十速科技股份有限公司
tenx technology inc.

TM8740

4-Bit Micro-Controller with LCD Driver

User's Manual

tenx technology, inc.

CONTENTS

Chapter 1 General Description

1-1. GENERAL DESCRIPTION

The TM8740 is an embedded high-performance 4-bit microcomputer with LCD driver. It contains all the of the following functions in a single chip : 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, Voltage to Frequency Converter(VFC), Low Battery Detect(LBD), LCD driver, look-up table, watchdog timer and key matrix scanning circuitry.

1-2. FEATURES

1. Low power dissipation.

2. Powerful instruction set (177 instructions).
   - Binary addition, subtraction, BCD adjust, logical operation in direct and index addressing mode.
   - Single-bit manipulation (set, reset, decision for branch).
   - Various conditional branches.
   - 8 working registers and manipulation.
   - Table look-up.
   - LCD driver data transfer.

3. Memory capacity.
   - ROM capacity 4096 x 16 bits.
   - RAM capacity 512 x 4 bits.

4. Input/output ports.
   - Port IOA3,4 2 pins (with internal pull-low).
   - Port IOB 4 pins (with internal pull-low).
   - Port IOC 4 pins (with internal pull-low, low-level-hold, chattering clock).
   - Port IOD 4 pins (with internal pull-low, chattering clock).

5. 8 level subroutine nesting.

6. Interrupt function.
   - External factor 4 (INT pin, Port IOC, IOD & KI input).
   - Internal factor 4 (Pre-Divider, Timer1,Timer2 & VFC).

7. Built in Alarm, clock or single tone melody generator.
   - BZB, BZ(Muxed with SEG30, SEG31).
8. Built-in 2 channels Voltage to Frequency Converter circuit.
   - PIN1, PIN2, NIN1, NIN2, OPO, VREF, AVDD, AGND

9. Built-in Low Battery Detect
10. Built in KEY_BOARD scanning function.
    - K1~K16 (Share with SEG1~SEG16).
    - KI1~KI4 (Mux with IOC1~IOC4).

11. Two 6-bit programmable timer with programmable clock source.

12. Watch dog timer.

13. LCD driver output.
    - 9 common output and 34 segment outputs (up to drive 306 LCD segments).

- 1/2 Duty, 1/3 Duty, 1/4 Duty,1/5 Duty,1/6Duty,1/7Duty,1/8Duty or 1/9Duty cab be selected by MASK option.
- 1/2 Bias,1/3 Bias or 1/4 Bias can be selected by MASK option.
- Single instruction to turn off all segments.
- Mask option is used to select COM5~9,SEG1~18,26~41 as DC outputs/P_open drain.

14. Built-in Voltage doubler, halver, tripler, quartic charge pump circuit.

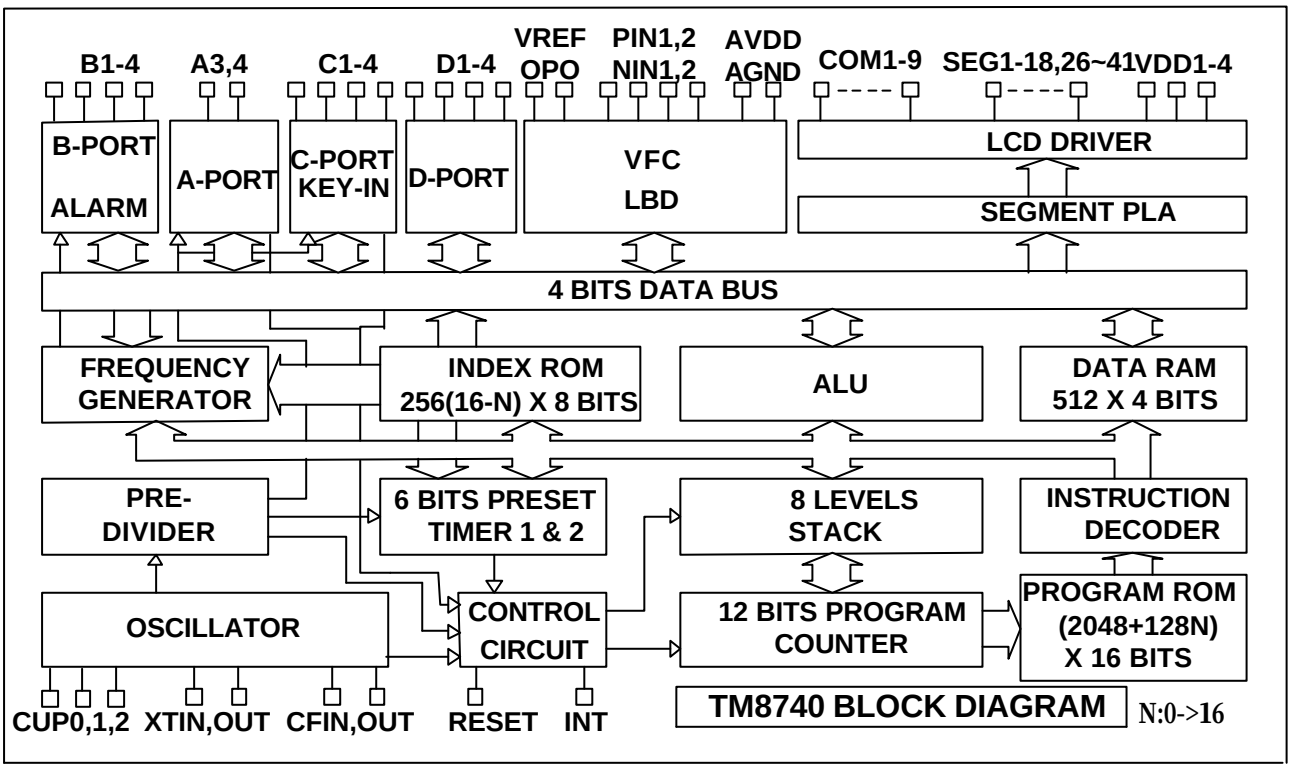
15. Dual clock operation.

- Slow clock oscillation can be defined as X'tal or external RC type oscillator by MASK option.
- Fast clock oscillation set to 3.58MHz ceramic resonator. For Fast only, PH0 set to BCLK/16.

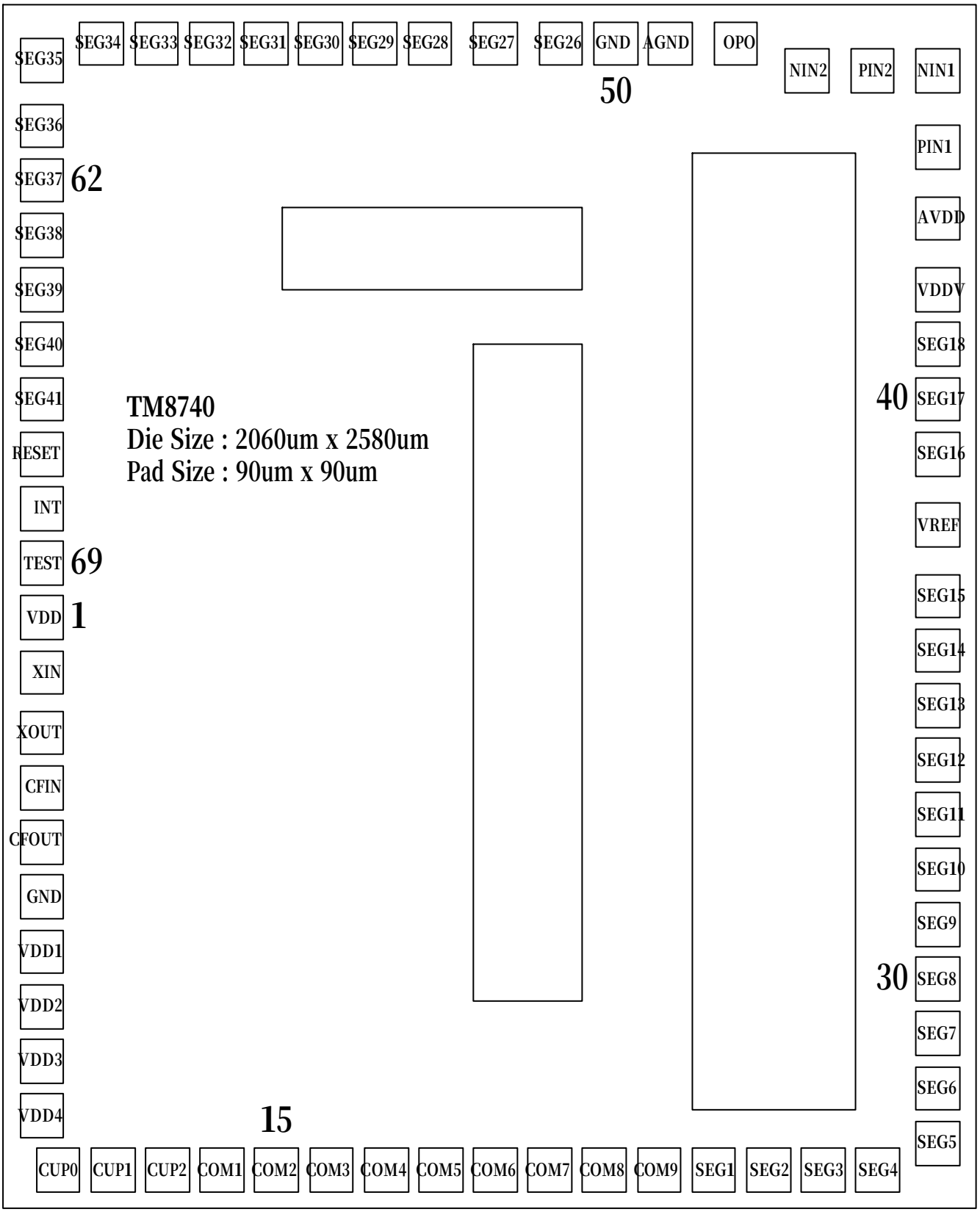
16. HALT function.

17. STOP function.

1-3. BLOCK DIAGRAM



1-4. PAD DIAGRAM



The substrate of the chip should be connected to the GND.

1-5. PAD COORDINATE

No	Name	X	Y	No	Name	X	Y
1	BAK	72.50	1229.50	36	SEG14/K14	1974.50	1149.40
2	XIN	72.50	1114.50	37	SEG15/K15	1974.50	1264.40
3	XOUT	72.50	999.50	38	VREF	1968.90	1443.60
4	CFIN	72.50	884.50	39	SEG16/K16	1968.90	1609.45
5	CFOUT	72.50	769.50	40	SEG17	1968.90	1724.45
6	GND	72.50	654.50	41	SEG18	1968.90	1839.45
7	VDD1	72.50	539.50	42	VDDV(VDD2)	1968.90	1954.45
8	VDD2	72.50	424.50	43	AVDD	1968.90	2111.30
9	VDD3	72.50	309.50	44	PIN1	1968.90	2259.35
10	VDD4	72.50	194.50	45	NIN1	1968.90	2454.05
11	CUP0	89.50	72.50	46	PIN2	1843.90	2454.45
12	CUP1	204.50	72.50	47	NIN2	1716.10	2454.05
13	CUP2	319.50	72.50	48	OPO	1556.35	2501.50
14	COM1	434.50	72.50	49	AGND	1399.50	2507.50
15	COM2	549.50	72.50	50	GND	1279.50	2507.50
16	COM3	669.50	72.50	51	SEG26/IOA3	1164.50	2507.50
17	COM4	789.50	72.50	52	SEG27/IOA4	1024.00	2507.50
18	COM5	909.50	72.50	53	SEG28/IOB1	881.50	2507.50
19	COM6	1029.50	72.50	54	SEG29/IOB2	766.50	2507.50
20	COM7	1149.50	72.50	55	SEG30/IOB3/BZB	651.50	2507.50
21	COM8	1269.50	72.50	56	SEG31/IOB4/BZ	536.50	2507.50
22	COM9	1389.50	72.50	57	SEG32/IOC1/KI1	421.50	2507.50
23	SEG1/K1	1509.50	72.50	58	SEG33/IOC2/KI2	306.50	2507.50
24	SEG2/K2	1629.50	72.50	59	SEG34/IOC3/KI3	191.50	2507.50
25	SEG3/K3	1744.50	72.50	60	SEG35/IOC4/KI4	72.50	2477.00
26	SEG4/K4	1859.50	76.20	61	SEG36/IOD1	72.50	2300.00
27	SEG5/K5	1974.50	114.40	62	SEG37/IOD2	72.50	2175.00
28	SEG6/K6	1974.50	229.40	63	SEG38/IOD3	72.50	2049.50
29	SEG7/K7	1974.50	344.40	64	SEG39/IOD4	72.50	1934.50
30	SEG8/K8	1974.50	459.40	65	SEG40	72.50	1819.50
31	SEG9/K9	1974.50	574.40	66	SEG41	72.50	1704.50
32	SEG10/K10	1974.50	689.40	67	RESET	72.50	1589.50
33	SEG11/K11	1974.50	804.40	68	INT	72.50	1474.50
34	SEG12/K12	1974.50	919.40	69	TEST	72.50	1359.50
35	SEG13/K13	1974.50	1034.40				

1-6. PIN DESCRIPTION

Name	I/O	Description
BAK	P	Positive Back-up voltage. Connect a 0.1u capacitance to GND.
VDD2	P	Positive supply voltage & LCD supply voltage.
VDD1,3,4	P	LCD supply voltage.
RESET	I	Input pin from LSI reset request signal, with internal pull-down resistor. Reset Time can select "PH15/2" or "PH12/2" by option. Reset Type can select "Level" or "Pulse" by mask option.
INT	I	Input pin for external INT request signal. Falling edge or rising edge triggered by mask option. Internal pull-down or pull-up resistor be selected by mask option.
TEST		Test signal input pin.
CUP0,1,2	O	Switching pins for supply the LCD driving voltage to the VDD1,2,3,4 pins. Connect the CUP1 and CUP2 pins with non-polarized electrolytic capacitor if 1/2, 1/3 or 1/4 bias mode has been selected, and also connect the CUP0 and CUP1 pins with non-polarized electrolytic capacitor if 1/4 bias mode has been selected. In no BIAS mode, these pins should be open
XIN XOUT	I O	Time base counter frequency (clock specified. LCD alternating frequency. Alarm signal frequency) or system clock oscillation for SLOW Only or DUAL by mask option.. 32KHz Crystal oscillator or external RC. If clock is 32Khz Crystal, XIN or XOUT must adder to capacitance.
CFIN CFOUT	I O	System clock oscillation for FAST Only or DUAL(execution of FAST instruction) Connected with 3.58MHz ceramic resonator or external R by mask option Oscillation stop at the execution of STOP,HALT,SF(X2=1&X3=1) or SLOW instruction
COM1~9	O	Output pins for driving the common pins of the LCD panel. COM5~9 can be defined as either COMS or Open Drain type output. (MASK option)
SEG1~18, 26~41	O	Output pins for driving the LCD panel segment.
IOA3,4	I/O	Input / Output port A, can use software to define internal pull-low resistor.. This port is muxed with SEG26,27, and set by mask option.
IOB1-4	I/O	Input / Output port B, can use software to define internal pull-low resistor. This port is muxed with SEG28~31 / BZB,BZ, and set by mask option.
IOC1-4	I/O	Input / Output port C, can use software to define internal pull-low / low-level-hold Resistor and Chattering clock to reduce input bounce. This port is muxed with SEG32~35 / KI1~4, and set by mask option.
IOD1~4	I/O	Input / Output port D, can use software to define internal pull-low Resistor, and Chattering clock to reduce input bounce. This port is muxed with SEG36~39, and set by mask option.
PIN1,2	I	Input port for VFC OP positive pins.
NIN1,2	I	Input port for VFC OP negative pins.
OPO	O	OP output voltage.
VREF	O	Reference voltage.
VDDV	P	Positive voltage supply VFC
AVDD	P	Analog Power Output
AGND	P	Analog Ground
(ALM) BZB/BZ	O	Output port for alarm, frequency or melody generator This port is muxed with SEG30~31 / IOB3~4, and set by mask option.
K1~K16	O	Output port for key matrix scanning (Shared with SEG1~SEG16).
KI1~4	I	Keyboard scanning input port. This port is muxed with SEG32~35 / IOC1~4, and set by mask option.
GND	P	Negative supply voltage.

1-7. CHARACTERIZATION

ABSOLOUTE MAXIMUM RATINGS

GND= 0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD1	-0.3 to 5.5	V
	VDD2	-0.3 to 5.5	V
	VDD3	-0.3 to 8.5	V
	VDD4	-0.3 to 8.5	V
Maximum Input Voltage	Vin	-0.3 to VDD1/2+0.3	V
Maximum output Voltage	Vout1	-0.3 to VDD1/2+0.3	V
	Vout2	-0.3 to VDD3+0.3	V
	Vout3	-0.3 to VDD4+0.3	V
Maximum Operating Temperature	Topg	-20 to +70	℃
Maximum Storage Temperature	Tstg	-25 to +125	℃

POWER CONSUMPTION

at Ta=-20℃ to 70℃,GND= 0V

Name	Sym.	Condition	Min.	Typ.	Max.	Unit
HALT mode	IHALT	Only 32.768KHz Crystal oscillator operating, without loading. VDD2=3.0V, BCF = 0		5		uA
STOP mode	ISTOP				1	uA

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

ALLOWABLE OPERATING CONDITIONS

at Ta=-20℃ to 70℃,GND= 0V

Name	Symb.	Condition	Min.	Max.	Unit
Supply Voltage	VDD1		1.2	5.25	V
	VDD2		2.4	5.25	V
	VDD3		2.4	8.0	V
	VDD4		2.4	8.0	V
Oscillator Start-Up Voltage	VDDB	Crystal Mode	1.3		V
Oscillator Sustain Voltage	VDDB	Crystal Mode	1.2		V
Supply Voltage	VDD2		2.4	5.25	V
Input "H" Voltage	Vih2		VDD2-0.7	VDD2+0.7	V
Input "L" Voltage	Vil2		-0.7	0.7	V
Operating Freq	Fopg1	Crystal Mode	32		KHZ
	Fopg2	RC Mode	10	1000	KHZ

DC Output Characteristics

Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit
Output "H" Voltage	Voh2c	Ioh=-1mA	COM5~9	1.5	1.8	2.1	V
Output "L" Voltage	Vol2c	Iol=2mA	SEG1~18,26~41	0.3	0.6	0.9	V

ELECTRICAL CHARACTERISTICS

Input Resistance

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
"L" Level Hold Tr(IOC)	Rllh2	Vi=0.2VDD2	10	40	100	Kohm
IOC Pull-Down Tr	Rmad2	Vi=VDD2	200	500	1000	Kohm
INT Pull-up Tr	Rintu2	Vi=VDD2	200	500	1000	Kohm
INT Pull-Down Tr	Rintd2	Vi=GND	200	500	1000	Kohm
RES Pull-Down R	Rres2	Vi=GND or VDD2	10	50	100	Kohm

Segment Driver Output Characteristics

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
Static Display Mode							
Output "H" Voltage	Voh2d	Ioh=-1uA	SEG-n	2.2			V
Output "L" Voltage	Vol2d	Iol=1uA				0.2	V
Output "H" Voltage	Voh2e	Ioh=-10uA	COM-n	2.2			V
Output "L" Voltage	Vol2e	Iol=10uA				0.2	V
1/2 Bias Display Mode							
Output "H" Voltage	Voh12f	Ioh=-1uA	SEG-n	2.2			V
Output "L" Voltage	Vol12f	Iol=1uA				0.2	V
Output "H" Voltage	Voh12g	Ioh=-10uA	COM-n	2.2			V
Output "M" Voltage	Vom12g	Iol/h=+/-10uA		1.0		1.4	V
1/3 Bias display Mode							
Output "H" Voltage	Voh12h	Ioh=-1uA	SEG-n	3.4			V
Output "M1" Voltage	Vom1h	Iol/h=+/-10uA		1.0		1.4	V
Output "M2" Voltage	Vom22h	Iol/h=+/-10uA		2.2		2.6	V
Output "L" Voltage	Vol12h	Iol=1uA				0.2	V
Output "H" Voltage	Voh12i	Ioh=-10uA	COM-n	3.4			V
Output "M1" Voltage	Vom12i	Iol/h=+/-10uA		1.0		1.4	V
Output "M2" Voltage	Vom22i	Iol/h=+/-10uA		2.2		2.6	V
Output "L" Voltage	Vol12i	Iol=10uA				0.2	V
1/4 Bias display Mode							
Output "H" Voltage	Voh12j	Ioh=-1uA	SEG-n	4.6			V
Output "M2" Voltage	Vom22j	Iol/h=+/-10uA		2.2		2.6	V
Output "L" Voltage	Vol12j	Iol=1uA				0.2	V
Output "H" Voltage	Voh12k	Ioh=-10uA	COM-n	4.6			V
Output "M1" Voltage	Vom12k	Iol/h=+/-10uA		1.0		1.4	V
Output "M3" Voltage	Vom22k	Iol/h=+/-10uA		3.4		3.8	V
Output "L" Voltage	Vol12k	Iol=10uA				0.2	V

RECOMMENDED OPERATING CONDITION

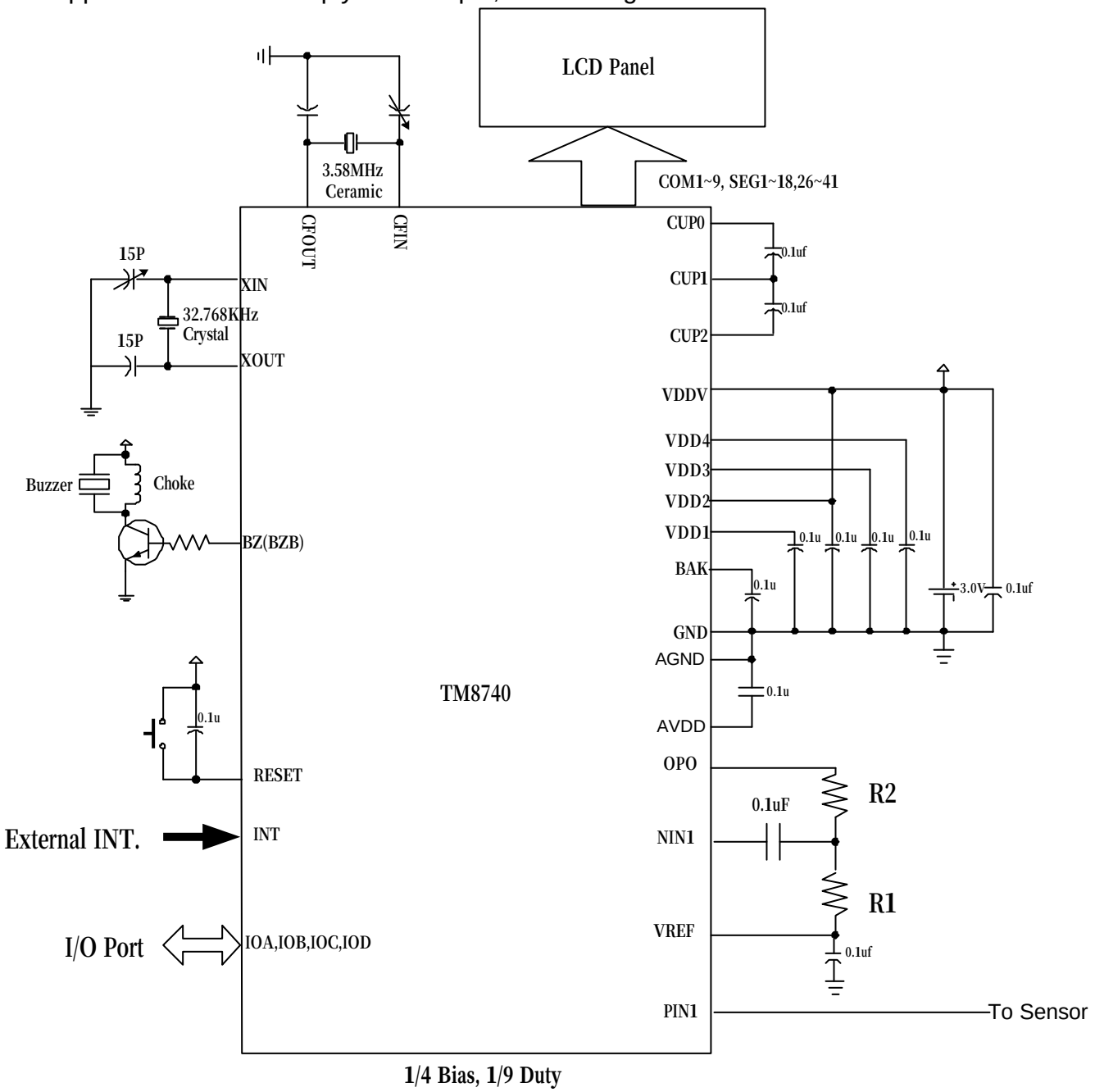
Symbol	Parameter	Min.	Typ.	Max.	Units
AZ	Auto-zero time	70			ms
VFCCK	VFC detect clock		125	200	KHz
LBDCK	Low-Battery detect clock		125		KHz
LBE	Low-Battery detect enable		1		u sec
VREF	VREF generate time	100			u sec

Analog Circuit Characteristics

Name	Symb	Condition	Min.	Typ.	Max.	Unit
VREF	V_{ref}	Connect 0.1uF capacitances between V_{ref} and AGND	1.10	1.14	1.17	V
LBD	V_{lbd}		2.25	2.40	2.55	V

1-8. TYPICAL APPLICATION CIRCUIT

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



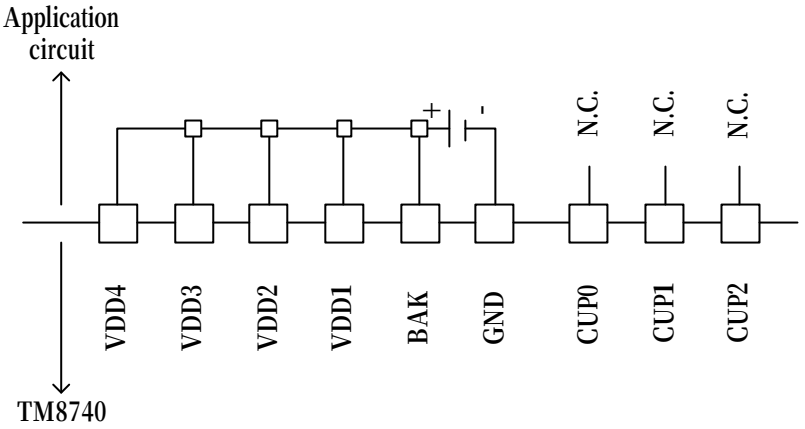
1/4 Bias, 1/9 Duty

Chapter 2 TM8740 Internal System Architecture

2-1 Power Supply

TM8740 can operate voltage range: 2.4V ~ 3.6V. The power supply circuitry also generates the necessary voltage level for driving the LCD panel with a different bias. Shown below are the connection diagrams for 1/2 bias, 1/3 bias, 1/4 bias, and no bias applications.

2-1-1. NO BIAS AT Li BATTERY POWER SUPPLY



MASK OPTION table :

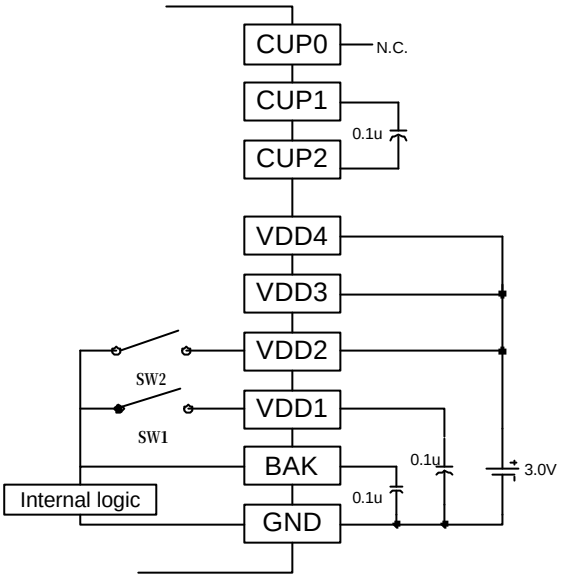
Mask Option name	Selected item
BIAS	(1) NO BIAS

Note 1: The input/output ports operate between GND and VDD2.

2-1-2. 1/2 BIAS AT Li BATTERY POWER SUPPLY

The backup flag (BCF) must be reset after the operation of the halver circuit is fully stabilized and a voltage of approximately 1/2 * VDD2 appears on the VDD1 pin.

Backup flag(BCF)	SW1	SW2
BCF=0	ON	OFF
BCF=1	OFF	ON



MASK OPTION table :

Mask Option name	Selected item
BIAS	(2) 1/2 BIAS

Note 1: The input/output ports operate between GND and VDD2.

Note 2: The backup flag (BCF) is set in the initial clear mode. When the backup flag is set, the internal logic operated on VDD2 and the oscillator circuit becomes large in driver size.

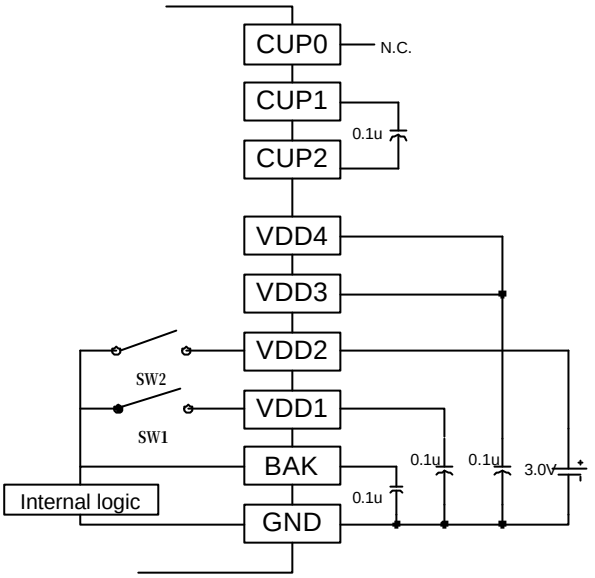
When the backup flag is set, the operating current is increased. Therefore, the backup flag should be reset unless otherwise required. For information on the backup flag, refer to 3-5.

Note 3: The VDD1 level (≈1/2 * VDD2) at the off-state of SW1 is used as an intermediate voltage level for the LCD driver.

2-1-3. 1/3 BIAS AT Li BATTERY POWER SUPPLY

The backup flag (BCF) must be reset after the operation of the halver circuit is fully stabilized and a voltage of approximately 1/2 * VDD2 appears on the VDD1 pin.

Backup flag(BCF)	SW1	SW2
BCF=0	ON	OFF
BCF=1	OFF	ON



MASK OPTION table :

Mask Option name	Selected item
BIAS	(3) 1/3 BIAS

Note 1: The input/output ports operate between GND and VDD2.

Note 2: The backup flag (BCF) is set in the initial clear mode. When the backup flag is set, the internal logic operated on VDD2 and the oscillator circuit becomes large in driver size.

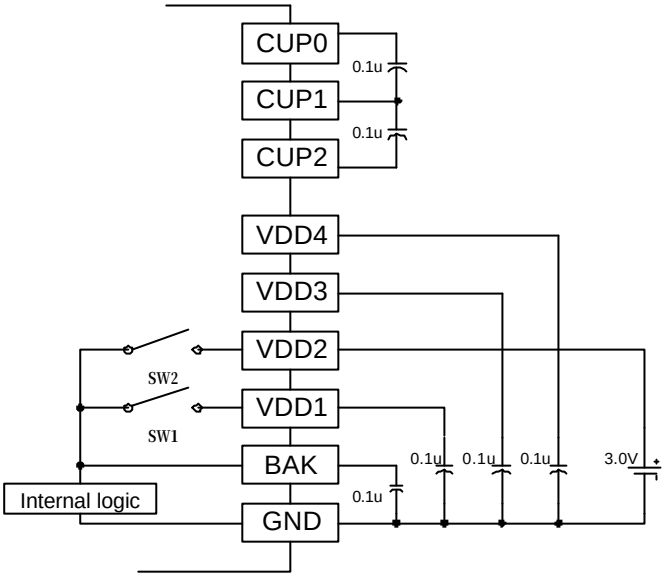
When the backup flag is set, the operating current is increased. Therefore, the backup flag should be reset unless otherwise required. For information on the backup flag, refer to 3-5.

Note 3: The VDD1 level (≈ 1/2 * VDD) at the off-state of SW1 is used as an intermediate voltage level for LCD driver.

2-1-4. 1/4 BIAS AT Li BATTERY POWER SUPPLY

The backup flag (BCF) must be reset after the operation of the halver circuit is fully stabilized and a voltage of approximately 1/2 * VDD2 appears on the VDD1 pin.

Backup flag(BCF)	SW1	SW2
BCF=0	ON	OFF
BCF=1	OFF	ON



It is recommanded that the option "LCD reset OFF" is not used in this power mode, as the LCD segments cannot be turned off completely in the RESET cycle.

MASK OPTION table :

Mask Option name	Selected item
BIAS	(4) 1/4 BIAS

Note 1: The input/output ports operate between GND and VDD2.

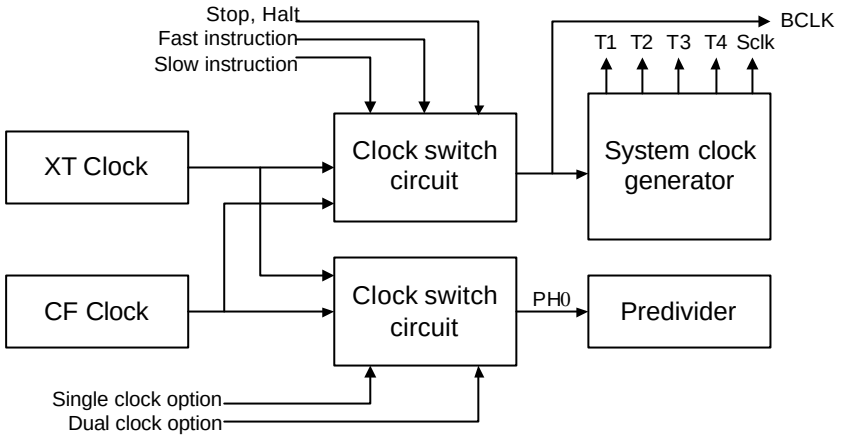
Note 2: The backup flag (BCF) is set in the initial clear mode. When the backup flag is set, the internal logic operated on VDD2 and the oscillator circuit becomes large in driver size.

When the backup flag is set, the operating current is increased. Therefore, the backup flag must be reset unless otherwise required. For information on the backup flag, refer to 3-5.

Note 3: The VDD1 level (≈ 1/2 * VDD) at the off-state of SW1 is used as an intermediate voltage level for LCD driver.

2-2. SYSTEM CLOCK

The XT clock (slow clock oscillator) and CF clock (fast clock oscillator) compose the clock oscillation circuitry and the block diagram is shown below.



The system clock generator provides the necessary clocks for execution of instruction. The pre-divider generates several clocks with different frequencies for the LCD driver, frequency generator, etc. to use.
The following table shows the clock sources of system clock generators and pre-divider under different conditions.

	PH0	BCLK
fast clock only option	CF clock	CF clock
Initial state(dual clock option)	XT clock	XT clock
Halt mode(dual clock option)	XT clock	XT clock
Slow mode(dual clock option)	XT clock	XT clock
Fast mode(dual clock option)	XT clock	CF clock

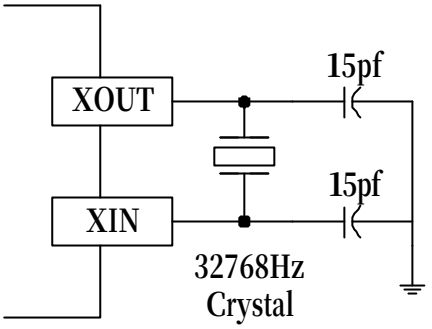
2-2-1 CONNECTION DIAGRAM OF SLOW CLOCK OSCILLATOR (XT CLOCK)

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, otherwise it will be active all the time after the initial reset. In stop mode, the oscillator will be stopped.
There are 2 types oscillators which can be used in slow clock oscillators; select with the mask option:

2-2-1-1. External 32.768KHz Crystal oscillator

MASK OPTION table :

Mask Option name	Selected item
SLOW CLOCK TYPE FOR DUAL	(1) X'tal



(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. This increases the power consumption. Therefore, the backup flag should be reset unless otherwise required.

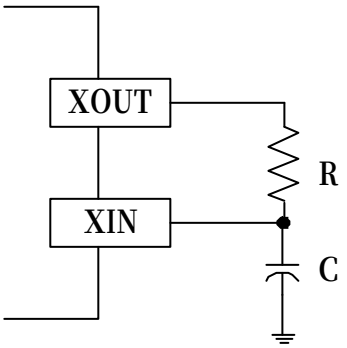
The following table shows the power consumption of Crystal oscillator under different conditions:

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

2-2-1-2. External RC oscillator

MASK OPTION table :

Mask Option name	Selected item
SLOW CLOCK TYPE FOR DUAL	(2) RC



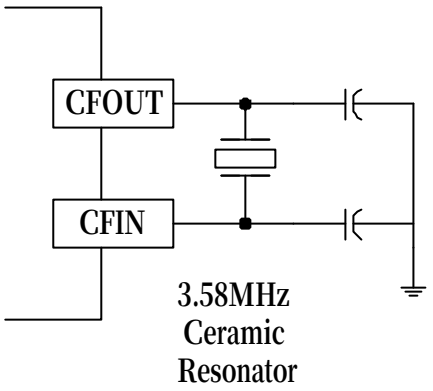
(2) RC

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 single clock operation (fast only), this oscillator will provide the clock to the system clock generator, pre-divider, timer, I/O port chattering prevention clock and LCD circuitry. In dual clock operation, CF clock provides the clock to the system clock generator only.

