contents of the data memory address 40H and AC are
                        ; binary-added; the result loads to AC & data memory address
                        ; 40H. (R10 = AC = AH, CF = 0)
DAA*    40h             ; Convert the content of AC to decimal format.
                        ; The result in the data memory address 40H is”0”and in
                        ; the CF is “1”. This represents the decimal number”10”.
```

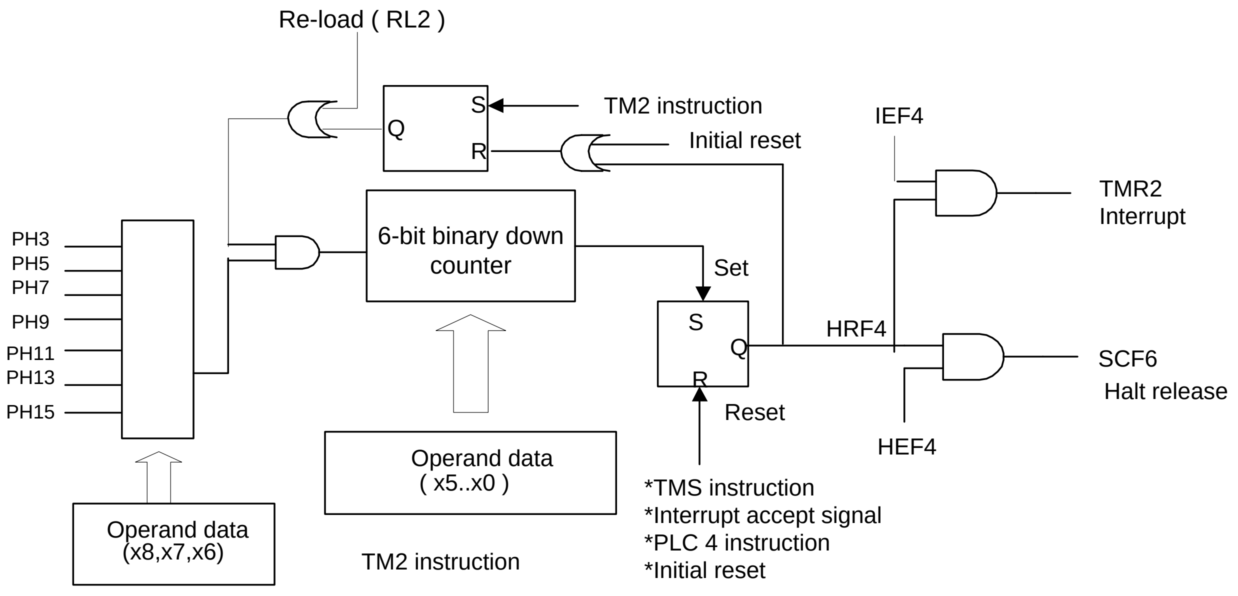
Instructions DAS, DAS* can convert the data from hexadecimal format to decimal format after any subtraction operation. The conversion rules are shown in the following table and illustrated in Example 2.

AC data before DAS execution	CF data before DAS execution	AC data after DAS execution	CF data after DAS execution
0 ≤ AC ≤ 9	CF = 1	No change	no change
6 ≤ AC ≤ F	CF = 0	AC= AC+A	no change

Example 2:

```
LDS    40h, 1      ; Load immediate data"1"to the data memory address 40H.
LDS    41h, 2      ; Load immediate data"2"to the data memory address 41H and
                   ;AC.
SF     1h          ; Set CF to 1, which means no borrowing has occurred.
SUB*   40h         ; Content of data memory address 40H is binary-subtracted;
                   ; the result loads to data memory address
                   ; 40H. (R10 = AC = FH, CF = 0)
DAS*   40h         ; Convert the content of the data memory address 40H to
                   ;decimal format.
                   ; The result in the data memory address 40H is"9"and in
                   ; the CF is "0". This represents the decimal number"–1".
```

2-11. TIMER 2 (TMR2)



This figure shows the TMR2 organization.

2-11-1. NORMAL OPERATION

TMR2 consists of a programmable 6-bit binary down counter, which is loaded and enabled by executing TM2 or TM2X instruction.

If reload function is not enabled, once the TMR2 counts down to 3Fh, it generates an underflow signal to set the halt release request flag 4 (HRF4) to 1 and then stops counting.

When HRF4 = 1, and the TMR2 interrupt enable flag (IEF4) = 1, the interrupt is generated.

When HRF4 = 1, if the IEF4 = 0 and the TMR2 halt release enable (HEF4) = 1, program will escapes from halt mode (if CPU is in halt mode) and then set the start condition flag 6 (SCF6) to 1 in the status register 3 (STS3).

The following table shows the definition of each bit in TMR2 instructions

OPCODE	Select clock			Initiate value of timer					
TM2X X	X8	X7	X6	X5	X4	X3	X2	X1	X0
TM2 Rx		AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

The following table shows the clock source setting for TMR2

X8	X7	X6	clock source
0	0	0	PH9
0	0	1	PH5
0	1	0	PH15
1	0	0	PH5
1	0	1	PH7
1	1	0	PH11
1	1	1	PH13

Notes:

1. When the TMR2 clock is PH3
   TMR2 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the TMR2 clock is PH5
   TMR2 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)
3. When the TMR2 clock is PH7
   TMR2 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)
4. When the TMR2 clock is PH9
   TMR2 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
5. When the TMR2 clock is PH11
   TMR2 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)
6. When the TMR2 clock is PH13
   TMR2 set time = (Set value + error) * 8192 * 1/fosc (KHz) (ms)
7. When the TMR2 clock is PH15
   TMR2 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
   Set value: Decimal number of timer set value
   error: the tolerance of set value, 0 < error <1.

fosc:	Input of the predivider
PH3:	The 3rd stage output of the predivider

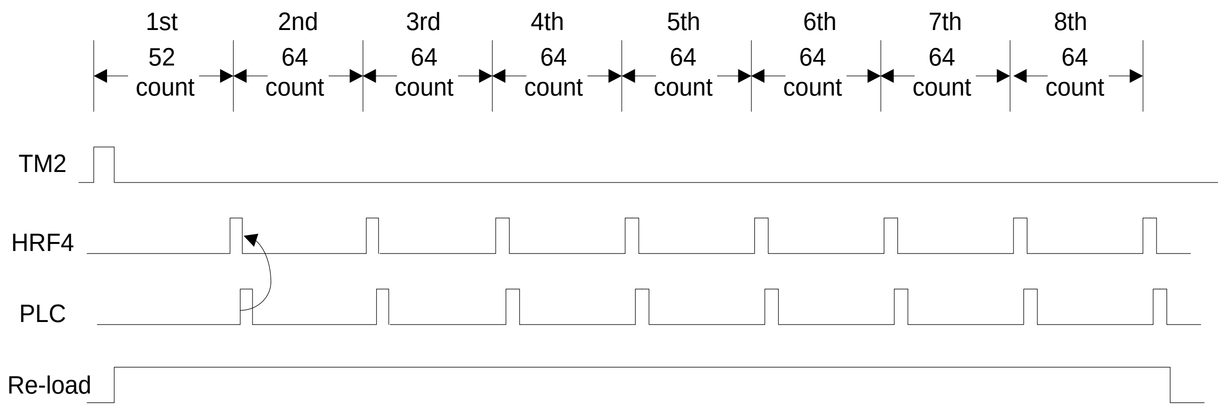
2-11-2. RE-LOAD OPERATION

TMR2 provides the re-load function which can extend any time interval greater than 3Fh. The SF2 01h instruction enables the re-load function and RF2 01h instruction disables it.

When the re-load function is enabled, the TMR2 will not stop counting until the re-load function is disabled and TMR2 underflows again. During this operation, the program must use the halt release request flag or interrupt to check the wanted counting value.

- It is necessary to execute the TM2 or TM2X instruction to set the down count value before the re-load function is enabled, because TMR2 will automatically count down with an unknown value once the re-load function is enabled.
- Never disable the re-load function before the last expected halt release or interrupt occurs. If TM2 related instructions are not executed after each halt release or interrupt occurs, the TMR2 will stop operating immediately after the re-load function is disabled.

For example, if the expected count down value is 500, it may be divided as 52 + 7 * 64. First, set the initiate count down value of TMR2 to 52 and start counting, then enable the TMR2 halt release or interrupt function. Before the first time underflow occurs, enable the re-load function. The TMR2 will continue operating even though TMR2 underflow occurs. When halt release or interrupt occurs, clear the HRF4 flag by PLC instruction. After halt release or interrupt occurs 8 times, disable the re-load function and the counting is completed.

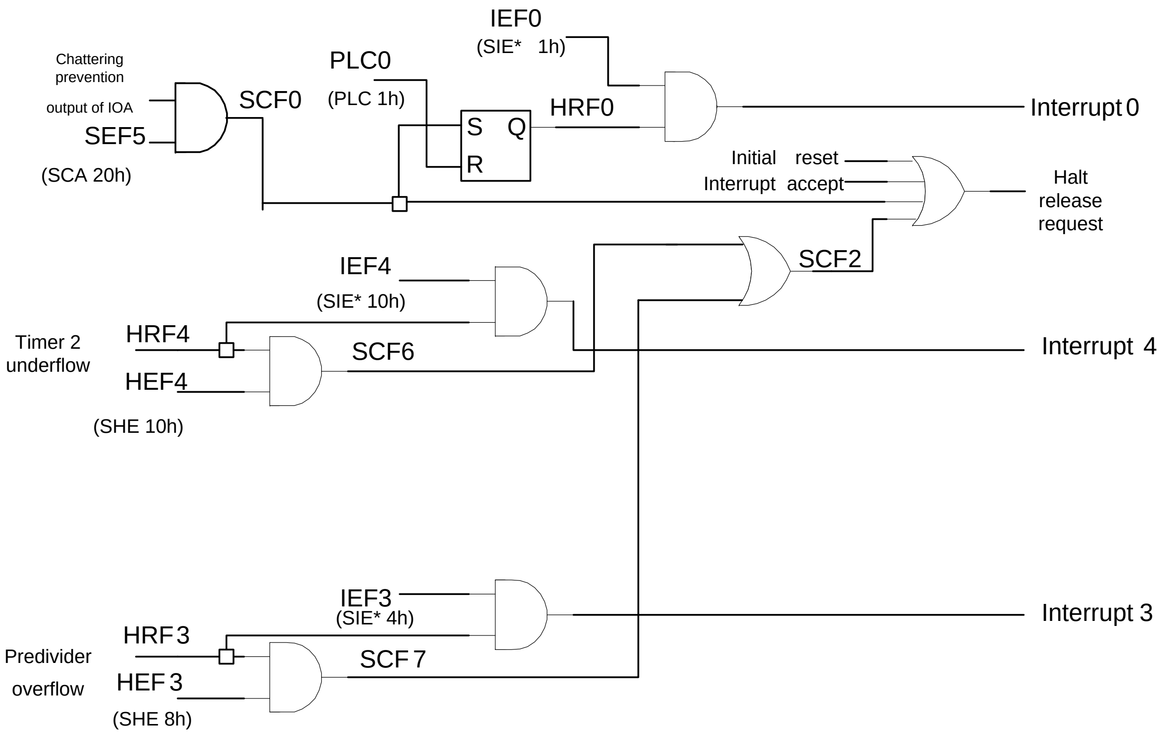


In the following example, S/W enters the halt mode to wait for the underflow of TMR2.

```
            LDS     0, 0            ;initiate the underflow counting register
            PLC     4
            SHE     10h             ;enable the HALT release caused by TMR2
            TM2X    034h            ;initiate the TMR2 value (52) and clock source is ϕ9
            SF2     01h             ;enable the re-load function
RE_LOAD:
            HALT
            INC*    0               ;increase the underflow counter
            PLC     4               ;clear HRF4
            JB3     END_TM2         ;if the TMR2 underflow counter is equal to 8, exit subroutine
            JMP     RE_LOAD
END_TM2
            RF2     01h             ;disable the re-load function
```

2-12. STATUS REGISTER (STS)

The status register (STS) is organized with 4 bits and comes in 4 types: status register 1 (STS1) to status register 3 (STS3) and status register 3X. The following figure shows the configuration of the start condition flags for TM8720.