When the dual clock option is selected by mask option, this oscillator will be inactive most of the time except when the FAST instruction is executed. After the FAST instruction is executed, the clock source (BCLK) of the system clock generator will be switched to CF clock, and the clock source for other functions will continue to come from XT clock. Halt mode, stop mode or SLOW instruction execution will stop this oscillator, after which the system clock (BCLK) will be switched to XT clock.

2-2-2-1. External 3.58MHz Ceramic Resonator oscillator



Notes : 1. Don't use 3.58MHz Ceramic Resonator as the oscillator when the Ag battery option is used.
2. When the program has to reset the BCF flag to 0 in Li battery power mode, don't use a 3.58MHz Ceramic Resonator as the oscillator.

2-2-3. COMBINATION OF THE CLOCK SOURCES

There are three combinations of clock sources that can be selected by mask option:

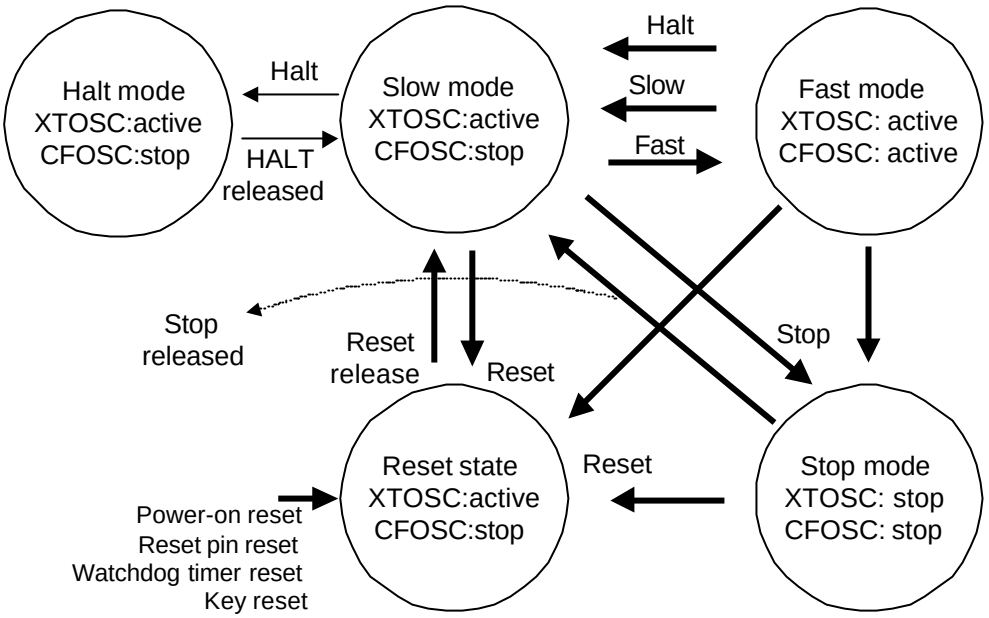
2-2-3-1 Dual Clock

MASK OPTION table :

Mask Option name	Selected item
CLOCK SOURCE	(2) DUAL

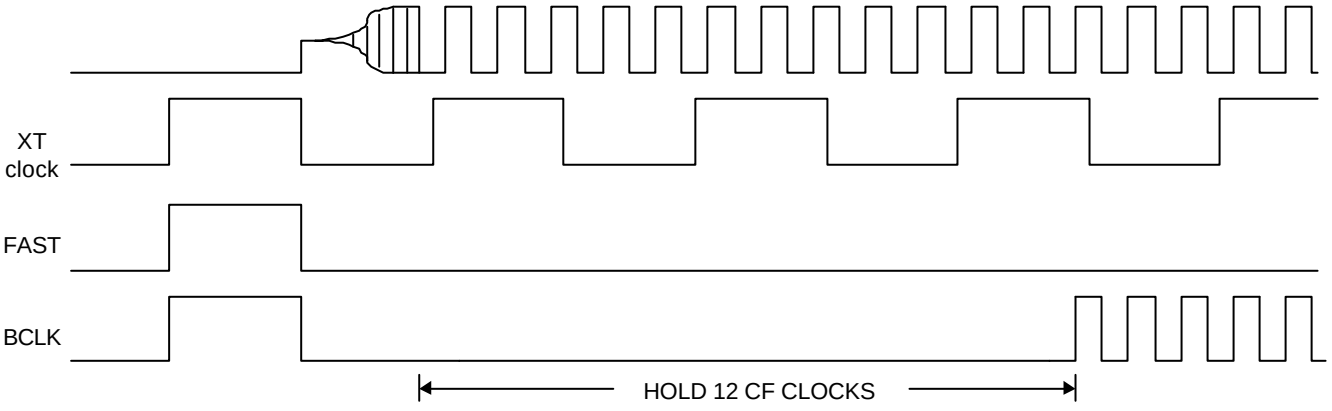
The operation of the dual clock option is shown in the following figure.
When this option is selected by mask option, the clock source (BCLK) of system clock generator will switch between the XT clock and the CF clock according to the user' s program. When the halt and stop instructions are executed, the clock source (BCLK) will switch to XT clock automatically.

In this option the XT clock provides the clock to the pre-divider, timer, I/O port chattering prevention and LCD circuitry in this option.



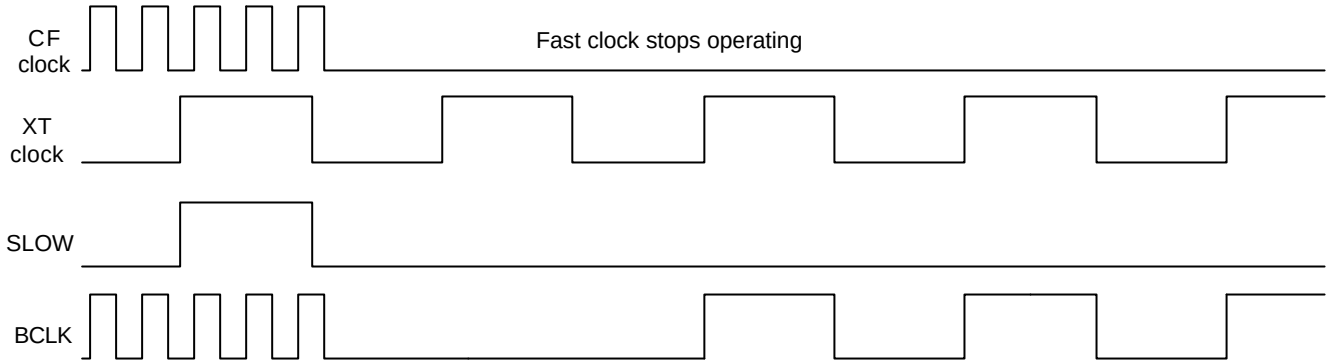
State Diagram of Dual Clock Option shown above.

After executing FAST instructions, the CF clock oscillator will start up and switch CF clock to BCLK, after which the system clock generator will hold 12 CF clocks. This will prevent the incorrect clock from reaching the system clock in the start-up duration of the fast clock oscillator.



This figure shows the System Clock Switching from Slow to Fast

After executing SLOW instruction, the system clock generator will hold for 2 XT clock cycles, then switch XT clock to BCLK.



This figure shows the System Clock Switching from Fast to Slow

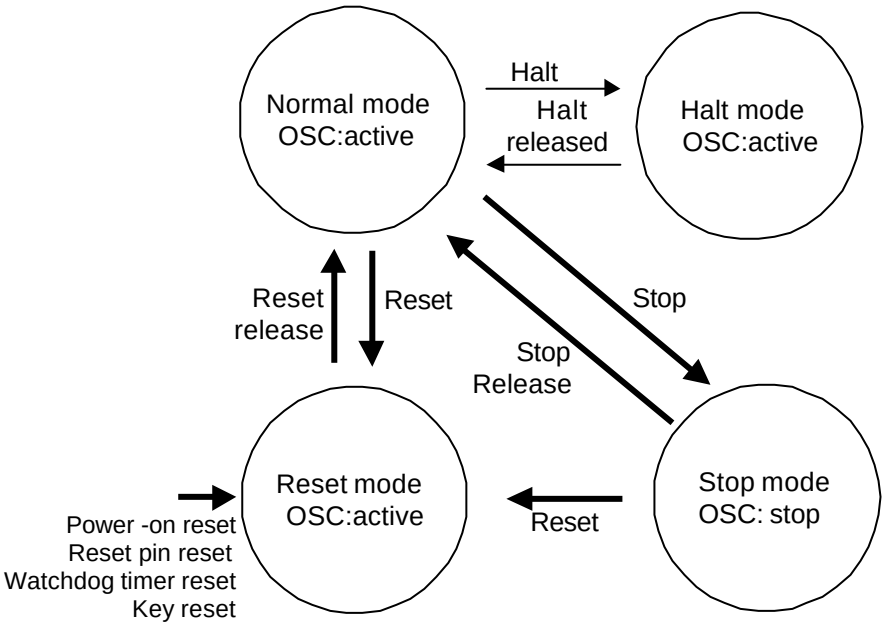
2-2-3-2 Single Clock

MASK OPTION table :

For Fast clock oscillator only

Mask Option name	Selected item
CLOCK SOURCE	(1) FAST ONLY

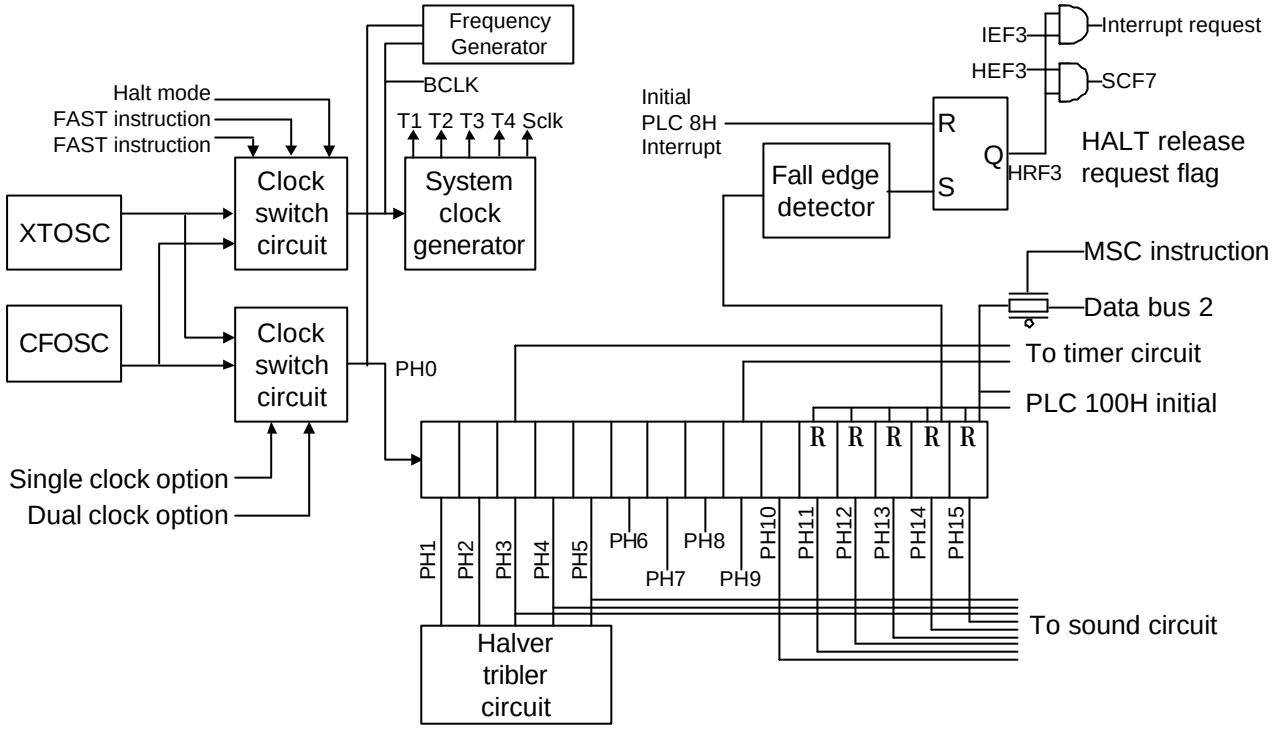
The operation of the single clock option is shown in the following figure.
Either the XT or the CF clock may be selected by mask option in this mode. The FAST instructions will perform as the NOP instruction in this option.
The backup flag (BCF) will be set to 1 automatically before the program enters the stop mode.



This figure shows the State Diagram of Single Clock Option

2-2-4 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 contents of this counter changes. The PH11 to PH15 of the pre-divider are reset to "0" when the PLC 100H instruction is executed or during the initial reset mode. The pre-divider delivers the signal to the halver / tripler circuit, alternating frequency for LCD display, system clock, sound generator and halt release request signal (I/O port chattering prevention clock).



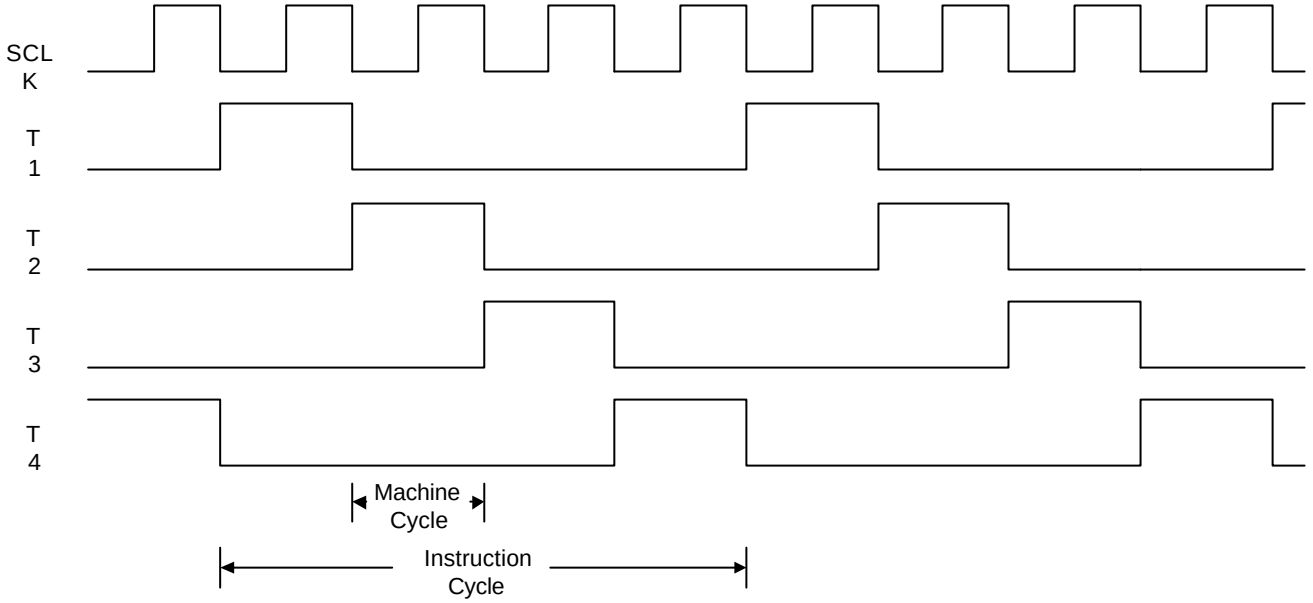
This figure shows the Pre-divider and its Peripherals

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 the pre-divider is PH0.

2-2-5 System Clock Generator

For the system clock, the clock switch circuit permits different clock inputs from XTOSC and CFOSC to be selected. The FAST and SLOW instructions can switch the clock input of the system clock generator (SGC).

The basic system clock is shown below:



2-3 PROGRAM COUNTER (PC)

This is a 12-bit counter, which addresses the program memory (ROM) for up to 4096 addresses. The MSB of program counter (PC11) is a page register. Only CALL and JMP instructions can address the whole address range (000h ~ FFFh), the rest jump relative instructions can address either page 0 (000h ~ 7ffh) or page 1 (800h ~ FFFh).

- The program counter (PC) is normally increased by one (+1) with every instruction execution.
  PC ← PC + 1
- When executing JMP instructions, subroutine call instructions (CALL), interrupt service routines or if reset occurs, the program counter (PC) loads the specified address corresponding to table 2-1.
  PC ← specified address shows in Table 2- 1
- When executing any jump instruction except JMP and CALL, the program counter (PC) loads the specified address in the operand of instruction. All of these jump relative instructions can only address the current page. That means when the current page is page 0 (PC11=0), only the range 000h ~ 7FFh is reachable; when the current page is page 1 (PC11=1), only the range 800h ~ FFFh is reachable.
  PC ← current page (PC11) + specified address in operand
- Return instruction (RTS)
  PC ← content of stack specified by the stack pointer
  Stack pointer ← stack pointer - 1

Table 2- 1

	PC11	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	0	0	0	0	0	0	0	0	0	0	0	0
Interrupt 2 (INT pin)	0	0	0	0	0	0	0	1	0	0	0	0
Interrupt 0 (input port C or D)	0	0	0	0	0	0	0	1	0	1	0	0
Interrupt 1 (timer 1 interrupt)	0	0	0	0	0	0	0	1	1	0	0	0
Interrupt 3 (pre-divider interrupt)	0	0	0	0	0	0	0	1	1	1	0	0
Interrupt 4 (timer 2 interrupt)	0	0	0	0	0	0	1	0	0	0	0	0
Interrupt 5 (Key Scanning interrupt)	0	0	0	0	0	0	1	0	0	1	0	0
Interrupt 6 (VFC counter interrupt)	0	0	0	0	0	0	1	0	1	0	0	0
Jump instruction	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
Subroutine call	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

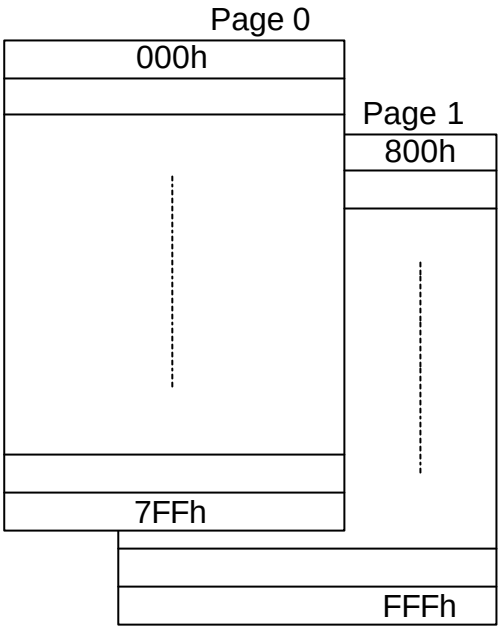
P10 to P0: Low-order 11 bits of instruction operand.

P11: page register

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

2-4 PROGRAM/TABLE MEMORY

The built-in mask ROM is organized with 4096 x 16 bits. There are 2 pages of memory space in this mask ROM. Page 0 covers the address range from 000h to 7FFh and page 1 covers 800h to FFFh.



Both instruction ROM (PROM) and table ROM (TROM) share this memory space. The partition formula for PROM and TROM is shown below:
Instruction ROM memory space = 2048 + (128 * N) words,
Table ROM memory space = 256(16 - N) bytes (N = 0 ~ 16).

Note: The data width of table ROM is 8-bit

The partition of memory space is defined by mask option, as shown in the table below:

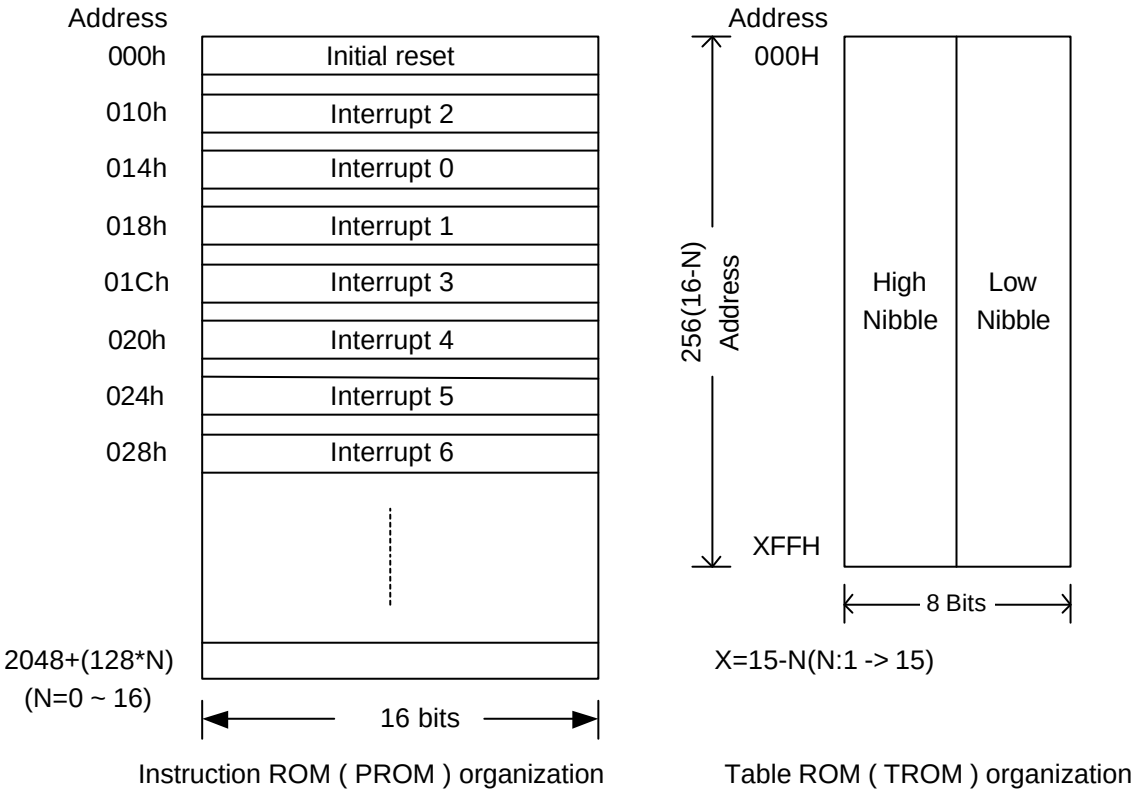
MASK OPTION table :

Mask Option name	Selected item	Instruction ROM memory space (Words)	Table ROM memory space (Bytes)
INSTRUCTION ROM <-> TABLE ROM	1 (N=0)	2048	4096
INSTRUCTION ROM <-> TABLE ROM	2 (N=1)	2176	3840
INSTRUCTION ROM <-> TABLE ROM	3 (N=2)	2304	3584
INSTRUCTION ROM <-> TABLE ROM	4 (N=3)	2432	3328
INSTRUCTION ROM <-> TABLE ROM	5 (N=4)	2560	3072
INSTRUCTION ROM <-> TABLE ROM	6 (N=5)	2688	2816
INSTRUCTION ROM <-> TABLE ROM	7 (N=6)	2816	2560
INSTRUCTION ROM <-> TABLE ROM	8 (N=7)	2944	2304
INSTRUCTION ROM <-> TABLE ROM	9 (N=8)	3072	2048
INSTRUCTION ROM <-> TABLE ROM	A (N=9)	3200	1792
INSTRUCTION ROM <-> TABLE ROM	B (N=10)	3328	1536
INSTRUCTION ROM <-> TABLE ROM	C (N=11)	3456	1280
INSTRUCTION ROM <-> TABLE ROM	D (N=12)	3584	1024
INSTRUCTION ROM <-> TABLE ROM	E (N=13)	3712	768
INSTRUCTION ROM <-> TABLE ROM	F (N=14)	3840	512

Mask Option name	Selected item	Instruction ROM memory space (Words)	Table ROM memory space (Bytes)
INSTRUCTION ROM <-> TABLE ROM	G (N=15)	3968	256
INSTRUCTION ROM <-> TABLE ROM	H (N=16)	4096	0

2-4-1. INSTRUCTION ROM (PROM)

There are some special locations that serve as interrupt service routines, such as reset address (000H), interrupt 0 address (014H), interrupt 1 address (018H), interrupt 2 address (010H), interrupt 3 address (01CH), interrupt 4 address (020H), interrupt 5 address (024H), and interrupt 6 address (028H), in the program memory.
When the useful address range of PROM exceeds 2048 addresses (800h), the memory space of PROM will automatically be defined as 2 pages. Refer to section 2-3.



This figure shows the Organization of ROM

2-4-2. TABLE ROM (TROM)

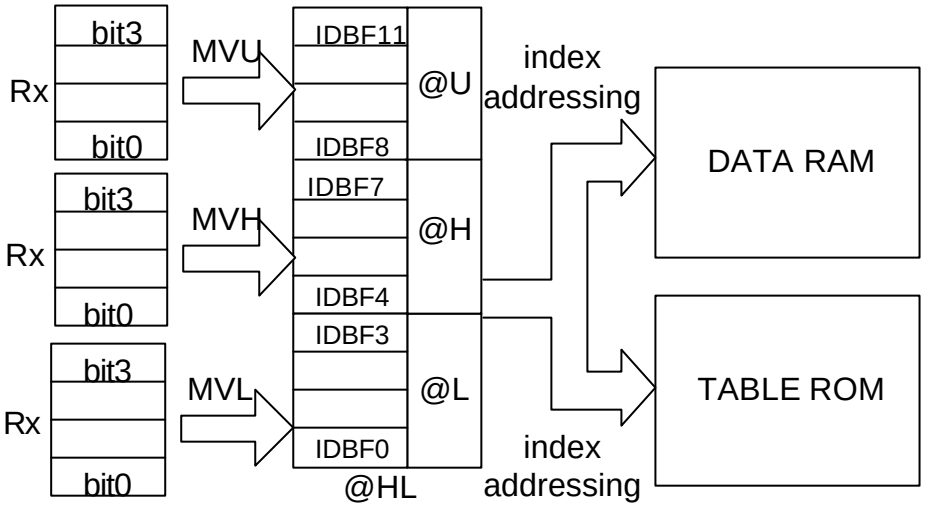
The table ROM is organized with 256(16-N) x 8 bits that share memory space with the instruction ROM (as shown in the figure above). This memory space stores the constant data or look up table for the usage of main program. All of the table ROM addresses are specified by the index address register (@HL). The data width can be 8 bits (256(16-N) x 8 bits) or 4 bits(512(16-N) x 4 bits) depending on usage. Refer to the explanation in the instruction chapter.

2-5 INDEX ADDRESS REGISTER (@HL)

This is a versatile address pointer for the data memory (RAM) and table ROM (TROM). The index address register (@HL) is a 12-bit register, and the contents of the register can be modified by executing MVH, MVL and MVU instructions. MVL instructions, when executed, will load the content of specified data memory to the lower nibble of the index register(@L). In the same manner, executing MVH and MVU instructions will load the content of the data RAM(Rx) to the higher nibble of the register @H and @U, respectively.

@U register				@H register				@L register			
Bit3	Bit2	Bit1	Bit0	Bit3	Bit2	Bit1	Bit0	Bit3	Bit2	Bit1	Bit0
IDBF11	IDBF10	IDBF9	IDBF8	IDBF7	IDBF6	IDBF5	IDBF4	IDBF3	IDBF2	IDBF1	IDBF0

The index address register can specify the full range addresses of the table ROM and data memory.



This figure shows the diagram of the index address register

The index address register is a write-only register, CPHL X instruction can specify 8-bit immediate data to compare with the content of @H and @L. When the result of comparison is equivalent, the instruction behind CPHL X will be skipped (NOP); if not equivalent, the instruction behind CPHL X will be executed normally.

Note: During the comparison of the index address, all the interrupt enable flags(IEF) have to be cleared to avoid malfunction.

The comparison bit pattern is shown below:

CPHL X	X7	X6	X5	X4	X3	X2	X1	X0
@HL	IDBF7	IDBF6	IDBF5	IDBF4	IDBF3	IDBF2	IDBF1	IDBF0

Example:

```
……                ; @HL = 30h
CPHL   30h
SIE*   0h         ; disable IEF
JMP    lable1     ; this instruction will be force as NOP
JMP    lable2     ; this instruction will be executed and than jump to lable2
……
lable1:
……
lable2:
```

2-6 STACK REGISTER (STACK)

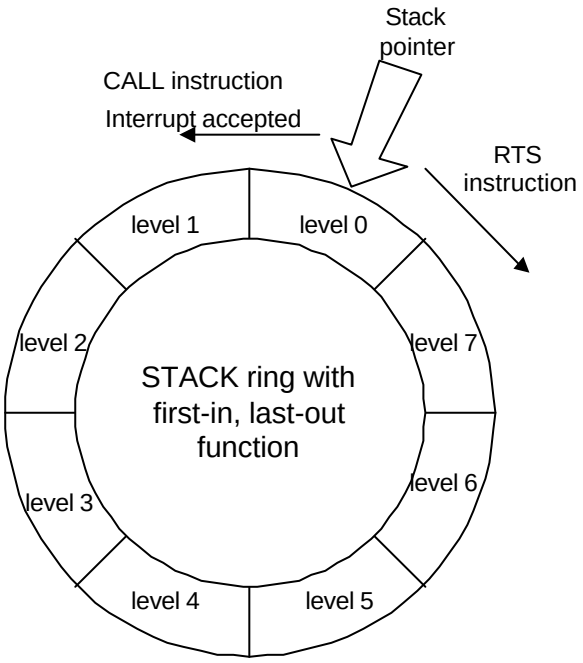
Stack is a specially designed register following the first-in-last-out rule. It is used to save the contents of the program counter sequentially during subroutine calls or execution of interrupt service routines.

The contents of the stack register are returned sequentially to the program counter (PC) when return instructions (RTS) are executed.

The stack register is organized by using 11 bits by 8 levels, but with no overflow flag; hence only 8 levels of subroutine calls or interrupts are allowed (If the stacks are full, and either an interrupt occurs or a subroutine call executes, the first level will be overwritten).

Once the subroutine call or interrupt causes a stack register (STACK) overflow, the stack pointer will return to 0 and the contents 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) when the RTS instruction is executed.
Once the RTS instruction causes a stack register (STACK) underflow, the stack pointer will return to level 7 and the content of the level 7 stack will be restored to the program counter.

The following figure shows the diagram of the stack.



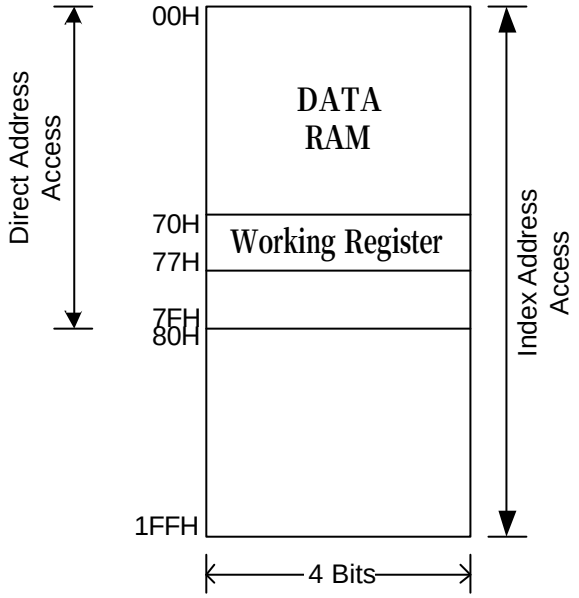
2-7 DATA MEMORY (RAM)

Static RAM is organized with 512 addresses x 4 bits and is used to store data.

The data memory may be accessed through two methods:

1. Direct addressing mode
   The address of the data memory is specified by the instructions and the addressing range is from 00H to 7FH.
2. Index addressing mode
   The index address register (@HL) specifies the address of the data memory and all address space from 00H to 1FFH can be accessed.

The 8 specified addresses (70H to 77H) in the direct addressing memory are also used as 8 working registers. The function of working registers will be described in detail in section 2-6.



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

2-8 WORKING REGISTER (WR)

The locations 70H to 77H of the data memory (RAM) are not only used as general-purpose data memory but also as working registers (WR). The following will introduce the general usage of working registers:

1. They can 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. They can be used to transfer data between the working register and any address in the direct addressing data memory (RAM). Such as: MWR Rx, Ry; MRW Ry, Rx

3. They can be used to decode (or directly transfer) the contents of the working register and output to the LCD PLA circuit. Such as: LCT, LCB, LCP.

2-9 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 is made possible.

2-10 ALU (Arithmetic and Logic Unit)

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

Binary addition/subtraction (INC, DEC, ADC, SBC, ADD, SUB, ADN, ADCI, SBUI, 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)

2-11 HEXADECIMAL CONVERT TO DECIMAL (HCD)

Decimal format is another number format for TM8740. When the contents of the data memory has been assigned as decimal format, it is necessary to convert the results to decimal format after the execution of ALU instructions. When the decimal converting operation is being processed, all of the operand data (including the contents of the data memory (RAM), accumulator (AC), immediate data, and look-up table) should be in the decimal format, or the results of conversion will be incorrect.

Instructions DAA, DAA*, DAA @HL can convert 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
$0 \le AC \le 9$	CF = 0	no change	no change
$A \le AC \le F$	CF = 0	AC= AC+ 6	CF = 1
$0 \le AC \le 3$	CF = 1	AC= AC+ 6	no change

Example 1:

```
LDS   10h, 9      ; Loads immediate data "9" to data memory address 10H.
LDS   11h, 1      ; Loads immediate data "1" to data memory address 11H
                  ; and AC.
RF    1h          ; Reset CF to 0.
ADD*  10h         ; Contents of the data memory address 10H and AC are
                  ; binary-added; the result loads to AC & data memory address
                  ; 10H. (R10 = AC = AH, CF = 0)
DAA*  10h         ; Converts the content of AC to
```

```
                ; decimal format.
                ; The result in the data memory address 10H is "0" and in
                ; the CF is "1". This represents the decimal number "10".
```

Instructions DAS, DAS*, DAS @HL 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.

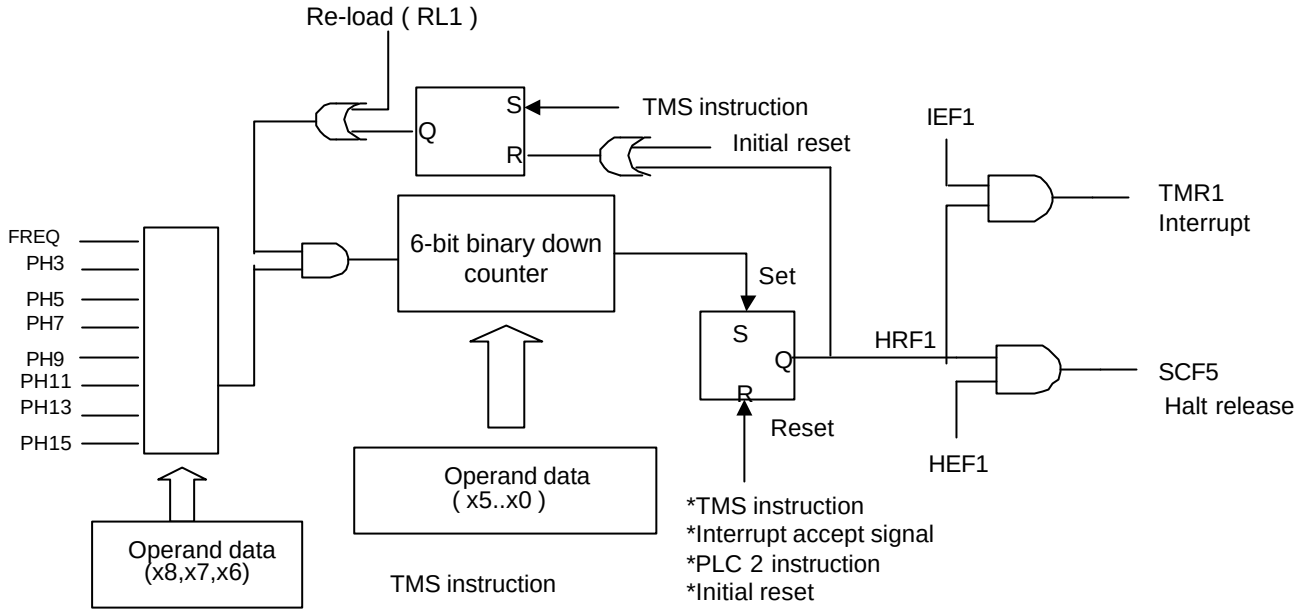
AC data before DAS execution	CF data before DAS execution	AC data after DAS execution	CF data after DAS execution
$0 \le AC \le 9$	CF = 1	No change	no change
$6 \le AC \le F$	CF = 0	AC= AC+A	no change

Example 2:

```
        LDS   10h, 1      ; Loads immediate data "1" to the data memory address 10H.
        LDS   11h, 2      ; Loads immediate data "2" to the data memory address 11H and
AC.
        SF    1h          ; Sets CF to 1, which means no borrowing has occurred.
        SUB*  10h         ; Content of data memory address 10H is binary-subtracted;
                          ; the result loads to data memory address
                          ; 10H. (R10 = AC = FH, CF = 0)
        DAS*  10h         ; Converts the content of the data memory address 10H to decimal
                          ; format. The result in the data memory address 10H is "9" and in
                          ; the CF is "0". This represents the decimal number "–1".
```

2-12 TIMER 1 (TMR1)

This figure shows the TMR1 organization.



2-12-1 NORMAL OPERATION

TMR1 consists of a programmable 6-bit binary down counter, which is loaded and enabled by executing the TMS or TMSX instructions.
Once the TMR1 counts down to 3Fh, it generates an underflow signal to set the halt release request flag 1 (HRF1) to 1 and then stops counting down.
When HRF1 = 1, and the TMR1 interrupt enable flag (IEF1) = 1, an interrupt is generated.
When HRF1 = 1, if the IEF1 = 0 and the TMR1 halt release enable (HEF1) = 1, the program will escape from halt mode (if CPU is in halt mode) and then set the start condition flag 5 (SCF5) to 1 in the status register 3 (STS3).
After power on reset, the default clock source of TMR1 is PH3.
If watchdog reset occurs, the clock source of TMR1 will remain the same.

The following table shows the definition of each bit in TMR1 instructions.

OPCODE	Select clock			Initiate value of timer					
TMSX X	X8	X7	X6	X5	X4	X3	X2	X1	X0
TMS Rx	0	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
TMS @HL	0	bit7	bit6	bit5	Bit4	bit3	bit2	bit1	bit0

The following table shows the clock source setting for TMR1.

X8	X7	X6	clock source
0	0	0	PH9
0	0	1	PH3
0	1	0	PH15
0	1	1	FREQ
1	0	0	PH5
1	0	1	PH7
1	1	0	PH11
1	1	1	PH13

Notes:

1. When the TMR1 clock is PH3
   TMR2 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the TMR1 clock is PH9
   TMR2 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the TMR1 clock is PH15
   TMR2 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
4. When the TMR1 clock is PH5
   TMR2 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)
5. When the TMR1 clock is PH7
   TMR2 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)
6. When the TMR1 clock is PH11
   TMR2 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)
7. When the TMR1 clock is PH13
   TMR2 set time = (Set value + error) * 8192 * 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

   PH5: The 5th stage output of the predivider

   PH7: The 7th stage output of the predivider

   PH9: The 9th stage output of the predivider

   PH11: The 11th stage output of the predivider

   PH13: The 13th stage output of the predivider

   PH15: The 15th stage output of the predivider

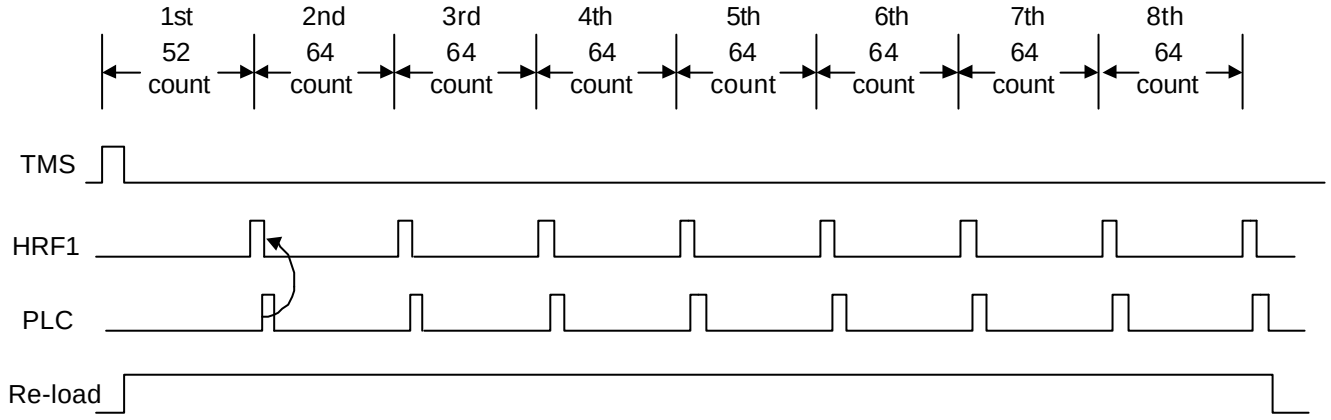
8. When the TMR1 clock is FREQ
   TMR1 set time = (Set value + error) * 1/FREQ (KHz) (ms).
   FREQ: refer to section 3-3-4.

2-12-2 RE-LOAD OPERATION

TMR1 provides the re-load function, which can extend any time interval greater than 3Fh. The SF 80h instruction enables the re-load function and RF 80h instruction disables it.
When the re-load function is enabled, the TMR1 will not stop counting until the re-load function is disabled and TMR1 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 either the TMS or the TMSX instructions to set the down count value before the re-load function is enabled, because TMR1 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 TMS related instructions are not executed after each halt release or interrupt occurs, the TMR1 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 TMR1 to 52 and start counting, then enable the TMR1 halt release or interrupt function. Before the first time underflow occurs, enable the re-load function. The TMR1 will continue operating even though TMR1 underflow occurs. When halt release or interrupt occurs, clear the HRF1 flag through a PLC instruction. After a halt release or interrupt occurs 8 times, disable the re-load function; counting is completed.

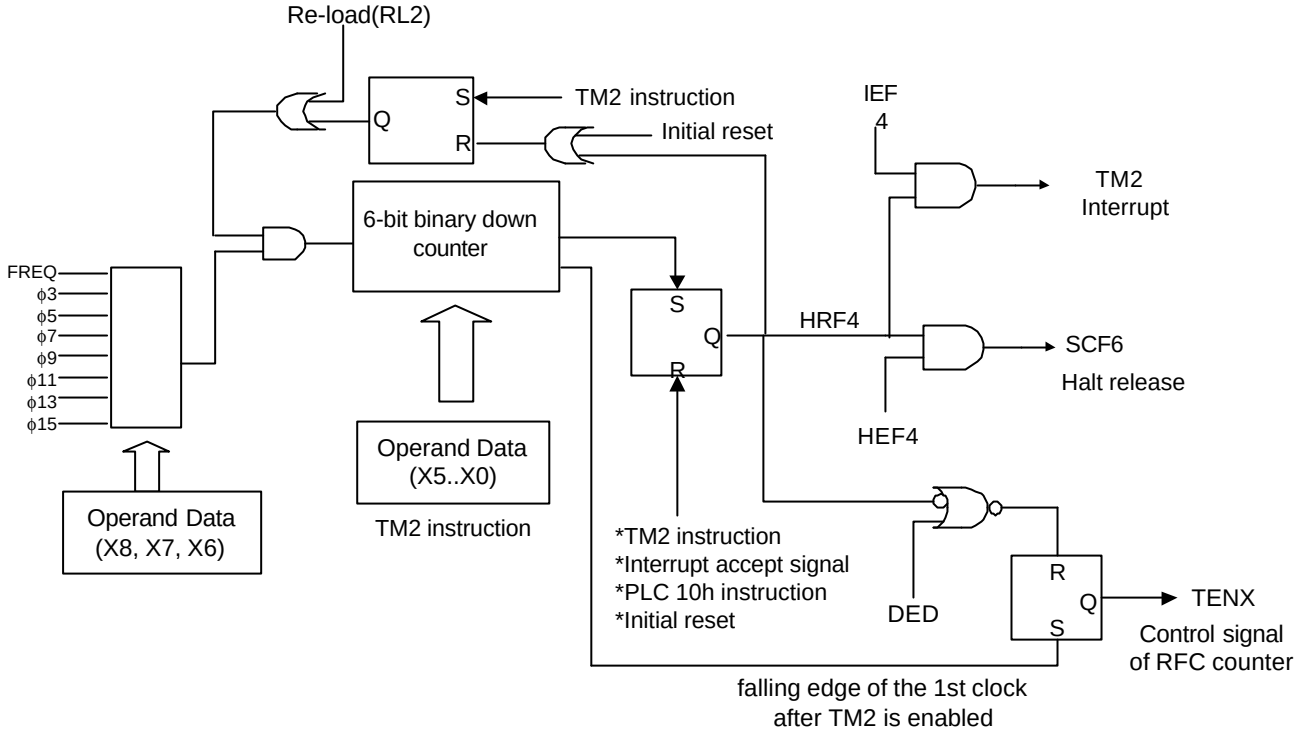


In this example, S/W enters the halt mode to wait for the underflow of TMR1.

```
            LDS   0, 0          ;initiates the underflow counting register
            PLC   2
            SHE   2             ;enables the HALT release caused by TMR1
            TMSX  34h           ;initiates the TMR1 value (52) and clock source is ϕ9
            SF    80h           ;enable the re-load function
RE_LOAD:
            HALT
            INC*  0             ;increases the underflow counter
            PLC   2             ;clears HRF1
            JB3   END_TM1       ;if the TMR1 underflow counter is equal to 8, exit subroutine
            JMP   RE_LOAD
END_TM1:
            RF    80h           ;disables the re-load function
```

2-13 TIMER 2 (TMR2)

The following figure shows the TMR2 organization.



2-13-1 NORMAL OPERATION

TMR2 consists of a programmable 6-bit binary down counter, which is loaded and enabled by executing either the TM2 or the TM2X instructions.
Once TMR2 counts down to 3Fh, it stops counting, then generates an underflow signal and sets the halt release request flag 4 (HRF4) to 1.

. When HRF4 = 1, and the TMR2 interrupt enabler (IEF4) is set to 1, the interrupt occurred.
. When HRF4 =1, IEF4 = 0, and the TMR2 halt release enabler (HEF4) is set to 1, the program will escape from halt mode (if CPU is in halt mode) and HRF4 sets the start condition flag 6 (SCF6) to 1 in the status register 4 (STS4).

After power on reset, the default clock source of TMR2 is PH7.
If watchdog reset occurs, the clock source of TMR2 will remain the same.

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	0	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
TM2 @HL	0	bit7	bit6	bit5	Bit4	bit3	bit2	bit1	bit0

The following table shows the clock source setting for TMR2.

X8	X7	X6	clock source
0	0	0	PH9
0	0	1	PH3
0	1	0	PH15
0	1	1	FREQ
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 PH9
   TMR2 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the TMR2 clock is PH15
   TMR2 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
4. When the TMR2 clock is PH5
   TMR2 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)
5. When the timer clock is PH7
   TMR2 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)
6. When the TMR2 clock is PH11
   TMR2 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)
7. When the TMR2 clock is PH13
   TMR2 set time = (Set value + error) * 8192 * 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

   PH5: The 5th stage output of the predivider

   PH7: The 7th stage output of the predivider

   PH9: The 9th stage output of the predivider

   PH11: The 11th stage output of the predivider

   PH13: The 13th stage output of the predivider

   PH15: The 15th stage output of the predivider

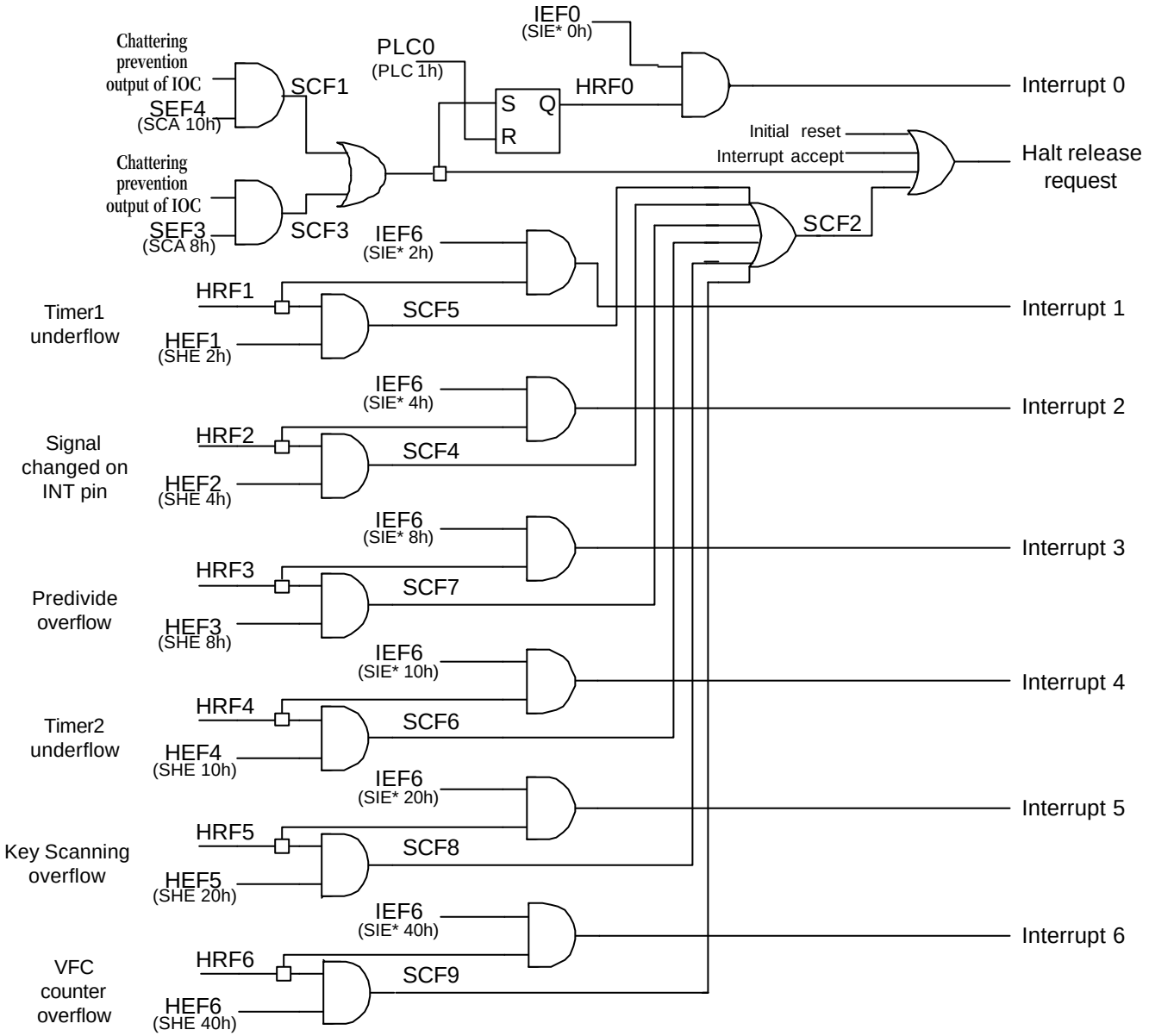
8. When the TMR2 clock is FREQ
   TMR2 set time = (Set value + error) * 1/FREQ (KHz) (ms).
   FREQ: refer to section 3-3-4.

2-13-2 RE-LOAD OPERATION

TMR2 also provides the re-load function, the same as TMR1. The instruction SF2 1 enables the re-load function, the instruction RF2 1 disables it.

2-14 STATUS REGISTER (STS)

The status register (STS) is organized with 4 bits and comes in 4 types: status register 1 (STS1) to status register 4 (STS4). The following figure shows the configuration of the start condition flags for TM8740.



2-14-1 STATUS REGISTER 1 (STS1)

Status register 1 (STS1) consists of 2 flags:

1. Carry flag (CF)

   The carry flag is used to save the results of the carry or borrow during the arithmetic operation.

2. Zero flag (Z)

   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 (AC)	Zero flag (Z)	NA	NA
Read / write	Read only	Read only	Read only

2-14-2 STATUS REGISTER 2 (STS2)

Status register 2 (STS2) consists of start condition flag 1, 2 (SCF1, SCF2) and the backup flag. The MSB instruction can be used to transfer data in status register 2 (STS2) to the accumulator (AC) and the data memory (RAM), but it is impossible 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
Start condition flag 3 (SCF3)	Start condition flag 2 (SCF2)	Start condition flag 1 (SCF1)	Backup flag (BCF)
Halt release caused by the IOD port	Halt release caused by SCF4, 5,6,7,9	Halt release caused by the IOC port	The back up mode status
Read only	Read only	Read only	Read only

Start condition flag 3 (SCF3)
When the SCA instruction specified signal change occurs at port IOD to release the halt mode, SCF3 will be set. Executing the SCA instruction will cause SCF3 to be reset to 0.

Start condition flag 1 (SCF1)
When the SCA instruction specified signal change occurs at port IOC to release the halt mode, SCF1 will be set. Executing the SCA instruction will cause SCF1 to be reset to 0.

Start condition flag 2 (SCF2)
When a factor other than port IOA and IOC causes the halt mode to be released, SCF2 will be set to1. In this case, if one or more start condition flags in SCF4, 5, 6, 7, 9 is set to 1, SCF2 will also be set to 1 simultaneously. When all of the flags in SCF4, 5, 6, 7, 9 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 can be set / reset by executing the SF 2h / RF 2h instruction.

2-14-3 STATUS REGISTER 3 (STS3)

When the halt mode is released by the 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 4 flags:

1. Start condition flag 4 (SCF4)

   Start condition flag 4 (SCF4) is set to 1 when the signal change at the INT pin causes the halt release request flag 2 (HRF2) to be outputted and the halt release enable flag 2 (HEF2) is set beforehand. To reset start condition flag 4 (SCF4), the PLC instruction must be used to reset the halt release request flag 2 (HRF2) otherwise the SHE instruction must be used to reset the halt release enable flag 2 (HEF2).

2. Start condition flag 5 (SCF5)

   Start condition flag 5 (SCF5) is set when an underflow signal from Timer 1 (TMR1) causes the halt release request flag 1 (HRF1) to be outputted and the halt release enable flag 1 (HEF1) is set beforehand. To reset start condition flag 5 (SCF5), the PLC instruction must be used to reset the halt release request flag 1 (HRF1) otherwise the SHE instruction must be used to reset the halt release enable flag 1 (HEF1).

3. 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) otherwise the SHE instruction must be used to reset the halt release enable flag 3 (HEF3).

4. Contents of the pre-divider on the 15th stage.

The MSC instruction is used to transfer the contents of status register 3 (STS3) to the 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	Start condition flag 5 (SCF5)	Start condition flag 4 (SCF4)
Halt release caused by pre-divider overflow		Halt release caused by TMR1 underflow	Halt release caused by INT pin
Read only	Read only	Read only	Read only

2-14-4 STATUS REGISTER 3X (STS3X)

When the halt mode is released with start condition flag 2 (SCF2), status register 3X (STS3X) will store the status of the factor in the release of the halt mode.

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

1. Start condition flag 8 (SCF8)

   SCF8 is set to 1 when any one of KI1~4 =1/0 (KI1~4=1 in LED mode / KI1~4=0 in LCD mode) causes the halt release request flag 5 (HRF5) to be outputted and the halt release enable flag 5 (HEF5) is set beforehand. To reset the start condition flag 8 (SCF8), the PLC instruction must be used to reset the halt release request flag 5 (HRF5) otherwise the SHE instruction must be used to reset the halt release enable flag 5 (HEF5).

2. Start condition flag 6 (SCF6)

SCF6 is set to 1 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 the start condition flag 6 (SCF6), the PLC instruction must be used to reset the halt release request flag 4 (HRF4) otherwise the SHE instruction must be used to reset the halt release enable flag 4 (HEF4).

3. Start condition flag 9 (SCF9)

SCF9 is set when a finish signal from mode 3 of VFC function causes the halt release request flag 6 (HRF6) to be outputted and the halt release enable flag 9 (HEF9) is set beforehand. In this case, the 16-counter of VFC function ; please refer to 2-16-9. To reset the start condition flag 9 (SCF9), the PLC instruction must be used to reset the halt release request flag 6 (HRF6) otherwise the SHE instruction must be used to reset the halt release enable flag 6 (HEF6).

The MCX instruction can be used to transfer the contents of status register 3X (STS3X) to the 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
Start condition flag 9 (SCF9)	NA	Start condition flag 6 (SCF6)	Start condition flag 8 (SCF8)
Halt release caused by VFC counter finish		Halt release caused by TMR2 underflow	Halt release caused by SKI underflow
Read only	Read only	Read only	Read only

2-14-5 STATUS REGISTER 4 (STS4)

Status register 4 (STS4) consists of 3 flags:

1. System clock selection flag (CSF)

The system clock selection flag (CSF) indicates which clock source of the system clock generator (SCG) is being used. Executing the SLOW instruction will change the clock source (BCLK) of the system clock generator (SCG) to the slow speed oscillator (XT clock), and resets the system clock selection flag (CSF) to 0. Executing the FAST instruction will change the clock source (BCLK) of the system clock generator (SCG) to the fast speed oscillator (CF clock), and sets the system clock selection flag (CSF) to 1. For the operation of the system clock generator, refer to 3-3.

2. Watchdog timer enable flag (WTEF)

The watchdog timer enable flag (WDF) indicates the operating status of the watchdog timer.

3. Overflow flag of 16-bit counter of VFC (VFOVF)

The overflow flag of 16-bit counter of VFC (VFOVF) is set to 1 when the overflow of the 16-bit counter of VFC occurs. The flag will reset to 0 when this counter is initiated by executing the SRF instruction.

The MSD instruction can be used to transfer the contents of status register 4 (STS4) to the accumulator (AC) and the data memory (RAM).

The following table shows the Bit Pattern of Status Register 4 (STS4)

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	The overflow flag of 16-bit counter of VFC (VFVOF)	Watchdog timer Enable flag (WDF)	System clock selection flag (CSF)
Read only	Read only	Read only	Read only

2-14-6 START CONDITION FLAG 11 (SCF11)

Start condition flag 11 (SCF11) will be set to 1 in STOP mode when the following conditions are met:

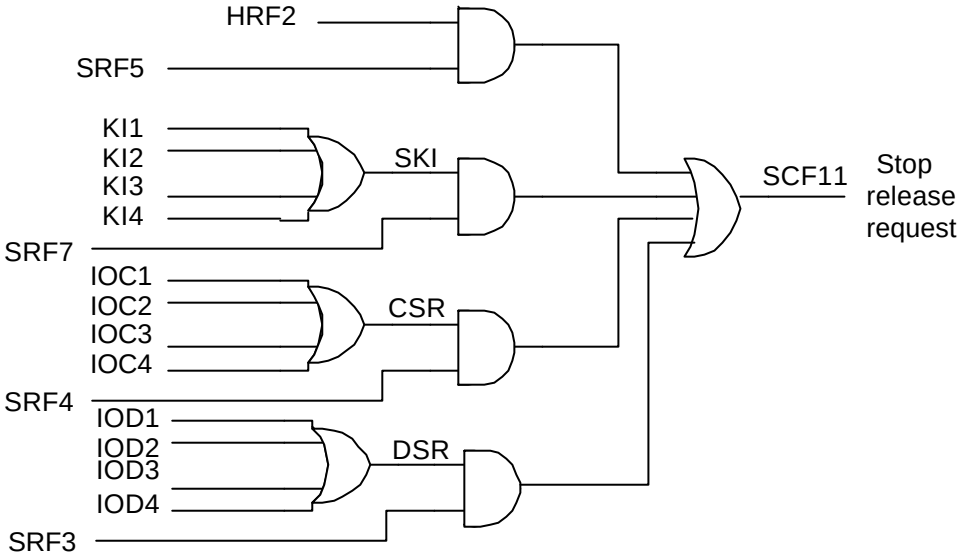
. A high level signal comes from the OR-ed output of the pins defined as input mode in IOC port, which causes the stop release flag of IOC port (CSR) to output. The stop release enable flag 4 (SRF4) is must be set beforehand.

. A high level signal comes from the OR-ed output of the pins defined as input mode in IOD port, which causes the stop release flag of IOD port (DSR) to output. The stop release enable flag 3 (SRF3) must be set beforehand.

. A high level signal comes from the OR-ed output of the signals latch for KI1~4, which causes the stop release flag of Key Scanning (SKI) to output. The stop release enable flag 4 (SRF7) must be set beforehand.

. The signal change from the INT pin causes the halt release flag 2 (HRF2) to output. The stop release enable flag 5 (SRF5) must be set beforehand.

The following figure shows the organization of start condition flag 11 (SCF 11).



The stop release flags (SKI, CSR, DSR, HRF2) were specified by the stop release enable flags (SRFx). These flags should be clear before the chip enters stop mode. All of the pins in the IOA and IOC ports have to be set in input mode and keep in 0 state before the chip enters the STOP mode, otherwise the program can not enter STOP mode.
Instruction SRE is used to set or reset the stop release enable flags (SRF4,5,7).

The following table shows the stop release request flags.

	The OR-ed latched signals for KI1~4	The OR-ed input mode pins of IOC(IOD) port	The rising or falling edge on INT pin
Stop release request flag	SKI	CSR(DSR)	HRF2
Stop release enable flag	SRF7	SRF4(SRF3)	SRF5

2-15 CONTROL REGISTER (CTL)

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

2-15-1 CONTROL REGISTER 1 (CTL1)

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

1. Switch enable flag 4 (SEF4)

   Stores the status of the input signal change at pins of IOC set in input mode that causes the halt mode or stop mode to be released.

2. Switch enable flag 3 (SEF3)

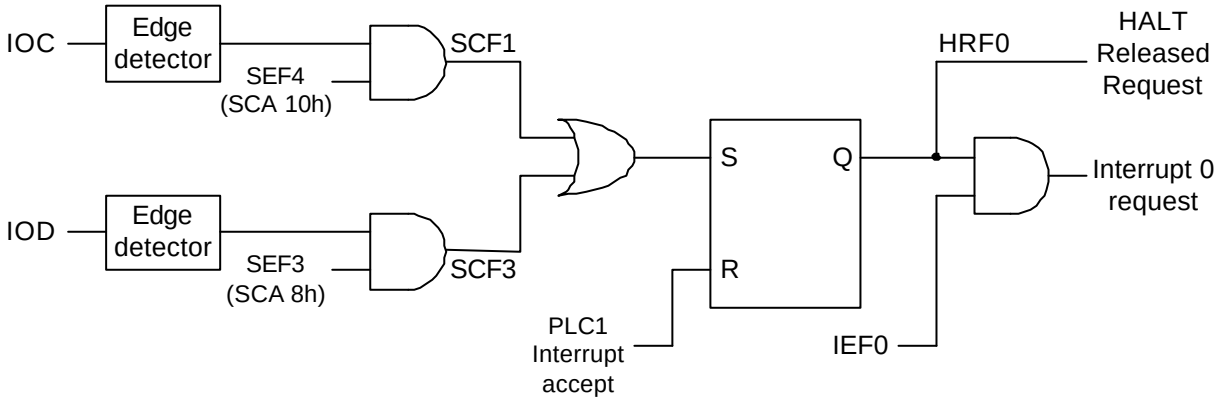
   Stores the status of the input signal change at pins of IOD set in input mode that causes the halt mode or stop mode to be released.

Executed the SCA instruction may set or reset these flags.

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

Bit 4	Bit3
Switch enable flag 4 (SEF4)	Switch enable flag 3 (SEF3)
Enables the halt release caused by the signal change on IOC port	Enables the halt release caused by the signal change on IOD port
Write only	Write only

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



2-15-1-1 The Settings for Halt Mode

If the SEF4 (SEF3) is set to 1, the signal changed on the IOC(IOD) port will cause the halt mode to be released and SCF1(SCF3) will be set to 1. Because the input signal of IOC(IOD) port are ORed, it is necessary to keep the unchanged input signals at " 0 " state; only one of the input signal can change state.

2-15-1-2 The Settings for Stop Mode

If SRF4(SRF3) and SEF4(SEF3) are set, the stop mode will be released to set the SCF1(SCF3) when a high level signal is applied to one of the input mode pins of IOC(IOD) port and the other pins stay in "0" state.
After the stop mode is released, TM8740 enters the halt condition.
The high level signal must hold for a while to allow the chattering prevention circuitry of IOC(IOD) port to detect this signal and then set SCF1(SCF3) to release the halt mode, otherwise the chip will return to stop mode again.

2-15-1-3 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 IOC(IOD) port will deliver the SCF1(SCF3) when SEF4(SEF3) has been set to 1 by executing the SCA instruction. Once the SCF1(SCF3) 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 the SIE instruction, the interrupt request flag 0 (interrupt 0) will be delivered to interrupt the program.
If the interrupt 0 is accepted by SEF4(SEF3) and IEF0, the interrupt 0 request to the next signal change at IOC(IOD) will be inhibited. To release this mode, the SCA instruction must be executed again. Refer to 2-16-1-1

2-15-2 CONTROL REGISTER 2 (CTL2)

Control register 2 (CTL2) consists of halt release enable flags 1, 2, 3, 4, 5, 6 (HEF1, 2, 3, 4, 5, 6) and is set by SHE instruction. The bit pattern of the control register (CTL2) is shown below.

Halt release enable flag	HEF6	HEF5	HEF4
Halt release condition	Enable the halt release caused by VFC counter to be finished (HRF6)	Enable the halt release caused by Key Scanning(HRF5)	Enable the halt release caused by TMR2 underflow (HRF4)
Halt release enable flag	HEF3	HEF2	HEF1
Halt release condition	Enable the halt release caused by pre-divider overflow (HRF3)	Enable the halt release caused by INT pin (HRF2)	Enable the halt release caused by TM1 underflow (HRF1)

When the halt release enable flag 6 (HEF6) is set, a finish signal from the 16-bit counter of VFC causes the halt mode to be released. In the same manner, when HEF1 to HEF4 are set to 1, the following conditions will cause the halt mode to be released, respectively : an underflow signal from TMR1, the signal change at the INT pin, an overflow signal from the pre-divider and an underflow signal from TMR2, a 'H' signal from OR-ed output of KI1~4 latch signals.
When the stop release enable flag 5 (SRF5) and the HEF2 are set, the signal change at the INT pin can cause the stop mode to be released.
When the stop release enable flag 7 (SRF7) and the HEF5 are set, the 'H' signal from OR-ed output of K1~4 latch signals can cause the stop mode to be released.

2-15-3 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 the SIE* instruction. The bit pattern of control register 3 (CTL3) is shown below.

Interrupt enable flag	IEF6	IEF5	IEF4
Interrupt request flag	Enable the interrupt request caused by VFC counter to be finished (HRF6)	Enable the interrupt request caused by Key Scanning (HRF5)	Enable the interrupt request caused by TMR2 underflow (HRF4)
Interrupt flag	Interrupt 6	Interrupt 4	Interrupt 4
Interrupt enable flag	IEF3	IEF2	IEF1
Interrupt request flag	Enable the interrupt request caused by predivider overflow (HRF3)	Enable the interrupt request caused by INT pin (HRF2)	Enable the interrupt request caused by TM1 underflow (HRF1)
Interrupt flag	Interrupt 3	Interrupt 2	Interrupt 1
Interrupt enable flag	IEF0		
Interrupt request flag	Enable the interrupt request caused by IOC or IOD 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.

2-15-4 CONTROL REGISTER 4 (CTL4)

Control register 4 (CTL4), being a 3-bit register, is set / reset by SRE instruction.

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

Stop release enable flag	SRF7	SRF5	SRF4 (SRF3)
Stop release request flag	Enable the stop release request caused by signal change on KI1~4 (SKI)	Enable the stop release request caused by signal change on INT pin (HRF2)	Enable the stop release request caused by signal change on IOC (IOD)

When the stop release enable flag 7 (SRF7) is set to 1, the input signal change at the KI1~4 pins causes the stop mode to be released. In the same manner, when SRF4 (SRF3) and SRF5 are set to 1, the input signal changes at the input mode pins of the IOC (IOD) port. The signal change on the INT pin causes the stop mode to be released as well.

Example:

This example illustrates the stop mode released by the port IOC, KI1~4 and INT pin. Assume all of the pins in IOD and IOC have been set to input mode.

```
PLC     25h     ; Resest the HRF0, HRF2 and HRF5.
SHE     24h     ; HEF2 and HEF5 is set so that the signal change at INT or KI1~4 pin
```

```
                        ; causes start condition flag 4 or 8 to be set.
SCA         10h         ; SEF4 is set so that the signal changes at port IOC
                        ; cause the start conditions SCF1 to be set.
SRE         0b0h        ; SRF7,5,4 are set so that the signal changes at KI1~4 pins, port
                        ; IOC and INT pin cause the stop mode to be released.
STOP                    ; Enters the stop mode.

        ……              ;STOP release

MSC         10h         ; Checks the signal change at INT pin that causes the stop mode to be
                        ; released.
MSB         11h         ; Check the signal change at port IOC that causes the stop mode to be
                        ; released.
MCX         12h         ; Checks the signal change at KI1~4 pins that causes the stop mode to
                        ; be released.
```

2-16 HALT FUNCTION

The halt function is provided to minimize the current dissipation of the TM8740 when the LCD is operating. During halt mode, the program memory (ROM) is not in operation; 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 (SCF1, SCF3, HRF1 ~ 6) is delivered, the CPU enters halt mode.

The following 3 conditions are available to release 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 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 IOC(SCF1) or IOD(SCF3).

(3) The halt release condition specified by the SHE instruction is met (HRF1 ~ HRF6).
When the halt mode is released in either (2) or (3), it is necessary that either the MSB, or the MSC, or the MCX instruction is executed in order to test the halt release signal. It is also necessary to execute the PLC instruction 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.

2-17 HEAVY LOAD FUNCTION

When heavy loading (lamp light-up, motor start, etc.) causes a temporary voltage drop in supply voltage, the heavy loading function (set BCF = 1) prevents TM8740 from malfunctioning, especially where a battery with high internal impedance, such as Li battery or alkali battery, is used.

During back up mode, the 32.768KHz Crystal oscillator will add an extra buffer in parallel and switch the internal power (BAK) from VDD1 to VDD2 (Li power option only). In this condition, all of the functions in TM8740 will work under the VDD voltage range, causing TM8740 to get better noise immunity.

To shorten the start-up time of 32.768KHz Crystal oscillator, TM8740 will set the BCF to 1 during the reset cycle and reset BCF to 0 after the reset cycle automatically when the Ag and Li power mode option is used. BCF will be reset to 0 by default setting during normal operation.

Table 3- 1 The back-up flag status.

	Li option	Remark
Reset cycle	BCF=1	large current
After reset cycle	BCF=1	large current
SF 2 executed	BCF=1	large current
RF 2 executed	BCF=0	

To shorten the start-up time of 32.768KHz Crystal oscillator, TM8740 will set the BCF to 1 during the reset cycle and reset BCF to 0 after the reset cycle automatically when the Li power mode option is used. BCF will be reset to 0 by default setting during normal operation. When the heavy load function is performed, the current dissipation will increase.

Table 3- 2 Li power option:

	Initial reset	After reset	Stop mode	SF 2	RF 2
BCF	1	1	1*	1	0
Internal logic	VDD	VDD	VDD	VDD	1/2 VDD
Peripheral logic	VDD	VDD	VDD	VDD	VDD

Note: When the program enters the stop mode, the BCF will set to 1 automatically to insure that the low speed oscillator will start up in a proper condition while stop release occurs.

2-18 STOP FUNCTION (STOP)

The stop function is another solution used to minimize the current dissipation for TM8740. In stop mode, all of the functions in TM8740 are held, including oscillators. All of the LCD corresponding signals (COM and Segment) will output "L" level. In this mode, TM8740 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 set to input modes on IOD and IOC ports must be in the "L" state, and no stop release signal (SRFn) should be delivered. The CPU will then enter stop mode.
The following conditions cause stop mode to be released.
. One of the signals on the input mode pin of IOD or IOC port is in "H" state and holds long enough to cause the CPU to be released from halt mode.
. A signal change in the INT pin.
. The stop release condition specified by the SRE instruction is met.

When the TM8740 is released from stop mode, the TM8740 enters the halt mode immediately and will process the halt release procedure. If the "H" signal on the IOC(IOD) port does not hold long enough to set the SCF1(SCF3), once the signal on the IOC port returns to "L", the TM8702 will enter stop mode. The backup flag (BCF) will be set to 1 automatically after the program enters stop mode.
The following diagram shows the stop release procedure:

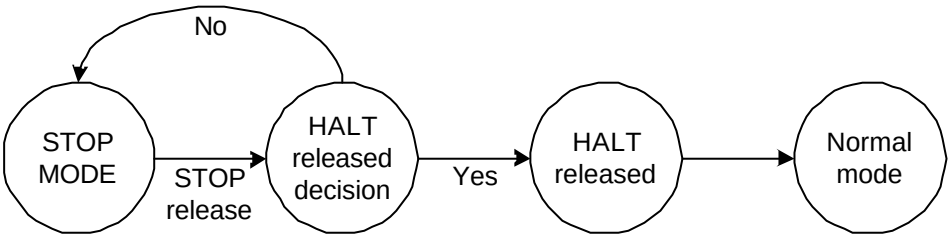


Figure 3-16 The stop release state machine

Before the stop instruction is executed, the following operations must be completed:
. Specify the stop release conditions through 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 stop mode is released by an interrupt request, the TM8740 will enter the halt mode immediately. While the interrupt is accepted, the halt mode will be released by the interrupt request. Stop mode returns by executing the RTS instruction after the completion of interrupt service.

After the stop release, it is necessary that either the MSB, or the MSC or the MCX instruction be executed to test the halt release signal. Then, the PLC instruction must 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 stop mode, but instead enters halt mode. When stop mode is released and an interrupt is accepted, the halt release signal (HRF) is reset automatically.

2-19 BACK UP FUNCTION

TM8740 provides a back up mode to avoid system malfunction when heavy loading occurs, such as buzzer activation... etc. Since heavy loading will cause a large voltage drop in the supply voltage, the system will malfunction in this condition.