2-12-1. STATUS REGISTER 1 (STS1)

Status register 1 (STS1) consists of 2 flags:

1. Carry flag (CF)
   The carry flag is used to save the result of the carry or borrow during the arithmetic operation.
2. Zero flag (Z)
   This flag indicates the accumulator (AC) status. When the content of the accumulator is 0, the Zero flag is set to 1. If the content of the accumulator is not 0, the zero flag is reset to 0.
3. The MAF instruction can be used to transfer data in status register 1 (STS1) to the accumulator (AC) and the data memory (RAM).
4. The MRA instruction can be used to transfer data of the data memory (RAM) to the status register 1 (STS1).

The bit pattern of status register 1 (STS1) is shown below.

Bit 3	Bit 2	Bit 1	Bit 0
Carry flag (CF)	Zero flag(Z)	NA	NA
Read / write	Read only	Read only	Read only

2-12-2. STATUS REGISTER 2 (STS2)

Status register 2 (STS2) consists of start condition flag 2 (SCF2) and the backup flag.

The MSB instruction can be used to transfer data of status register 2 (STS2) to both accumulator (AC) and the data memory (RAM), but it is not available to transfer data of the data memory (RAM) to status register 2 (STS2).

The following table shows the bit pattern of each flag in status register 2 (STS2).

Bit 3	Bit 2	Bit 1	Bit 0
NA	Start condition flag 2 (SCF2)	NA	Backup flag (BCF)
NA	Halt release caused by SCF4,5,7	NA	The back up mode status
NA	Read only	NA	Read only

Start condition flag 2 (SCF2)

When a factor other than port IOA causes the halt mode to be released, SCF2 will be set to1. In this case, if one or more start condition flags in SCF6, 7 is set to 1, SCF2 will also be set to 1 simultaneously. When all of the flags in SCF6, 7 are clear, start condition flag 2 (SCF2) is reset to 0. Note: If start condition flag is set to 1, the program will not be able to enter halt mode.

Backup flag (BCF)

This flag could be set / reset by executing the SF 2h / RF 2h instruction.

2-12-3. STATUS REGISTER 3 (STS3)

When the halt mode is released by start condition flag 2 (SCF2), status register 3 (STS3) will store the status of the factor in the release of the halt mode.

Status register 3 (STS3) consists of 2 flags:

1. Start condition flag 7 (SCF7)

Start condition flag 7 (SCF7) is set when an overflow signal from the pre-divider causes the halt release request flag 3 (HRF3) to be outputted and the halt release enable flag 3 (HEF3) is set beforehand. To reset start condition flag 7 (SCF7), the PLC instruction must be used to reset the halt release request flag 3 (HRF3) or the SHE instruction must be used to set the halt release enable flag 3 (HEF3).

2. The 15th stage's content of the pre-divider.

The MSC instruction is used to transfer the contents of status register 3 (STS3) to both accumulator (AC) and the data memory (RAM).

The following table shows the Bit Pattern of Status Register 3 (STS3)

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 7 (SCF7)	15th stage of the pre-divider	NA	NA
Halt release caused by pre-divider overflow		NA	NA
Read only	Read only	NA	NA

2-12-4. STATUS REGISTER 3X (STS3X)

Status register **3X (STS3X)** consists of 2 flags:

1. Start condition flag 0 (SCF0)
   When the SCA instruction specified signal change occurs at port IOA to release the halt mode, SCF0 will be set. Executing the SCA instruction will cause SCF0 to be reset to 0
2. Start condition flag 6 (SCF6)
   Start condition flag 6 (SCF6) is set when an underflow signal from Timer 2 (TMR2) causes the halt release request flag 4 (HRF4) to be outputted and the halt release enable flag 4 (HEF4) is set beforehand. To reset start condition flag 6 (SCF6), the PLC instruction must be used to reset the halt release request flag 4 (HRF4) or the SHE instruction must be used to reset the halt release enable flag 4 (HEF4).

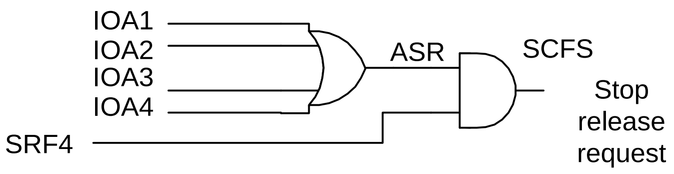
The MCX instruction can be used to transfer the contents of status register 3X (STS3X) both accumulator (AC) and the data memory (RAM).

The following table shows the Bit Pattern of Status Register 3X (STS3X)

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Start condition flag 0 (SCF0)	Start condition flag 6 (SCF6)	Reserved
Reserved	Halt release caused by the IOA port	Halt release caused by TMR2 underflow	Reserved

2-12-5. START CONDITION FLAGS (SCFS)

Start condition flags (SCFS) will be set to 1 in STOP mode when the following conditions are met :
. A high level signal comes from the OR-ed output of the pins defined as input mode in IOA port, which causes the stop release flag of IOA port (ASR) to output, and stop release enable flag 4 (SRF6) is set beforehand.
The following figure shows the organization of start condition flag S (SCF S).



The stop release flags (ASR) were specified by the stop release enable flags (SRFx) and these flags should be cleared before the chip enters the stop mode. All of the pins in IOA port had to be defined as the input mode and keep in 0 state before the chip enters the STOP mode, or the program can not enter the STOP mode.

Instruction SRE is used to set or reset the stop release enable flags (SRF6).

The following table shows the stop release request flags.

	The OR-ed input mode pins of IOA port
Stop release request flag	ASR
Stop release enable flag	SRF6

2-13. CONTROL REGISTER (CTL)

The control register (CTL) comes in 4 types: control register 1 (CTL1) to control register 4 (CTL4).

2-13-1. CONTROL REGISTER 1 (CTL1)

The control register 1 (CTL1), being a 1-bit register:

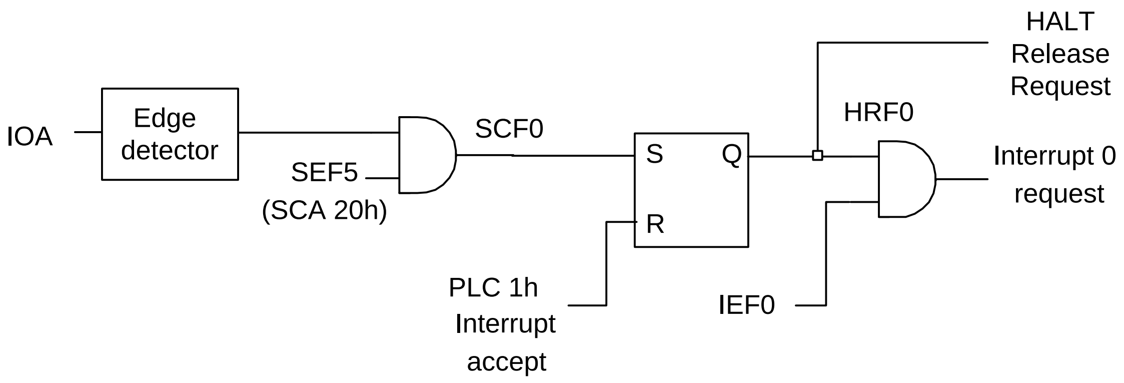
1. Switch enable flag 5 (SEF5)
   Stores the status of the input signal change at pins of IOA defined as input mode that causes the halt mode or stop mode to be released.

Executed SCA instruction may set or reset these flags.

The following table shows Bit Pattern of Control Register 1 (CTL1)

Bit 4
Switch enable flag 5 (SEF5)
Enables the halt release caused by the signal change on IOA port
Write only

The following figure shows the organization of control register 1 (CTL1).



2-13-1-1. The Setting for Halt Mode

If the SEF5 is set to 1, the signal changed on IOA port will cause the halt mode to be released, and set SCF0 to 1. Because the input signal of IOA port were ORed, so it is necessary to keep the unchanged input signals at “0” state and only one of the input signal could change state.

2-13-1-2. The Setting for Stop Mode

If SRF5 and SEF4 are set, the stop mode will be released to set the SCF0 when a high level signal is applied to one of the input mode pins of IOA port and the other pins stay in ”0” state.
After the stop mode is released, TM8720 enters the halt condition.
The high level signal must hold for a while to cause the chattering prevention circuitry of IOA port to detect this signal and then set SCF0 to release the halt mode, or the chip will return to the stop mode again.

2-13-1-3. Interrupt for CTL1

The control register 1 (CTL1) performs the following function in the execution of the SIE instruction to enable the interrupt function.
The input signal changes at the input pins in IOA port will deliver the SCF0 when SEF5 has been set to 1 by executing SCA instruction. Once the SCF0 is delivered, the halt release request flag (HRF0) will be set to 1. In this case, if the interrupt enable flag 0 (IEF0) is set to 1 by executing SIE instruction, the interrupt request flag 0 (interrupt 0) will be delivered to interrupt the program.
If the interrupt 0 is accepted by SEF5 and IEF0, the interrupt 0 request to the next signal change at IOA will be inhibited.

2-13-2. CONTROL REGISTER 2 (CTL2)

Control register 2 (CTL2) consists of halt release enable flags 3, 4 (HEF3, 4) and is set by SHE instruction. The bit pattern of the control register (CTL2) is shown below.

Halt release enable flag	HEF4	HEF3
Halt release condition	Enable the halt release caused by TM2 underflow (HRF4)	Enable the halt release caused by pre-divider overflow (HRF3)

When the halt release enable flag 3 (HEF3) is set, an overflow signal from the pre-divider causes the halt mode to be released. In the same manner, when HEF4 is set to 1, the following condition will cause the halt mode to be released: an underflow signal from TMR2 .

2-13-3. CONTROL REGISTER 3 (CTL3)

Control register 3 (CTL3) is organized with 7 bits of interrupt enable flags (IEF) to enable / disable interrupts.

The interrupt enable flag (IEF) is set / reset by SIE* instruction. The bit pattern of control register 3 (CTL3) is shown below.

Interrupt enable flag	IEF4	IEF3
Interrupt request flag	Enable the interrupt request caused by TM2 underflow (HRF4)	Enable the interrupt request caused by predivider overflow (HRF3)
Interrupt flag	Interrupt 4	Interrupt 3
Interrupt enable flag		IEF0
Interrupt request flag		Enable the interrupt request caused by IOA port signal to be changed (HRF0)
Interrupt flag		Interrupt 0

When any of the interrupts are accepted, the corresponding HRFx and the interrupt enable flag (IEF) will be reset to 0 automatically. Therefore, the desirable interrupt enable flag (IEFx) must be set again before exiting from the interrupt routine.

2-13-4. CONTROL REGISTER 4 (CTL4)

Control register 4 (CTL4), being a 2-bit register, is set / reset by SRE instruction.

The following table shows the Bit Pattern of Control Register 4 (CTL4)

Stop release enable flag	SRF6
Stop release request flag	Enable the stop release request caused by signal change on IOA

When SRF6 is set to 1, the input signal change at the input mode pins of IOA port and causes the stop mode to be released.

Example:

This example illustrates the stop mode released by port IOA. Assume all of the pins in IOA have been defined as input mode.

```
PLC                 01h             ; Reset HRF0
SCA         20h             ; SEF5 is set so that the signal changes at port IOA
                            ; cause the start conditions SCF0 to be set.
SRE         040h            ; SRF6 are set so that the signal changes at
                            ; port IOA  cause the stop mode to be released.
STOP                                ; Enter the stop mode.

   ……………                    ;STOP release
MCX         11h             ; Check the signal change at port IOA that causes the stop
                            ; mode to be released.
```

2-14. HALT FUNCTION

The halt function is provided to minimize the current dissipation of the TM8720 when LCD is operating. During the halt mode, the program memory (ROM) is not in operation and only the oscillator circuit, pre-divider circuit, sound circuit, I/O port chattering prevention circuit, and LCD driver output circuit are in operation. (If the timer has started operating, the timer counter still operates in the halt mode).

After the HALT instruction is executed and no halt release signal (SCF0, HRF3,4) is delivered, the CPU enters the halt mode.

The following 3 conditions are available to release the halt mode.

(1) An interrupt is accepted.
When an interrupt is accepted, the halt mode is released automatically, and the program will enter halt mode again by executing the RTS instruction after completion of the interrupt service.