Once the program enters back up mode (BCF = 1), 32.768KHz Crystal oscillator will operate in a large driver condition and the internal logic function operates with a higher supply voltage. TM8740 will get a higher power supply noise margin while back up mode is active, but it will also receive an increase in power consumption.

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

The back up function show in the following table.

TM8740 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)

	BCF = 0	BCF = 1
32.768KHz Crystal Oscillator	Small driver	Large driver
Voltage on BAK pin	VDD2	VDD2
Internal operating voltage	VDD2	VDD2

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

Chapter 3 Control Function

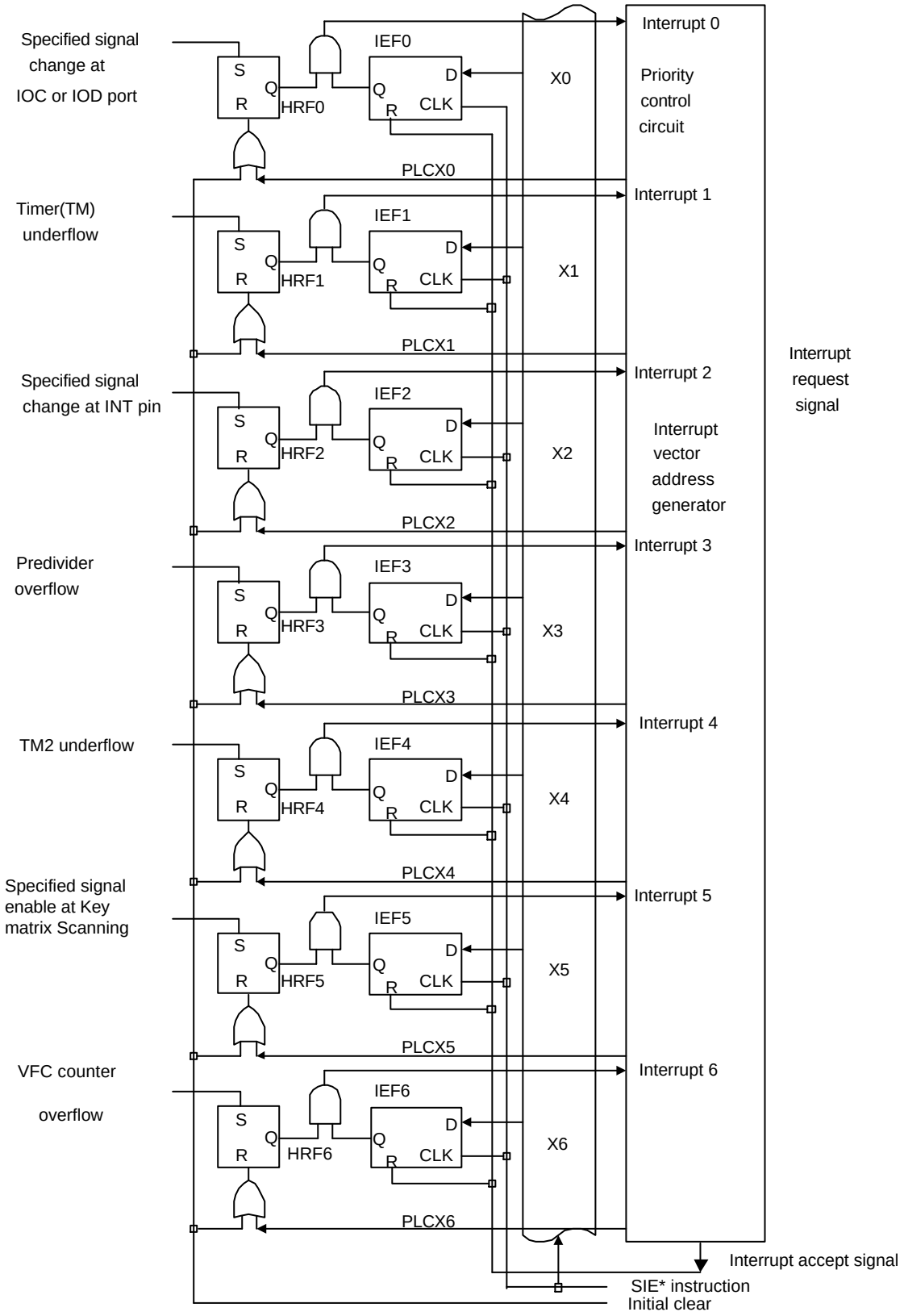
3-1 INTERRUPT FUNCTION

There are 7 interrupt resources: 3 external interrupt factors and 4 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 3-3 Interrupt information

Interrupt source	INT pin	IOC or IOD port	TMR1 underflow	Pre-divider overflow	TMR2 underflow	Key matrix Scanning	VFC counter overflow
Interrupt vector	010H	014H	018H	01CH	020H	024H	028H
Interrupt enable flag	IEF2	IEF0	IEF1	IEF3	IEF4	IEF5	IEF6
Interrupt priority	6^{th}	5^{th}	2^{nd}	1^{st}	3^{rd}	7^{th}	4^{th}
Interrupt request flag	Interrupt 2	Interrupt 0	Interrupt 1	Interrupt 3	Interrupt 4	Interrupt 5	Interrupt 6

The following figure shows the Interrupt Control Circuit



3-1-1 INTERRUPT REQUEST AND SERVICE ADDRESS

3-1-1-1 External interrupt factor

The external interrupt factor involves the use of the INT pin, IOC or IOD ports, or Key matrix Scanning.

1. External INT pin interrupt request

By using the mask option, either a rise or fall of the signal at the INT pin can be selected for applying an interrupt. If the interrupt enable flag 2 (IEF2) is set and the change signal on the INT pin matches the mask option, it will issue the HRF2. Interrupt 2 is accepted and the instruction at address10H is executed automatically. It is necessary to apply level "L" before the signal rises and level "H" after the signal rises to the INT pin for at least 1 machine cycle.

2. I/O port IOC(IOD) interrupt request.

An interrupt request signal (HRF0) is delivered when the input signal changes at the I/O port IOC(IOD) 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. Key matrix Scanning interrupt request.

An interrupt request signal (HRF5) is delivered when the input signal is generated in the scanning interval. If the interrupt enable flag 5 (IEF5) is set to 1 and interrupt 5 is accepted, the instruction at address 24H will be executed automatically.

3-1-1-2 Internal interrupt factor

The internal interrupt factor involves the use of timer 1 (TMR1), timer 2 (TMR2), RFC counter and the pre-divider.

1. Timer1 / 2 (TMR1 / 2) interrupt request

An interrupt request signal (HRF1 / 4) is delivered when timer1 / 2 (TMR1/ 2) underflows. In that case, if the interrupt enable flag 1 / 4 (IEF1 / 4) is set, interrupt 1 / 4 is accepted and the instruction at address 18H / 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. 16-bit counter of VFC interrupt request

An interrupt request signal (HRF6) is delivered when the 2nd falling edge applied on the VFC and the 16-bit counter stops operating. In this case, if the interrupt enable flag6 (IEF6) is set, interrupt 6 is accepted and the instruction at address 28H 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 put on hold. When the interrupt service routine is initiated, all of the interrupt enable flags (IEF0 ~ IEF6) are cleared and should be set on 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 IOC have been set to input mode.

```
PLC   7Fh        ;Clear all of the HRF flags
SCA   10h        ;enable the interrupt request of IOC
SIE*  7Fh        ;enable all interrupt requests

;……;all interrupts are requested simultaneously.

;An interrupt caused by the predivider overflow occurs, and interrupt  service is concluded.

SIE*  77h        ;Enable the interrupt request (except  the predivider).

;An interrupt caused by TM1 underflow occurs, and interrupt service is concluded.

SIE*  75h        ;Enable the interrupt request (except  the predivider and TMR1).

;An interrupt caused by TM2 underflow occurs, and interrupt service is concluded.

SIE*  65h        ;Enable the interrupt request(except the predivider, TMR1
and ;TMR2).

;An interrupt caused by RFC counter overflow occurs, and interrupt service is concluded.

SIE*  25h        ;Enable the interrupt request (except the predivider, TMR1,
                        ;TMR2, and the VFC counter).

;An interrupt is caused by IOC port, and interrupt service is concluded.

SIE*  24h        ;Enable the interrupt request (except  the predivider, TMR1,
                 ;TMR2, VFC counter, and IOC port)

;An interrupt is caused by the INT pin, and interrupt service is concluded.

SIE*  20h        ;Enable the interrupt request (except  the predivider, TMR1,
                 ;TMR2, VFC counter, IOC port, and INT)
;An interrupt is caused by Key matrix Scanning, 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) The return address of the interrupt service routine and 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 TM8740 will follow the procedure below:

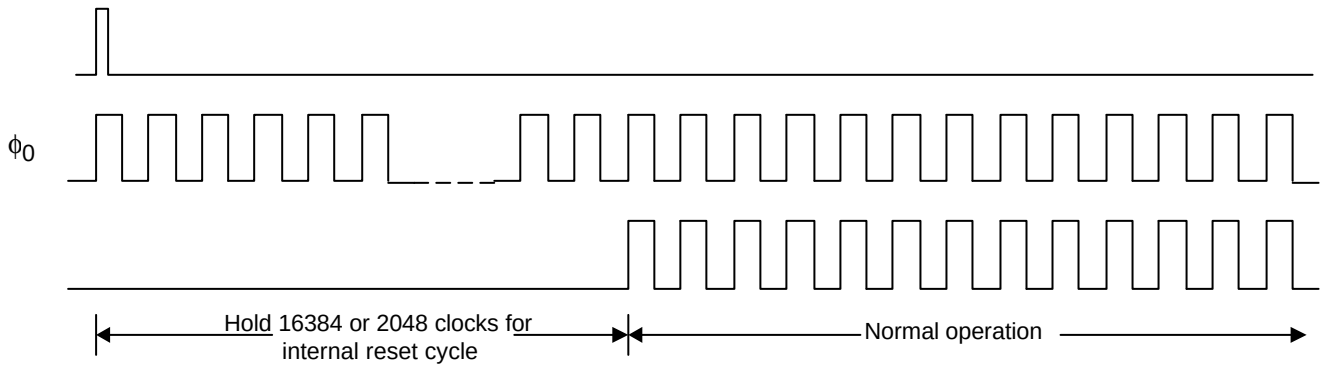
```
Instruction 1        ;In this instruction, interrupts are accepted.
NOP                  ;TM8740 stores the program counter data into the STACK. At this time,
                     ;no instructions will be executed, as with NOP instructions.
Instruction A        ;The program jumps to the interrupt 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

TM8740 contains four reset sources: power-on reset, RESET pin reset, IOC port reset and watchdog timer reset.
When a reset signal is accepted, TM8740 will generate a time period for its internal reset cycle. There are two types of internal reset cycle times that can be selected by mask option. One is PH15/2 and the other is PH12/2.



Internal reset cycle time is PH15/2

MASK OPTION table :

Mask Option name	Selected item
RESET TIME	(1) PH15/2

In this option, the reset cycle time will be extended at least 16384 clocks (clock source comes form pre-divider).

. Internal reset cycle time is PH12/2

MASK OPTION table :

Mask Option name	Selected item
RESET TIME	(2) PH12/2

In this option, the reset cycle time will be extended at least 2048 clocks (clock source comes form pre-divider).

3-2-1 POWER-ON RESET

TM8740 provides a power-on reset function. If the power (VDD) is turned on or the power supply drops below 0.6V, it will generate a power-on reset signal.
The power-on reset function can be disabled through the mask option.

MASK OPTION table :

Mask Option name	Selected item
POWER ON RESET	(1) USE
POWER ON RESET	(2) NO USE

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.
Two types of reset methods for the RESET pin and the type can be set with the mask option. One is level reset and other is pulse reset.
It is recommended that you connect a capacitor (0.1uf) between the RESET pin and the VDD. This connection will prevent the issuance of the bounce signal on the RESET pin.

3-2-2-1 Level Reset

Once a "H" signal is applied on the RESET pin, TM8740 will not release the reset cycle until the signal on the RESET pin is returned to "0". After the signal on the reset pin is cleared to 0, TM8740 begins the internal reset cycle and then releases the reset status automatically.

MASK OPTION table :

Mask Option name	Selected item
RESET PIN TYPE	(1) LEVEL

3-2-2-2 Pulse Reset

Once a "H" signal is applied on the RESET pin, TM8740 will escape from the reset state and begin normal operation after the internal reset cycle automatically, no matter whether the signal on the RESET pin is returned to "0" or not.

MASK OPTION table :

Mask Option name	Selected item
RESET PIN TYPE	(2) PULSE

The following table shows the initial conditions of TM8740 in reset cycle.

Program counter	(PC)	Address 000H
Start condition flags 1 to 7	(SCF1-7)	0
Backup flag	(BCF)	1
Stop release enable flags 4,5,7	(SRF3,4,5,7)	0
Switch enable flags 4	(SEF3,4)	0
Halt release request flag	(HRF 0~6)	0
Halt release enable flags 1 to 3	(HEF1-6)	0
Interrupt enable flags 0 to 3	(IEF0-6)	0
Alarm output	(ALARM)	DC 0
Pull-down flags in I/OC, I/OD port		1(with pull-down resistor)
Input/output ports I/OA, I/OB, I/OC, I/OD	(PORT I/OA, I/OB, I/OC, I/OD)	Input mode
I/OC, I/OD port chattering clock	Cch	PH10*
Frequency generator clock source and duty cycle	Cfq	PH0, duty cycle is 1/4, output is inactive
LCD driver output		All lighted (mask option)*
Timer 1/2		Inactive
Watchdog timer	(WDT)	Reset mode, WDF = 0
Clock source	(BCLK)	XT clock (slow speed clock in dual clock option)

Notes: PH3: the 3rd output of predivider

PH10: the 10th output of predivider

Mask option can unlighted all of the LCD output

3-2-3 IOC Port / Key Matrix RESET

The key reset function can be selected by mask option. When the IOC port or key matrix scanning input (KI1~4) is in use, the '0' signal is applied to all these pins that were set to the input mode at the same time (KI1~4 pins need to wait scanning time), the reset signal is delivered.

MASK OPTION table :

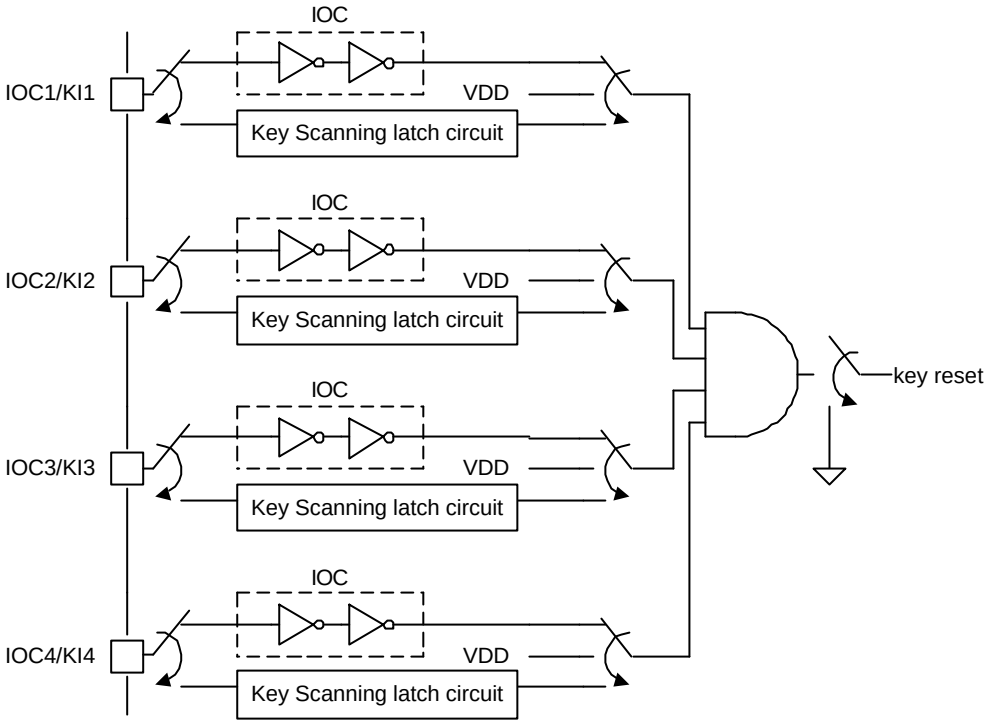
IOC or KI pins are used as key reset :

Mask Option name	Selected item
IOC1/KI1 FOR KEY RESET	(1) USE
IOC2/KI2 FOR KEY RESET	(1) USE
IOC3/KI3 FOR KEY RESET	(1) USE
IOC4/KI4 FOR KEY RESET	(1) USE

IOC or KI pins aren't used as key reset :

Mask Option name	Selected item
IOC1/KI1 FOR KEY RESET	(2) NO USE
IOC2/KI2 FOR KEY RESET	(2) NO USE
IOC3/KI3 FOR KEY RESET	(2) NO USE
IOC4/KI4 FOR KEY RESET	(2) NO USE

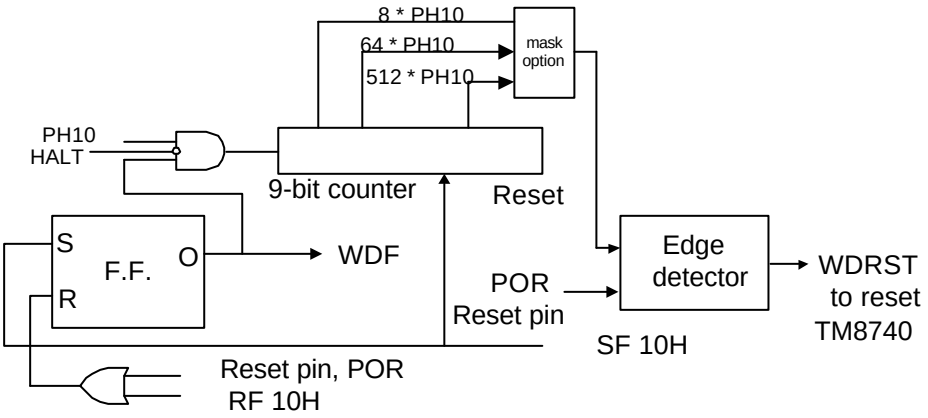
The following figure shows the key reset organization.



3-2-4 WATCHDOG RESET

The timer is used to detect unexpected execution sequences caused by software run-away. The watchdog timer consists of a 9-bit binary counter. The timer input (PH10) is the 10th stage output of the pre-divider.
When the watchdog timer overflows, it generates a reset signal to reset TM8740. Most of the functions in TM8740 will be initiated except for the watchdog timer (which is still active); The WDF flag will not be affected and PH0 ~ PH10 of the pre-divider will not be reset.

The following figure shows the organization of the watchdog timer.



During initial reset (power on reset [POR] or reset pin), the timer is inactive and the watchdog flag (WDF) is reset. Instruction SF 10h will enable the watchdog timer and set the watchdog flag (WDF) to 1. At the same time, the contents of the timer will be cleared. Once the watchdog timer is enabled, the timer will be paused when the program enters halt or stop mode. When the TM8740 wakes up from halt or stop mode, the timer operates continuously. It is recommended that you execute the SF 10h instruction before the program enters the halt or stop mode in order to initialize the watchdog timer.
Once the watchdog timer is enabled, the program must execute the SF 10h instruction periodically to prevent timer overflow.

The overflow time interval of the watchdog timer is selected by mask option :

MASK OPTION table :

Mask Option name	Selected item
WATCHDOG TIMER OVERFLOW TIME INTERVAL	(1) 8 x PH10
WATCHDOG TIMER OVERFLOW TIME INTERVAL	(2) 64 x PH10
WATCHDOG TIMER OVERFLOW TIME INTERVAL	(3) 512 x PH10

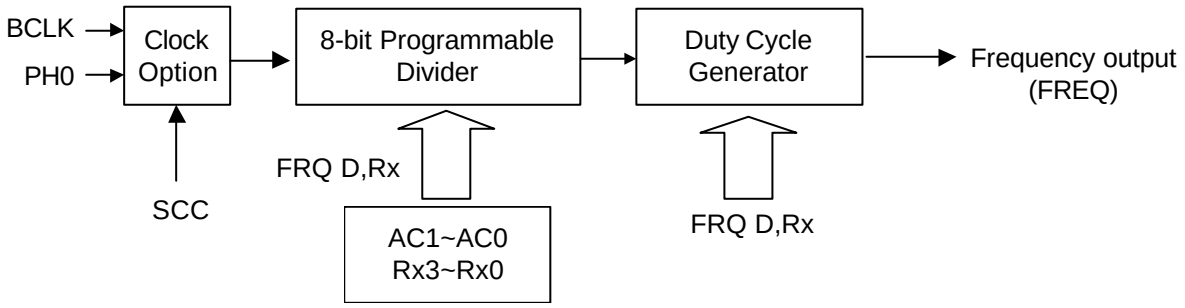
Note : timer overflow time interval is about 16 seconds when PH0 = 32.768KHz

3-3 CLOCK GENERATOR

3-3-1 FREQUENCY GENERATOR

The Frequency Generator is a versatile programmable divider that is capable of delivering a clock with wide frequency range and different duty cycles. The output of the frequency generator may be the clock source for the alarm function, timer1, timer2 and VFC counter.

The following shows the organization of the frequency generator.



The SCC instruction may specify the clock source selection for the frequency generator. The frequency generator outputs the clock with different frequencies and duty cycles corresponding to the preset data of FRQ related instructions. The FRQ related instructions preset a letter N into the programming divider and the letter D into the duty cycle generator. The frequency generator will then output the clock using the following formula:

FREQ=(clock source) / ((N+1) * X) Hz. (X=1,2,3,4 for 1/1,1/2,1/3,1/4 duty)

This letter N is a combination of the data memory and the accumulator (AC), or the table ROM data or the operand data specified in the FRQX instruction. The following table shows the bit pattern of the combination.

The following table shows the bit pattern of the preset letter N

	The bit pattern of preset letter N							
Programming divider	bit7	Bit6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0
FRQ D,Rx	AC3	C2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
FRQ D,@HL	T7	T6	T5	T4	T3	T2	T1	T0
FRQX D,X	X7	X6	X5	X4	X3	X2	X1	X0

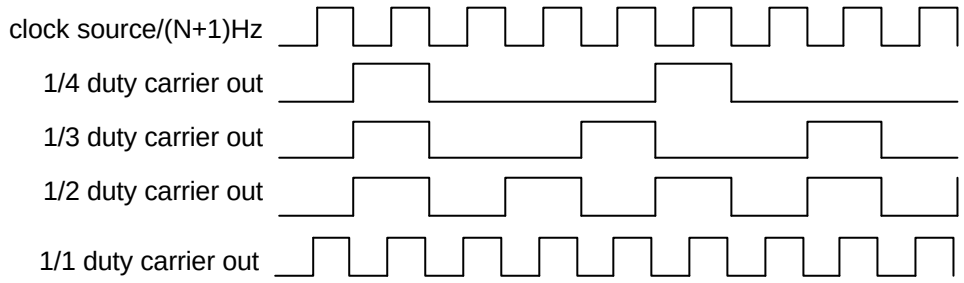
Notes: 1. T0 ~ T7 represents the data of table ROM.

2. X0 ~ X7 represents the data specified in operand X.

The following table shows the bit pattern of the preset letter D

Preset Letter D		Duty Cycle
D1	D0	
0	0	1/4 duty
0	1	1/3 duty
1	0	1/2 duty
1	1	1/1 duty

The following diagram shows the output waveform for different duty cycles.



3-3-2 Melody Output

The frequency generator may generate frequencies for melody usage. When the frequency generator is used to generate melody output, the tone table is shown below:

1. The clock source is PH0, i.e. 32,768 Hz
2. The duty cycle is 1/2 Duty (D=2)
3. "FREQ" is the output frequency
4. "ideal" is the ideal tone frequency
5. "%" is the frequency deviation

The following table shows the note table for melody application

Tone	N	FREQ	Ideal	%	Tone	N	FREQ	Ideal	%
C2	249	65.5360	65.4064	0.19	C4	62	260.063	261.626	-0.60
#C2	235	69.4237	69.2957	0.18	#C4	58	277.695	277.183	0.18
D2	222	73.4709	73.4162	0.07	D4	55	292.571	293.665	-0.37
#D2	210	77.6493	77.7817	-0.17	#D4	52	309.132	311.127	-0.64
E2	198	82.3317	82.4069	-0.09	E4	49	327.680	329.628	-0.59
F2	187	87.1489	87.3071	-0.18	F4	46	348.596	349.228	-0.18
#F2	176	92.5650	92.4986	0.07	#F4	43	372.364	369.994	0.64
G2	166	98.1078	97.9989	0.11	G4	41	390.095	391.995	-0.48
#G2	157	103.696	103.826	-0.13	#G4	38	420.103	415.305	1.16
A2	148	109.960	110.000	-0.04	A4	36	442.811	440.000	0.64
#A2	140	116.199	116.541	-0.29	#A4	34	468.114	466.164	0.42
B2	132	123.188	123.471	-0.23	B4	32	496.485	493.883	0.53
C3	124	131.072	130.813	0.20	C5	30	528.516	523.251	1.01
#C3	117	138.847	138.591	0.19	#C5	29	546.133	554.365	-1.48
D3	111	146.286	146.832	-0.37	D5	27	585.143	587.330	-0.37
#D3	104	156.038	155.563	0.31	#D5	25	630.154	622.254	1.27
E3	98	165.495	164.814	0.41	E5	24	655.360	659.255	-0.59
F3	93	174.298	174.614	-0.18	F5	22	712.348	698.456	1.99
#F3	88	184.090	184.997	-0.49	#F5	21	744.727	739.989	0.64
G3	83	195.048	195.998	-0.48	G5	20	780.190	783.991	-0.48
#G3	78	207.392	207.652	-0.13	#G5	19	819.200	830.609	-1.37
A3	73	221.405	220.000	0.64	A5	18	862.316	880.000	-2.01
#A3	69	234.057	233.082	0.42	#A5	17	910.222	932.328	-2.37
B3	65	248.242	246.942	0.53	B5	16	963.765	987.767	-2.43

Note:

1. Above variation does not include X'tal variation.
2. If PH0 = 65536Hz, C3 - B5 may have more accurate frequency.

During the application for melody output, sound effect output or carrier output for remote controls, the frequency generator needs to combine with the alarm function (BZB, BZ). For detailed information about this application, refer to section 3-4.

3-3-3 Halver / Doubler / Tripler

The halver / doubler / tripler circuits are used to generate the bias voltage for LCD and are composed of a combination of PH2, PH3, PH4, PH5. When the Li battery application is used, the 1/2 VDD voltage generated by the halver operation is supplied to the circuits which are not related to input / output operation.

3-3-4 Alternating Frequency for LCD

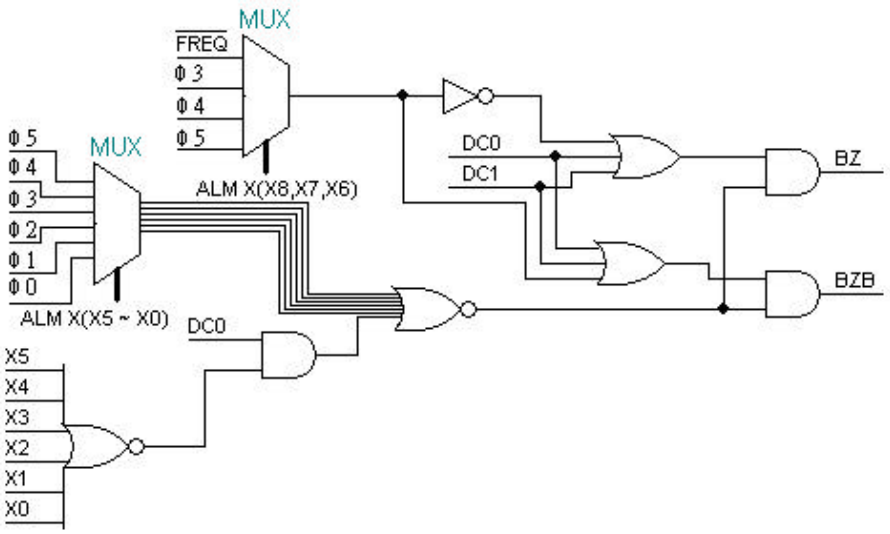
The alternating frequency for LCDs is a frequency used to make the LCD waveform.

3-4 BUZZER OUTPUT PINS

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

MASK OPTION table :

Mask Option name	Selected item
SEG30/IOB3/BZB	(3) BZB
SEG31/IOB4/BZ	(3) 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 FREQ, PH3(1024Hz), 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:

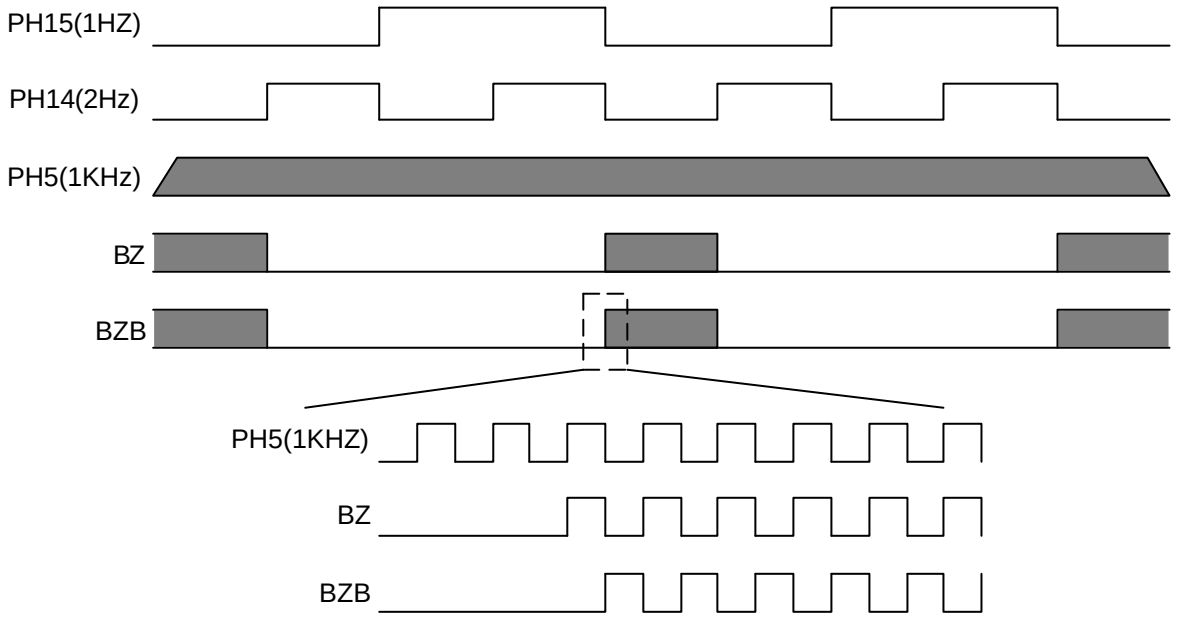
1. The high frequency clock source should only be one of PH3, PH4, PH5 or FREQ, and the lower frequency may be any/all of the combinations from PH10 ~ PH15.
2. The frequency in parentheses corresponding to the input clock of the pre-divider (PH0) is 32768Hz.
3. The BZ and BZB pins will output DC0 after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

Example:

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

```
LDS   20h, 0Ah
…….
ALM   70h          ; Output the waveform.
……
```

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



3-4-2 THE CARRIER FOR REMOTE CONTROL

If buzzer output combines with the timer and frequency generator, the output of the BZ pin may deliver waveforms for IR remote controllers. For remote control usage, the setting value of the frequency generator must be greater than or equal to 3, and the ALM instruction must be executed immediately after the FRQ related instructions in order to deliver the FREQ signal to the BZ pin as the carrier for IR remote controller.

Example:

```
SHE         1       ;Enable timer 1 halt release enable flag.
TMSX        3Fh     ;Set value for timer 1 is 3Fh and the clock source is PH9.
SCC         40h     ;Set the clock source of  the frequency generator as BCLK.
FRQX        2, 3    ;FREQ = BCLK / (4*2), setting value for the frequency generator
                    ;is 3 and duty cycle is 1/2.
ALM         100h    ;FREQ signal is outputted. This instruction must be executed
                    ;after the FRQ related instructions.
HALT                ;Wait for the halt release caused by timer 1.
……                  ;Halt released.
ALM         0       ;Stop the buzzer output.
```

3-5 INPUT / OUTPUT PORTS

Four I/O ports are available in TM8740 : IOA, IOB, IOC and IOD. Each I/O port is composed of 4 bits and has the same basic function.
When the I/O pins are defined as non-IO functions by mask option, the input / output function of the pins will be disabled.

3-5-1 IOA PORT

IOA3 ~ IOA4 pins are MUX with SEG26 and SEG27 pins respectively by mask option.

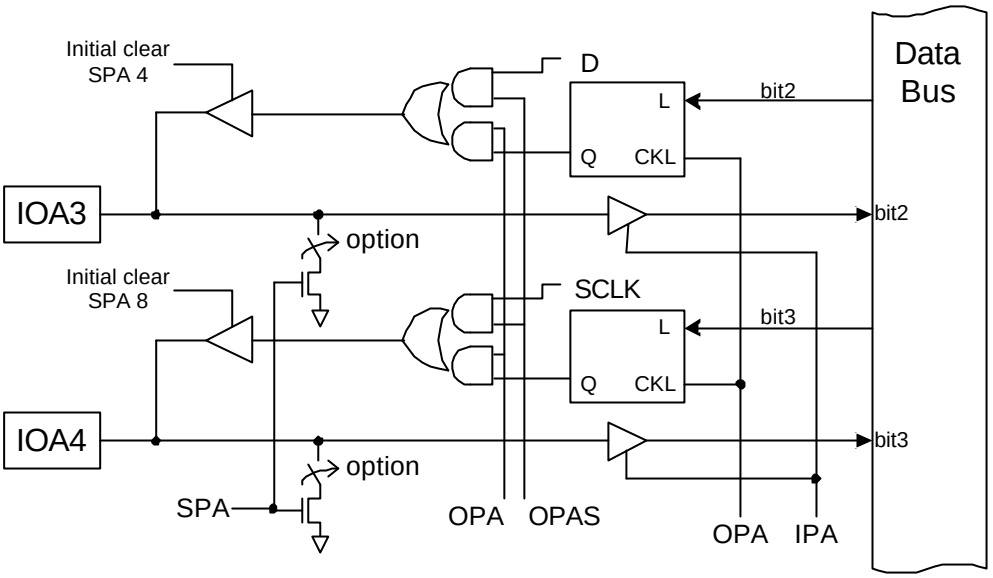
MASK OPTION table :

Mask Option name	Selected item
SEG26/IOA3	(2) IOA3
SEG27/IOA4	(2) IOA4

In initial reset cycle, the IOA port is set to input mode and each bit of port can be set to input 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 set to input mode will still remain in the input mode.
Executing IPA instructions may store the signals applied to the IO pins into specified data memory locations. When the IO pins are set to output mode, executing IPA instructions will store the contents of the latch of the output pin into the specified data memory location.
Before executing the SPA instruction to set the I/O pins to 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 changes.
The IOA port has a built-in pull-down resistor. The pull-low device for each pin is selected by mask option and then executing the SPA instruction to enable / disable the device.

Pull-low function option

Mask Option name	Selected item
IOA PULL LOW RESISTOR	(1) USE
IOA PULL LOW RESISTOR	(2) NO USE



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

IOB1 ~ IOB4 pins are MUXed with SEG28, SEG29, BZB / SEG30 and BZ / SEG31 pins respectively by mask option.

MASK OPTION table :

Mask Option name	Selected item
SEG28/IOB1	(2) IOB1
SEG29/IOB2	(2) IOB2
SEG30/IOB3/BZB	(2) IOB3
SEG31/IOB4/BZ	(2) IOB4

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.

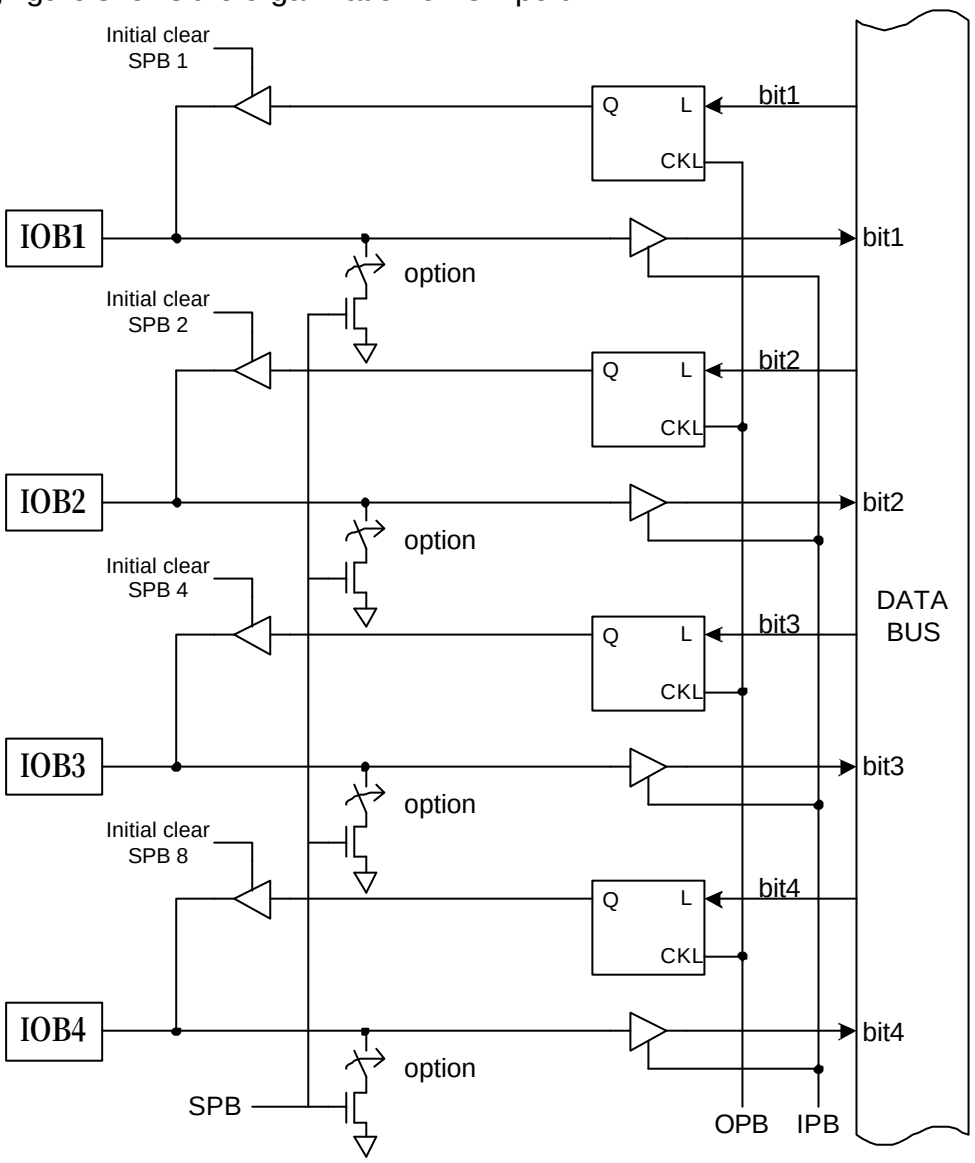
After the reset cycle, the IOB port is set as input and each bit of port can be defined as input or output individually by executing SPB instructions. Executing OPB instructions may output the contents of specified data memory to the pins set to output mode; the other pins which are set to input will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

If IPB instructions are executed, they may store the signals applied on IOB pins into specified data memory locations. When the IOB pins are set as the output, executing the IPB instruction will save the data stored in the output latch into the specified data memory location.

Before executing the SPB instruction to set the I/O pins to 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 changes.

IOB port has a built-in pull-down resistor. The pull-low device for each pin is selected by mask option and executing the SPB instruction to enable / disable the device.

The following figure shows the organization of IOB port.



Pull-low function option

Mask Option name	Selected item
IOB PULL LOW RESISTOR	(1) USE
IOB PULL LOW RESISTOR	(2) NO USE

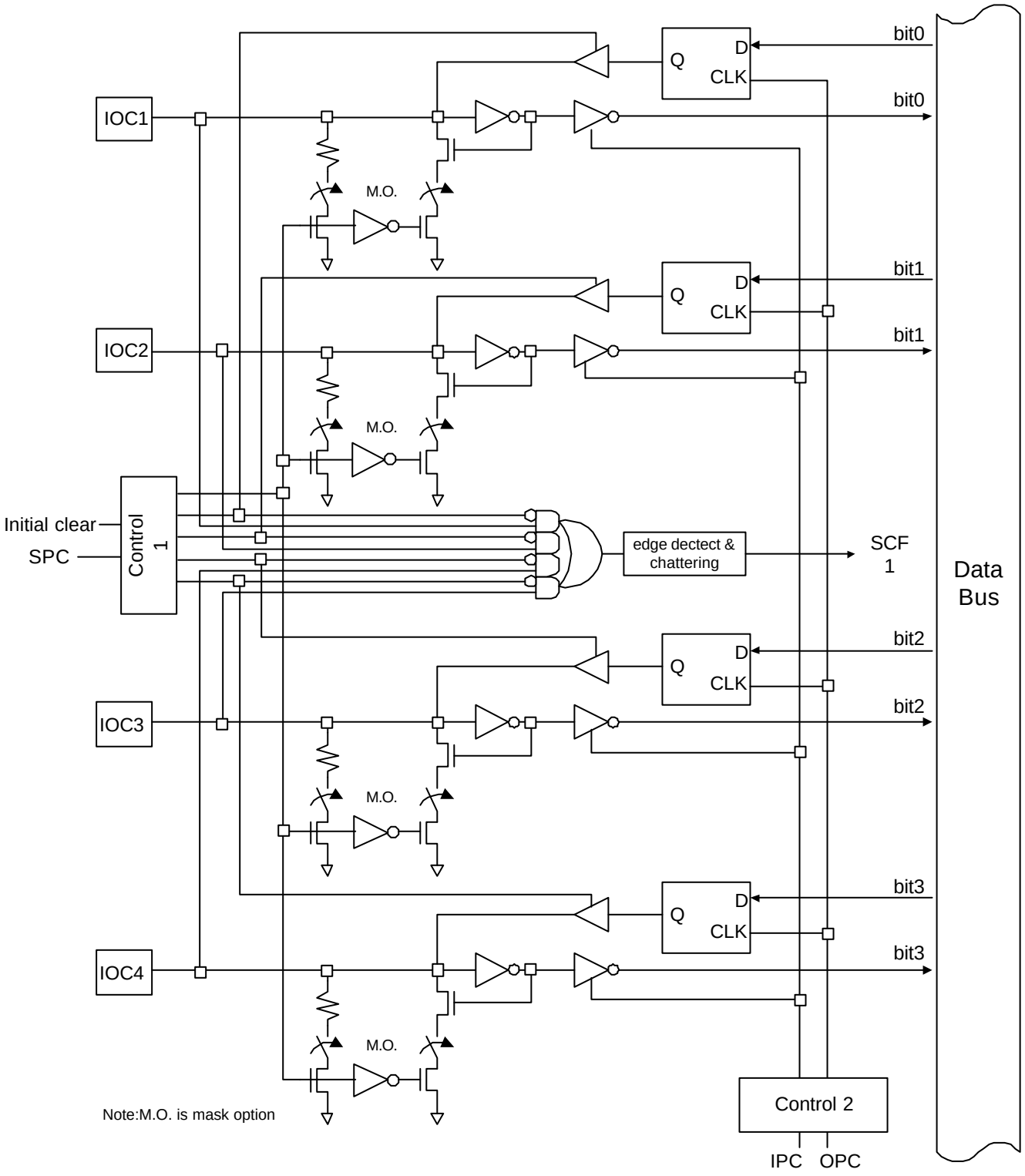
3-5-3 IOC PORT

IOC1 ~ IOC4 pins are MUXed with KI1 / SEG32, KI2 / SEG33, KI3 / SEG34 and KI4 / SEG35 pins respectively by mask option.

MASK OPTION table :

Mask Option name	Selected item
SEG32/IOC1/KI1	(2) IOC1
SEG33/IOC2/KI2	(2) IOC2
SEG34/IOC3/KI3	(2) IOC3
SEG35/IOC4/KI4	(2) IOC4

After the reset cycle, the IOC port is set to input mode. Each bit of port can be set to input or output mode individually by executing SPC instructions. Executing the OPC instruction may output the contents of specified data memory to the pins set as output; the other pins which are set to input will still remain in the input mode.



This figure shows the organization of IOC port.

When IPC instructions are executed, they may store the signals applied to the IOC pins in specified data memory locations. When the IOC pins are set as output, executing IPC instructions will save the data stored in the output latches in the specified data memory location.
Before executing SPC instructions to set the IOC pins as output, the OPC instruction must be executed to output data to the output latches. This will prevent the occurrence of the chattering signal when the IOC pins change to output mode.

Note: If the input level is in the floating state, a large current (straight-through current) flows to the input buffer when both the pull low and L-level hold devices are disabled. The input level must not be in the floating state

The IOC port may select the pull-low device or the low-level hold device for each pin through the mask option or enable / disable this device by program setting. When the pull-low device and low-level hold device are both enabled through the mask option, the reset will enable the pull-low device and disable the low-level hold device. Executing the SPC 10h instruction will also enable the pull-low device and disable the low-level hold device. Executing the SPC 0h instruction will disable the pull-low device and enable the low-level hold device. When the IOC pin has been set to output mode, both the pull-low and low-level hold devices will be disabled.

MASK OPTION table :

Pull-low function option

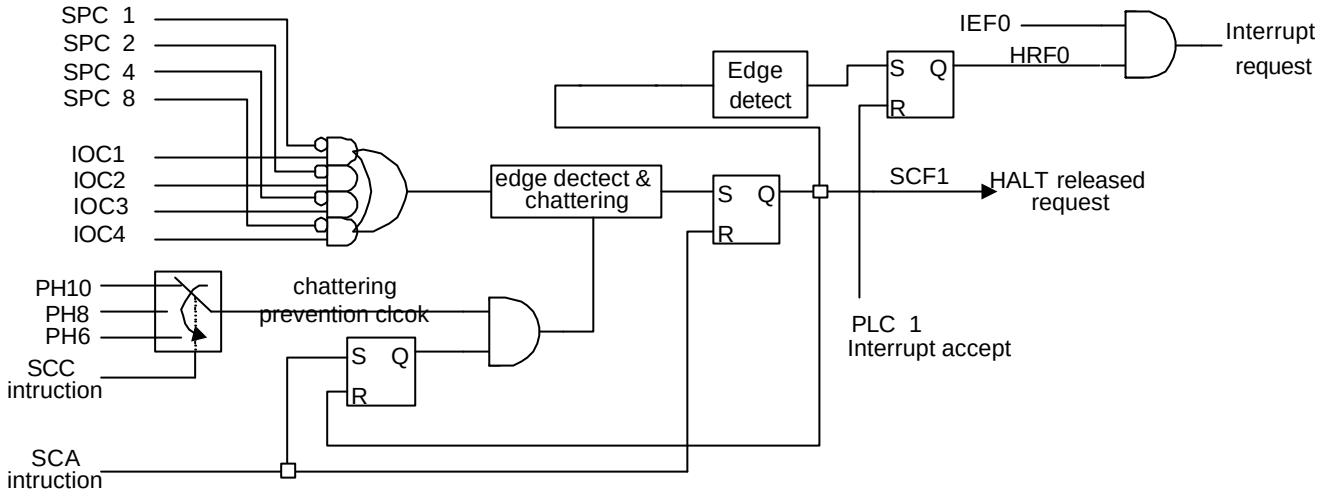
Mask Option name	Selected item
IOC PULL LOW RESISTOR	(1) USE
IOC PULL LOW RESISTOR	(2) NO USE

The low-level-hold function will not be available when pull-low function is not activated.
Low-level-hold function option

Mask Option name	Selected item
C PORT LOW LEVEL HOLD	(1) USE
C PORT LOW LEVEL HOLD	(2) NO USE

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

The port IOC is capable of preventing high / low chattering of the switch signal applied on IOC1 to IOC4 pins. The chattering prevention time can be selected as PH10 (32ms), PH8 (8ms) or PH6 (2ms) by executing the SCC instruction. The default selection is PH10 after the reset cycle. When the pins of the IOC port are set to 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. IOC1) is changed from "L" level to "H" level or from "H" level to "L" level, and the remaining pins (ex, IOC2 to IOC4) are held at "L" level.

When the signal changes at the input pins of the IOC port specified by the SCA instruction occur and stay in that 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 (SCF1). At that time, the chattering prevention clock will stop due to the delivery of SCF1. The SCF1 will be reset to 0 by executing the SCA instruction; the chattering prevention clock will be enabled at the same time. If the SCF1 has been set to 1, the halt release request flag 0 (HRF0) will be delivered. In this case, if the port IOC 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 IOC1 to IOC4, the input data at the port IOC must be read into the RAM immediately after the halt mode is released.

3-5-4 IOD PORT

IOD1 ~ IOD4 pins are MUXed with SEG36, SEG37, SEG38 and SEG39 pins respectively by mask option.

MASK OPTION table :

Mask Option name	Selected item
SEG36/IOD1	(2) IOD1
SEG37/IOD2	(2) IOD2
SEG38/IOD3	(2) IOD3
SEG39/IOD4	(2) IOD4

After the reset cycle, the IOD port is set to input mode; each bit of port can be set to input or output mode individually by executing SPD instructions. Executing the OPD instruction outputs the contents of specified data memory locations to the pins set as output; the other pins which are set as input will still remain the in the input mode.

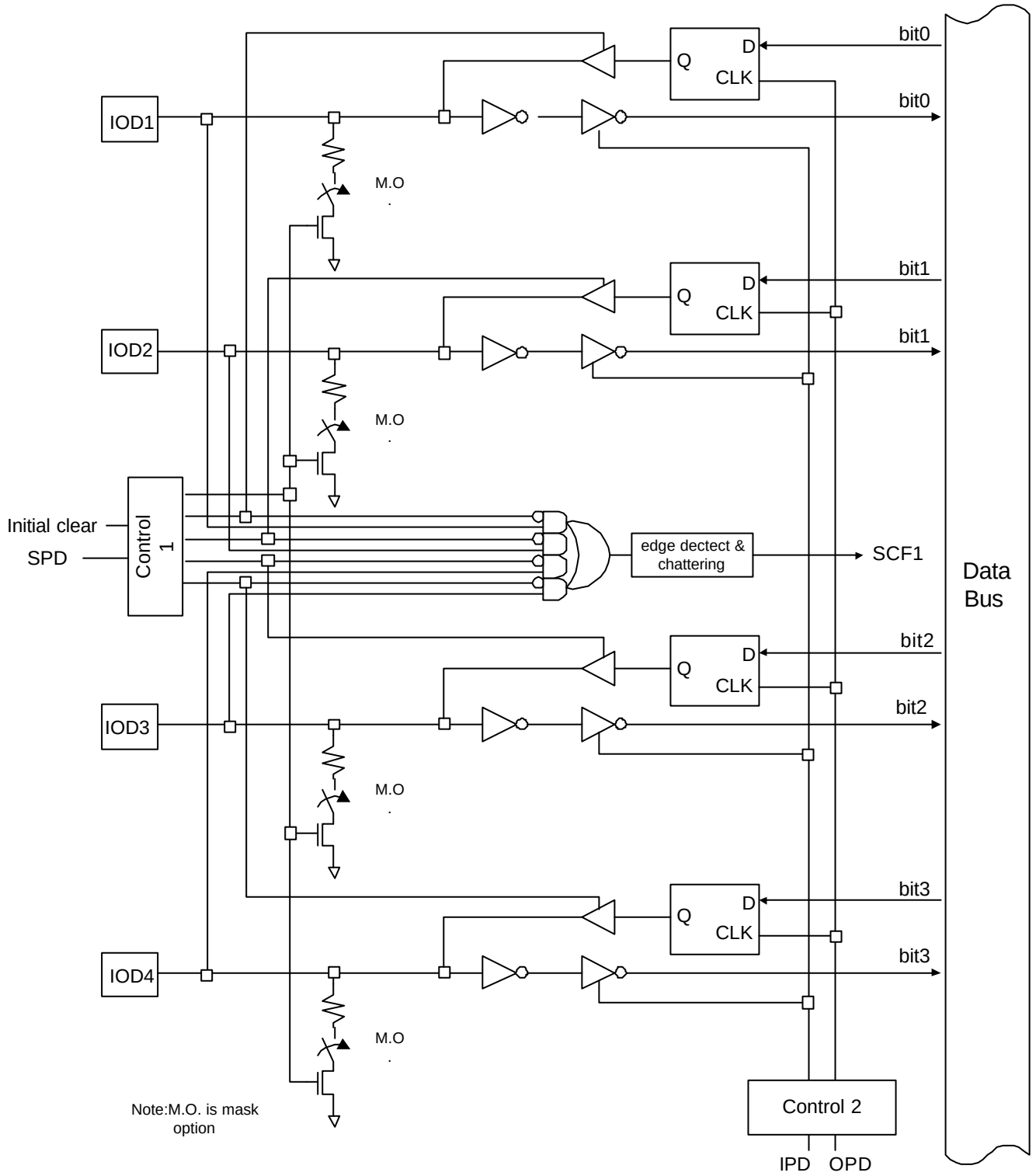
Executing IPD instructions will store the signals applied to the IOD pins in the specified data memory locations. When the IOD pins are set as output, executing IPD instructions will save the data stored in the output latches in the specified data memory locations.
Before executing SPD instructions to define the IOD pins as output, the OPD instructions must be executed to output the data to those output latches. This will prevent the chattering signal when the IOD pins change to output mode.

IOD port has a built in pull-low device for each pin that is selected by mask option. To enable or disable this device, execute the SPD instruction.
When the IOD pin has been set to the output mode, the pull-low device will be disabled.

MASK OPTION table :
Pull-low function option

Mask Option name	Selected item
IOC PULL LOW RESISTOR	(1) USE
IOC PULL LOW RESISTOR	(2) NO USE

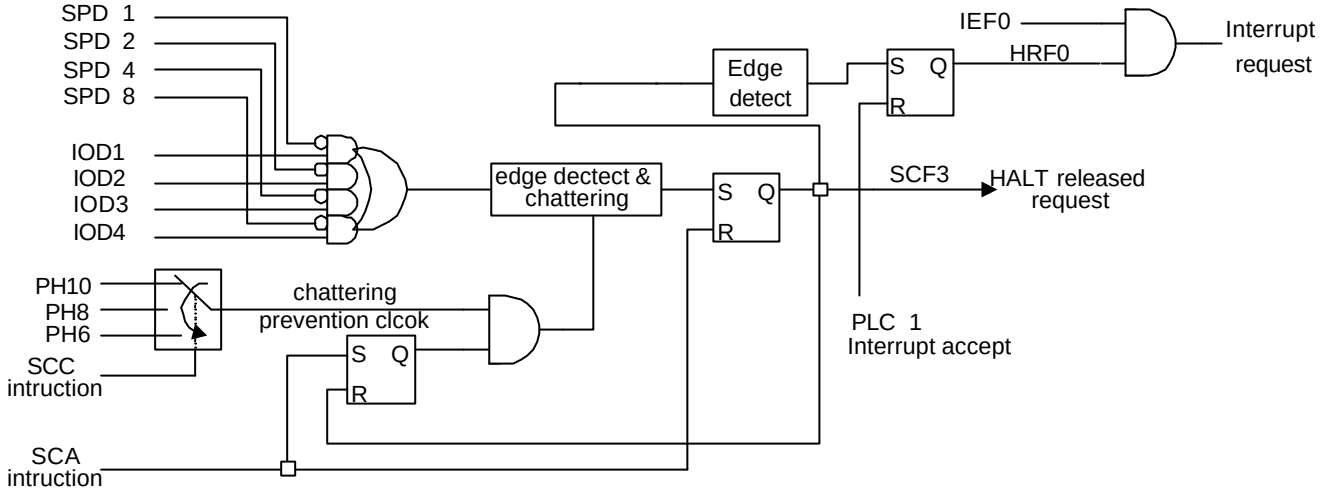
Note: If the input level is in the floating state, a large current (straight-through current) flows to the input buffer when both the pull low and L-level hold devices are disabled. The input level must not be in the floating state



This figure shows the organization of IOD port.

3-5-4-1 Chattering Prevention Function and Halt Release

The port IOD is capable of preventing high / low chattering of the switch signal applied on the IOD1 to IOD4 pins. Chattering prevention time can be selected as PH10 (32ms), PH8 (8ms) or PH6 (2ms) by executing the SCC instruction; the default selection is PH10 after the reset cycle. When the pins of the IOD port are set 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.



This 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. IOD1) is changed from "L" level to "H" level or from "H" level to "L" level, and the remaining pins (ex, IOD2 to IOD4) are held at "L" level.

When the signal changes at the input pins of IOD 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 (SCF3). At that time, the chattering prevention clock will stop due to the delivery of SCF3. The SCF3 will be reset to 0 by executing SCA instruction and the chattering prevention clock will be enabled at the same time. If the SCF3 has been set to 1, the halt release request flag 0 (HRF0) will be delivered. In this case, if the port IOD 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 IOD1 to IOD4, the input data at the port IOD must be read into the RAM immediately after the halt mode is released.

3-6 EXTERNAL INT PIN

The INT pin can be selected as a pull-up, pull-down, or open type by mask option. The signal change (either rising edge or falling edge by mask option) sets the interrupt flag, delivering the halt release request flag 2 (HRF2). In this case, if the halt release enable flag (HEF2) is provided, the start condition flag 2 is delivered. If the INT pin interrupt enable mode (IEF2) is provided, the interrupt is accepted.

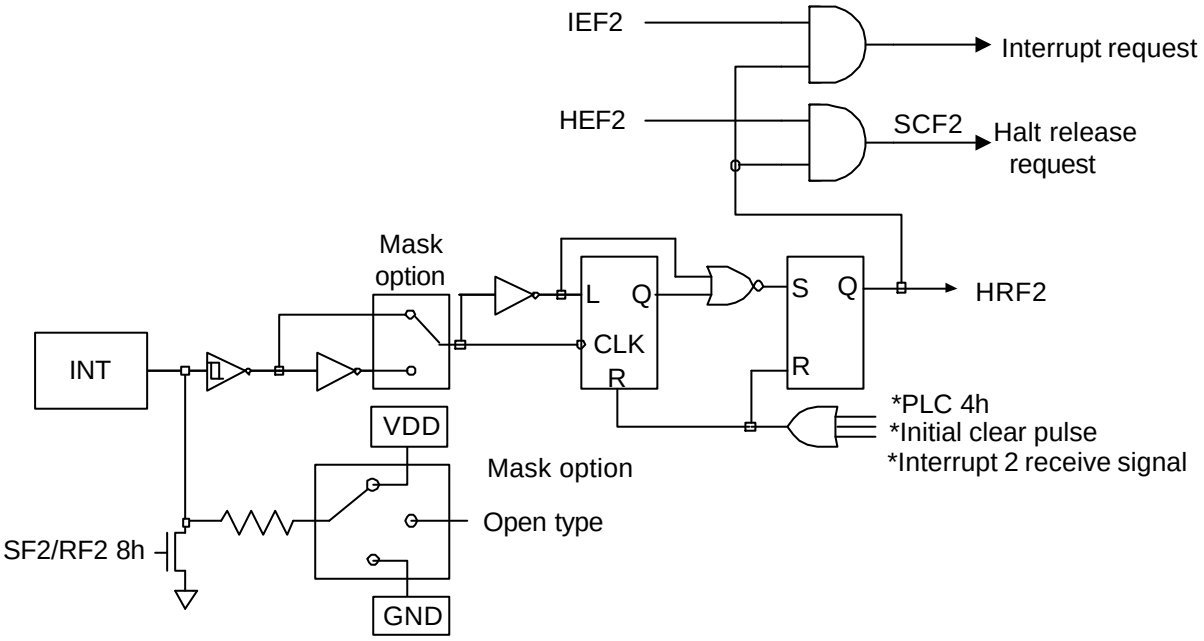
MASK OPTION table :

For internal resistor type :

Mask Option name	Selected item
INT PIN INTERNAL RESISTOR	(1) PULL HIGH
INT PIN INTERNAL RESISTOR	(2) PULL LOW
INT PIN INTERNAL RESISTOR	(3) OPEN TYPE

For input triggered type :

Mask Option name	Selected item
INT PIN TRIGGER MODE	(1) RISING EDGE
INT PIN TRIGGER MODE	(2) FALLING EDGE



This figure shows the INT Pin Configuration

Note: For Ag battery power supply, positive power is connected to VDD1; for anything other than Ag battery power supply, it is connected to VDD2.

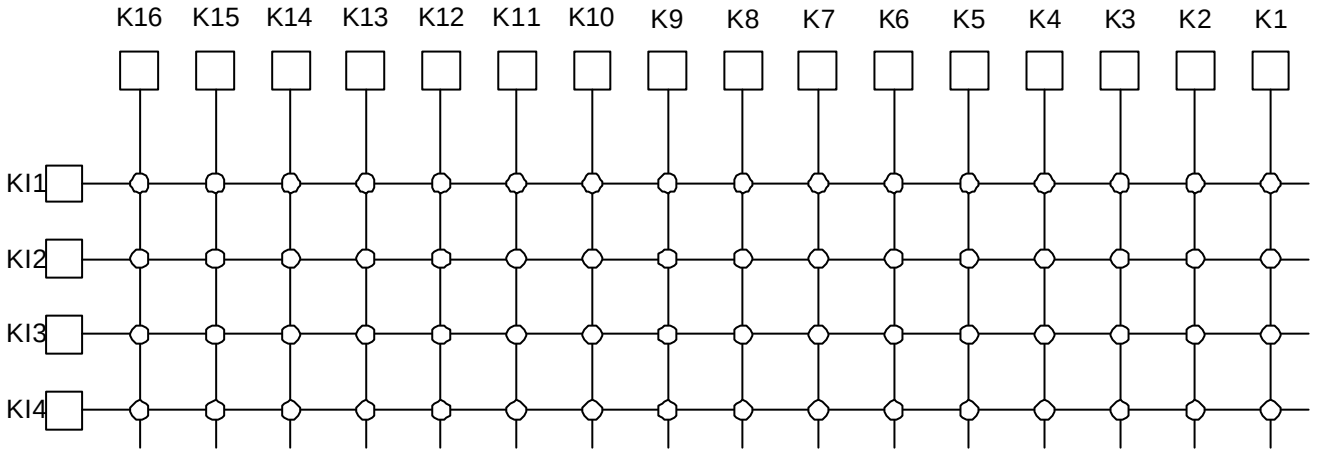
3-7 Key Matrix Scanning

TM8740 shares the timing of the LCD waveform to scan the key matrix circuitry. These scanning output pins are SEG1~16(for easy to understand, named these pins as K1 ~ K16). The time sharing of the LCD waveform will not affect the display of the LCD panel. The input port of the key matrix circuitry is composed of KI1 ~ KI4 pins (these pins are muxed with SEG32 ~ SEG35 pins and selected by mask option).

MASK OPTION table :

Mask Option name	Selected item
SEG32/IOC1/KI1	(3) KI1
SEG33/IOC2/KI2	(3) KI2
SEG34/IOC3/KI3	(3) KI3
SEG35/IOC4/KI4	(3) KI4

The typical application circuit of the key matrix scanning is shown below:



Executing SPKX X, SPK Rx, and SPK @HL instructions could set the scanning type of the key matrix. The bit pattern of these 3 instructions are shown below :

Instruction	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPKX X	X7	X6	X5	X4	X3	X2	X1	X0
SPK Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
SPK @HL	T@HL7	T@HL6	T@HL5	T@HL4	T@HL3	T@HL2	T@HL1	T@HL0

The following description shows the bit definitions in the operand of the SPKX instruction.

X_6 = " 0 ", when HEF5 is set to 1, the HALT release request (HRF5) will be set to 1 after the key depressed on the key matrix, and then SCF7 will be set to 1.

" 1 ", when HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after each scanning cycle regardless of key depression, and then SCF7 will be set to 1.

$X_7X_5X_4$ = 000, in this setting, each scanning cycle only checks one specified column (K1 ~ K16) on the key matrix. The specified column is defined by the setting of X_3 ~ X_0.

X_3 ~ X_0 = 0000, activates K1 column

X_3 ~ X_0 = 0001, activates K2 column

……..

X_3 ~ X_0 = 1110, activates K15 column

X_3 ~ X_0 = 1111, activates K16 column

$X_7X_5X_4$ = 001, in this setting, all of the matrix columns (K1 ~ K16) will be checked simultaneously in each scanning cycle. X_3 ~ X_0 are not a factor.

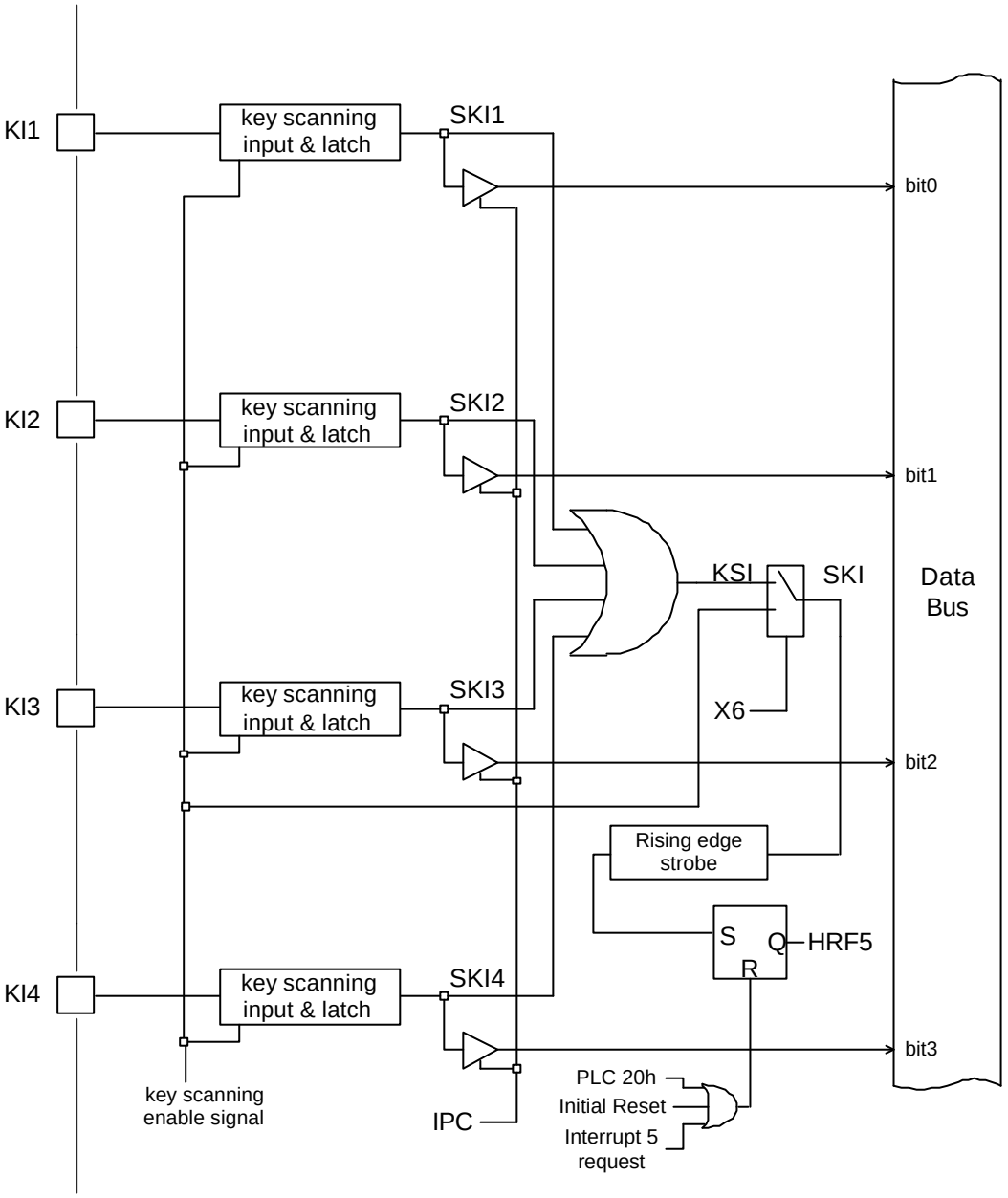
$X_7X_5X_4$ = 010, in this setting, the key matrix scanning function will be disabled. X_3 ~ X_0 are not a factor.

$X_7X_5X_4$ = 10X, in this setting, each scanning cycle checks 8 specified columns on the key matrix. The specified column is defined by the setting of X_3.
X_3 = 0, activates K1 ~ K8 columns simultaneously
X_3 = 1, activates K9 ~ K16 columns simultaneously
X_2 ~ X_0 don't care.

$X_7X_5X_4$ = 110, in this setting, each scanning cycle checks four specified columns on key matrix. The specified columns are defined by the setting of X_3 and X_2.
X_3X_2 = 00, activates K1 ~ K4 columns simultaneously
X_3X_2 = 01, activates K5 ~ K8 columns simultaneously
X_3X_2 = 10, activates K9 ~ K12 columns simultaneously
X_3X_2 = 11, activates K13 ~ K16 columns simultaneously
X_1, X_0 don't care.

$X_7X_5X_4$ = 111, in this setting, each scanning cycle checks two specified columns on key matrix. The specified columns are defined by the setting of X_3, X_2 and X_1.
$X_3X_2X_1$ = 000, activates K1 ~ K2 columns simultaneously
$X_3X_2X_1$ = 001, activates K3 ~ K4 columns simultaneously
……
$X_3X_2X_1$ = 110, activates K13 ~ K14 columns simultaneously
$X_3X_2X_1$ = 111, activates K15 ~ K16 columns simultaneously
X_0 is not a factor.

When KI1~4 is defined for the Key matrix scanning input by mask option, it is necessary to execute the SPC instruction to set the internal unused IOC port to output mode before the key matrix scanning function is activated. Fig 2-27 shows the organization of the Key matrix scanning input port. Each one of the SKI1~4 changed to "High" will set HRF5 to 1. If HEF5 has been set to 1 beforehand, this will cause SCF7 to be set, as well as releasing the HALT mode. After the key scanning cycle, the states of SKI1 ~ 4 will be latched and executing the IPC instruction could store these states into data RAM. Executing the PLC 20h instruction clears the HRF5 flag.
Since the key matrix scanning function shares the timing of LCD waveform, the scanning frequency corresponds to the LCD frame frequency and the LCD duty cycle. The formula for the key matrix scanning frequency is shown below :
key matrix scanning frequency (Hz) = (LCD frame frequency) x (LCD duty cycle) x 2
Note : "2" is a factor
For example, if the LCD frame frequency is 32Hz, and duty cycle is 1/5 duty, the scanning frequency for the key matrix is : 320Hz(32 x 5 x 2).



This figure shows the organization of Key matrix scanning input

Example:

```
        SPC     0fh     ; Disables all the pull-down devices on the internal IOC port.
                        ; Sets all of the IOC pins as the output mode.
        SPKX    10h     ;Generates HALT release request when a key is depressed
                        ; Scanns every column simultaneously in each cycle.
        PLC     20h     ; Clears HRF5
        SHE     20h     ;Sets HEF5.
        HALT            ;waits for the halt release caused by the key matrix.
        MCX     10h     ;Checks SCF8 (SKI).
        JB0     ski_release
        ……
        ……
ski_release:
        IPC     10h     ;reads the KI1~4 input latch state.
        JB0     ki1_release
        JB1     ki2_release
        JB2     ki3_release
        JB3     ki4_release
        .
        .
ki1_release:
        SPKX    40h     ; Checks the key depressed on K1 column.
        PLC     20h     ; Clears HRF5 to avoid the false HALT release
        CALL    wait_scan_again; Waits for the next key matrix scanning cycle.
                        ; The waiting period must be longer than the key matrix
                    scanning
                        ; cycle.
        IPC     10h     ; Reads the KI1 input latch state.
        JB0     ki1_seg1
        ……
        ……
        SPK     4fh     ; Enables only the SEG16 scanning output.
        PLC     20h     ; Clear HRF5 to avoid the false HALT released
        CALL    wait_scan_again     ; Waits for the time over the halt LCD clock cycle to
        ensure,  and scans again.
        IPC     10h     ; Reads the KI1 input latch state.
        JB0     kil_seg16
        ……
        ……
wait_scan_again:
        HALT
        PLC  20h
        RTS
```

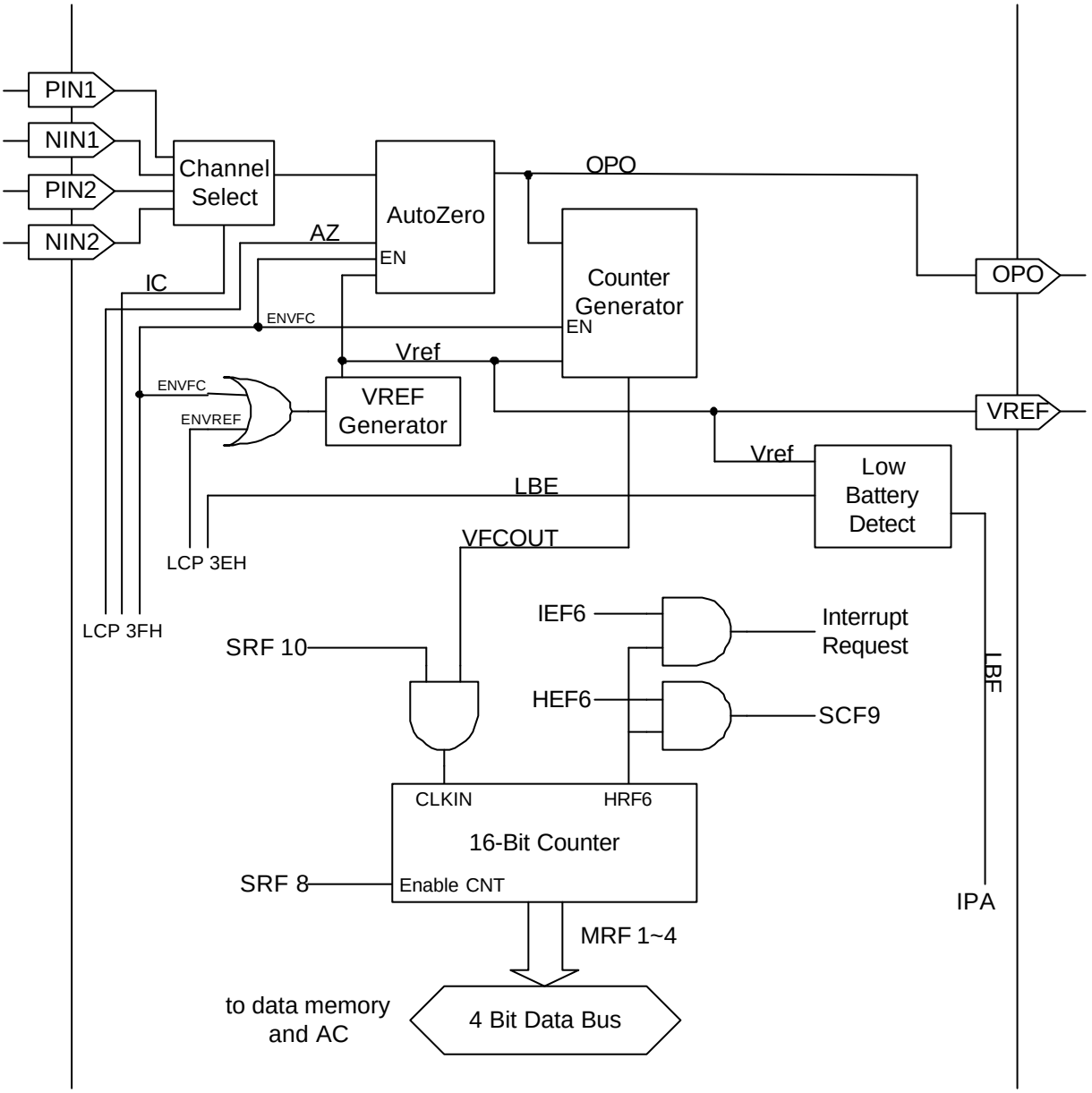
3-8 Voltage to Frequency Converter (VFC) & Low Battery Detect

Control Signal Table

Lz	DBUSD	DBUSC	DBUSB	DBUSA
3Eh	-	-	ENVREF	LBE
3Fh	-	ENVFC	IC	AZ

* Lz : LCD Address
ENVREF : Only Enable VREF for Low Battery Detect
LBE : Low Battery Detect Enable Signal
ENVFC : Enable VREF,AVDD, and VFC function
IC : VFC Channel Mode Control
IC=1 => PIN1,NIN1
IC=0 => PIN2,NIN2
AZ : Auto Zero Regulate flag

There is a Voltage to Frequency Converter, it contains a RC oscillation circuit and output VFCOUT to a 16 bits counter, which enable by SRF Instruction and read count data by MRF1~4 instruction. There are two kinds of methodology for measuring the input frequency; first, set VFCOUT as clock input and using Timer 2 as interval control, or use software direct control the time interval; second, if the VFCOUT frequency is too low, either a poor resolution for a fixed interval, or a long period for better resolution but a longer read-out rate, for example, 10 seconds per read-out, in such condition, you can switch the measuring mode to be setting the VFCOUT as interval control, it will enable the counter from the first VFCOUT risingedge till next rising edge, then generate a HALT release signal HRF6, and using FREQ (internal frequency generator output) as clock input, hence you can count the interval of VFCOUT.