When the halt mode is released and an interrupt is accepted, the halt release signal is reset automatically.

(2) The signal change specified by the SCA instruction is applied to port IOA (SCF0).

(3) The halt release condition specified by the SHE instruction is met (HRF3,4).
When the halt mode is released in either (2) or (3), it is necessary that the MSB, MSC, or MCX instruction is executed in order to test the halt release signal and that the PLC instruction is then executed to reset the halt release signal (HRF).

Even when the halt instruction is executed in the state where the halt release signal is delivered, the CPU does not enter the halt mode.

2-15. STOP FUNCTION (STOP)

The stop function is another solution to minimize the current dissipation for TM8720. In stop mode, all of functions in TM8720 are held including oscillators. All of the LCD corresponding signals (COM and Segment) will output "L" level. In this mode, TM8720 does not dissipate any power in the stop mode. Because the stop mode will set the BCF flag to 1 automatically, it is recommended to reset the BCF flag after releasing the stop mode in order to reduce power consumption.

Before the stop instruction is executed, all of the signals on the pins defined as input mode of IOA port must be in the "L" state, and no stop release signal (SRFn) should be delivered. The CPU will then enter the stop mode.

The following condition causes the stop mode to be released.

. One of the signals on the input mode pin of IOA port is in "H" state and holds long enough to cause the CPU to be released from halt mode.

When the TM8720 is released from the stop mode, the TM8720 enters the halt mode immediately and will process the halt release procedure. If the "H" signal on the IOA port does not hold long enough to set the SCFS, once the signal on the IOA port returns to "L", the TM8720 will enter the stop mode immediately. The backup flag (BCF) will be set to 1 automatically after the program enters the stop mode.

The following diagram shows the stop release procedure:

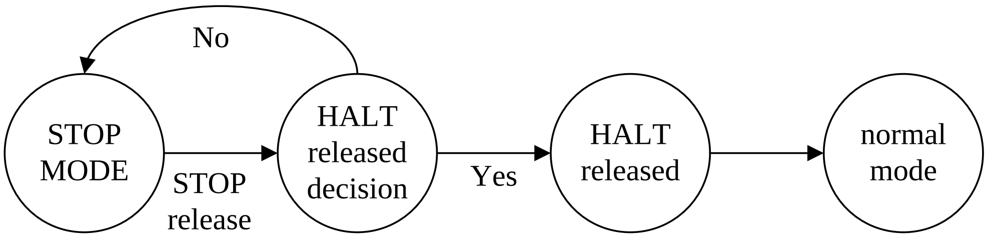


Figure 3- 16 The stop release state machine

Before the stop instruction is executed, the following operations must be completed:

. Specify the stop release conditions by the SRE instruction.

. Specify the halt release conditions corresponding to the stop release conditions if needed.

. Specify the interrupt conditions corresponding to the stop release conditions if needed.

When the stop mode is released by an interrupt request, the TM8720 will enter the halt mode immediately. While the interrupt is accepted, the halt mode will be released by the interrupt request. The stop mode returns by executing the RTS instruction after completion of interrupt service.

After the stop release, it is necessary that the MSB, MSC or MCX instruction be executed to test the halt release signal and that the PLC instruction then be executed to reset the halt release signal. Even when the stop instruction is executed in the state where the stop release signal (SRF) is delivered, the CPU does not enter the stop mode but the halt mode. When the stop mode is released and an interrupt is accepted, the halt release signal (HRF) is reset automatically.

2-16. BACK UP FUNCTION

Once the program enter back up mode (BCF = 1), 32.768 KHz Crystal oscillator will operate in a large driver condition and consumes more power.

The back up flag (BCF) indicates the status of back up function. BCF flag could be set or reset by executing SF or RF instruction respectively.

The back up function has different performance corresponding to different power mode option, shown in the following table.

1.5V battery mode:

TM8720 status	BCF flag status
Initial reset cycle	BCF = 1 (hardware controlled)
After initial reset cycle	BCF = 1 (hardware controlled)
Executing SF 2h instruction	BCF = 1
Executing RF 2h instruction	BCF = 0
HALT mode	Previous state
STOP mode	BCF = 1 (hardware controlled)

Note : For power saving reason, it is recommend to reset BCF flag to 0 when back up mode is not used.

Chapter 3 Control Function

3-1. INTERRUPT FUNCTION

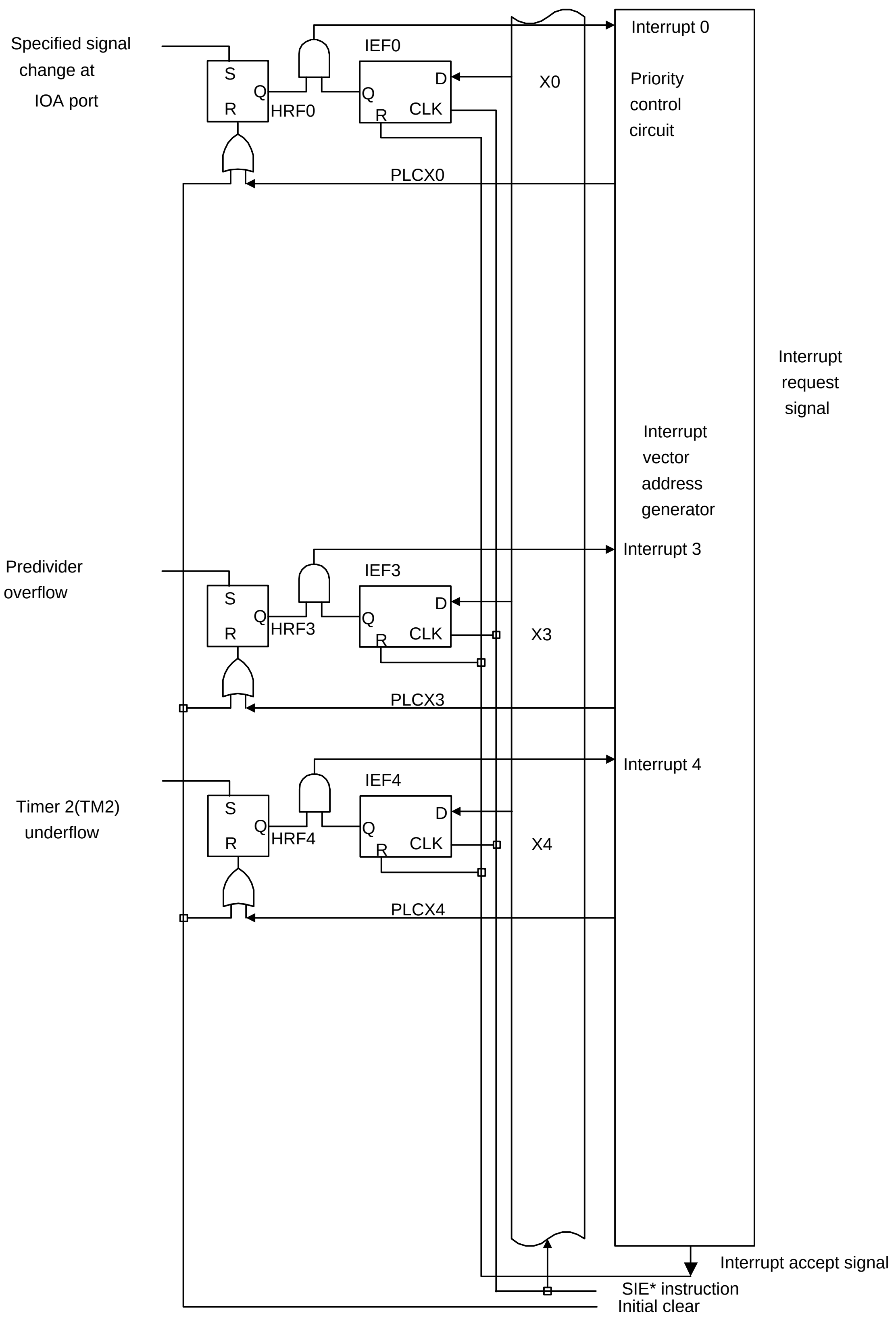
There are 3 interrupt resources: 1 external interrupt factor and 2 internal interrupt factors. When an interrupt is accepted, the program in execution is suspended temporarily and the corresponding interrupt service routine specified by a fix address in the program memory (ROM) is called.

The following table shows the flag and service of each interrupt:

Table 3-1 Interrupt information

Interrupt source	IOA port	TMR2 underflow	Pre-divider overflow
Interrupt vector	014H	020H	01CH
Interrupt enable flag	IEF0	IEF4	IEF3
Interrupt priority	5th	3nd	1st
Interrupt request flag	Interrupt 0	Interrupt 4	Interrupt 3

The following figure shows the Interrupt Control Circuit


Specified signal change at IOA port
S
Q
R
HRF0
IEF0
D
Q
R
CLK
X0
Interrupt 0
Priority control circuit
PLCX0
Interrupt request signal
Interrupt vector address generator
Predivider overflow
S
Q
R
HRF3
IEF3
D
Q
R
CLK
X3
Interrupt 3
PLCX3
Timer 2(TM2) underflow
S
Q
R
HRF4
IEF4
D
Q
R
CLK
X4
Interrupt 4
PLCX4
Interrupt accept signal
SIE* instruction
Initial clear

3-1-1. INTERRUPT REQUEST AND SERVICE ADDRESS

3-1-1-1. External interrupt factor

The external interrupt factor involves the use of the IOA ports.

I/O port IOA interrupt request.
An interrupt request signal (HRF0) is delivered when the input signal changes at I/O port IOA specified by the SCA instruction. In this case, if the interrupt enabled by flag 0 (IEF0) is set to 1, interrupt 0 is accepted and the instruction at address 14H is executed automatically.

3-1-1-2. Internal interrupt factor

The internal interrupt factor involves the use of timer 2 (TMR2) and the pre-divider.

1. Timer2 (TMR2) interrupt request
   An interrupt request signal (HRF4) is delivered when timer2 (TMR2) underflows. In this case, if the interrupt enable flag 1 (IEF4) is set, interrupt 4 is accepted and the instruction at address 20H is executed automatically.

2. Pre-divider interrupt request
   An interrupt request signal (HRF3) is delivered when the pre-divider overflows. In this case, if the interrupt enable flag3 (IEF3) is set, interrupt 3 is accepted and the instruction at address 1CH is executed automatically.

3-1-2. INTERRUPT PRIORITY

If all interrupts are requested simultaneously during a state when all interrupts are enabled, the pre-divider interrupt is given the first priority and other interrupts are held. When the interrupt service routine is initiated, all of the interrupt enable flags (IEF0 ~ IEF3) are cleared and should be set with the next execution of the SIE instruction. Refer to Table 3-1.

Example:
; Assume all interrupts are requested simultaneously when all interrupts are enabled, and all of the the pins of IOA have been defined as input mode.

```
PLC     19h          ;Clear all of the HRF flags
SCA     20h          ;enable the interrupt request of IOA
SIE*    19h          ;enable all interrupt requests

                     ;…………………… all interrupts are requested
                     ;simultaneously.
                     ;Interrupt caused by the predivider overflow occurs, and
                     ;interrupt  service is concluded.
SIE*    11h          ;Enable the interrupt request (except  the predivider).
                     ;Interrupt caused by the TM2 underflow occurs, and interrupt
                     ;service is concluded.
SIE*    01h          ;Enable the interrupt request (except  the predivider and TMR2).
                     ;Interrupt caused by the IOA port, and interrupt service is
                     ;concluded.  All interrupt requests have been processed.
```

3-1-3. INTERRUPT SERVICING

When an interrupt is enabled, the program in execution is suspended and the instruction at the interrupt service address is executed automatically (Refer to Table 3-1). In this case, the CPU performs the following services automatically.

(1) As for the return address of the interrupt service routine, the addresses of the program counter (PC) installed before interrupt servicing began are saved in the stack register (STACK).

(2) The corresponding interrupt service routine address is loaded in the program counter (PC).
The interrupt request flag corresponding to the interrupt accepted is reset and the interrupt enable flags are all reset.