This figure shows the VFC circuitry.

3-8-1 OPERATE Of VFC

Before the VFC function is enabled, the following seting is necessary,
-ENVFC flag has to be set to 1,
-Set the timer to generate a 125KHz clock for the usage of VFC,
-Selected input port, PIN1/NIN1 or PIN2/NIN2

The operating procedures of VFC are shown below and refer to the example program for detail.

Step 1 : Set ENVFC flag to 1

Step 2 : Enable the autozero(AZ) procedure of VFC function.
Set AZ flag to 1 and keep this flag at least 70mS and then set AZ flag to 0 to complete this procedure. After AZ flag retrun to 0, the output voltage on OPO pad will get to around 0.2V(virtual zero).

Step 3 : Delay 100mS to make sure OPO output is stable.

Step 4 : Retting timer in order to enable the measurement period.

Step 5 : Start the counting procedure.

Step 6 : Read out the counting data from register.

Note:

1.The value of OPO was kept on 100m second after AUTOZERO.

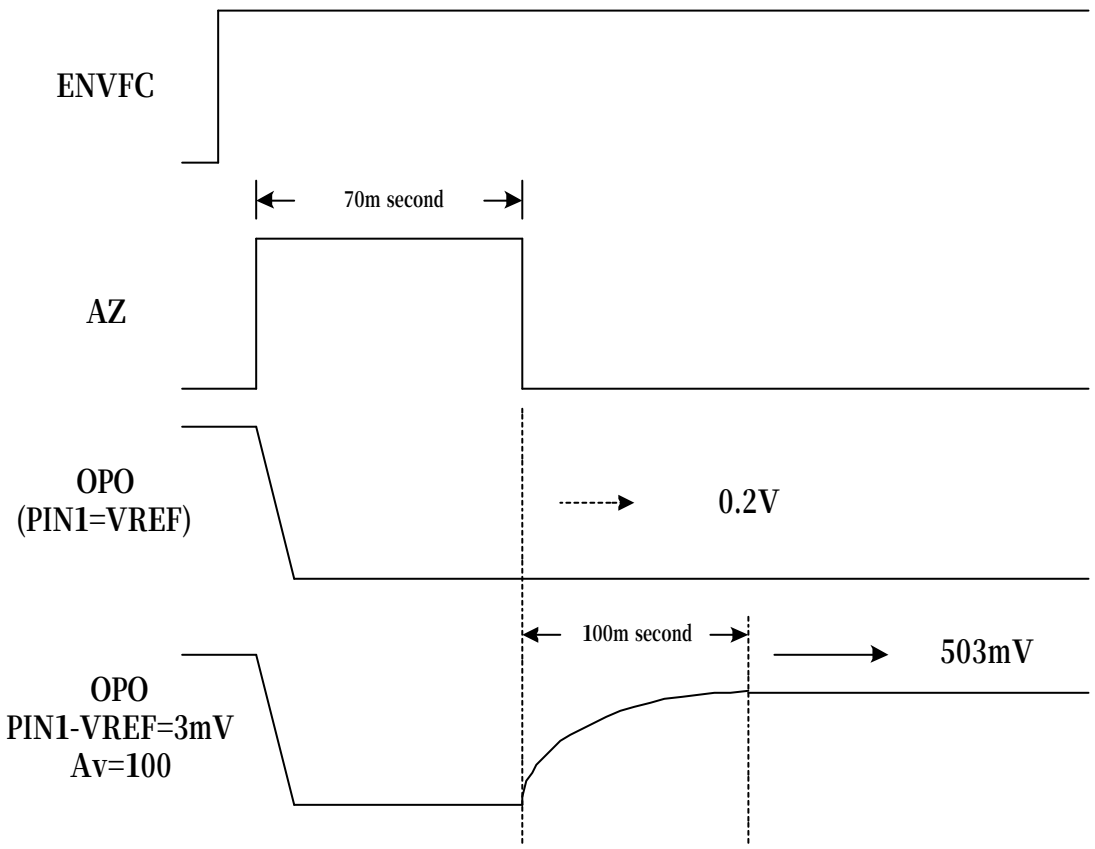
2.VFC block Vin (i.e. OPO) voltage range : 0.2V ~ (AVDD-0.2V)

3.ENVFC & ENVREF difference

a. ENVFC=1 or ENVREF=1 that VFCREF active.

b. When ENVFC=1, we got VREF & AVDD 2*VREF. The VFC function can work.

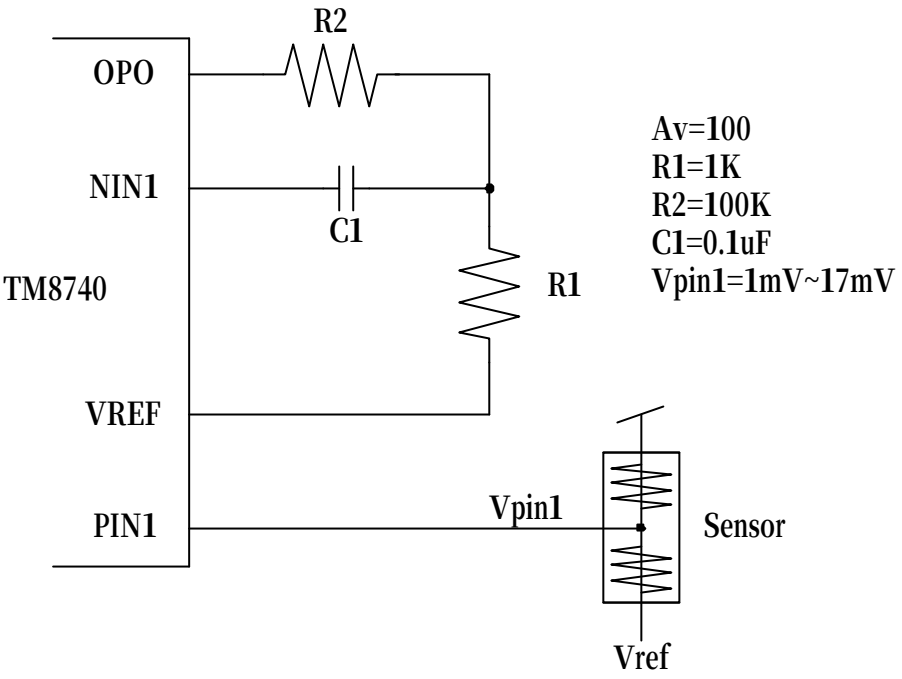
c. When ENVREF=1, we got VREF & AVDD VDDV. The Low-Battery Detect function can work.



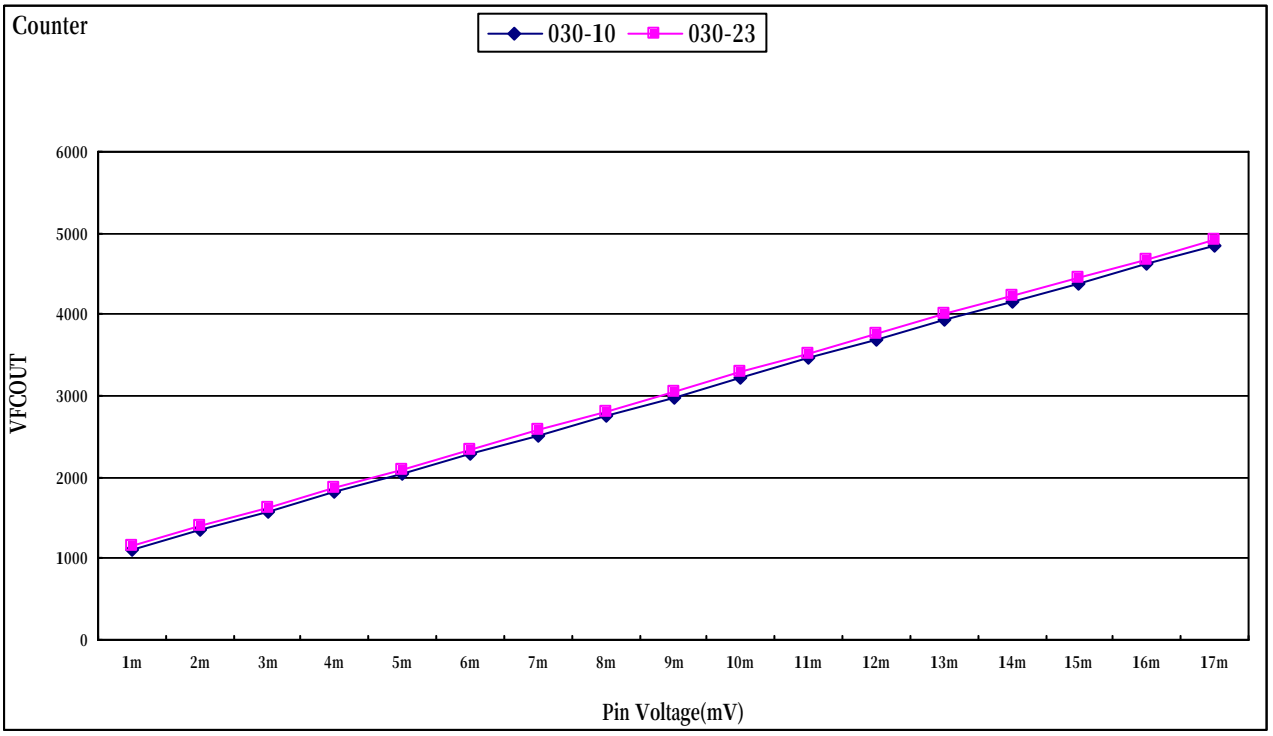
This figure shows the operating timing of VFC

Example:

```
        .
        .
        call  Vfc_function
        .
        .
; VFC Function
Vfc_function:
        plc  012h               ; Reset HRF4,2(Time2,Time1)
        she  12h                ; Enable HEF4,2(Time2,Time1)
        mrw  71h,06h            ; To choose channel_1, Enable ENVFC
        lcp  3fh,71h
        mrw  71h,07h            ; Enable Auto-Zero
        lcp  3fh,71h
        tmsx 152h               ; Delay 70ms to Auto-Zero high pulse(PH7,value=18,fosc=32.768KHz)
time1_AZ:
        msc  71h                ; Detect Time1 flag
        jb1  AZ_end
        jmp  time1_AZ
AZ_end:
        mrw  71h,06h            ; Disable Auto_Zero
        lcp  3fh,71h
        plc  002h               ; Reset HRF2(Time1)
        tmsx 158h               ; Delay 100ms
Time1_Delay:
        msc  71h
        jb1  Time1_Delay_End
        jmp  Time1_Delay
Time1_Delay_End:
        plc  002h
        srf  18h                ; Enable VFCOUT Control, TM2 Control
        she  10h                ; Enable HEF4(Time2)
        fast
        tm2x 159h               ; Delay 100ms to counter(PH7,value=25,fosc=32.768KHz)
time2_end:
        mcx  71h                ; Detect Timer2 flag
        jb1  time_end
        jmp  0,time2_end
time_end:
        mrf1 33h                ; move counter to register
        mrf2 32h
        mrf3 31h
        mrf4 30h
; Display VFC counter
        .
        .
vfc_end:
        mrw  71h,00h            ; Disable ENVFC
        lcp  3fh,71h
        plc  012h
        rts
        .
        .
```



Sample	PIN Input Voltage(mV)																Vref
	1m	2m	3m	4m	5m	6m	7m	8m	9m	10m	11m	12m	13m	14m	15m	16m	17m	
030-10	1107	1347	1583	1818	2052	2281	2513	2765	2983	3221	3458	3694	3926	4156	4389	4617	4836	1.151
030-23	1164	1391	1619	1874	2095	2347	2577	2812	3058	3290	3528	3770	4010	4240	4460	4677	4912	1.124



3-8-2 Low Battery Detect

When VREF is stable, we can operate low-battery detect function.

Step 1:Must let ENVREF=' 1' (ENVFC don' t care).

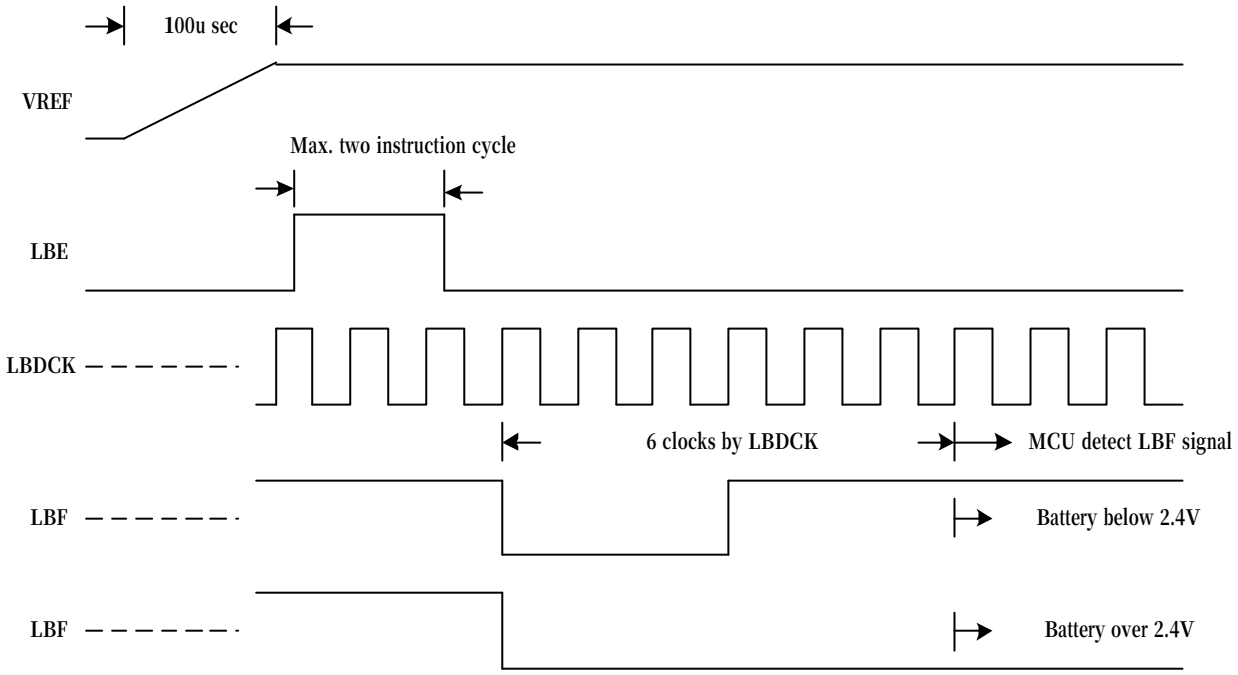
Step 2:Send 1 hi-pulse that pulse width about 1u second by LBE.

Step 3:After LBE about 48u second (about 6 LBDCK clock), that MCU detect LBF pin.

1. If LBF=' 0' , that to represent battery over 2.4V
2. If LBF=' 1' , that to represent battery below 2.4V

Note: When the LBE signal is enabled, the maximum time of high level should be kept 2-instruction cycle.

WAVE



This figure shows the operating timing of LBD

Example:

```
      …..
      lds  72h,02h
      lcp  3eh,72h              ; Enable ENVREF
      …..                       ; Delay 100us
      …..
      lds  73h,03h
      lcp  3eh,73h              ; Enable LBE
      lds  72h,02h
      lcp  3eh,72h              ; Disable LBE
      …..
      …..                       ; Wait 6 clock
      …..
      ipa  17h                  ; Check Low Battery
      jb1  Low_Battery
      …..                       ; No Low Battery
Low_Battery:
      …..                       ; Low Battery
      …..
```

CHAPTER 4 LCD DRIVER OUTPUT

There are 34 segment pins with 9 common pins in the LCD driver outputs in TM8740. All of these output pins can also be used as DC output ports (through the mask option). If more than one LCD driver output pin is defined as DC output, the following mask option must be selected.

MASK OPTION table :

When all of SEG and COM pins have been used to drive LCD panel

Mask Option name	Selected item
LCD ACTIVE TYPE	(1) LCD

When more than one of SEG or COM pins had been used for DC output port :

Mask Option name	Selected item
LCD ACTIVE TYPE	(2) O/P

During the initial reset cycle, the LCD lighting system may be lit or extinguished by mask option. All of the LCD output will remain in the initial setting until instructions relative to the LCD are executed to change the output data.

MASK OPTION table :

Mask Option name	Selected item
LCD DISPLAY IN RESET CYCLE	(1) ON
LCD DISPLAY IN RESET CYCLE	(2) OFF

4-1. LCD LIGHTING SYSTEM IN TM8740

There are several LCD lighting systems that can be selected by mask option in TM8740, they are :

- 1/2 bias 1/2 duty, 1/2 bias 1/3 duty, 1/2 bias 1/4 duty, 1/2bias 1/5duty, 1/2bias 1/6duty, 1/2bias 1/7duty, 1/2bias 1/8duty, 1/2bias 1/9duty,
- 1/3 bias 1/3 duty, 1/3 bias 1/4 duty, 1/3 bias 1/5duty, 1/3 bias 1/6duty, 1/3 bias 1/7duty, 1/3 bias 1/8duty, 1/3 bias 1/9duty,
- 1/4 bias 1/3 duty, 1/4 bias 1/4 duty, 1/4 bias 1/5duty, 1/4 bias 1/6duty, 1/4 bias 1/7duty, 1/4 bias 1/8duty, 1/4 bias 1/9duty,

All of these lighting systems are combined with 2 kinds of mask options; one is "LCD DUTY CYCLE" and the other is "BIAS".

MASK OPTION table :

LCD duty cycle option

Mask Option Name	Selected Item
LCD DUTY CYCLE	(1) O/P
LCD DUTY CYCLE	(2) DUPLEX (note : 1/2 duty)
LCD DUTY CYCLE	(3) 1/3 DUTY
LCD DUTY CYCLE	(4) 1/4 DUTY
LCD DUTY CYCLE	(5) 1/5 DUTY
LCD DUTY CYCLE	(6) 1/6 DUTY
LCD DUTY CYCLE	(7) 1/7 DUTY
LCD DUTY CYCLE	(8) 1/8 DUTY
LCD DUTY CYCLE	(9) 1/9 DUTY

LCD bias option

Mask Option name	Selected item
BIAS	(1) NO BIAS
BIAS	(2) 1/2 BIAS
BIAS	(3) 1/3 BIAS
BIAS	(4) 1/4 BIAS

The frame frequency for each lighting system is shown below; these frequencies can be selected by mask option. (All of the LCD frame frequencies in the following tables are based on the clock source frequency of the pre-divider (PH0) is 32768Hz).

The LCD alternating frequency in **duplex (1/2 duty)** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	16Hz
LCD frame frequency	(2) TYPICAL	32Hz
LCD frame frequency	(2) FAST	64Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/3 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	21Hz
LCD frame frequency	(2) TYPICAL	42Hz
LCD frame frequency	(2) FAST	85Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/4 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	16Hz
LCD frame frequency	(2) TYPICAL	32Hz
LCD frame frequency	(2) FAST	64Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/5 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	25Hz
LCD frame frequency	(2) TYPICAL	51Hz
LCD frame frequency	(2) FAST	102Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/6 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	21Hz
LCD frame frequency	(2) TYPICAL	42Hz
LCD frame frequency	(2) FAST	85Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/7 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	18Hz
LCD frame frequency	(2) TYPICAL	36Hz
LCD frame frequency	(2) FAST	73Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/8 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	32Hz
LCD frame frequency	(2) TYPICAL	64Hz
LCD frame frequency	(2) FAST	128Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The LCD alternating frequency in **1/9 duty** type

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	28Hz
LCD frame frequency	(2) TYPICAL	56Hz
LCD frame frequency	(2) FAST	113Hz
LCD frame frequency	(2) O/P	0Hz (LCD not used)

The following table shows the relationship between the LCD lighting system and the maximum number of driving LCD segments.

LCD Lighting System	Maximum Number of Driving LCD Segments	Remarks
Duplex(1/2 bias,1/2 duty)	82	Connect VDD3 and VDD4 to VDD2
1/2bias 1/3duty	123	Connect VDD3 and VDD4 to VDD2
1/2bias 1/4duty	164	Connect VDD3 and VDD4 to VDD2
1/2bias 1/5duty	205	Connect VDD3 and VDD4 to VDD2
1/2bias 1/6duty	246	Connect VDD3 and VDD4 to VDD2
1/2bias 1/7duty	287	Connect VDD3 and VDD4 to VDD2
1/2bias 1/8duty	328	Connect VDD3 and VDD4 to VDD2
1/2bias 1/9duty	369	Connect VDD3 and VDD4 to VDD2
1/3 bias 1/3 duty	123	Connect VDD4 to VDD3
1/3 bias 1/4 duty	164	Connect VDD4 to VDD3
1/3 bias 1/5 duty	205	Connect VDD4 to VDD3
1/3 bias 1/6 duty	246	Connect VDD4 to VDD3
1/3 bias 1/7 duty	287	Connect VDD4 to VDD3
1/3 bias 1/8 duty	328	Connect VDD4 to VDD3
1/3 bias 1/9 duty	369	Connect VDD4 to VDD3
1/4 bias 1/3 duty	123	
1/4 bias 1/4 duty	164	
1/4 bias 1/5 duty	205	
1/4 bias 1/6 duty	246	
1/4 bias 1/7 duty	287	
1/4 bias 1/8 duty	328	
1/4 bias 1/9 duty	369	

When choosing the LCD frame frequency, it is recommended to chose a frequency higher than 24Hz. If the frame frequency is lower than 24Hz, the pattern on the LCD panel will start to flash.

4-2. DC OUTPUT

TM8740 permits LCD driver output pins (COM5 ~ COM9 and SEG1 ~ SEG18, SEG26 ~ SEG41) to be defined as CMOS type DC output or P open-drain DC output ports by mask option. In these cases, it is possible to use some LCD driver output pins as DC output and the rest of the LCD driver output pins as LCD drivers. Refer to 4-3-4.
The configurations of CMOS output type and P open-drain type are shown below.
When the LCD driver output pins (SEG) are defined as DC output ports, the output data on those ports will not be affected when the program enters stop mode or LCD turn-off mode.

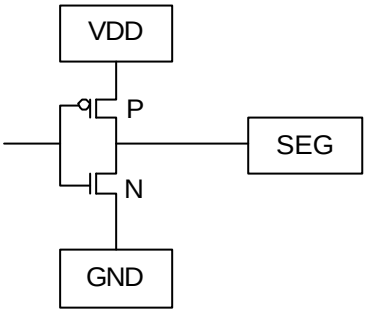


Figure 5-1 CMOS Output Type

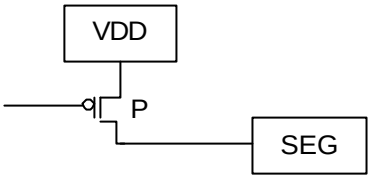


Figure 5-2 P Open-Drain Output Type

Only unused COM and SEG pads can be defined as DC output pins. The COM pad sequence for LCD drivers cannot be interrupted when the COM pads are defined as DC output ports.

For example, when the LCD lighting system is specified as 1/5 duty, the COM pad used for LCD driver must be COM1 ~ COM5. Each of COM6 ~ COM9 pad can be defined as DC output ports.

4-3. SEGMENT PLA CIRCUIT FOR LCD DISPLAY

4-3-1. PRINCIPLE OF OPERATION OF LCD DRIVER SECTION

The explanation below explains how the LCD driver section operates when the instructions are executed.

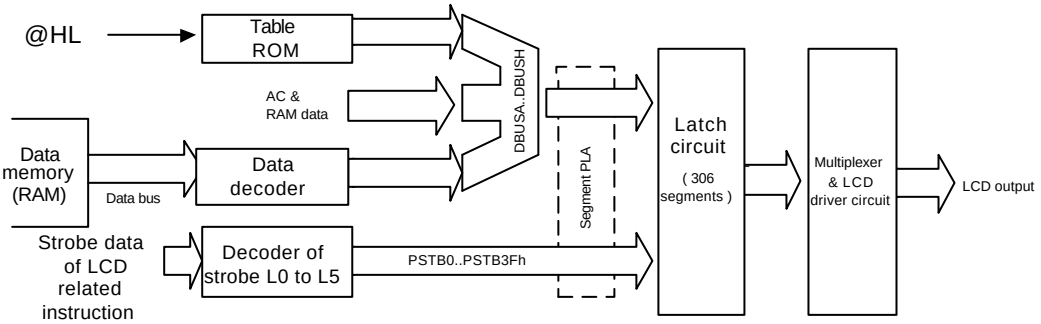


Figure 5-3 Principal Drawing of LCD Driver Section

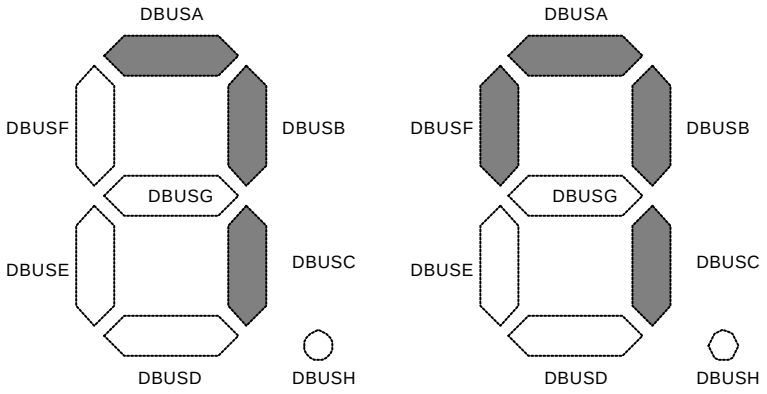
The LCD driver section consists of the following units:

- Data decoder to decode data supplied from RAM or table ROM
- Latch circuit to store LCD lighting information
- L0 to L5 decoder to decode the Lz-specified data in the LCD-related instructions which specifies the strobe of the latch circuit
- Multiplexer to select 1/2duty, 1/3duty, 1/4duty, 1/5duty, 1/6duty, 1/7duty, 1/8duty and 1/9duty
- LCD driver circuitry
- Segment PLA circuit connected between data decoder, L0 to L5 decoder and latch circuit.

The data decoder is used for decoding the contents of the working registers as specified in LCD-related instructions. They are decoded as 7-segment patterns on the LCD panel. The decoding table is shown below:

Content of data memory	Output of data decoder							
	DBUSA	DBUSB	DBUSC	DBUSD	DBUSE	DBUSF	DBUSG	DBUSH
0	1	1	1	1	1	1	0	1
1	0	1	1	0	0	0	0	1
2	1	1	0	1	1	0	1	1
3	1	1	1	1	0	0	1	1
4	0	1	1	0	0	1	1	1
5	1	0	1	1	0	1	1	1
6	1	0	1	1	1	1	1	1
7	1	1	1	0	0	*note	0	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1	1
A-F	0	0	0	0	0	0	0	0

* Note : The DBUSF of decoded output can be selected as 0 or 1 by mask option. The LCD pattern of this option is shown below :



DBUSF=0 DBUSF=1

The following table shows the options table for displaying the digit "7" pattern:

MASK OPTION table :

Mask Option name	Selected item
F SEGMENT FOR DISPLAY " 7 "	(1) ON
F SEGMENT FOR DISPLAY " 7 "	(2) OFF

Both LCT and LCB instructions use the data-decoder table to decode the content of the specified data memory location. When the content of the data memory location specified by the LCB instruction is "0", the decoded outputs of DBUSA ~ DBUSH are all "0". (this is used for blanking the leading digit "0" on the LCD panel).
The LCP instruction transfers data about the RAM (Rx) and accumulator (AC) directly from "DBUSA" to "DBUSH" without passing through the data decoder.
The LCD instruction transfers the table ROM data (T@HL) directly from "DBUSA" to "DBUSH" without passing through the data decoder.

Table 2- 2 The mapping table of LCP and LCD instructions

	DBUSA	DBUSB	DBUSC	DBUSD	DBUSE	DBUSF	DBUSG	DBUSH
LCP	Rx0	Rx1	Rx2	Rx3	AC0	AC1	AC2	AC3
LCD	T@HL0	T@HL1	T@HL2	T@HL3	T@HL4	T@HL5	T@HL6	T@HL7

There are 8 data decoder outputs from "DBUSA" to "DBUSH" and 64 L0 to L5 decoder outputs from PSTB 0h to PSTB 3Fh. The input data and clock signal of the latch circuit are "DBUSA" to "DBUSH" and PSTB 0h to PSTB 1Fh, respectively. Each segment pin has 9 latches corresponding to COM1-9.
The segment PLA performs the function of combining "DBUSA" outputs to "DBUSH" inputs and then sending them to each latch and strobe; PSTB 0h to PSTB3Fh is selected freely by mask option.
Of the 512 signals obtainable by combining "DBUSA" to "DBUSH" and PSTB 0h to PSTB 3Fh, any one of 369 (corresponding to the number of latch circuits incorporated in the hardware) signals can be selected by programming the aforementioned segment PLA. Table 2-7 shows the PSTB 0h to PSTB 3Fh signals.

Table 2- 3 Strobe Signal for LCD Latch in Segment PLA and Strobe in LCT Instruction

strobe signal for LCD latch	Strobe in LCT, LCB, LCP, LCD instructions The values of Lz in"LCT Lz, Q": *
PSTB0	0H
PSTB1	1H
PSTB2	2H
PSTB3	3H
PSTB4	4H
PSTB5	5H
……	……
PSTB3Ah	3AH
PSTB3Bh	3BH
PSTB3Ch	3CH
PSTB3Dh	3DH
PSTB3Eh	3EH
PSTB3Fh	3FH

Note: 1.The values of Q are the addresses of the working register in the data memory (RAM). In the LCD instruction, Q is the index address in the table ROM.

2. The Address of 3EH and 3FH is stationary, This are controlled address by VFC and LBD.

The LCD outputs can be turned off without changing segment data. The execution of the SF2 4h instruction may turn off the displays simultaneously. The execution of the RF2 4h instruction may turn on the display with the patterns turned off. These two instructions will not affect the data stored in the latch circuitry. When executing the RF2 4h instruction to turn off the LCD, the program can still execute LCT, LCB, LCP and LCD instructions to update the data in the latch circuitry. The new content will be outputted to the LCD while the display is being turned on again.
In the stop state, all COM and SEG outputs of LCD drivers will automatically switch to the GND state to avoid DC voltage bias on the LCD panel.

4-3-2. Relative Instructions

1. LCT Lz, Ry

Decodes the content specified in Ry with the data decoder and transfers the DBUSA ~ H to the LCD latch specified by Lz.

2. LCB Lz, Ry

Decodes the content specified in Ry with the data decoder and transfers the DBUSA ~ H to the LCD latch specified by Lz. "DBUSA" to "DBUSH" are all set to 0 when the input data of the data decoder is 0.

3. LCD Lz, @HL

Transfers the table ROM data specified by @HL directly to "DBUSA" through "DBUSH" without passing through the data decoder. The mapping table is shown in table 2-32.

4. LCP Lz, Ry

The data in the RAM and accumulator (AC) are transferred directly to "DBUSA" through "DBUSH" without passing through the data decoder. The mapping table is shown below:

5. LCT Lz, @HL

Decodes the index RAM data specified in @HL with the data decoder and transfers DBUSA ~ H to the LCD latch specified by Lz.

6. LCB Lz, @HL

Decodes the index RAM data specified in @HL with the data decoder and transfers the DBUSA ~ H to the LCD latch specified by Lz. The "DBUSA" to "DBUSH" are all set to 0 when the input data of the data decoder is 0.

7. LCP Lz, @HL

The data of the index RAM and accumulator (AC) are transferred directly to "DBUSA" through "DBUSH" without passing through the data decoder. The mapping table is shown below:

Table 2- 4 The mapping table of LCP and LCD instructions

	DBUSA	DBUSB	DBUSC	DBUSD	DBUSE	DBUSF	DBUSG	DBUSH
LCP	Rx0	Rx1	Rx2	Rx3	AC0	AC1	AC2	AC3
LCD	T@HL0	T@HL1	T@HL2	T@HL3	T@HL4	T@HL5	T@HL6	T@HL7

5. SF2 4h

Turns off the LCD display.

6. RF2 4h

Turns on the LCD display.

4-3-3. CONCRETE EXPLANATION

Each LCD driver output corresponds to the LCD 1/9 duty panel and has 9 latches (refer to Figure : Sample Organization of Segment PLA Option). Since the latch input and the signal to be applied to the clock (strobe) are selected with the segment PLA, the combination of segments in the LCD driver outputs is flexible. In other words, one of the data decoder outputs from "DBUSA" to "DBUSH" is applied to the latch input L, and one of the PSTB0 to PSTB3Fh outputs is applied to clock CLK.

TM8740 provide a flash type instruction to update the LCD pattern. When the LCTX D, LCBX D, LCPX D and LCDX D instructions are executed, the pattern of DBUS will be outputted to the16 latches (Lz) specified by D simultaneously.

D	Specified range of latched
00	Lz = 00h ~ 0Fh
01	Lz = 10h ~ 1Fh
10	Lz = 20h ~ 2Fh
11	Lz = 30h ~ 3Fh

Refer to Chapter 5 for detailed description of these instructions.