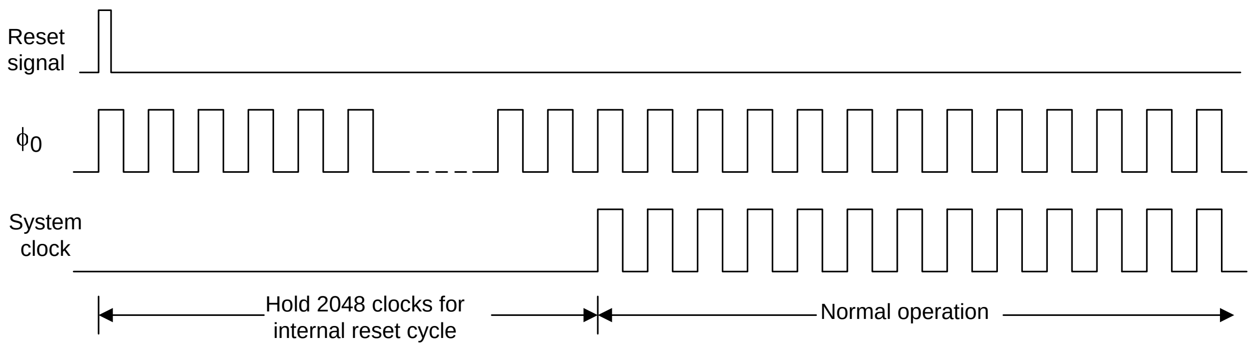
When the interrupt occurs, the TM8720 will follow the procedure below:

```
Instruction 1          ;In this instruction, interrupt is accepted.
NOP                    ;TM8720 stores the program counter data into the STACK. At this time,
      ;no instruction will be executed, as with NOP instruction.
Instruction A          ;The program jumps to the interrupt service routine.
Instruction B
Instruction C
 .............
RTS                    ;Finishes the interrupt service routine
Instruction 1*         ;re-executes the instruction which was interrupted.
Instruction 2
```

Note: If instruction 1 is "halt" instruction, the CPU will return to "halt" after interrupt.
When an interrupt is accepted, all interrupt enable flags are reset to 0 and the corresponding HRF flag will be cleared; the interrupt enable flags (IEF) must be set again in the interrupt service routine as required.

3-2. RESET FUNCTION

TM8720 contains two reset sources: power-on reset, RESET pin reset.
When reset signal is accepted, TM8720 will generate a time period (PH12/2) for internal reset cycle.



3-2-1. POWER ON RESET

TM8720 provides a power on reset function. If the power (VDD) is turned on, it will generate a power-on reset signal.

Note : When the power on reset option is selected, connected a capacitor between VDD and GND is necessary.

3-2-2. RESET PIN RESET

When "H" level is applied to the reset pin, the reset signal will be issued. There is a built-in pull down resistor on this pin.

It is recommended to connect a capacitor (0.1uf) between RESET pin and VDD. This connection will prevent the bounce signal on RESET pin.

Once a “1” signal applied on the RESET pin, TM8720’s internal circuit generates pulse reset. TM8720 begins the internal reset cycle and then release the reset status automatically.

The following table shows the initial condition of TM8720 in reset cycle.

Program counter	(PC)	Address 000H
Start condition flags	(SCF0,2,6,7)	0
Stop release enable flags	(SRF6)	0
Switch enable flags	(SEF5)	0
Halt release request flag	(HRF0,3,4)	0
Halt release enable flags	(HEF3,4)	0
Interrupt enable flags	(IEF0,3,4)	0
Alarm output	(ALARM)	DC 0
Pull-down flags in I/OA port		without pull down resistor
Input/output ports I/OA, I/OB,	(PORT I/OA, I/OB,)	Input mode
I/OA port chattering clock	Cch	PH10*
LCD driver output		All off
Timer 2		Inactive
Clock source	(BCLK)	Internal/External RC or XTAL (Mask option)

Notes:

PH10: the 10th output of predivider
Mask option can unlighted all of the LCD output

3-3. Clock Generator and Predivider

3-3-1. Doubler

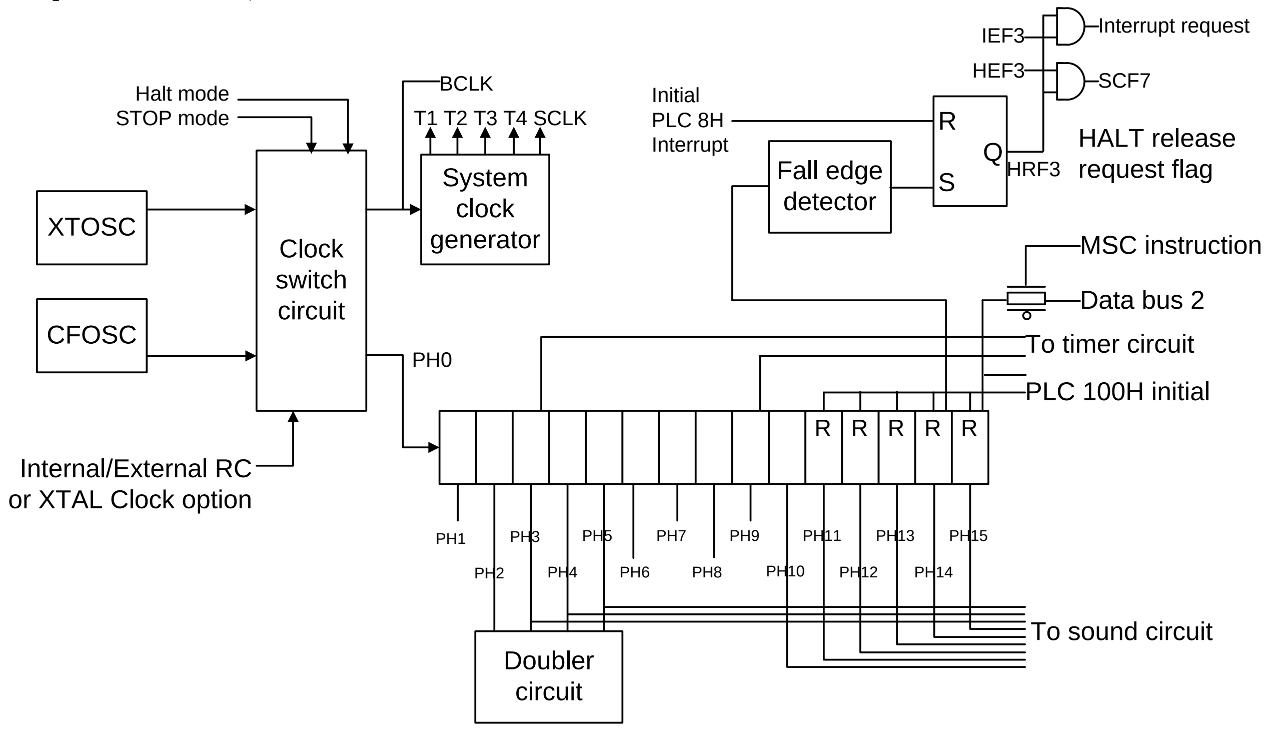
The doubler circuits are used to generate the bias voltage for LCD and are composed of a combination of PH2, PH3, PH4 and PH5.

3-3-2. Alternating Frequency for LCD

The alternating frequency for LCD is a frequency used to make the LCD waveform.

3-3-3. PREDIVIDER

The pre-divider is a 15-stage counter that receives the clock from the output of clock switch circuitry (PH0) as input. When PH0 is changed from "H" level to "L" level, the content of this counter changes. The PH11 to PH15 of the pre-divider are reset to "0" when the PLC 100H instruction is executed or at the initial reset mode. The pre-divider delivers the signal to the doubler circuit, alternating frequency for LCD display, system clock, sound generator and halt release request signal (I/O port chattering prevention clock).



This figure shows the Pre-divider and its Peripherals

The PH14 delivers the halt mode release request signal, setting the halt mode release request flag (HRF3). In this case, if the pre-divider interrupt enable mode (IEF3) is provided, the interrupt is accepted; and if the halt release enable mode (HEF3) is provided, the halt release request signal is delivered, setting the start condition flag 7 (SCF7) in status register 3 (STS3).
The clock source of pre-divider is PH0, and 2 kinds of frequency of PH0 could be selected by mask option:
MASK OPTION table :

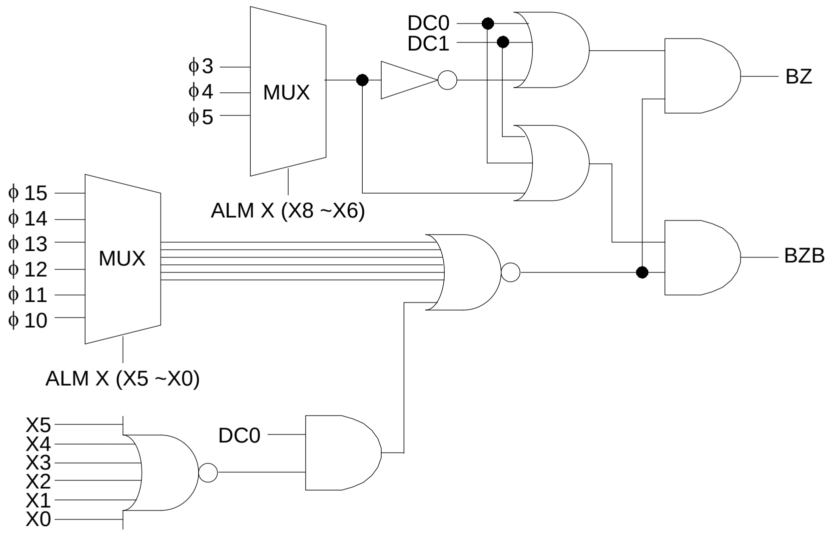
Mask Option name	Selected item
PH0<->Internal/External RC oscillator	(1) PH0 = Internal/External RC oscillator
PH0<->XTAL oscillator	(2) PH0 = XTAL oscillator

3-4. BUZZER OUTPUT PINS

There are two output pins, BZB and BZ. Each is MUXed with IOB3 and IOB4 by mask option, respectively. BZB and BZ pins are versatile output pins with complementary output polarity. When buzzer output function combined with the clock source comes from the frequency generator, this output function may generate melody, sound effect or carrier output of remote control.

MASK OPTION table :

Mask Option name	Selected item
IOB3/BZB	(2) BZB
IOB4/BZ	(2) BZ



This figure shows the organization of the buzzer output.

3-4-1 BASIC BUZZER OUTPUT

The buzzer output (BZ, BZB) is suitable for driving a transistor for the buzzer with one output pin or driving a buzzer with BZ and BZB pins directly. It is capable of delivering a modulation output in any combination of one signal of PH3 (4096Hz), PH4 (2048Hz), PH5 (1024Hz) and multiple signals of PH10 (32Hz), PH11 (16Hz), PH12 (8Hz), PH13 (4Hz), PH14 (2Hz), PH15 (1Hz). The ALM instruction is used to specify the combination. The higher frequency clock is the carrier of modulation output and the lower frequency clock is the envelope of the modulation output.

Note:

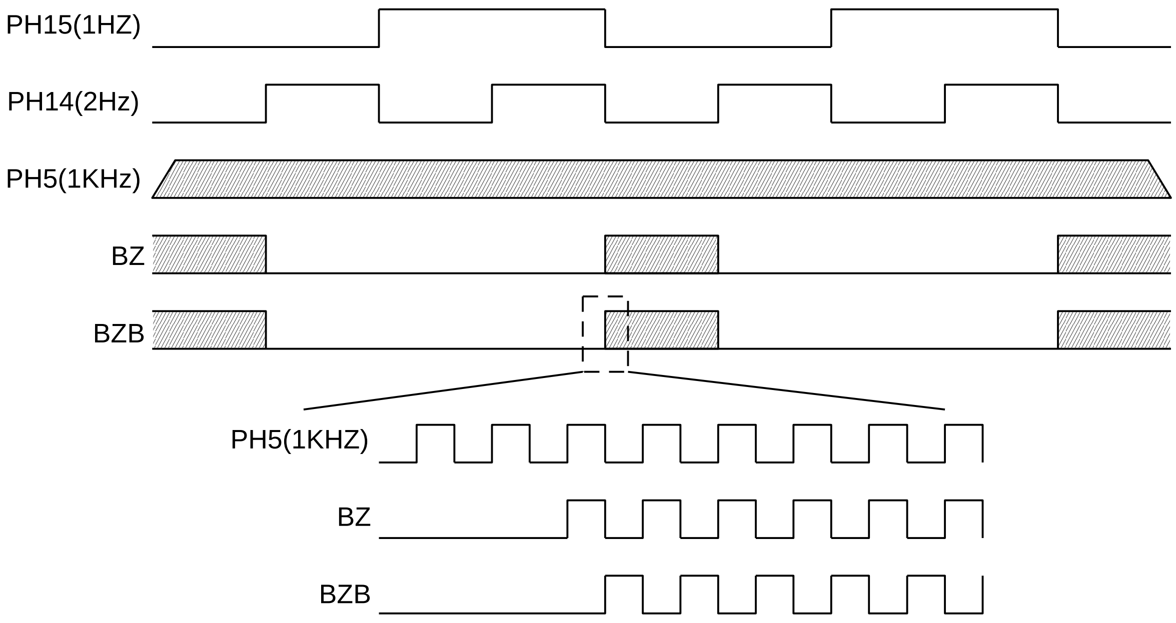
1. The high frequency clock source should only be one of PH3, PH4, PH5 and the lower frequency may be any/all of the combinations from PH10 ~ PH15.
2. The frequencies in () corresponding to the input clock of the pre-divider (PH0) is 32768Hz.
3. The BZ and BZB pins will output DC0 after the initial reset.

Example:

Buzzer output generates a waveform with 1K Hz carrier and (PH15 + PH14) envelope.

```
    LDS     40h, 0Ah
    ……….
    ALM     70h                 ; Output the waveform.
    ………
```

In this example, the BZ and BZB pins will generate the waveform as shown in the following figure:



3-5. INPUT / OUTPUT PORTS

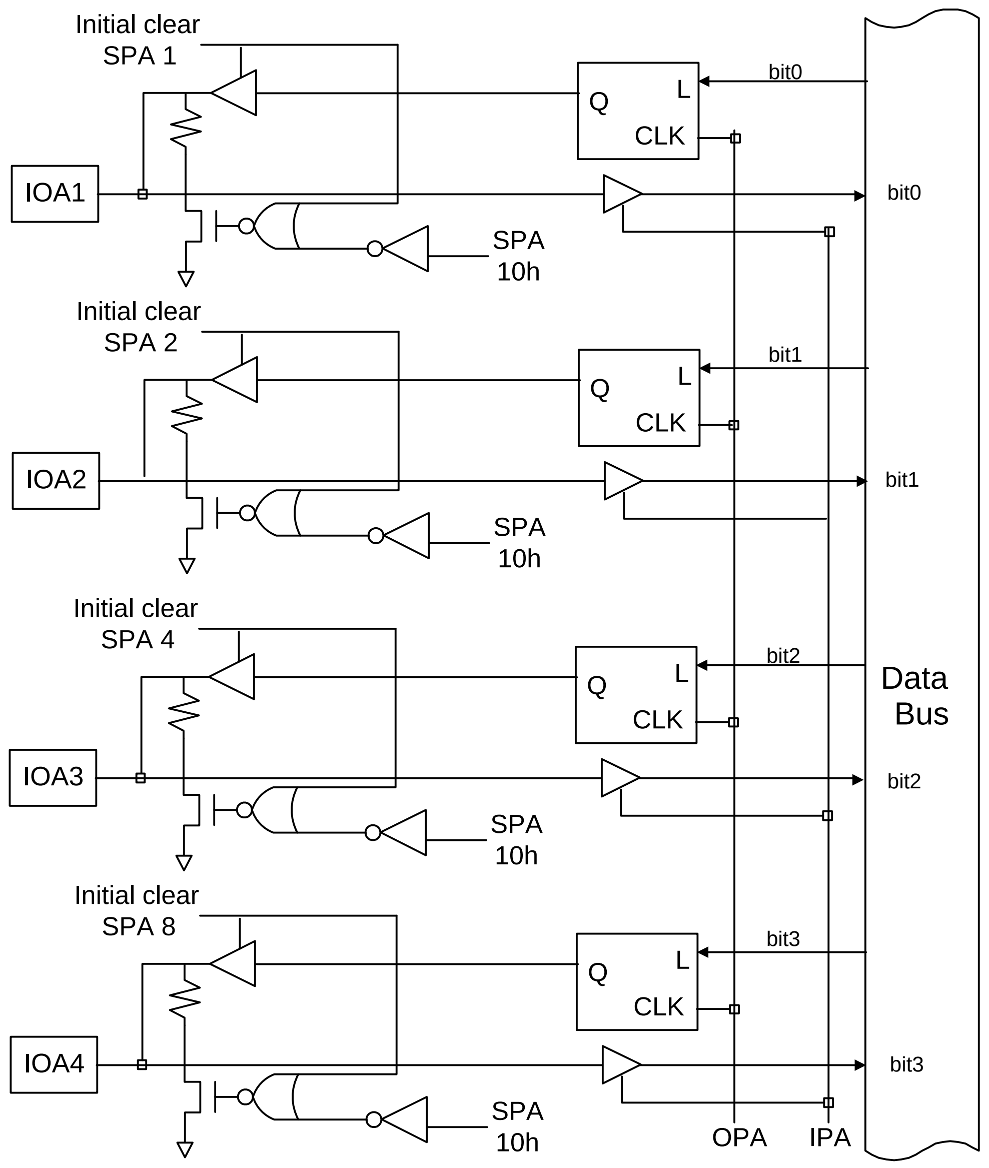
Three I/O ports are available in TM8720: IOA, IOB.

3-5-1 IOA PORT

In initial reset cycle, the IOA port is set as input mode and each bit of port can be defined as input mode or output mode individually by executing SPA instructions. Executing OPA instructions may output the content of specified data memory to the pins defined as output mode; the pins defined as the input mode will still remain the input mode.

Executing IPA instructions may store the signals applied to the IO pins into the specified data memory. When the IO pins are defined as the output mode, executing IPA instruction will store the content that stored in the latch of the output pin into the specified data memory.

Before executing SPA instruction to define the I/O pins as the output mode, the OPA instruction must be executed to output the data to those output latches beforehand. This will prevent the chattering signal on the I/O pin when the I/O mode changed.



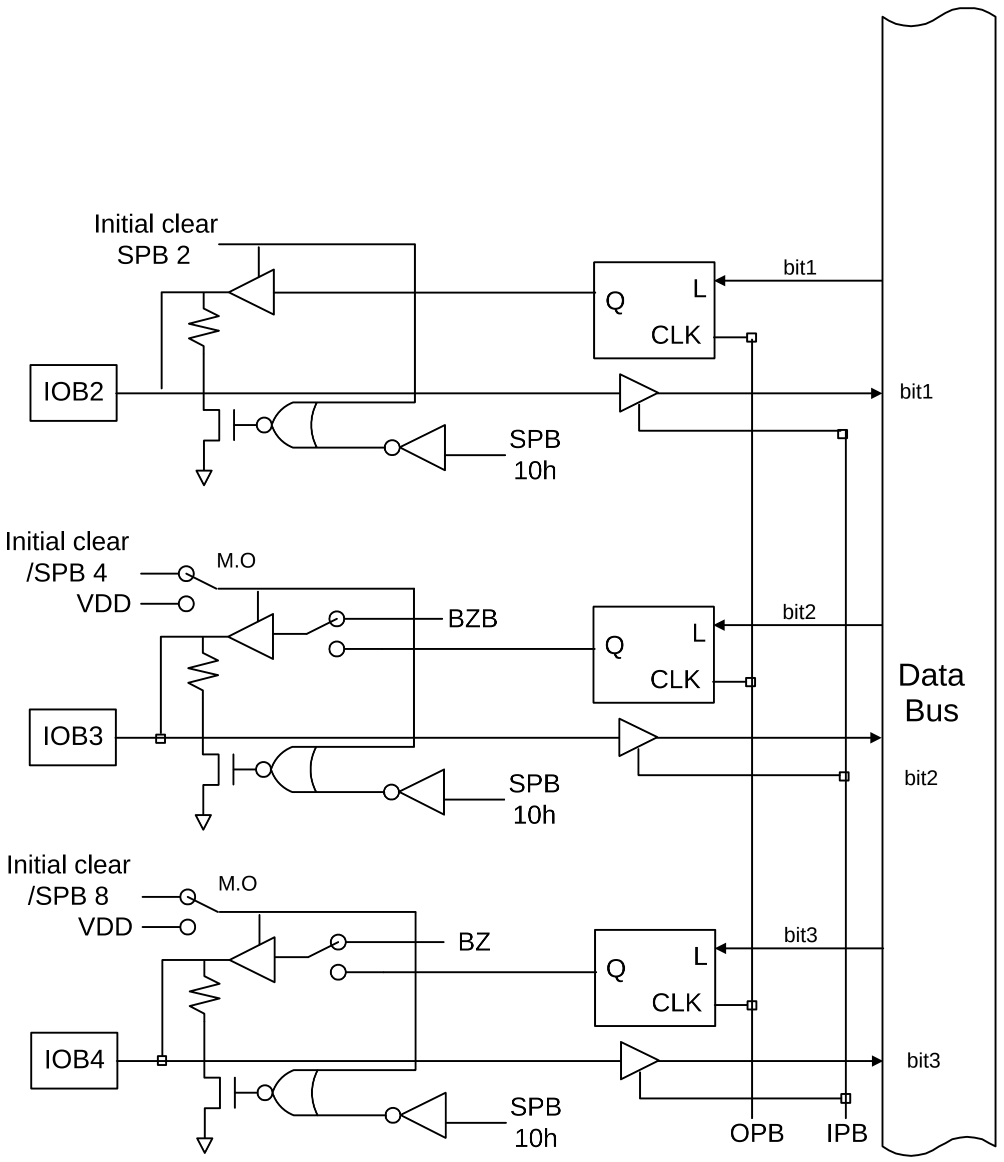
This figure shows the organization of IOA port.

Note: If the input level is in the floating state, a large current (straight-through current) flows to the input buffer. The input level must not be in the floating state.

3-5-2 IOB PORT

IOB has only 3 pins, i.e... IOB2, 3, 4. IOB3, 4 are MUXed with BZB and BZ.

The following figure shows the organization of IOB port.



Note: If the input level is in the floating state, a large current (straight-through current) flows to the input buffer. The input level must not be in the floating state.

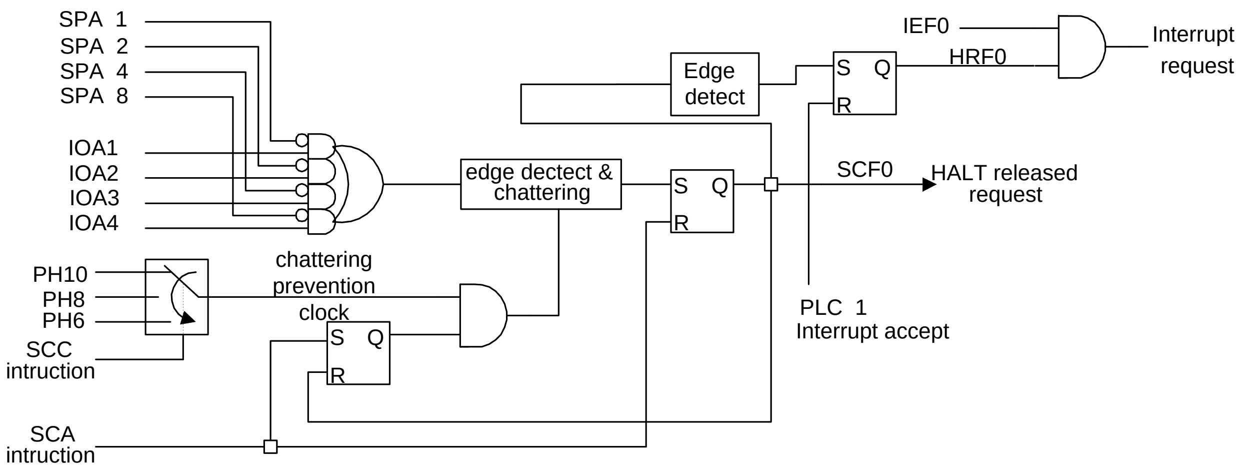
After the reset cycle, the IOB port is set as input and each bit of port can be defined as input or output individually by executing SPB instructions. Executing OPB instructions may output the content of specified data memory to the pins defined as output mode; the other pins which are defined as the input will still be input.