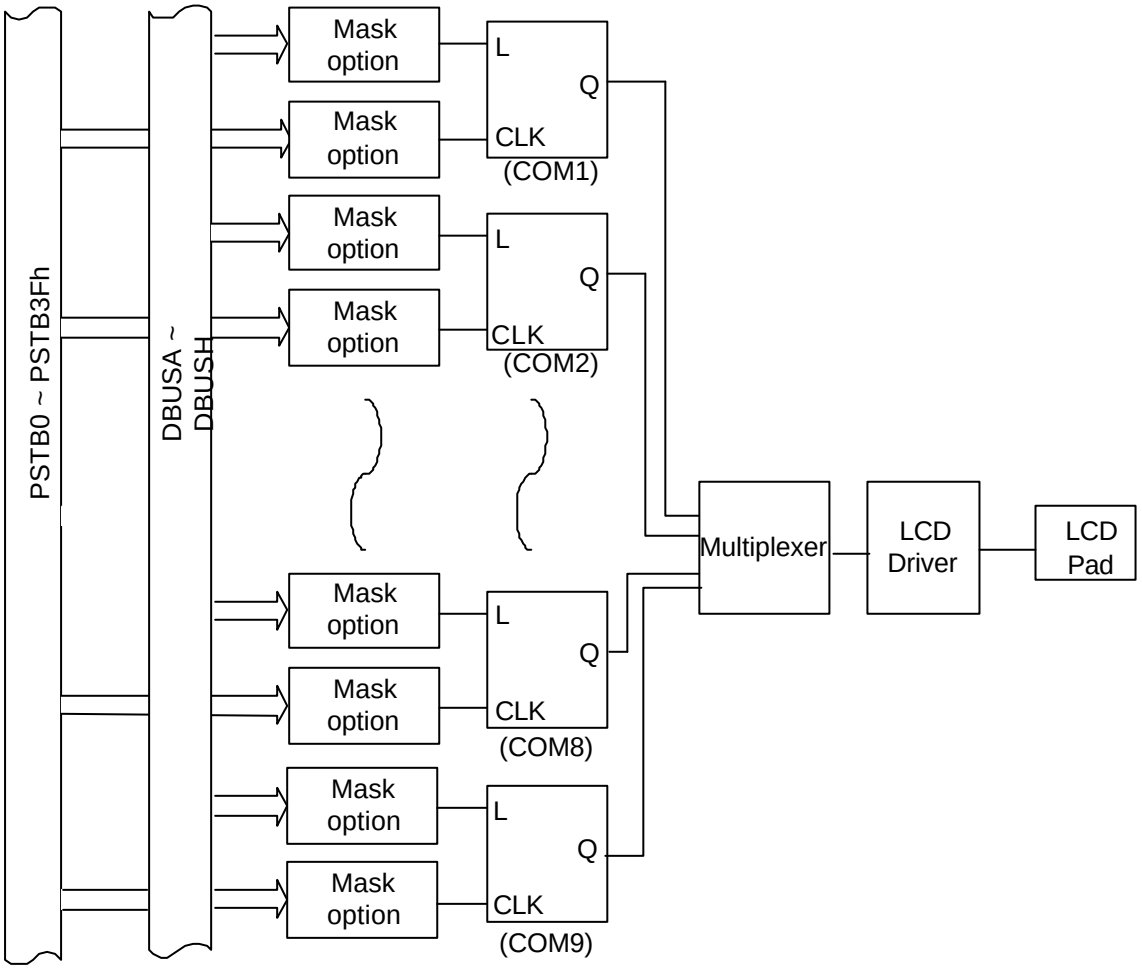


Figure : Sample Organization of Segment PLA Option

4-3-4. THE CONFIGURATION FILE FOR MASK OPTION

When configuring the mask option of LCD PLA, the *.cfg file provides the necessary format for editing the LCD configuration.

The syntax in the *.cfg file is as follows:

SEG COM PSTB DBUS

SEG : Specifies the segment pin No.
"1" ~ "41" represents segment pin No., "C1" ~ "C9" represents common pin No.
When the common pin (COM) is specified as the DC output pin, assign "C1" ~ "C9" in this column. "C1" ~ "C9" represents COM1 ~ COM9 respectively.

COM : Specifies the corresponding latch in each segment pin. Only 0, 1, 2, 3, 4 ~ 10 can be specified in this column. "1" ~ "9" represents COM1 latch ~ COM9 latch respectively.
"0" is for CMOS type DC output option and "10" is for P open-drain DC output option.

PSTB : Specifies the strobe data for the latch.

DBUS : Specifies the DBUS data for the latch.

Chapter 5 Detailed Explanation of the TM8740 Instructions

- It is necessary to initialize the content of the data memory after the initial reset, because the initial content of the data memory is unknown.
- The working registers are part of the data memory (RAM), and the relationship between them is shown as follows:

[The absolute address of working register Rx=Ry+70H]*

Address of working registers specified by Ry	Absolute address of data memory (Rx)
0H	70H
1H	71H
2H	72H
3H	73H
4H	74H
5H	75H
6H	76H
7H	77H

- Lz represents the address of the latch of LCD PLA (PSTB data in *.cfy file); the address range specified by Lz is from 00H to 3DH.

5-1 INPUT / OUTPUT INSTRUCTIONS

LCT Lz, Ry

Function : LCD latch [Lz] ← data decoder ← [Ry]

Description : The working register contents, specified by Ry, are loaded to the LCD latch, specified by Lz, through the data decoder.
Lz : 00 ~ 3DH, Ry : 0 ~ 7H.

LCB Lz, Ry

Function : LCD latch [Lz] ← data decoder ← [Ry]

Description : The working register contents, specified by Ry, are loaded to the LCD latch, specified by Lz, through the data decoder.
If the content of Ry is "0", the output of the data decoder will consist entirely of "0"s.
Lz : 00 ~ 3DH, Ry : 0 ~ 7H.

LCP Lz, Ry

Function : LCD latch [Lz] ← [Ry],AC

Description : The working register contents, specified by Ry, and the contents of AC are loaded to the LCD latch, specified by Lz.
Lz : 00 ~ 3DH, Ry : 0 ~ 7H.

	DBUSA	DBUSB	DBUSC	DBUSD	DBUSE	DBUSF	DBUSG	DBUSH
LCP	Rx0	Rx1	Rx2	Rx3	AC0	AC1	AC2	AC3
LCD	T@HL0	T@HL1	T@HL2	T@HL3	T@HL4	T@HL5	T@HL6	T@HL7

Table 4-2 The mapping table of LCD latches with the contents of AC and Ry.

LCD Lz, @HL

Function : LCD latch [Lz] ← TAB[@HL]

Description : @HL indicates an index address of table ROM.
The contents of table ROM, specified by @HL, are loaded to the LCD latch, specified by Lz, directly. Refer to Table 4-2.
Lz : 00 ~ 3DH.

LCT Lz, @HL

Function : LCD latch [Lz] ← data decoder ← [@HL]

Description : The contents of index RAM, specified by @HL, are loaded to the LCD latch, specified by Lz, through the data decoder. Refer to Table 4-2.
Lz : 00 ~ 3DH.

LCB Lz, @HL

Function : LCD latch [Lz] ← data decoder ← [@HL]

Description : The contents of index RAM, specified by @HL, are loaded to the LCD latch, specified by Lz, through the data decoder. Refer to Table 4-2.
If the content of @HL is "0", the output of the data decoder will consist entirely of "0"s.
Lz : 00 ~ 3DH.

LCP Lz, @HL

Function : LCD latch [Lz] ← [@HL],AC

Description : The contents of index RAM, specified by @HL, and the contents of AC are loaded to the LCD latch, specified by Lz. Refer to Table 4-2.
Lz : 00 ~ 3DH.

LCDX D

Function : Mullti-LCD latches [Lz(s)] ← TAB[@HL]

Description : @HL indicates an index address of table ROM.
The content of table ROM, specified by @HL, are loaded to several LCD latches(Lz) simultaneously. Refer to Table 4-2. The range of multi-Lz is specified by data "D".
D : 0 ~ 3.

D=0	Multi-Lz=00H~0FH
D=1	Multi-Lz=10H~1FH
D=2	Multi-Lz=20H~2FH
D=3	Multi-Lz=30H~3FH

Table 4-3 The range of multi-Lz latches

LCTX D

Function : Mullti-LCD latch [Lz] ← data decoder ← [@HL]

Description : The contents of index RAM, specified by @HL, are loaded to several LCD latches(Lz) simultaneously. The range of multi-Lz is specified by data "D". Refer to Table 4-3.
D : 0 ~ 3.

LCBX D

Function : Mullti- LCD latch [Lz] ← data decoder ← [@HL]

Description : The contents of index RAM, specified by @HL, are loaded to the LCD latch specified by Lz through the data decoder. The range of multi-Lz is specified by data "D". Refer to Table 4-3.
D : 0 ~ 3.

LCPX D

Function : Mullti- LCD latch [Lz] ← [@HL],AC

Description : The contents of index RAM, specified by @HL, and the contents of AC are loaded to several LCD latches(Lz) simultaneously. Refer to Table 4-2. The range of multi-Lz is specified by data "D". Refer to Table 4-3.
D : 0 ~ 3.

SPA X

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

Description : Sets the I/O mode and turns the pull-low device on or off. The meaning of each bit of X(X4, X3, X2) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable the pull-low device on IOA1~IOA4 simultaneously	X4=0	Disable the pull-low device on IOA1~IOA4 simultaneously
X3=1	IOA4 as output mode	X3=0	IOA4 as input mode
X2=1	IOA3 as output mode	X2=0	IOA3 as input mode

OPA Rx

Function : I/OA ← [Rx]

Description : The content of Rx is outputted to I/OA port.

OPAS D

Function : IOA3 ← D, IOA4 ← pulse

Description : D is outputted to IOA3, and pulse is outputted to IOA4.
D = 0 or 1

IPA Rx

Function : [Rx], AC ← [I/OA]

Description : The data of I/OA port is loaded to AC and the data memory Rx.

SPB X

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

Description : Sets the I/O mode and turns the pull-low device on or off. The meaning of each bit of X(X4, X3, X2, X1, X0) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable the pull-low device on IOB1~IOB4 simultaneously	X4=0	Disable the pull-low device on IOB1~IOB4 simultaneously
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
X0=1	IOB1 as output mode	X0=0	IOB1 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 ← [I/OB]

Description : The data of I/OB port is loaded to AC and data memory Rx.

SPC X

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

Description : The meaning of each bit of X(X4, 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	IOC4 as output mode	X3=0	IOC4 as input mode
X2=1	IOC3 as output mode	X2=0	IOC3 as input mode
X1=1	IOC2 as output mode	X1=0	IOC2 as input mode
X0=1	IOC1 as output mode	X0=0	IOC1 as input mode

OPC Rx

Function : I/OC ← [Rx]

Description : The content of Rx is outputted to I/OC port.

IPC Rx

Function : [Rx],AC ← [I/OC]

Description : The data of the I/OC port is loaded to AC and data memory Rx.

SPD X

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

Description : Sets the I/O mode and turns the pull-low device on or off. The meaning of each bit of X(X4, X3, X2, X1, X0) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable the pull-low device on IOD1~IOD4 simultaneously	X4=0	Disable the pull-low device on IOD1~IOD4 simultaneously
X3=1	IOD4 as output mode	X3=0	IOD4 as input mode
X2=1	IOD3 as output mode	X2=0	IOD3 as input mode
X1=1	IOD2 as output mode	X1=0	IOD2 as input mode
X0=1	IOD1 as output mode	X0=0	IOD1 as input mode

OPD Rx

Function : I/OD ← [Rx]

Description : The content of Rx is outputted to I/OD port.

IPD Rx

Function : [Rx], AC ← [I/OD]

Description : The data of the I/OD port is loaded to AC and data memory Rx.

SPKX X

Function : Sets the Key matrix scanning output state.

Description : When SEG1~16 is(are) used for LCD driver pin(s), set X(X7~0) to specify the key matrix scanning output state for each SEGn pin in the scanning interval.

X_6 = " 0 ", when HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after the key is depressed on the key matrix, and then SCF7 will be set to 1.

" 1 ", when HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after each scanning cycle regardless of key depression, and then SCF7 will be set to 1.

$X_7X_5X_4$ = 000, in this setting, each scanning cycle only checks one specified column (K1 ~ K16) on the key matrix. The specified column is defined by the setting of $X_3 \sim X_0$.

$X_3 \sim X_0$ = 0000, activates the K1 column

$X_3 \sim X_0$ = 0001, activates the K2 column

……..

$X_3 \sim X_0$ = 1110, activates the K15 column

$X_3 \sim X_0$ = 1111, activates the K16 column

$X_7X_5X_4$ = 001, in this setting, all of the matrix columns (K1 ~ K16) will be checked simultaneously in each scanning cycle. X_3 ~ X_0 are not a factor.

$X_7X_5X_4$ = 010, in this setting, the key matrix scanning function will be disabled. X_3 ~ X_0 are not a factor.

$X_7X_5X_4$ = 10X, in this setting, each scanning cycle checks 8 specified columns on the key matrix. The specified column is defined by the setting of X_3.

X_3 = 0, activates the K1 ~ K8 columns simultaneously

X_3 = 1, activates the K9 ~ K16 columns simultaneously

(X_2 ~ X_0 are not a factor)

$X_7X_5X_4$ = 110, in this setting, each scanning cycle checks four specified columns on the key matrix. The specified columns are defined by the setting of X_3 and X_2.

X_3X_2 = 00, activates the K1 ~ K4 columns simultaneously

X_3X_2 = 01, activates the K5 ~ K8 columns simultaneously

X_3X_2 = 10, activates the K9 ~ K12 columns simultaneously

X_3X_2 = 11, activates the K13 ~ K16 columns simultaneously

(X_1, X_0 are not a factor)

$X_7X_5X_4$ = 111, in this setting, each scanning cycle checks two specified columns on the key matrix. The specified columns are defined by the setting of X_3, X_2 and X_1.

$X_3X_2X_1$ = 000, activates the K1 ~ K2 columns simultaneously

$X_3X_2X_1$ = 001, activates the K3 ~ K4 columns simultaneously

……

$X_3X_2X_1$ = 110, activates the K13 ~ K14 columns simultaneously

$X_3X_2X_1$ = 111, activates the K15 ~ K16 columns simultaneously

(X_0 is not a factor)

SPK Rx

Function : Sets the Key matrix scanning output state.

Description : When SEG1~16 is(are) used for LCD driver pin(s), sets the contents of AC and Rx to specify the key matrix scanning output state for each SEGn pin in the scanning interval.

The bit setting is the same as the SPKX instruction. The bit patterns of AC and Rx corresponding to SPKX are shown below:

Instruction	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPK Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
SPKX X	X7	X6	X5	X4	X3	X2	X1	X0

SPK @HL

Function : Sets the Key matrix scanning output state.

Description : When SEG1~16 is(are) used for LCD driver pin(s), sets the content of table ROM([@HL]) to specify the key matrix scanning output state for each SEGn pin in the scanning interval.
The bit setting is the same as the SPKX instruction. The bit pattern of the table ROM corresponding to SPKX is shown below:

Instruction	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPK @HL	(T@HL)7	(T@HL)6	(T@HL)5	(T@HL)4	(T@HL)3	(T@HL)2	(T@HL)1	(T@HL)0
SPKX X	X7	X6	X5	X4	X3	X2	X1	X0

ALM X

Function : Sets the buzzer output frequency.

Description : The waveform specified by X(X8 ~ X0) is delivered to the BZ and BZB pins.The 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	1	1	FREQ*
1	0	0	DC1
0	1	1	PH3(4KHz)
0	1	0	PH4(2KHz)
0	0	1	PH5(1KHz)
0	0	0	DC0

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

Bit	clock source(lower frequency)
X5	PH15(1Hz)
X4	PH14(2Hz)
X3	PH13(4Hz)
X2	PH12(8Hz)
X1	PH11(16Hz)
X0	PH10(32Hz)

Notes: 1. FREQ is the output of the 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 PH0 is 32768Hz.

SRF X

Function : The operation control for RFC.

Description : The meaning of each control bit(X5 ~ X4) is shown below:

X3=1	enables the 16-bit counter	X3=0	disables the 16-bit counter
X4=1	Timer 2 controls the 16-bit counter. X3 must be set to 1 when this bit is set to 1.	X4=0	Disables timer 2 to control the 16-bit counter.
X5=1	The 16-bit counter is controlled by the VFC. X3 must be set to 1 when this bit is set to 1.	X5=0	Disables the VFC the 16-bit counter.

Note: X4 and X5 can not be set to 1 at the same time.

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.

MRW @HL, Rx

Function : AC, R[@HL] ← [Rx]

Description : The content of the data memory specified by Rx is loaded to AC and the data memory specified by @HL.

MRW# @HL, Rx

Function : AC, R[@HL] ← [Rx], @HL ← HL + 1

Description : The content of data memory specified by Rx is loaded to AC and the data memory specified by @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.

MWR Rx, Ry

Function : AC,[Rx] ← [Ry]

Description : The content of the working register specified by Ry is loaded to AC and the data memory specified by Rx.

MWR Rx, @HL

Function : AC, [Rx] ← R[@HL]

Description : The content of the data memory specified by @HL is loaded to AC and the data memory specified by Rx.

MWR# Rx, @HL

Function : AC, [Rx] ← R[@HL] , @HL ← HL + 1

Description : The content of the data memory specified by @HL is loaded to AC and the data memory specified by Rx.
The content of the index register (@HL) will be incremented automatically after executing this instruction.

SR0 Rx

Function : [Rx]n, ACn ← [Rx](n+1),AC(n+1)
[Rx]3, AC3 ← 0

Description : The Rx content is shifted right and 0 is loaded to the MSB.
The result is loaded to the AC.

Content of Rx	Bit3	Bit2	Bit1	Bit0
Before	Rx3	Rx2	Rx1	Rx0
After	0	Rx3	Rx2	Rx1

SR1 Rx

Function : [Rx]n, ACn ← [Rx](n+1),AC(n+1)
[Rx]3, AC3 ← 1

Description : The Rx content is shifted right and 1 is loaded to the MSB. The result is loaded to the AC.

Content of Rx	Bit3	Bit2	Bit1	Bit0
Before	Rx3	Rx2	Rx1	Rx0
After	1	Rx3	Rx2	Rx1

SL0 Rx

Function : [Rx]n, ACn ← [Rx](n-1),[ACn-1]
[Rx]0, AC0 ← 0

Description : The Rx content is shifted left and 0 is loaded to the LSB. The results are loaded to the AC.

Content of Rx	Bit3	Bit2	Bit1	Bit0
Before	Rx3	Rx2	Rx1	Rx0
After	Rx2	Rx1	Rx0	0

SL1 Rx

Function : [Rx]n, ACn ← [Rx](n-1),AC(n-1)
[Rx]0, AC0 ← 1

Description : The Rx content is shifted left and 1 is loaded to the LSB. The results are loaded to the AC.

Content of Rx	Bit3	Bit2	Bit1	Bit0
Before	Rx3	Rx2	Rx1	Rx0
After	Rx2	Rx1	Rx0	1

MRA Rx

Function : CF ← [Rx]3

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

MAF Rx

Function : AC,[Rx] ← CF, Zero flag

Description : The content of CF is loaded to AC and Rx. The content of AC and the meaning of the bit after execution of this instruction are as follows:

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

5-3 OPERATION INSTRUCTIONS

INC* Rx

Function : [Rx],AC ← [Rx]+1

Description : Adds 1 to the content of Rx; the result is loaded to the data memory Rx and AC.
* The carry flag (CF) will be affected.

INC* @HL

Function : [@HL],AC ← R[@HL]+1

Description : Adds 1 to the content of @HL; the result is loaded to the data memory @HL and AC.
* the carry flag (CF) will be affected.
• @HL indicates an index address of data memory.

INC*# @HL

Function : [@HL],AC ← R[@HL]+1, @HL ← HL + 1

Description : Adds 1 to the content of @HL; the result is loaded to the data memory @HL and AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The carry flag (CF) will be affected.
• @HL indicates an index address of data memory.

DEC* Rx

Function : [Rx], AC ← [Rx] -1

Description : Substrates 1 from the content of Rx; the result is loaded to the data memory Rx and AC.
• The carry flag (CF) will be affected.

DEC* @HL

Function : R@HL, AC ← R[@HL] -1

Description : Substrates 1 from the content of @HL; the result is loaded to the data memory @HL and AC.
* The carry flag (CF) will be affected.
• @HL indicates an index address of data memory.

DEC*# @HL

Function : R@HL, AC ← R[@HL] -1, @HL ← HL + 1

Description : Substrates 1 from the content of @HL; the result is loaded to the data memory @HL and AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The carry flag (CF) will be affected.
• @HL indicates an index address of data memory.

ADC Rx

Function : AC ← [Rx]+AC+CF

Description : Binary-adds the contents of Rx, AC and CF; the result is loaded to AC.
* The carry flag (CF) will be affected.

ADC @HL

Function : AC ← [@HL]+AC+CF

Description : Binary-adds the contents of @HL,AC and CF; the result is loaded to AC.
* The carry flag (CF) will be affected.
. @HL indicates an index address of data memory.

ADC# @HL

Function : AC ← [@HL]+AC+CF, @HL ← HL + 1

Description : Binary-adds the contents of @HL,AC and CF; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The carry flag (CF) will be affected.
. @HL indicates an index address of data memory.

ADC* Rx

Function : AC, [Rx] ← [Rx]+AC+CF

Description : Binary-adds the contents of Rx, AC and CF; the result is loaded to AC and the data memory Rx.
* The carry flag (CF) will be affected.

ADC* @HL

Function : AC,[@HL] ← [@HL]+AC+CF

Description : Binary-adds the contents of @HL,AC and CF; the result is loaded to AC and the data memory @HL.
* The carry flag (CF) will be affected.
. @HL indicates an index address of data memory.

ADC*# @HL

Function : AC, [@HL] ← [@HL]+AC+CF, @HL ← HL + 1

Description : Binary-adds the contents of @HL,AC and CF; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The carry flag (CF) will be affected.
. @HL indicates an index address of data memory.

SBC Rx

Function : AC ← [Rx]+ (AC)B+CF

Description : Binary-subtracts the contents of AC and CF are from the content of Rx; the result is loaded to AC.
* The carry flag (CF) will be affected.

SBC @HL

Function : AC ← [@HL]+ (AC)B+CF

Description : Binary-subtracts the contents of AC and CF from the content of @HL; the result is loaded to AC.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SBC# @HL

Function : AC ← [@HL]+ (AC)B+CF, @HL ← HL + 1

Description : Binary-subtracts the contents of AC and CF from the content of @HL; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SBC* Rx

Function : AC, [Rx] ← [Rx]+(AC)B+CF

Description : Binary-subtracts the contents of AC and CF from the content of Rx; the result is loaded to AC and the data memory Rx.
* The carry flag (CF) will be affected.

SBC* @HL

Function : AC,[@HL] ← [@HL]+ (AC)B+CF

Description : Binary-subtracts the contents of AC and CF from the content of @HL; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SBC*# @HL

Function : AC,[@HL] ← [@HL]+ (AC)B+CF, @HL ← HL + 1

Description : Binary-subtracts the contents of AC and CF from the content of @HL; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

ADD Rx

Function : AC ← [Rx]+AC

Description : Binary-adds the contents of Rx and AC; the result is loaded to AC.
* The carry flag (CF) will be affected.

ADD @HL

Function : AC ← [@HL]+AC

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

ADD# @HL

Function : AC ← [@HL]+AC, @HL ← HL + 1

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

ADD* Rx

Function : AC, [Rx] ← [Rx]+AC

Description : Binary-adds the contents of Rx and AC; the result is loaded to AC and the data memory Rx.
* The carry flag (CF) will be affected.

ADD* @HL

Function : AC,[@HL] ← [@HL]+AC

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

ADD*# @HL

Function : AC,[@HL] ← [@HL]+AC, @HL ← HL + 1

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SUB Rx

Function : AC ← [Rx]+ (AC)B+1

Description : Binary-subtracts the content of AC from the content of Rx; the result is loaded to AC.
* The carry flag (CF) will be affected.

SUB @HL

Function : AC ← [@HL]+ (AC)B+1

Description : Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SUB# @HL

Function : AC ← [@HL]+ (AC)B+1, @HL ← HL + 1

Description : Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SUB* Rx

Function : AC,[Rx] ← [Rx]+ (AC)B+1

Description : Binary-subtracts the content of AC from the content of Rx; the result is loaded to AC and Rx.
* The carry flag (CF) will be affected.

SUB* @HL

Function : AC, [@HL] ← [@HL]+ (AC)B+1

Description : Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

SUB*# @HL

Function : AC, [@HL] ← [@HL]+ (AC)B+1, @HL ← HL + 1

Description : Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The carry flag (CF) will be affected.

ADN Rx

Function : AC ← [Rx]+AC

Description : Binary-adds the contents of Rx and AC; the result is loaded to AC.
* The result will not affect the carry flag (CF).

ADN @HL

Function : AC ← [@HL]+AC

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC.
* The result will not affect the carry flag (CF).
. @HL indicates an index address of data memory.

AND# @HL

Function : AC ← [@HL]+AC, @HL ← HL + 1

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The result will not affect the carry flag (CF).
. @HL indicates an index address of data memory.

ADN* Rx

Function : AC, [Rx] ← [Rx]+AC

Description : Binary-adds the contents of Rx and AC; the result is loaded to AC and data memory Rx.
* The result will not affect the carry flag (CF).

ADN* @HL

Function : AC, [@HL] ← [@HL]+AC

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
* The result will not affect the carry flag (CF).
. @HL indicates an index address of data memory.

ADN*# @HL

Function : AC, [@HL] ← [@HL]+AC, @HL ← HL + 1

Description : Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The result will not affect the carry flag (CF).
. @HL indicates an index address of data memory.

AND Rx

Function : AC ← [Rx] & AC

Description : Binary-ANDs the contents of Rx and AC; the result is loaded to AC.

AND @HL

Function : AC ← [@HL] & AC

Description : Binary-ANDs the contents of @HL and AC; the result is loaded to AC.
. @HL indicates an index address of data memory.

AND# @HL

Function : AC ← [@HL] & AC, @HL ← HL + 1

Description : Binary-ANDs the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

AND* Rx

Function : AC, [Rx] ← [Rx] & AC

Description : Binary-ANDs the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

AND* @HL

Function : AC, [@HL] ← [@HL] & AC

Description : Binary-ANDs the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

AND*# @HL

Function : AC, [@HL] ← [@HL] & AC, @HL ← HL + 1

Description : Binary-ANDs the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

EOR Rx

Function : AC ← [Rx] ⊕ AC

Description : Exclusive-Ors the contents of Rx and AC; the result is loaded to AC.

EOR @HL

Function : AC ← [@HL] ⊕ AC

Description : Exclusive-Ors the contents of @HL and AC; the result is loaded to AC.
. @HL indicates an index address of data memory.

EOR# @HL

Function : AC ← [@HL] ⊕ AC, @HL ← HL + 1

Description : Exclusive-Ors the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

EOR* Rx

Function : AC, Rx ← [Rx] ⊕ AC

Description : Exclusive-Ors the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

EOR* @HL

Function : AC, [@HL] ← [@HL] ⊕ AC

Description : Exclusive-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

EOR*# @HL

Function : AC, [@HL] ← [@HL] ⊕ AC, @HL ← HL + 1

Description : Exclusive-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

OR Rx

Function : AC ← [Rx] | AC

Description : Binary-Ors the contents of Rx and AC; the result is loaded to AC.

OR @HL

Function : AC ← [@HL] | AC

Description : Binary-Ors the contents of @HL and AC; the result is loaded to AC.
. @HL indicates an index address of data memory.

OR# @HL

Function : AC ← [@HL] | AC, @HL ← HL + 1

Description : Binary-Ors the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

OR* Rx

Function : AC, Rx ← [Rx] | AC

Description : Binary-Ors the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

OR* @HL

Function : AC,[@HL] ← [@HL] | AC

Description : Binary-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

OR*# @HL

Function : AC,[@HL] ← [@HL] | AC, @HL ← HL + 1

Description : Binary-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

ADCI Ry, D

Function : AC ← [Ry]+D+CF

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

SBCI Ry, D

Function : AC ← [Ry]+#(D)+CF

Description : . D represents the immediate data.
Binary-subtracts the CF and immediate data D from the 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)+CF

Description : . D represents the immediate data.
Binary-subtracts the CF and immediate data D from the working register Ry; the result is loaded to AC and the working register Ry.
* The carry flag (CF) will be affected.
D = 0H ~ FH

ADDI Ry, D

Function : AC ← [Ry]+D

Description : . D represents the immediate data.
Binary-ADDs the contents of Ry and D; 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 the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC and the working register Ry.
* The carry flag (CF) will be affected.
D = 0H ~ FH

SUBI Ry, D

Function : AC ← [Ry]+#(D)+1

Description : . D represents the immediate data.
Binary-subtracts the immediate data D from the 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]+#(Y)+1

Description : . D represents the immediate data.
Binary-subtracts the immediate data D from the working register Ry; the result is loaded to AC and the working register Ry.
* The carry flag (CF) will be affected.
D = 0H ~ FH

ADNI Ry, D

Function : AC ← [Ry]+D

Description : . D represents the immediate data.
Binary-ADDs the contents of Ry and D; 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 the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC and the 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 the immediate data.
Binary-ANDs the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

ANDI* Ry, D

Function : AC,[Ry] ← [Ry] & D

Description : . D represents the immediate data.
Binary-ANDs the contents of Ry and D; the result is loaded to AC and the working register Ry.
D = 0H ~ FH

EORI Ry, D

Function : AC ← [Ry] EOR D

Description : . D represents the immediate data.
Exlusive-Ors the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

EORI* Ry, D

Function : AC,[Ry] ← [Ry] ⊕ D

Description : . D represents the immediate data.
Exclusive-Ors the contents of Ry and D; the result is loaded to AC and the working register Ry.
D = 0H ~ FH

ORI Ry, D

Function : AC ← [Ry] | D

Description : . D represents the immediate data.
Binary-Ors the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

ORI* Ry, D

Function : AC,[Ry] ← [Ry] | D

Description : . D represents the immediate data.
Binary-Ors the contents of Ry and D; the result is loaded to AC and the working register Ry.
D = 0H ~ FH

5-4 LOAD/STORE INSTRUCTIONS

STA Rx

Function : [Rx] ← AC

Description : The content of AC is loaded to the data memory specified by Rx.

STA @HL

Function : [@HL] ← AC

Description : The content of AC is loaded to the data memory specified by @HL.
. @HL indicates an index address of data memory.

STA# @HL

Function : [@HL] ← AC, @HL ← HL + 1

Description : The content of AC is loaded to the data memory specified by @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

LDS Rx, D

Function : AC,[Rx] ← D

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

LDA Rx

Function : AC ← [Rx]

Description : The content of Rx is loaded to AC.

LDA @HL

Function : AC ← [@HL]

Description : The content specified by @HL is loaded to AC.
. @HL indicates an index address of data memory.

LDA# @HL

Function : AC ← [@HL] , @HL ← HL + 1

Description : The content specified by @HL is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

LDH Rx, @HL

Function : [Rx] , AC ← TAB[@HL] high nibble*

Description : The higher nibble data of the look-up table, specified by @HL, is loaded to the data memory specified by Rx.

LDH* Rx, @HL

Function : [Rx] , AC ← TAB[@HL] high nibble, @HL=@HL+1

Description : The higher nibble data of the look-up table, specified by @HL, is loaded to the data memory specified by Rx, and then is increased in @HL.

LDL Rx, @HL

Function : [Rx] , AC ← TAB[@HL] low nibble

Description : The lower nibble data of the look-up table, specified by @HL, is loaded to the data memory specified by Rx.

LDL* Rx, @HL

Function : [Rx], AC ← TAB[@HL] low nibble, @HL=@HL+1

Description : The lower nibble data of the look-up table, specified by @HL, is loaded to the data memory specified by Rx, and then is increased in @HL.

MRF1 Rx

Function : [Rx] , AC ← VFC[3 ~ 0]

Description : Loads the lowest nibble data of the 16-bit counter of VFC to AC and the data memory specified by Rx.
Bit 3 ← VFC[3]
Bit 2 ← VFC[2]
Bit 1 ← VFC[1]
Bit 0 ← VFC[0]

MRF2 Rx

Function : [Rx] , AC ← VFC[7 ~ 4]

Description : Loads the 2nd nibble data of the 16-bit counter of VFC to AC and the data memory specified by Rx.
Bit 3 ← VFC[7]
Bit 2 ← VFC[6]
Bit 1 ← VFC[5]
Bit 0 ← VFC[4]

MRF3 Rx

Function : [Rx] , AC ← VFC[11 ~ 8]

Description : Loads the 3rd nibble data of the 16-bit counter of VFC to AC and the data memory specified by Rx.
Bit 3 ← VFC[11]
Bit 2 ← VFC[10]
Bit 1 ← VFC[9]
Bit 0 ← VFC[8]

MRF4 Rx

Function : [Rx] , AC ← VFC[15 ~ 12]

Description : Loads the highest nibble data of the 16-bit counter of VFC to AC and the data memory specified by Rx.
Bit 3 ← VFC[15]
Bit 2 ← VFC[14]
Bit 1 ← VFC[13]
Bit 0 ← VFC[12]

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 the ports IOC(SCF1) or IOD(SCF3).
3) The halt release condition specified by the SHE instruction is met (HRF1 ~ HRF6).
When an interrupt is accepted to release the halt mode, the halt mode returns by executing the RTS instruction after the completion of the interrupt service.

STOP

Function : Enters stop mode and stops all oscillators

Description : Before executing this instruction, all signals on IOC port must be set to low.
The following 3 conditions cause the stop mode to be released.
1) One of the signals on the input mode pin of IOD or IOC port is in "H" state and holds long enough to cause the CPU to be released from halt mode.
2) A signal change in the INT pin.
3) The stop release condition specified by the SRE instruction is met.

SCA X

Function : The data specified by X causes the halt mode to be released.

Description : The signal change at port IOC,IOD is specified. The bit meaning of X(X4, X3) is shown below:

Bit pattern	Description
X4=1	Releases halt mode when signal is applied to IOC
X3=1	Releases halt mode when signal is applied to IOD

X2~0 don't care.

SIE* X

Function : Set/Reset interrupt enable flag

Description :

X0=1	The IEF0 is set so that interrupt 0(Signal change at port IOC or IOD specified by SCA) is accepted.
X1=1	The IEF1 is set so that interrupt 1 (underflow from timer 1) is accepted.
X2=1	The IEF2 is set so that interrupt 2(the signal change at the INT pin) 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.
X5=1	The IEF5 is set so that interrupt 5(key scanning) is accepted.
X6=1	The IEF6 is set so that interrupt 6(overflow from the VFC counter) is accepted.

SHE X

Function : Set/Reset halt release enable flag

Description :

X1=1	The HEF1 is set so that the halt mode is released by TMR1 underflow.
X2=1	The HEF2 is set so that the halt mode is released by signal changed on INT pin.
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 TMR2 underflow.
X5=1	The HEF5 is set so that the halt mode is released by the signal "L", applied on KI1~4 during scanning interval.
X6=1	The HEF6 is set so that the halt mode is released by VFC counter overflow.

Note : X0 don't care

SRE X

Function : Set/Reset stop release enable flag

Description :

X3=1	The SRF3 is set so that the stop mode is released by the signal changed on IOD port.
X4=1	The SRF4 is set so that the stop mode is released by the signal changed on IOC port.
X5=1	The SRF5 is set so that the stop mode is released by the signal changed on INT pin.
X7=1	The SRF7 is set so that the stop mode is released by the signal is "L" applied on KI1~4 in scanning interval.

Note : X2~0 are not a factor.

FAST

Function : Switches the system clock to CFOSC clock.

Description : Starts up the CFOSC(high speed osc.), and then switches the system clock to the high speed clock.

SLOW

Function : Switches the system clock to XTOSC clock(low speed osc).

Description : Switches the system clock to low speed clock, and then stops the CFOSC.

MSB Rx

Function : AC,[Rx] ← SCF3,SCF2,BCF1,BCF

Description : The SCF1, SCF2, SCF3 and BCF flag contents are loaded to AC and the data memory specified by Rx.
The content of AC and the meaning of the bits after the execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 3 (SCF3)	Start condition flag 2 (SCF2)	Start condition flag 1 (SCF1)	Backup flag (BCF)
Halt release caused by the lOD port	Halt release caused by SCF4,5,6,7,8,9	Halt release caused by the IOC port	The backup mode status in TM8740

MSC Rx

Function : AC,[Rx] ← SCF4, SCF5, SCF7, PH15

Description : The SCF4 to SCF7 contents are loaded to AC and the data memory specified by Rx.
The content of AC and the meaning of the bit after the 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	Start condition flag 5 (SCF5)	Start condition flag 4 (SCF4)
Halt release caused by predivider overflow		Halt release caused by TM1 underflow	Halt release caused by INT pin

MCX Rx

Function : AC,[Rx] ← SCF8,SCF6,SCF9

Description : The SCF8,SCF6,SCF9 contents are loaded to AC and the data memory specified by Rx.
The content of AC and the meaning of the bit after the execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 9 (SCF9)	NA	Start condition flag 6 (SCF6)	Start condition flag 8 (SCF8)
Halt release caused by VFC counter overflow	NA	Halt release caused by TM2 underflow	Halt release caused by the signal change to "L" applied on KI1~4 in scanning interval

MSD Rx

Function : Rx, AC ← WDF,CSF,RFOVF

Description : The watchdog flag, system clock status and overflow flag of RFC counter are loaded to data memory specified by Rx and AC.
The content of AC and the meaning of the bit after the execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	The overflow flag of 16-bit counter of VFC (VFVOF)	Watchdog timer enable flag (WDF)	System clock selection flag (CSF)

5-6 INDEX ADDRESS INSTRUCTIONS

MVU Rx

Function : [@U] ← [Rx],AC

Description : Loads the content of Rx to the index address buffer @U.
U3=[Rx]3, U2=[Rx]2, U1=[Rx]1, U0=[Rx]0,

MVH Rx

Function : [@H] ← [Rx],AC

Description : Loads the content of Rx to the index address buffer @H.
H3=[Rx]3, H2=[Rx]2, H1=[Rx]1, H0=[Rx]0,

MVL Rx

Function : [@L] ← [Rx]

Description : Loads the content of Rx to the index address buffer @L.
L3=[Rx]3, L2=[Rx]2, L1=[Rx]1, L0=[Rx]0

CPHL X

Function : If @HL = X, force the next instruction as NOP.

Description : Compare the content of the index register @HL in lower 8 bits (@h and @L) with the immediate data X.