Executing IPB instructions may store the signals applied on the IOB pins into the specified data memory. When the IOB pins are defined as the output, executing IPB instruction will save the data stored in the output latch into the specified data memory.

Before executing SPB instruction to define the I/O pins as output, the OPB instruction must be executed to output the data to the output latches. This will prevent the chattering signal on the I/O pin when the I/O mode changed.

3-5-3-1 Chattering Prevention Function and Halt Release

The port IOA is capable of preventing high / low chattering of the switch signal applied on IOA1 to IOA4 pins. The chattering prevention time can be selected as PH10 (32ms), PH8 (8ms) or PH6 (2ms) by executing SCC instruction, and the default selection is PH10 after the reset cycle. When the pins of the IOA port are defined as output, the signals applied to the output pins will be inhibited for the chattering prevention function. The following figure shows the organization of chattering prevention circuitry.



Note: The default prevention clock is PH10

This chattering prevention function works when the signal at the applicable pin (ex. IOA1) is changed from ”L” level to ”H” level or from ”H” level to ”L” level, and the remaining pins (ex, IOA2 to IOA4) are held at ”L” level.

When the signal changes at the input pins of IOA port specified by the SCA instruction occur and keep the state for at least two chattering clock (PH6, PH8, PH10) cycles, the control circuit at the input pins will deliver the halt release request signal (SCF0). At that time, the chattering prevention clock will stop due to the delivery of SCF0. The SCF1 will be reset to 0 by executing SCA instruction and the chattering prevention clock will be enabled at the same time. If the SCF0 has been set to 1, the halt release request flag 0 (HRF0) will be delivered. In this case, if the port IOA interrupt enable mode (IEF0) is provided, the interrupt is accepted.

Since no flip-flop is available to hold the information of the signal at the input pins IOA1 to IOA4, the input data at the port IOA must be read into the RAM immediately after the halt mode is released.

CHAPTER 4 LCD DRIVER OUTPUT

4-1. LCD DRIVER OUTPUT

The number of the LCD driver outputs in TM8720 is 12 segment pins with 4 common pins.

4-1-1. LCD LIGHTING SYSTEM IN TM8720

TM8720 may only output 1/2 bias 1/4 duty waveform.

Frame frequency MASK option (PH0 = 32 KHz)

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	16Hz
LCD frame frequency	(2) TYPICAL	32Hz
LCD frame frequency	(3) FAST	64Hz

The LCP instruction transfers the data of the RAM (Rx) to LCD RAM directly.

FIXED LCD TABLE

SEG	**Lz**	**Bit0**	**Bit1**	**Bit2**	**Bit3**	SEG	**Lz**	**Bit0**	**Bit1**	Bit2	**Bit4**
SEG1	00H	com1	com2	com3	com4	**SEG7**	06H	com1	com2	com3	com4
SEG2	01H	com1	com2	com3	com4	**SEG8**	07H	com1	com2	com3	com4
SEG3	02H	com1	com2	com3	com4	**SEG9**	08H	com1	com2	com3	com4
SEG4	03H	com1	com2	com3	com4	**SEG10**	09H	com1	com2	com3	com4
SEG5	04H	com1	com2	com3	com4	**SEG11**	0AH	com1	com2	com3	com4
SEG6	05H	com1	com2	com3	com4	**SEG12**	0BH	com1	com2	com3	com4

The LCD outputs could be turned off without changing the segment data. Executing SF2 4h instruction could turn off the display simultaneously and RF2 4h could turn on the display with the patterns before turned off. These two instructions will not affect the content stored in the latch circuitry. When the LCD is turned off by executing RF2 4h instruction, the program could still execute LCP instruction to update the content in the latch circuitry and the new content will be outputted to the LCD while the display is turned on again.

In stop state, all COM and SEG outputs of LCD driver will automatically switch to the GND state to avoid the DC voltage bias on the LCD panel.

4-1-3-2. Relative Instructions

1. LCP   Lz, Ry

The data of the RAM is transferred directly to LCD memories.

2. SF2   4h

Turn off the LCD display.

3. RF2   4h

Turn on the LCD display.

Chapter 5 Detail Explanation of TM8720 Instructions

- The working registers are part of the data memory (RAM), and the relationship between them can be shown as follows:
  [The absolute address of working register Rx=Ry+70H]*

Note: Ry: Address of working register, the range of addresses specified by Rx is from 00H to 7FH.
Rx: Address of data memory, the range of addresses specified by Ry is from 0H to FH.

Address of working registers specified by Ry	Absolute address of data memory (Rx)
0H	70H
1H	71H
2H	72H
. . . .	. . . .
DH	7DH
EH	7EH
FH	7FH

- Lz represents the address of the latch of LCD PLA; the address range specified by Lz is from 00H to 1FH.

5-1. INPUT / OUTPUT INSTRUCTIONS

LCP Lz, Ry

Function: LCD latch Lz ← (Ry)

Description: The working register contents specified by Ry is loaded to the LCD latch specified by Lz.

SPA X

Function: Defines the input/output mode of each pin for IOA port and enables / disables the pull-low device.

Description: Sets the I/O mode and turns on/off the pull-low device. The input pull-low device will be enabled when the I/O pin was set as input mode. The meaning of each bit of X (X3 X2 X1 X0) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enables all of the pull-low and disables the low-level hold devices	X4=0	Disables all of the pull-low and enables the low-level hold devices
X3=1	IOA4 as output mode	X3=0	IOA4 as input mode
X2=1	IOA3 as output mode	X2=0	IOA3 as input mode
X1=1	IOA2 as output mode	X1=0	IOA2 as input mode
X0=1	IOA1 as output mode	X0=0	IOA1 as input mode

OPA Rx

Function:	I/OA ← (Rx)
Description:	The content of Rx is outputted to I/OA port.

IPA Rx

Function:	Rx, AC ← (IOA)
Description:	The data of I/OA port is loaded to AC and data memory Rx.

SPB X

Function: Defines the input/output mode of each pin for IOB port and enables / disables the pull-low device.

Description: Sets the I/O mode and turns on/off the pull-low device. The input pull-low device will be enabled when the I/O pin was set as input mode. The meaning of each bit of X (X3 X2 X1) is shown below:

Bit pattern	**Setting**	**Bit pattern**	**Setting**
X4=1	Enables all of the pull-low and disables the low-level hold devices	X4=0	Disables all of the pull-low and enables the low-level hold devices
X3=1	IOB4 as output mode	X3=0	IOB4 as input mode
X2=1	IOB3 as output mode	X2=0	IOB3 as input mode
X1=1	IOB2 as output mode	X1=0	IOB2 as input mode

OPB Rx

Function:	I/OB ← (Rx)
Description:	The contents of Rx are outputted to I/OB port.

IPB Rx

Function:	Rx, AC ← (IOB)
Description:	The data of I/OB port is loaded to AC and data memory Rx.

ALM X

Function: Sets buzzer output frequency.

Description: The waveform specified by X (X8 ~ X0) is delivered to the BZ and BZB pins. output frequency could be any combination in the following table.

The bit pattern of X (for higher frequency clock source):

X8	**X7**	**X6**	**clock source (higher frequency)**
1	0	0	DC1
0	1	1	ϕ3(4KHz)
0	1	0	ϕ4(2KHz)
0	0	1	ϕ5(1KHz)
0	0	0	DC0

The bit pattern of X (for lower frequency clock source)*:

Bit	**clock source(lower frequency)**
X5	ϕ15(1Hz)
X4	ϕ14(2Hz)
X3	ϕ13(4Hz)
X2	ϕ12(8Hz)
X1	ϕ11(16Hz)
X0	ϕ10(32Hz)

Notes:

1. FREQ is the output of frequency generator.
2. When the buzzer output does not need the envelope waveform, X5 ~ X0 should be set to 0.
3. The frequency inside the () bases on the ϕ0 is 32768Hz.

5-2. ACCUMULATOR MANIPULATION INSTRUCTIONS AND MEMORY MANIPULATION INSTRUCTIONS

MRW Ry, Rx

Function:	AC, Ry ← (Rx)
Description:	The content of Rx is loaded to AC and the working register specified by Ry.

MWR Rx, Ry

Function:	AC, Rx ← (Ry)
Description:	The content of working register specified by Ry is loaded to AC and data memory specified by Rx.

SR0 Rx

Function:	Rxn, ACn ← Rx(n+1),AC(n+1) Rx3, AC3 ← 0
Description:	The Rx content is shifted right and 0 is loaded to the MSB. The result is loaded to the AC. 0 → Rx3 → Rx2 → Rx1 → Rx0 →

SR1 Rx

Function:	Rxn, ACn ← Rx(n+1),AC(n+1) Rx3, AC3 ← 1
Description:	The Rx content is shifted right and 1 is loaded to the MSB. The result is loaded to the AC. 1 → Rx3 → Rx2 → Rx1 → Rx0 →

SL0 Rx

Function:	Rxn, ACn ← Rx(n-1),AC(n-1) Rx0, AC0 ← 0
Description:	The Rx content is shifted left and 0 is loaded to the LSB. The results are loaded to the AC. ← Rx3 ← Rx2 ← Rx1 ← Rx0 ← 0

SL1 Rx

Function:	Rxn, ACn ← Rx(n-1),AC(n-1) Rx0, AC0 ← 1
Description:	The Rx content is shifted left and 1 is loaded to the LSB. The results are loaded to the AC. ← Rx3 ← Rx2 ← Rx1 ← Rx0 ← 1

MRA Rx
Function: CF ← (Rx)3
Description: Bit3 of the content of Rx is loaded to carry flag(CF).

MAF Rx
Function: AC,Rx ← STS1
Description: The content of STS1 is loaded to AC and Rx. The content of AC and meaning of bit after execution of this instruction are as follows:

Bit 3 .... CF
Bit 2 .... (AC)=0, zero flag
Bit 1 .... (No Use)
Bit 0 .... (No Use)

5-3. OPERATION INSTRUCTIONS

INC* Rx
Function: Rx,AC ← (Rx)+1
Description: Add 1 to the content of Rx; the result is loaded to data memory Rx and AC.
* Carry flag (CF) will be affected.

DEC* Rx
Function: Rx, AC ← (Rx)-1
Description: Substrate 1 from the content of Rx; the result is loaded to data memory Rx and AC.
• Carry flag (CF) will be affected.

ADC Rx
Function: AC ← (Rx)+(AC)+CF
Description: The contents of Rx, AC and CF are binary-added; the result is loaded to AC.
* Carry flag (CF) will be affected.

ADC* Rx
Function: AC, Rx ← (Rx)+(AC)+CF
Description: The contents of Rx, AC and CF are binary-added; the result is loaded to AC and data memory Rx.
* Carry flag (CF) will be affected.

SBC Rx
Function: AC ← (Rx)+ (AC)B+CF
Description: The contents of AC and CF are binary-subtracted from content of Rx; the result is loaded to AC.
. Carry flag (CF) will be affected.

SBC* Rx

Function:	AC, Rx ← (Rx)+(AC)B+CF
Description:	The contents of AC and CF are binary-subtracted from content of Rx; the result is loaded to AC and data memory Rx. . Carry flag (CF) will be affected.

ADD Rx

Function:	AC ← (Rx)+(AC)
Description:	The contents of Rx and AC are binary-added; the result is loaded to AC. * Carry flag (CF) will be affected.

ADD* Rx

Function:	AC, Rx ← (Rx)+(AC)
Description:	The contents of Rx and AC are binary-added; the result is loaded to AC and data memory Rx. . Carry flag (CF) will be affected.

SUB Rx

Function:	AC ← (Rx)+ (AC)B+1
Description:	The content of AC is binary-subtracted from content of Rx; the result is loaded to AC. * Carry flag (CF) will be affected.

SUB* Rx

Function:	AC,Rx ← (Rx)+ (AC)B+1
Description:	The content of AC is binary-subtracted from content of Rx; the result is loaded to AC and Rx. * Carry flag (CF) will be affected.

ADN Rx

Function:	AC ← (Rx)+(AC)
Description:	The contents of Rx and AC are binary-added; the result is loaded to AC. * The result will not affect the carry flag (CF).

ADN* Rx

Function:	AC, Rx ← (Rx)+(AC)
Description:	The contents of Rx and AC are binary-added; the result is loaded to AC and data memory Rx. * The result will not affect the carry flag (CF).

AND Rx

Function:	AC ← (Rx) & (AC)
Description:	The contents of Rx and AC are binary-ANDed; the result is loaded to AC.

AND* Rx

Function:	AC, Rx ← (Rx) & (AC)
Description:	The contents of Rx and AC are binary-ANDed; the result is loaded to AC and data memory Rx.

EOR Rx

Function:	AC ← (Rx) ⊕ (AC)
Description:	The contents of Rx and AC are exclusive-Ored; the result is loaded to AC.

EOR* Rx

Function:	AC, Rx ← (Rx) ⊕ (AC)
Description:	The contents of Rx and AC are exclusive-Ored; the result is loaded to AC and data memory Rx.

OR Rx

Function:	AC ← (Rx) | (AC)
Description:	The contents of Rx and AC are binary-Ored; the result is loaded to AC.

OR* Rx

Function:	AC, Rx ← (Rx) | (AC)
Description:	The contents of Rx and AC are binary-Ored; the result is loaded to AC data memory Rx.

ADCI Ry, D

Function:	AC ← (Ry)+D+CF
Description:	D represents an immediate data. The contents of Ry, D and CF are binary-ADDed; the result is loaded to AC. *The carry flag (CF) will be affected. D = 0H ~ FH

ADCI* Ry, D

Function:	AC,Ry ← (Ry)+D+CF
Description:	D represents an immediate data. The contents of Ry, D and CF are binary-ADDed; the result is loaded to AC and working register Ry. * The carry flag (CF) will be affected. D = 0H ~ FH

SBCI Ry, D

Function:	AC ← (Ry)+ (D)B+CF
Description:	D represents an immediate data. The CF and immediate data D are binary-subtracted from working register Ry; the result is loaded to AC. * The carry flag (CF) will be affected. D = 0H ~ FH

SBCI* Ry, D

Function:	AC,Ry ← (Ry)+(D)B+CF
Description:	D represents an immediate data. The CF and immediate data D are binary-subtracted from working register Ry; the result is loaded to AC and working register Ry. * The carry flag (CF) will be affected. D = 0H ~ FH

ADDI Ry, D

Function:	AC ← (Ry)+D
Description:	D represents an immediate data. The contents of Ry and D are binary-ADDed; the result is loaded to AC. * The carry flag (CF) will be affected. D = 0H ~ FH

ADDI* Ry, D

Function:	AC, Ry ← (Ry)+D
Description:	D represents an immediate data. The contents of Ry and D are binary-ADDed; the result is loaded to AC and working register Ry. * The carry flag (CF) will be affected. D = 0H ~ FH

SUBI Ry, D

Function:	AC ← (Ry)+(D)B+1
Description:	D represents an immediate data. The immediate data D is binary-subtracted from working register Ry; the result is loaded to AC. * The carry flag (CF) will be affected. D = 0H ~ FH

SUBI* Ry, D

Function:	AC,Ry ← (Ry)+(D)B+1
Description:	D represents an immediate data. The immediate data D is binary-subtracted from working register Ry; the result is loaded to AC and working register Ry. * The carry flag (CF) will be affected. D = 0H ~ FH

ADNI Ry, D

Function:	AC ← (Ry)+D
Description:	D represents an immediate data. The contents of Ry and D are binary-ADDed; the result is loaded to AC. * The result will not affect the carry flag (CF). D = 0H ~ FH

ADNI* Ry, D

Function:	AC, Ry ← (Ry)+D
Description:	D represents an immediate data.
	The contents of Ry and D are binary-ADDed; the result is loaded to AC and working register Ry.
	* The result will not affect the carry flag (CF).
	D = 0H ~ FH

ANDI Ry, D

Function:	AC ← (Ry) & D
Description:	D represents an immediate data.
	The contents of Ry and D are binary-ANDed; the result is loaded to AC.
	D = 0H ~ FH

ANDI* Ry, D

Function:	AC, Ry ← (Ry) & D
Description:	D represents an immediate data.
	The contents of Ry and D are binary-ANDed; the result is loaded to AC and working register Ry.
	D = 0H ~ FH

EORI Ry, D

Function:	AC ← (Ry) ⊕ D
Description:	D represents an immediate data.
	The contents of Ry and D are exclusive-OREd; the result is loaded to AC.
	D = 0H ~ FH

EORI* Ry, D

Function:	AC, Ry ← (Ry) ⊕ D
Description:	D represents an immediate data.
	The contents of Ry and D are exclusive-OREd; the result is loaded to AC and working register Ry.
	D = 0H ~ FH

ORI Ry, D

Function:	AC ← (Ry) | D
Description:	D represents an immediate data.
	The contents of Ry and D are binary-OREd; the result is loaded to AC.
	D = 0H ~ FH

ORI* Ry, D

Function:	AC, Ry ← (Ry) | D
Description:	D represents an immediate data.
	The contents of Ry and D are binary-OREd; the result is loaded to AC and working register Ry.
	D = 0H ~ FH

5-4. LOAD/STORE INSTRUCTIONS

STA Rx

Function:	Rx ← (AC)
Description:	The content of AC is loaded to data memory specified by Rx.

LDS Rx, D

Function:	AC,Rx ← D
Description:	Immediate data D is loaded to the AC and data memory specified by Rx. D = 0H ~ FH

LDA Rx

Function:	AC ← (Rx)
Description:	The content of Rx is loaded to AC.

5-5. CPU CONTROL INSTRUCTIONS

NOP

Function:	no operation
Description:	no operation

HALT

Function:	Enters halt mode
Description:	The following 3 conditions cause the halt mode to be released. 1) An interrupt is accepted. 2) The signal change specified by the SCA instruction is applied to IOA. 3) The halt release condition specified by SHE instruction is met. When an interrupt is accepted to release the halt mode, the halt mode returns by executing the RTS instruction after completion of interrupt service.

STOP

Function:	Enters stop mode and stops all oscillators
Description:	Before executing this instruction, all signals on IOA port must be set to low. The following 2 conditions cause the stop mode to be released. 1) One of the signals on IOA port is "H".

SCA X

Function:	The data specified by X causes the halt mode to be released.
Description:	The signal change at ports IOA is specified. The bit meaning of X (X4) is shown below:

Bit pattern	Description
X5=1	Halt mode is released when signal applied to IOA

X7~6,X4~0 is reserved

SIE*　X

Function: Set/Reset interrupt enable flag

Description:

X0=1	The IEF0 is set so that interrupt 0(Signal change at port IOA specified by SCA) is accepted.
X3=1	The IEF3 is set so that interrupt 3(overflow from the predivider) is accepted.
X4=1	The IEF4 is set so that interrupt 4(underflow from Timer 2) is accepted.

X7, 6, 5, 2, 1 is reserved

SHE　X

Function: Set/Reset halt release enables flag

Description:

X3=1	The HEF3 is set so that the halt mode is released by predivider overflow.
X4=1	The HEF4 is set so that the halt mode is released by timer 2 underflow.

SRE　X

Function: Set/Reset stop release enable flag

Description:

X6=1	The SRF6 is set so that the stop mode is released by the signal changed on IOA pin.

MSB　Rx

Function: AC, Rx ← STS2

Description: The SCF2 and BCF flag contents are loaded to AC and the data memory specified by Rx.
The content of AC and meaning of bit after execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
NA	Start condition flag 2 (SCF2)	NA	Backup flag (BCF)
NA	Halt release caused by SCF6, 7	NA	Backup mode status

MSC　Rx

Function: AC, Rx ← STS3

Description: The PH15 and SCF7 contents are loaded to AC and the data memory specified by Rx.
The content of AC and meaning of bit after execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 7 (SCF7)	The content of 15th stage of the predivider	NA	NA
Halt release caused by predivider overflow		NA	NA

MCX Rx

Function: Rx, AC ← STS3X

Description: The contents of status register 3X (STS3X) is transferred to both accumulator (AC) and the data memory (RAM).
The meaning of each bit after execution is as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Start condition flag 0 (SCF0)	Start condition flag 6 (SCF6)	Reserved
Reserved	Halt release caused by the IOA port	Halt release caused by TMR2 underflow	Reserved
Reserved	Reserved	Read only	Reserved

5-6. DECIMAL ARITHMETIC INSTRUCTIONS

DAA

Function: AC ← BCD(AC)

Description: Converts the content of AC to decimal format, and then restores to AC.
When this instruction is executed, the AC must be the result of any added instruction.
* The carry flag (CF) will be affected.

DAA* Rx

Function: AC, Rx ← BCD(AC)

Description: Converts the content of AC to decimal format, and then restores to AC and data memory specified by Rx.
When this instruction is executed, the AC must be the result of any added instruction.
* The carry flag (CF) will be affected.

AC data before DAA execution	CF data before DAA execution	AC data after DAA execution	CF data after DAA execution
0 ≤ AC ≤ 9	CF = 0	no change	no change
A ≤ AC ≤ F	CF = 0	AC= AC+ 6	CF = 1
0 ≤ AC ≤ 3	CF = 1	AC= AC+ 6	no change

DAS

Function: AC ← BCD(AC)

Description: Converts the content of AC to decimal format, and then restores to AC.
When this instruction is executed, the AC must be the result of any subtracted instruction.
* The carry flag (CF) will be affected.

DAS* Rx

Function: AC, Rx ← BCD(AC)

Description: Converts the content of AC to decimal format, and then restores to AC and data memory specified by Rx.
When this instruction is executed, the AC must be the result of any subtracted instruction.

* The carry flag (CF) will be affected.

AC data before DAS execution	CF data before DAS execution	AC data after DAS execution	CF data after DAS execution
0 ≤ AC ≤ 9	CF = 1	No change	no change
6 ≤ AC ≤ F	CF = 0	AC= AC+A	no change

5-7. JUMP INSTRUCTIONS

JB0 X

Function:	Program counter jumps to X if AC0=1.
Description:	If bit0 of AC is 1, jump occurs.
	If 0, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JB1 X

Function:	Program counter jumps to X if AC1=1.
Description:	If bit1 of AC is 1, jump occurs.
	If 0, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JB2 X

Function:	Program counter jumps to X if AC2=1.
Description:	If bit2 of AC is 1, jump occurs.
	If 0, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JB3 X

Function:	Program counter jumps to X if AC3=1.
Description:	If bit3 of AC is 1, jump occurs.
	If 0, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JNZ X

Function:	Program counter jumps to X if (AC)! = 0.
Description:	If the content of AC is not 0, jump occurs.
	If 0, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JNC X

Function:	Program counter jumps to X if CF=0.
Description:	If the content of CF is 0, jump occurs.
	If 1, the PC increases by 1.
	The range of X is from 000H to 3FFH.

JZ X

Function:	Program counter jumps to X if (AC)=0.
Description:	If the content of AC is 0, jump occurs.

If 1, the PC increases by 1.
The range of X is from 000H to 3FFH.

JC X

Function:	Program counter jumps to X if CF=1.
Description:	If the content of CF is 1, jump occurs. If 0, the PC increases by 1. The range of X is from 000H to 3FFH.

JMP X

Function:	Program counter jumps to X.
Description:	Unconditional jump. The range of X is from 000H to 3FFH.

CALL X

Function:	STACK ← (PC)+1 Program counter jumps to X.
Description:	A subroutine is called. The range of X is from 000H to 3FFH.

RTS

Function:	PC ← (STACK)
Description:	A return from a subroutine occurs.

5-8. MISCELLANEOUS INSTRUCTIONS

SCC X

Function:	Setting the clock source for IOA chattering prevention.
Description:	The following table shows the meaning of each bit for this instruction:

Bit pattern	Clock source setting	Bit pattern	Clock source setting
(X2,X1,X0)=001	Chattering prevention clock = $\phi10$	(X2,X1,X0)=010	Chattering prevention clock = $\phi8$
(X2,X1,X0)=100	Chattering clock = $\phi6$		

X7, 5,4,3 is reserved

TM2 Rx

Function:	Select timer 2 clock source and preset timer 2.