Note : In the duration of the comparison of the index address, all the interrupt enable flags(IEF) have to be cleared to avoid malfunction.If the compared result is equal, the next executed instruction that is behind the CPHL instruction will be forced as NOP.If the compared result is not equal, the next executed instruction that is behind CPHL instruction will operate normally.

The comparison bit pattern is shown below :

CPHL X	X7	X6	X5	X4	X3	X2	X1	X0
@HL	IDBF7	IDBF6	IDBF5	IDBF4	IDBF3	IDBF2	IDBF1	IDBF0

5-7 DECIMAL ARITHMETIC INSTRUCTIONS

DAA

Function : AC ← BCD[AC]

Description : Converts the content of AC to binary 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 binary format, and then restores to AC and the 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.

DAA* @HL

Function : AC,[@HL] ← BCD[AC]

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

DAA*# @HL

Function : AC,[@HL] ← BCD[AC], @HL = @HL + 1

Description : Converts the content of AC to binary format, and then restores to AC and the data memory specified by @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
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 binary 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 binary format, and then restores to AC and the 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.

DAS* @HL

Function : AC, @HL ← BCD[AC]

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

DAS*# @HL

Function : AC, @HL ← BCD[AC], @HL = @HL + 1

Description : Converts the content of AC to binary format, and then restores to AC and the data memory @HL.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
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

5-8 JUMP INSTRUCTIONS

JB0 X

Function : Program counter jumps to X in current page, if AC0=1.

Description : If bit0 of AC is 1 , a jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 7FFH.

JB1 X

Function : Program counter jumps to X in current page, if AC1=1.

Description : If bit1 of AC is 1 , a jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 7FFH.

JB2 X

Function : Program counter jumps to X in current page, if AC2=1.
Description : If bit2 of AC is 1 , a jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 7FFH.

JB3 X

Function : Program counter jumps to X in current page, 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 7FFH.

JNZ X

Function : Program counter jumps to X in current page, if AC!=0.
Description : If the content of AC is not 0 , a jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 7FFH.

JNC X

Function : Program counter jumps to X in current page, if CF=0.
Description : If the content of CF is 0 , a jump occurs.
If 1, the PC increases by 1.
The range of X is from 000H to 7FFH.

JZ X

Function : Program counter jumps to X in current page, if AC=0.
Description : If the content of AC is 0 , a jump occurs.
If 1, the PC increases by 1.
The range of X is from 000H to 7FFH.

JC X

Function : Program counter jumps to X in current page, if CF=1.
Description : If the content of CF is 1 , a jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 7FFH.

JMP X

Function : Program counter jumps to X.
Description : Unconditional jump.

CALL X

Function : STACK ← (PC)+1,
Program counter jumps to X.
Description : A subroutine is called.

RTS

Function : PC ← (STACK)

Description : A return from a subroutine occurs.

5-9 MISCELLANEOUS INSTRUCTIONS

SCC X

Function : Sets the clock source for IOD and IOC chattering prevention, PWM output and frequency generator.

Description : The following table shows the meaning of each bit for this instruction:

Bit pattern	Clock source setting	Bit pattern	Clock source setting
X6=1	The clock source of frequency generator comes from the system clock (BCLK).	X6=0	The clock source of frequency generator comes from the PH0. Refer to section 3-3-4 for ϕ0.

Bit pattern	Clock source setting	Bit pattern	Clock source setting
(X4,X3) = 01 (X2,X1,X0)=001	Chattering prevention clock of IOD port = PH0	(X4,X3) = 10 (X2,X1,X0)=001	Chattering prevention clock of IOC port = PH0
(X4,X3) = 01 (X2,X1,X0)=010	Chattering prevention clock of IOD port = PH8	(X4,X3) = 10 (X2,X1,X0)=010	Chattering prevention clock of IOC port = PH8
(X4,X3) = 01 (X2,X1,X0)=100	Chattering prevention clock of IOD port = PH6	(X4,X3) = 10 (X2,X1,X0)=100	Chattering prevention clock of IOC port = PH6

X5 is reserved

FRQ D, Rx

Function : Frequency generator ← D, [Rx], AC

Description : Loads the content of AC and the data memory specified by Rx and D(D1, D0) to the frequency generator to set the duty cycle and initial value. The following table shows the preset data and the duty cycle setting:

	The bit pattern of preset letter N							
Programming divider	Bit7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRQ D, Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

Preset Letter D		Duty Cycle
D1	D0	
0	0	1/4 duty
0	1	1/3 duty
1	0	1/2 duty
1	1	1/1 duty

FRQ D, @HL

Function : Frequency generator ← D, T[@HL]

Description : Loads the content of Table ROM specified by @HL and D(D1, D0) to the frequency generator to set the duty cycle and initial value. The following table shows the preset data and the duty cycle setting:

	The bit pattern of preset letter N							
Programming divider	Bit7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRQ D,@HL	T7	T6	T5	T4	T3	T2	T1	T0

Note: T0 ~ T7 represents the data of table ROM.

Preset Letter D		Duty Cycle
D1	D0	
0	0	1/4 duty
0	1	1/3 duty
1	0	1/2 duty
1	1	1/1 duty

FRQX D, X

Function : Frequency generator ← D, X

Description : Loads the data X(X7 ~ X0) and D(D1, D0) to the frequency generator to set the duty cycle and initial value. The following table shows the preset data and the duty cycle setting:

	The bit pattern of preset letter N							
Programming divider	Bit7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	bit 1	bit 0
FRQX D,X	X7	X6	X5	X4	X3	X2	X1	X0

Note: X0 ~ X7 represents the data specified in operand X.

Preset Letter D		Duty Cycle
D1	D0	
0	0	1/4 duty
0	1	1/3 duty
1	0	1/2 duty
1	1	1/1 duty

1. FRQ D, Rx

The content of Rx and AC as preset data N.

2. FRQ D, @HL

The content of table ROM specified by @HL as preset data N.

3. FRQX D, X

The data of operand in the instruction assigned as preset data N.

TMS Rx

Function : Select the timer 1 clock source and preset timer 1.

Description : The content of the data memory specified by Rx and AC are loaded to timer 1 to start the timer.

The following table shows the bit pattern for this instruction:

	Select clock		Presetting value of timer 1					
TMS Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

The clock source selection for timer 1

AC3	AC2	Clock source
0	0	PH9
0	1	PH3
1	0	PH15
1	1	Output of frequency generator (FREQ)

TMS @HL

Function : Select the timer 1 clock source and preset timer 1.

Description : The content of the table ROM specified by @Hl is loaded to timer 1 to start the timer.

The following table shows the bit pattern for this instruction:

	Select clock		Presetting value of timer 1					
TMS @HL	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

The clock source selection for timer 1

Bit7	Bit6	Clock source
0	0	PH9
0	1	PH3
1	0	PH15
1	1	Output of frequency generator (FREQ)

TMSX X

Function : Selects the timer 1 clock source and preset timer 1.

Description : The data specified by X(X7 ~ X0) is loaded to timer 1 to start the timer.

The following table shows the bit pattern for this instruction:

OPCODE	Select clock			Presetting value of timer 1					
TMSX X	X8	X7	X6	X5	X4	X3	X2	X1	X0

The clock source selection for timer 1

X8	X7	X6	clock source
0	0	0	PH9
0	0	1	PH3
0	1	0	PH15
0	1	1	Output of frequency generator (FREQ)
1	0	0	PH5
1	0	1	PH7
1	1	0	PH11
1	1	1	PH13

TM2 Rx

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

Description : The content of data memory specified by Rx and AC is loaded to timer 2 to start the timer.

The following table shows the bit pattern for this instruction:

OPCODE	Select clock		Presetting value of timer 2					
TM2 Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

The clock source selection for timer 2

AC3	AC2	clock source
0	0	PH9
0	1	PH3
1	0	PH15
1	1	Output of frequency generator (FREQ)

TM2 @HL

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

Description : The content of the Table ROM specified by @HL is loaded to timer 2 to start the timer.

The following table shows the bit pattern for this instruction:

OPCODE	Select clock		Presetting value of timer 2					
TM2 @HL	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

The clock source selection for timer 2

Bit7	Bit6	clock source
0	0	PH9
0	1	PH3
1	0	PH15
1	1	Output of frequency generator (FREQ)

TM2X X

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

Description : The data specified by X(X8 ~ X0) is loaded to timer 2 to start the timer.

The following table shows the bit pattern for this instruction:

OPCODE	Select clock			Presetting value of timer 2					
TM2X X	X8	X7	X6	X5	X4	X3	X2	X1	X0

The clock source selection for timer 2

X8	X7	X6	clock source
0	0	0	PH9
0	0	1	PH3
0	1	0	PH15
0	1	1	Output of frequency generator (FREQ)
1	0	0	PH5
1	0	1	PH7
1	1	0	PH11
1	1	1	PH13

SF X

Function : Sets flag

Description : Description of each flag
X0 : "1" The CF flag is set to 1.
X1 : "1" The chip enters backup mode and the BCF flag is set to 1.
X4 : "1" The watchdog timer is initiated and active and WDF flag is to 1.
X7 : "1" Enables the re-load function of timer 1.
X6, 5, 3 and 2 is reserved

RF X

Function : Resets flag

Description : Description of each flag
X0 : "1" The CF flag is reset to 0.
X1 : "1" The chip escapes from backup mode and BCF flag is reset to 0.
X4 : "1" The watchdog timer is disabled and WDF flag is reset to 0.
X7 : "1" Disables the re-load function of timer 1.
X6, 5, 3 and 2 are reserved

SF2 X

Function : Sets flag

Description : Description of each flag
X3 : "1" Enables the strong pull-low device on INT pin.
X2 : "1" Turns off the LCD display temporarily.
X1 : "1" Sets the DED flag. Refer to 2-12-3 for detail.
X0 : "1" Enables the re-load function of timer 2.

RF2 X

Function : Resets flag

Description : Description of each flag
X3 : "1" Disables the strong pull-low device on INT pin.
X2 : "1" Turns on the LCD display.
X1 : "1" Resets the DED flag. Refer to 2-12-3 for detail.
X0 : "1" Disables the re-load function of timer 2.

PLC

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 C is reset.

X1 : "1" Halt release request flag HRF1 caused by underflow from the timer 1 is reset, and stops the operating of timer 1(TM1).

X2 : "1" Halt or stop release request flag HRF2 caused by the signal change at the INT pin is reset.

X3 : "1" Halt release request flag HRF3 caused by overflow from the predivider is 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).

X5 : "1" Halt release request flag HRF5 caused by the signal change to "L" on KI1~4 in scanning interval is reset.

X6 : "1" Halt release request flag HRF6 caused by overflow from the RFC counter is reset.

X8 : "1" The last 5 bits of the predivider (15 bits) are reset. When executing this instruction, X3 must be set to "1" simultaneously.

Appendix A TM8740 Instruction Table

Instruction		Machine Code	Function		Flag/Remark
NOP		0000 0000 0000 0000	No Operation		
LCT	Lz,Ry	0000 001Z ZZZZ ZYYY	(Lz)	← 7SEG ← (Ry)	Ry=70H~77H
LCB	Lz,Ry	0000 010Z ZZZZ ZYYY	(Lz)	←7SEG ← (Ry)	Blank Zero
LCP	Lz,Ry	0000 011Z ZZZZ ZYYY	(Lz)	← (Ry) , (AC)	
LCD	Lz,@HL	0000 100Z ZZZZ Z000	(Lz)	← (R@HL)	
LCT	Lz,@HL	0000 100Z ZZZZ Z001	(Lz)	←7SEG ← (R@HL)	
LCB	Lz,@HL	0000 100Z ZZZZ Z010	(Lz)	←7SEG ← (R@HL)	Blank Zero
LCP	Lz,@HL	0000 100Z ZZZZ Z011	(Lz)	← (R@HL) , (AC)	
LCDX	D	0000 100D D000 0100	(Multi-Lz) D=00 D=01 D=10 D=11	← (T@HL) : Multi-Lz=00H~0FH : Multi-Lz=10H~1FH : Multi-Lz=20H~2FH : Multi-Lz=30H~3FH	
LCTX	D	0000 100D D000 0101	((Multi-Lz) D=00 D=01 D=10 D=11	←7SEG ←(R @HL) : Multi-Lz=00H~0FH : Multi-Lz=10H~1FH : Multi-Lz=20H~2FH : Multi-Lz=30H~3FH	
LCBX	D	0000 100D D000 0110	(Multi-Lz) D=00 D=01 D=10 D=11	←7SEG ←(R @HL) : Multi-Lz=00H~0FH : Multi-Lz=10H~1FH : Multi-Lz=20H~2FH : Multi-Lz=30H~3FH	Blank Zero
LCPX	D	0000 100D D000 0111	(Multi-Lz) D=00 D=01 D=10 D=11	←(R@HL) , (AC) : Multi-Lz=00H~0FH : Multi-Lz=10H~1FH : Multi-Lz=20H~2FH : Multi-Lz=30H~3FH	
OPA	Rx	0000 1010 0XXX XXXX	(IOA)	← (Rx)	
OPAS	D	0000 1011 D000 0000	IOA 3, 4	← D,Pulse	
OPB	Rx	0000 1100 0XXX XXXX	(IOB)	← (Rx)	
OPC	Rx	0000 1101 0XXX XXXX	(IOC)	← (Rx)	
OPD	Rx	0000 1110 0XXX XXXX	(IOD)	← (Rx)	
FRQ	D,Rx	0001 00DD 0XXX XXXX	FREQ D=00 D=01 D=10 D=11	← (Rx) , (AC) : 1/4 Duty : 1/3 Duty : 1/2 Duty : 1/1 Duty	
FRQ	D,@HL	0001 01DD 0000 0000	FREQ	←(T@HL)	
FRQX	D,X	0001 10DD XXXX XXXX	FREQ	← X	
MVL	Rx	0001 1100 0XXX XXXX	(@L)0~3	← (Rx)	
MVH	Rx	0001 1101 0XXX XXXX	(@H)4~7	← (Rx)	
MVU	Rx	0001 1110 0XXX XXXX	(@U)8~11	← (Rx)	
ADC	Rx	0010 0000 0XXX XXXX	(AC)	← (Rx) + (AC) + CF	CF
ADC	@HL	0010 0000 1000 0000	(AC)	← (R@HL) + (AC) + CF	CF
ADC#	@HL	0010 0000 1100 0000	(AC) (@HL)	←(R@HL) + (AC) + CF ←(@HL)+1	CF
ADC*	Rx	0010 0001 0XXX XXXX	(AC),(Rx)	←(Rx) + (AC) + CF	CF
ADC*	@HL	0010 0001 1000 0000	(AC), (R@HL)	←(R@HL) + (AC) + CF	CF
ADC*#	@HL	0010 0001 1100 0000	(AC), (R@HL) (@HL)	←(R@HL) + (AC) + CF ←(@HL)+1	CF
SBC	Rx	0010 0010 0XXX XXXX	(AC)	←(Rx) + (AC)B + CF	CF
SBC	@HL	0010 0010 1000 0000	(AC)	←(R@HL) + (AC)B + CF	CF
SBC#	@HL	0010 0010 1100 0000	(AC) (@HL)	←(R@HL) + (AC)B + CF ←(@HL)+1	CF
SBC*	Rx	0010 0011 0XXX XXXX	(AC), (R@HL)	←(R@HL) + (AC)B + CF	CF
SBC*	@HL	0010 0011 1000 0000	(AC), (R@HL)	←(R@HL) + (AC)B + CF	CF
SBC*#	@HL	0010 0011 1100 0000	(AC),(R@HL) (@HL)	←(R@HL) + (AC)B + CF ←(@HL)+1	CF
ADD	Rx	0010 0100 0XXX XXXX	(AC)	← (Rx) + (AC)	CF
ADD	@HL	0010 0100 1000 0000	(AC)	←(R@HL) + (AC)	CF

Instruction		Machine Code	Function		Flag/Remark
ADD#	@HL	0010 0100 1100 0000	(AC) (@HL)	←(R@HL) + (AC) ←(@HL)+1	CF
ADD*	Rx	0010 0101 0XXX XXXX	(AC),(Rx)	← (Rx) + (AC)	CF
ADD*	@HL	0010 0101 1000 0000	(AC), (R@HL)	←(R@HL) + (AC)	CF
ADD*#	@HL	0010 0101 1100 0000	(AC), (R@HL) (@HL)	←(R@HL) + (AC) ←(@HL)+1	CF
SUB	Rx	0010 0110 0XXX XXXX	(AC)	← (Rx) + (AC)B + 1	CF
SUB	@HL	0010 0110 1000 0000	(AC)	←(R@HL) + (AC)B + 1	CF
SUB#	@HL	0010 0110 1100 0000	(AC) (@HL)	←(R@HL) + (AC)B + 1 ←(@HL)+1	CF
SUB*	Rx	0010 0111 0XXX XXXX	(AC),(Rx)	← (Rx) + (AC)B + 1	CF
SUB*	@HL	0010 0111 1000 0000	(AC), (R@HL)	←(R@HL) + (AC)B + 1	CF
SUB*#	@HL	0010 0111 1100 0000	(AC), (R@HL) (@HL)	←(R@HL) + (AC)B + 1 ←(@HL)+1	CF
ADN	Rx	0010 1000 0XXX XXXX	(AC)	← (Rx) + (AC)	
ADN	@HL	0010 1000 1000 0000	(AC)	←(R@HL) + (AC)	
ADN#	@HL	0010 1000 1100 0000	(AC) (@HL)	←(R@HL) + (AC) ←(@HL)+1	
ADN*	Rx	0010 1001 0XXX XXXX	(AC),(Rx)	← (Rx) + (AC)	
ADN*	@HL	0010 1001 1000 0000	(AC), (R@HL)	←(R@HL) + (AC)	
ADN*#	@HL	0010 1001 1100 0000	(AC), (R@HL) (@HL)	←(R@HL) + (AC) ←(@HL)+1	
AND	Rx	0010 1010 0XXX XXXX	(AC)	← (Rx) AND (AC)	
AND	@HL	0010 1010 1000 0000	(AC)	←(R@HL) AND (AC)	
AND#	@HL	0010 1010 1100 0000	(AC) (@HL)	←(R@HL) AND (AC) ←(@HL)+1	
AND*	Rx	0010 1011 0XXX XXXX	(AC),(Rx)	← (Rx) AND (AC)	
AND*	@HL	0010 1011 1000 0000	(AC), (R@HL)	←(R@HL) AND (AC)	
AND*#	@HL	0010 1011 1100 0000	(AC),(R@HL) (@HL)	← (R@HL) AND (AC) ←(@HL)+1	
EOR	Rx	0010 1100 0XXX XXXX	(AC)	← (Rx) EOR (AC)	
EOR	@HL	0010 1100 1000 0000	(AC)	← (R@HL) EOR (AC)	
EOR#	@HL	0010 1100 1100 0000	(AC) (@HL)	← (R@HL) EOR (AC) ←(@HL)+1	
EOR*	Rx	0010 1101 0XXX XXXX	(AC),(Rx)	← (Rx) EOR (AC)	
EOR*	@HL	0010 1101 1000 0000	(AC),(R@HL)	← (R@HL) EOR (AC)	
EOR*#	@HL	0010 1101 1100 0000	(AC),(R@HL) (@HL)	← (R@HL) EOR (AC) ←(@HL)+1	
OR	Rx	0010 1110 0XXX XXXX	(AC)	← (Rx) OR (AC)	
OR	@HL	0010 1110 1000 0000	(AC)	← (R@HL) OR (AC)	
OR#	@HL	0010 1110 1100 0000	(AC) (@HL)	← (R@HL) OR (AC) ←(@HL)+1	
OR*	Rx	0010 1111 0XXX XXXX	(AC),(Rx)	← (Rx) OR (AC)	
OR*	@HL	0010 1111 1000 0000	(AC),(R@HL)	← (R@HL) OR (AC)	
OR*#	@HL	0010 1111 1100 0000	(AC),(R@HL) (@HL)	← (R@HL) OR (AC) ←(@HL)+1	
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) + D(B) + CF	CF
SBCI*	Ry,D	0011 0011 DDDD YYYY	(AC),(Ry)	← (Ry) + D(B) + 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) + D(B) + 1	CF
SUBI*	Ry,D	0011 0111 DDDD YYYY	(AC),(Ry)	← (Ry) + D(B) + 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	

Instruction		Machine Code	Function		Flag/Remark
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 0XXX XXXX	(AC),(Rx)	← (Rx) + 1	CF
INC*	@HL	0100 0000 1000 0000	(AC),(R@HL)	← (R@HL) + 1	CF
INC*#	@HL	0100 0000 1100 0000	(AC),(R@HL) (@HL)	← (R@HL) + 1 ←(@HL)+1	CF
DEC*	Rx	0100 0001 0XXX XXXX	(AC),(Rx)	← (Rx) – 1	CF
DEC*	@HL	0100 0001 1000 0000	(AC),(R@HL)	← (R@HL) - 1	CF
DEC*#	@HL	0100 0001 1100 0000	(AC),(R@HL) (@HL)	← (R@HL) - 1 ←(@HL)+1	CF
IPA	Rx	0100 0010 0XXX XXXX	(AC),(Rx)	← (IOA)	B1 : LBF B0 : VFCOUT
IPB	Rx	0100 0100 0XXX XXXX	(AC),(Rx)	← (IOB)	
IPC	Rx	0100 0111 0XXX XXXX	(AC),(Rx)	← (IOC)	
IPD	Rx	0100 1000 0XXX XXXX	(AC),(Rx)	← (IOD)	
MAF	Rx	0100 1010 0XXX XXXX	(AC),(Rx)	← (STS1)	B3 : CF B2 : ZERO B1 : (No use) B0 : (No use)
MSB	Rx	0100 1011 0XXX XXXX	(AC),(Rx)	← (STS2)	B3 : SCF3(DPT) B2 : SCF2(HRx) B1 : SCF1(CPT) B0 : BCF
MSC	Rx	0100 1100 0XXX XXXX	(AC),(Rx)	← (STS3)	B3 : SCF7(PDV) B2 : PH15 B1 : SCF5(TM1) B0 : SCF4(INT)
MCX	Rx	0100 1101 0XXX XXXX	(AC),(Rx)	← (STS3X)	B3 : SCF9(VFC) B2 : (No use) B1 : SCF6(TM2) B0 : SCF8(SKI)
MSD	Rx	0100 1110 0XXX XXXX	(AC),(Rx)	← (STS4)	B3 : (No use) B2 : VFOVF B1 : WDF B0 : CSF
SR0	Rx	0101 0000 0XXX XXXX	(AC)n, (Rx)n (AC)3, (Rx)3	← (Rx)(n+1) ← 0	
SR1	Rx	0101 0001 0XXX XXXX	(AC)n, (Rx)n (AC)3, (Rx)3	← (Rx)(n+1) ← 1	
SL0	Rx	0101 0010 0XXX XXXX	(AC)n, (Rx)n (AC)0, (Rx)0	← (Rx)(n-1) ← 0	
SL1	Rx	0101 0011 0XXX XXXX	(AC)n, (Rx)n (AC)0, (Rx)0	← (Rx)(n-1) ← 1	
DAA		0101 0100 0000 0000	(AC)	← BCD((AC))	CF
DAA*	Rx	0101 0101 0XXX XXXX	(AC),(Rx)	← BCD((AC))	CF
DAA*	@HL	0101 0101 1000 0000	(AC),(R@HL)	← BCD((AC))	CF
DAA*#	@HL	0101 0101 1100 0000	(AC),(R@HL) (@HL)	← BCD((AC)) ←(@HL)+1	CF
DAS		0101 0110 0000 0000	(AC)	← BCD((AC))	CF
DAS*	Rx	0101 0111 0XXX XXXX	(AC),(Rx)	← BCD((AC))	CF
DAS*	@HL	0101 0111 1000 0000	(AC),(R@HL)	← BCD((AC))	CF
DAS*#	@HL	0101 0111 1100 0000	(AC),(R@HL) (@HL)	← BCD((AC)) ←(@HL)+1	CF
LDS	Rx,D	0101 1DDD DXXX XXXX	(AC),(Rx)	← D	
LDH	Rx,@HL	0110 0000 0XXX XXXX	(AC),(Rx)	← H(T@HL)	
LDH*	Rx,@HL	0110 0001 0XXX XXXX	(AC),(Rx) (@HL)	← H(T@HL) ←(@HL) + 1	
LDL	Rx,@HL	0110 0010 0XXX XXXX	(AC),(Rx)	← L(T@HL)	
LDL*	Rx,@HL	0110 0011 0XXX XXXX	(AC),(Rx) (@HL)	← L(T@HL) ←(@HL) + 1	
MRF1	Rx	0110 0100 0XXX XXXX	(AC),(Rx)	← (VFC)3-0	
MRF2	Rx	0110 0101 0XXX XXXX	(AC),(Rx)	← (VFC)7-4	
MRF3	Rx	0110 0110 0XXX XXXX	(AC),(Rx)	← (VFC)11-8	
MRF4	Rx	0110 0111 0XXX XXXX	(AC),(Rx)	← (VFC)15-12	

Instruction		Machine Code	Function		Flag/Remark
STA	Rx	0110 1000 0XXX XXXX	(Rx)	← (AC)	
STA	@HL	0110 1000 1000 0000	@HL	← (AC)	
STA#	@HL	0110 1000 1100 0000	@HL (@HL)	← (AC) ←(@HL)+1	
LDA	Rx	0110 1100 0XXX XXXX	(AC)	← (Rx)	
LDA	@HL	0100 1100 1000 0000	(AC)	← (R@HL)	
LDA#	@HL	0100 1100 1100 0000	(AC) (@HL)	← (R@HL) ←(@HL)+1	
MRA	Rx	0110 1101 0XXX XXXX	CF	← (Rx)3	
MRW	@HL,Rx	0110 1110 0XXX XXXX	(AC),(R@HL)	← (Rx)	
MRW#	@HL,Rx	0110 1110 1XXX XXXX	(AC),(R@HL) (@HL)	← (Rx) ←(@HL)+1	
MWR	Rx,@HL	0110 1111 0XXX XXXX	(AC),(Rx)	← (R@HL)	
MWR#	Rx,@HL	0110 1111 1XXX XXXX	(AC),(Rx) (@HL)	← (R@HL) ←(@HL)+1	
MRW	Ry,Rx	0111 0YYY YXXX XXXX	(AC),(Ry)	← (Rx)	
MWR	Rx,Ry	0111 1YYY YXXX XXXX	(AC),(Rx)	← (Ry)	
JB0	X	1000 0XXX XXXX XXXX	(PC)	← X	if (AC)0 = 1
JB1	X	1000 1XXX XXXX XXXX	(PC)	← X	if (AC)1 = 1
JB2	X	1001 0XXX XXXX XXXX	(PC)	← X	if (AC)2 = 1
JB3	X	1001 1XXX XXXX XXXX	(PC)	← X	if (AC)3 = 1
JNZ	X	1010 0XXX XXXX XXXX	(PC)	← X	if (AC) ≠ 0
JNC	X	1010 1XXX XXXX XXXX	(PC)	← X	if CF = 0
JZ	X	1011 0XXX XXXX XXXX	(PC)	← X	if (AC) = 0
JC	X	1011 1XXX XXXX XXXX	(PC)	← X	if CF = 1
CALL	X	1100 XXXX XXXX XXXX	(STACK) (PC)	← (PC) + 1 ← X	
JMP	X	1101 XXXX XXXX XXXX	(PC)	← X	
TMS	Rx	1110 0000 0XXX XXXX	(AC)3,2 = 11 (AC)3,2 = 10 (AC)3,2 = 01 (AC)3,2 = 00 (AC)1~0, (Rx)3~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	Setting of Timer 1
TMS	@HL	1110 0001 0000 0000	(T@HL)7,6 = 11 (T@HL)7,6 = 10 (T@HL)7,6 = 01 (T@HL)7,6 = 00 (T@HL)5~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	Setting of Timer 1
TMSX	X	1110 001X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	Setting of Timer 1
TM2	Rx	1110 0100 0XXX XXXX	(AC)3,2 = 11 (AC)3,2 = 10 (AC)3,2 = 01 (AC)3,2 = 00 (AC)1~0, (Rx)3~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	Setting of Timer 2
TM2	@HL	1110 0101 0000 0000	(T@HL)7,6 = 11 (T@HL)7,6 = 10 (T@HL)7,6 = 01 (T@HL)7,6 = 00 (T@HL)5~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	Setting of Timer 2
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=011 X8,7,6=010 X8,7,6=001	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3	Setting of Timer 2

Instruction		Machine Code	Function		Flag/Remark
			X8,7,6=000 X5~0	: Ctm = PH9 : Set Timer2 Value	
SHE	X	1110 1000 0XXX XXX0	X6 X5 X4 X3 X2 X1	: Enable HEF6 : Enable HEF5 : Enable HEF4 : Enable HEF3 : Enable HEF2 : Enable HEF1	VFC KEY_S TMR2 PDV INT TMR1
SIE*	X	1110 1001 0XXX XXXX	X6 X5 X4 X3 X2 X1 X0	: Enable IEF6 : Enable IEF5 : Enable IEF4 : Enable IEF3 : Enable IEF2 : Enable IEF1 : Enable IEF0	VFC KEY_S TMR2 PDV INT TMR1 C,DPT
PLC	X	1110 101X 0XXX XXXX	X8 X6-0	: Reset PH15~11 : Reset HRF6-0	
SRF	X	1110 1100 00XX X000	X5 X4 X3	: Enable Cx Control : Enable TM2 Control : Enable Counter	 ENX
SRE	X	1110 1101 X0XX X000	X7 X5 X4 X3	: Enable SRF7 : Enable SRF5 : Enable SRF4 : Enable SRF3	SRF7(KEY_S) SRF5 (INT) SRF4 (C Port) SRF3 (D port)
FAST		1110 1110 0000 0000		: Switch to High Speed Clock	
SLOW		1110 1110 1000 0000		: Switch to Low Speed Clock	
CPHL	X	1110 1111 XXXX XXXX		Skip next instruction when X7~0=(@HL)7~0	
SPKX	X	1111 0010 XXXX XXXX	X6=1 X6=0 X7,5,4=000 X7,5,4=001 X7,5,4=010 X7,5,4=10X X7,5,4=110 X7,5,4=111	: KEY_S release by scanning cycle : KEY_S release by normal key scanning : Set one of KO1~16 =1 by X3~0 : Set all = 1 : Set all Hi-z : Set eight of KO1~16 =1 by X3 X3=0 => KO1~8 X3=1 => KO9~16 : Set four of KO1~16 =1 by X3,2 X3,2=00 => KO1~4 X3,2=01 => KO5~8 X3,2=10 => KO9~12 X3,2=11 => KO13~16 : Set two of KO1~16 =1 by X3,2,1 X3~1=000=>KO1,2 X3~1=001=>KO3,4 X3~1=010=>KO5,6 X3~1=011=>KO7,8 X3~1=100=>KO9,10 X3~1=101=>KO11,12 X3~1=110=>KO13,14 X3~1=111=>KO15,16	
SPK	Rx	1111 0000 0XXX XXXX	(AC)2=1 (AC)2=0 (AC)7,5,4=000 (AC)7,5,4=001 (AC)7,5,4=010 (AC)7,5,4=10X (AC)7,5,4=110	: KEY_S release by scanning cycle : KEY_S release by normal key scanning : Set one of KO1~16 =1 by (Rx)3~0 : Set all = 1 : Set all Hi-z : Set eight of KO1~16 =1 by (Rx)3 (Rx)3=0 => KO1~8 (Rx)3=1 => KO9~16 : Set four of KO1~16 =1 by (Rx)3,2 (Rx)3,2=00 => KO1~4 (Rx)3,2=01 => KO5~8 (Rx)3,2=10 => KO9~12	

Instruction		Machine Code	Function		Flag/Remark
			(AC)7,5,4=111	(Rx)3,2=11 => KO13~16 : Set two of KO1~16 =1 by X3,2,1 (Rx)3~1=000=>KO1,2 (Rx)3~1=001=>KO3,4 (Rx)3~1=010=>KO5,6 (Rx)3~1=011=>KO7,8 (Rx)3~1=100=>KO9,10 (Rx)3~1=101=>KO11,12 (Rx)3~1=110=>KO13,14 (Rx)3~1=111=>KO15,16	
SPK	@HL	1111 0001 0000 0000	(T@HL)6=1 (T@HL)6=0 (T@HL)7,5,4 = 000 (T@HL)7,5,4 = 001 (T@HL)7,5,4 = 010 (T@HL)7,5,4 = 10X (T@HL)7,5,4 = 110 (T@HL)7,5,4 = 111	: KEY_S release by scanning cycle : KEY_S release by normal key scanning : Set one of KO1~16 =1 by (T@HL)3~0 : Set all = 1 : Set all Hi-z : Set eight of KO1~16 =1 by (T@HL)3 (T@HL)3=0 => KO1~8 (T@HL)3=1 => KO9~16 : Set four of KO1~16 =1 by (T@HL)3,2 (T@HL)3,2=00 => KO1~4 (T@HL)3,2=01 => KO5~8 (T@HL)3,2=10 => KO9~12 (T@HL)3,2=11 => KO13~16 : Set two of KO1~16 =1 by (T@HL)3,2,1 (T@HL)3~1=000=>KO1,2 (T@HL)3~1=001=>KO3,4 (T@HL)3~1=010=>KO5,6 (T@HL)3~1=011=>KO7,8 (T@HL)3~1=100=>KO9,10 (T@HL)3~1=101=>KO11,12 (T@HL)3~1=110=>KO13,14 (T@HL)3~1=111=>KO15,16	
RTS		1111 0100 0000 0000	(PC)	← STACK	CALL Return
SCC	X	1111 0100 1X0X XXXX	X6 = 1 X6 = 0 X4=1 X3=1 X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cfq = BCLK : Cfq = PH0 Set IOC Cch Set IOD Cch : Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1111 0101 000X X000	X4 X3	: Enable SEF4 : Enable SEF3	C1-4 D1-4
SPA	X	1111 0101 100X XX00	X4 X3~2	: Enable IOA4-3 Pull-Low : Set IOA4-3 I/O mode	
SPB	X	1111 0101 101X XXXX	X4 X3~0	: Enable IOB4-1 Pull-Low : Set IOB4-1 I/O mode	
SPC	X	1111 0101 110X XXXX	X4 X3-0	: Enable IOC4-1 Pull-Low / Low -Level-Hold : Set IOC4-1 I/O mode	
SPD	X	1111 0101 111X XXXX	X4 X3-0	: Enable IOD4-1 Pull-Low : Set IOD4-1 I/O mode	
SF	X	1111 0110 X00X 00XX	X7 X4 X1 X0	: Enable TM1 Reload function : Enable watchdog timer : Set BCF flag : Set CF	RL1 WDF BCF CF
RF	X	1111 0111 X00X 00XX	X7 X4 X1 X0	: Disable TM1 Reload function : Disable watchdog timer : Reset BCF : Reset CF	RL1 WDF BCF CF
ALM	X	1111 110X XXXX XXXX	X8,7,6=111 X8,7,6=100	: FREQ : DC "1"	

Instruction		Machine Code	Function		Flag/Remark
			X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: PH3 : PH4 : PH5 : DC "0" ← PH15~10	
SF2	X	1111 1110 0000 XXXX	X3 X2 X1 X0	: Enable INT strong Pull-low dev. : Turn off all Segments : Set DED flag : Enable TM2 Reload function	INTPL RSOFF DED RL2
RF2	X	1111 1110 1000 XXXX	X3 X2 X1 X0	: Disable INT powerful Pull-low : Release Segments : Reset DED flag : Disable TM2 Reload function	INTPL RSOFF DED RL2
HALT		1111 1111 0000 0000	Halt Operation		
STOP		1111 1111 1000 0000	Stop Operation		

Symbol Description

Symbol	Description	Symbol	Description
( )	Content of Register	D	Immediate Data
AC	Accumulator	(D)B	Complement of Immediate Data
(AC)n	Content of Accumulator (bit n)	PC	Program Counter
(AC)B	Complement of content of Accumulator	CF	Carry Flag
X	Address of program or control data	ZERO	Zero Flag
Rx	Address X of data RAM	WDF	Watch-Dog Timer Enable Flag
(Rx)n	Bit n content of Rx	7SEG	7 segment decoder for LCD
Ry	Address Y of working register	BCLK	System clock for instruction
R@HL	Address of data RAM specified by @HL	IEFn	Interrupt Enable Flag
BCF	Back-up Flag	HRFn	HALT Release Flag
@HL	Generic Index address register	HEFn	HALT Release Enable Flag
(@HL)	Content of generic Index address register	Lz	Address of LCD PLA Latch
(@L)	Content of lowest nibble Index register	SRFn	STOP Release Enable Flag
(@H)	Content of middle nibble Index register	SCFn	Start Condition Flag
(@U)	Content of highest nibble Index register	Cch	Clock Source of Chattering prevention ckt.
T@HL	Address of Table ROM	Cfq	Clock Source of Frequency Generator
H(T@HL)	High Nibble content of Table ROM	SEFn	Switch Enable Flag
L(T@HL)	Low Nibble content of Table ROM	FREQ	Frequency Generator setting Value
TMR	Timer Overflow Release Flag	CSF	Clock Source Flag
Ctm	Clock Source of Timer	P	Program Page
PDV	Pre-Divider	VFOVF	VFC Overflow Flag
STACK	Content of stack	VFC	Voltage to Frequency counter
TM1	Timer 1	(VFC)n	Bit data of Voltage to Frequency counter
TM2	Timer 2	LBF	Low Battery Detect Flag