Description:	The content of data memory specified by Rx and AC are loaded to timer 2 to start the timer. The following table shows the bit pattern for this instruction:

			Select clock			Setting value		
TM2 Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

The clock source option for timer 2

AC3	AC2	Clock source
0	0	ϕ9
0	1	ϕ3
1	0	ϕ15

TM2X X

Function: Selects timer 2 clock source and preset timer 2.

Description: The data specified by X (X5 ~ X0) is loaded to timer 2 to start the timer.
The following table shows the bit pattern for this instruction:

	Select clock			Setting value					
TM2X X	X8	X7	X6	X5	X4	X3	X2	X1	X0

The clock source option for timer 2

X7	X7	X6	Clock source
0	0	0	ϕ9
0	0	1	ϕ3
0	1	0	ϕ15
1	0	0	ϕ5
1	0	1	ϕ7
1	1	0	ϕ11
1	1	1	ϕ13

SF X

Function: Sets flag

Description: Description of each flag
X0 : "1" The CF is set to 1.
X1 : "1" The chip enters backup mode and BCF is set to 1.
X7, 6, 5, 4, 3, 2 is reserved

RF X

Function: Resets flag

Description: Description of each flag
X0 : "1" The CF is reset to 0.
X1 : "1" The chip is out of backup mode and BCF is reset to 0.
X7, 6, 5, 4, 3, 2 is reserved

SF2 X

Function: Sets flag

Description: Description of each flag
X2 : "1" Disables the LCD segment output.
X0 : "1" Reload timer 2 set.
X7~3,1 is reserved

RF2 X

Function:	Resets flag
Description:	Description of each flag X2 : "1" Enables the LCD segment output. X0 : "1" Reload timer 2 reset. X7~3,1 is reserved

PLC X

Function:	Pulse control
Description:	The pulse corresponding to the data specified by X is generated. X0 : "1" Halt release request flag HRF0 caused by the signal at I/O port A is reset. X3 : "1" Halt release request flag HRF3 caused by overflow from the predivider be reset. X4 : "1" Halt release request flag HRF4 caused by underflow from the timer 2 is reset and stops the operating of timer 2(TM2). X8 : "1" The last 5 bits of the predivider (15 bits) are reset. When executing this instruction, X3 must be set to "1". X7 ~ 5, 2, 1 are reserved.

ORDERING INFORMATION

The ordering information:

Ordering number	Package
TM8720-COD	Wafer / Dice with code

Appendix A TM8720 Instruction Table

Instruction		Machine Code	Function		Flag/Remark
NOP		0000 0000 0000 0000	No Operation		
LCP	Lz,Ry	0000 0110 ZZZZ YYYY	Lz	← Ry	Lz : 00h~0Bh
OPA	Rx	0000 1010 01XX XXXX	PortA(IOA)	← Rx	
OPB	Rx	0000 1100 01XX XXXX	PortB(IOB)	← Rx	
ADC	Rx	0010 0000 01XX XXXX	AC	← Rx + AC + CF	CF
ADC*	Rx	0010 0001 01XX XXXX	AC,Rx	← Rx + AC + CF	CF
SBC	Rx	0010 0010 01XX XXXX	AC	← Rx + ACB + CF	CF
SBC*	Rx	0010 0011 01XX XXXX	AC,Rx	← Rx + ACB + CF	CF
ADD	Rx	0010 0100 01XX XXXX	AC	← Rx + AC	CF
ADD*	Rx	0010 0101 01XX XXXX	AC,Rx	← Rx + AC	CF
SUB	Rx	0010 0110 01XX XXXX	AC	← Rx + ACB + 1	CF
SUB*	Rx	0010 0111 01XX XXXX	AC,Rx	← Rx + ACB + 1	CF
ADN	Rx	0010 1000 01XX XXXX	AC	← Rx + AC	
ADN*	Rx	0010 1001 01XX XXXX	AC,Rx	← Rx + AC	
AND	Rx	0010 1010 01XX XXXX	AC	← Rx AND AC	
AND*	Rx	0010 1011 01XX XXXX	AC,Rx	← Rx AND AC	
EOR	Rx	0010 1100 01XX XXXX	AC	← Rx EOR AC	
EOR*	Rx	0010 1101 01XX XXXX	AC,Rx	← Rx EOR AC	
OR	Rx	0010 1110 01XX XXXX	AC	← Rx OR AC	
OR*	Rx	0010 1111 01XX XXXX	AC,Rx	← Rx OR AC	
ADCI	Ry,D	0011 0000 DDDD YYYY	AC	← Ry + D + CF	CF
ADCI*	Ry,D	0011 0001 DDDD YYYY	AC,Ry	← Ry + D + CF	CF
SBCI	Ry,D	0011 0010 DDDD YYYY	AC	← Ry + DB + CF	CF
SBCI*	Ry,D	0011 0011 DDDD YYYY	AC,Ry	← Ry + DB + CF	CF
ADDI	Ry,D	0011 0100 DDDD YYYY	AC	← Ry + D	CF
ADDI*	Ry,D	0011 0101 DDDD YYYY	AC,Ry	← Ry + D	CF
SUBI	Ry,D	0011 0110 DDDD YYYY	AC	← Ry + DB + 1	CF
SUBI*	Ry,D	0011 0111 DDDD YYYY	AC,Ry	← Ry + DB + 1	CF
ADNI	Ry,D	0011 1000 DDDD YYYY	AC	← Ry + D	
ADNI*	Ry,D	0011 1001 DDDD YYYY	AC,Ry	← Ry + D	
ANDI	Ry,D	0011 1010 DDDD YYYY	AC	← Ry AND D	
ANDI*	Ry,D	0011 1011 DDDD YYYY	AC,Ry	← Ry AND D	
EORI	Ry,D	0011 1100 DDDD YYYY	AC	← Ry EOR D	
EORI*	Ry,D	0011 1101 DDDD YYYY	AC,Ry	← Ry EOR D	
ORI	Ry,D	0011 1110 DDDD YYYY	AC	← Ry OR D	
ORI*	Ry,D	0011 1111 DDDD YYYY	AC,Ry	← Ry OR D	
INC*	Rx	0100 0000 01XX XXXX	AC,Rx	← Rx + 1	CF
DEC*	Rx	0100 0001 01XX XXXX	AC,Rx	← Rx - 1	CF
IPA	Rx	0100 0010 01XX XXXX	AC,Rx	← PortA(IOA4)	
IPB	Rx	0100 0100 01XX XXXX	AC,Rx	← PortB(IOB4~2)	
MAF	Rx	0100 1010 01XX XXXX	AC,Rx	← STS1	B3 : CF B2 : ZERO B1, B0 :(Unused)
MSB	Rx	0100 1011 01XX XXXX	AC,Rx	← STS2	B3 : (Unused) B2 : SCF2(HRx) B1 : (Unused) B0 : BCF
MSC	Rx	0100 1100 01XX XXXX	AC,Rx	← STS3	B3 : SCF7(PDV) B2 : PH15 B1, B0: (Unused)
MCX	Rx	0100 1101 01XX XXXX	AC,Rx	← STS3X	B3 : (Unused) B2 : SCF0(APT)

Instruction		Machine Code	Function		Flag/Remark
					B1 : SCF6(TM2) B0 : (Unused)
SR0	Rx	0101 0000 01XX XXXX	ACn, Rxn AC3, Rx3	← Rx(n+1) ← 0	
SR1	Rx	0101 0001 01XX XXXX	ACn, Rxn AC3, Rx3	← Rx(n+1) ← 1	
SL0	Rx	0101 0010 01XX XXXX	ACn, Rxn AC0, Rx0	← Rx(n-1) ← 0	
SL1	Rx	0101 0011 01XX XXXX	ACn, Rxn AC0, Rx0	← Rx(n-1) ← 1	
DAA		0101 0100 0000 0000	AC	← BCD(AC)	
DAA*	Rx	0101 0101 01XX XXXX	AC,Rx	← BCD(AC)	
DAS		0101 0110 0000 0000	AC	← BCD(AC)	
DAS*	Rx	0101 0111 01XX XXXX	AC,Rx	← BCD(AC)	
LDS	Rx,D	0101 1DDD D1XX XXXX	AC,Rx	← D	
STA	Rx	0110 1000 01XX XXXX	Rx	← AC	
LDA	Rx	0110 1100 01XX XXXX	AC	← Rx	
MRA	Rx	0110 1101 01XX XXXX	CF	← Rx3	
MRW	Ry,Rx	0111 0YYY Y1XX XXXX	AC,Ry	← Rx	
MWR	Rx,Ry	0111 1YYY Y1XX XXXX	AC,Rx	← Ry	
JB0	X	1000 00XX XXXX XXXX	PC	← X	if AC0 = 1
JB1	X	1000 10XX XXXX XXXX	PC	← X	if AC1 = 1
JB2	X	1001 00XX XXXX XXXX	PC	← X	if AC2 = 1
JB3	X	1001 10XX XXXX XXXX	PC	← X	if AC3 = 1
JNZ	X	1010 00XX XXXX XXXX	PC	← X	if AC ≠ 0
JNC	X	1010 10XX XXXX XXXX	PC	← X	if CF = 0
JZ	X	1011 00XX XXXX XXXX	PC	← X	if AC = 0
JC	X	1011 10XX XXXX XXXX	PC	← X	if CF = 1
CALL	X	1100 00XX XXXX XXXX	STACK PC	← PC + 1 ← X	
JMP	X	1101 00XX XXXX XXXX	PC	← X	
RTS		1101 1000 0000 0000	PC	← STACK	CALL Return
SCC	X	1101 1001 0010 0XXX	X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1101 1010 00X0 0000	X5	: Enable SEF5	IOA4~1
SPA	X	1101 1100 000X XXXX	X4 X3~0	: Set A4-1 Pull-Low : Set A4-1 I/O	 1:Output, 0: Input
SPB	X	1101 1101 000X XXX1	X4 X3~1	: Set B4-2 Pull-Low : Set B4-2 I/O	 1:Output, 0: Input
TM2	Rx	1110 0100 01XX XXXX	Timer2	← Rx & AC	
TM2X	X	1110 011X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer2 Value	
SHE	X	1110 1000 000X X000	X4 X3	: Enable HEF4 : Enable HEF3	TMR2 PDV
SIE*	X	1110 1001 000X X00X	X4 X3 X0	: Enable IEF4 : Enable IEF3 : Enable IEF0	TMR2 PDV APT

Instruction		Machine Code	Function		Flag/Remark
PLC	X	1110 101X 000X X00X	X8	: Reset PH15~11	
			X4	: Reset HRF4	TMR2
			X3	: Reset HRF3	PDV
			X0	: Reset HRF0	APT
SRE	X	1110 1101 0X00 0000	X6	: Enable SRF6	SRF6(APT)
SF		1111 0000 0000 00XX	X1	: BCF Set	BCF
			X0	: CF Set	CF
RF		1111 0100 0000 00XX	X1	: BCF Reset	BCF
			X0	: CF Reset	CF
SF2	X	1111 1000 0000 0X0X	X2	: Close all Segments	RSOFF
			X0	: Reload 2 Set	RL2
RF2	X	1111 1001 0000 0X0X	X2	: Release Segments	RSOFF
			X0	: Reload 2 Reset	RL2
ALM	X	1111 101X XXXX XXXX	X8,7,6=100	: DC1	
			X8,7,6=011	: PH3	
			X8,7,6=010	: PH4	
			X8,7,6=001	: PH5	
			X8,7,6=000	: DC0	
			X5~0	← PH15~10	
HALT		1111 1110 0000 0000	Halt Operation		
STOP		1111 1111 0000 0000	Stop Operation		

Symbol Description

AC	: Accumulator	ACn	: Accumulator Bit n
BCF	: Back-up Flag	BCLK	: System clock, stop only in STOP condition
CF	: Carry Flag	Cch	: Clock source of Chattering Detector
D	: Immediate Data	HEFn	: Halt Release Enable Flag
HRFn	: HALT Release Flag	IEFn	: Interrupt Enable Flag
Lz	: LCD Latch	PC	: Program Counter
PDV	: Pre-Divider	Rx	: Memory of Address X
Rxn	: Memory Bit n of Address X	Ry	: Memory of working register Y
SCFn	: Start Condition Flag	SEFn	: Switch Enable Flag
SRFn	: STOP Release Enable Flag	TMR	: Timer Overflow Release Flag
X	: Address	ZERO	: Zero Flag
()	: Content of Register		