十速

TM8724

4-Bit Microcontroller with LCD Driver

Rev 1.4

AMENDMENT HISTORY

Version	Date	Description
V1.0	Sept, 2010	New release
V1.1	Nov, 2007	Add External RC oscillator
V1.2	Dec, 2011	Add Ordering Information table
V1.3	Oct, 2018	Corrected LZ from 5 bits to 4 bits, PSTB and other related changes are P.73, P.75~77, and the changed instructions are: LCT, LCB, LCP, LCD (P.103)
V1.4	Apr, 2022	Modify bit6 of SCC machine code to “0”

CONTENTS

1. Chapter 1 General Description

1-1 GENERAL DESCRIPTION

The TM8724 is an embedded high-performance 4-bit microcomputer with LCD driver. It contains all the necessary functions, such as 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, LCD driver, look-up table and watchdog timer in a single chip.

1-2 FEATURE

(1). 1.5V and 3V operations and with low power dissipation.

(2). Powerful instruction set (129 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.
- 16 working registers and manipulation.
- Table look-up.
- LCD driver data transfer.

(3). Memory capacity.

- ROM capacity 1152 x 16 bits.
- RAM capacity 80 x 4 bits.

(4). LCD driver output.

- 5 common outputs and 24 segment outputs (up to drive 120 LCD pixels).
- 1/1 Duty, 1/2 Duty, 1/3 Duty, 1/4 Duty or 1/5 Duty is selected by MASK option.
- 1/2 Bias or 1/3 Bias is selected by MASK option.
- Single instruction to turn off all segments and coms.
- Segment output pins (SEG1~12) could be defined as CMOS or P_open drain output type by mask option.
- Input/output ports.
- Port IOA 4 pins, muxed with SEG24~SEG27.
- Port IOB 4 pins (with internal pull-low), muxed with SEG28~SEG31.
- Port IOC 4 pins (with internal pull-low), muxed with SEG32~SEG35.

  IOC port had built in the input signal chattering prevention circuitry.

(5). 8 level subroutine nesting.

(6). Interrupt function.

- External factor 2 (INT pin, Port IOC).
- Internal factors 2 (Pre-Divider, Timer1).

(7). Built in Alarm generator.

- BZB, BZ (Muxed with SEG30, SEG31).

(8). One 6-bit programmable timer with programmable clock source.

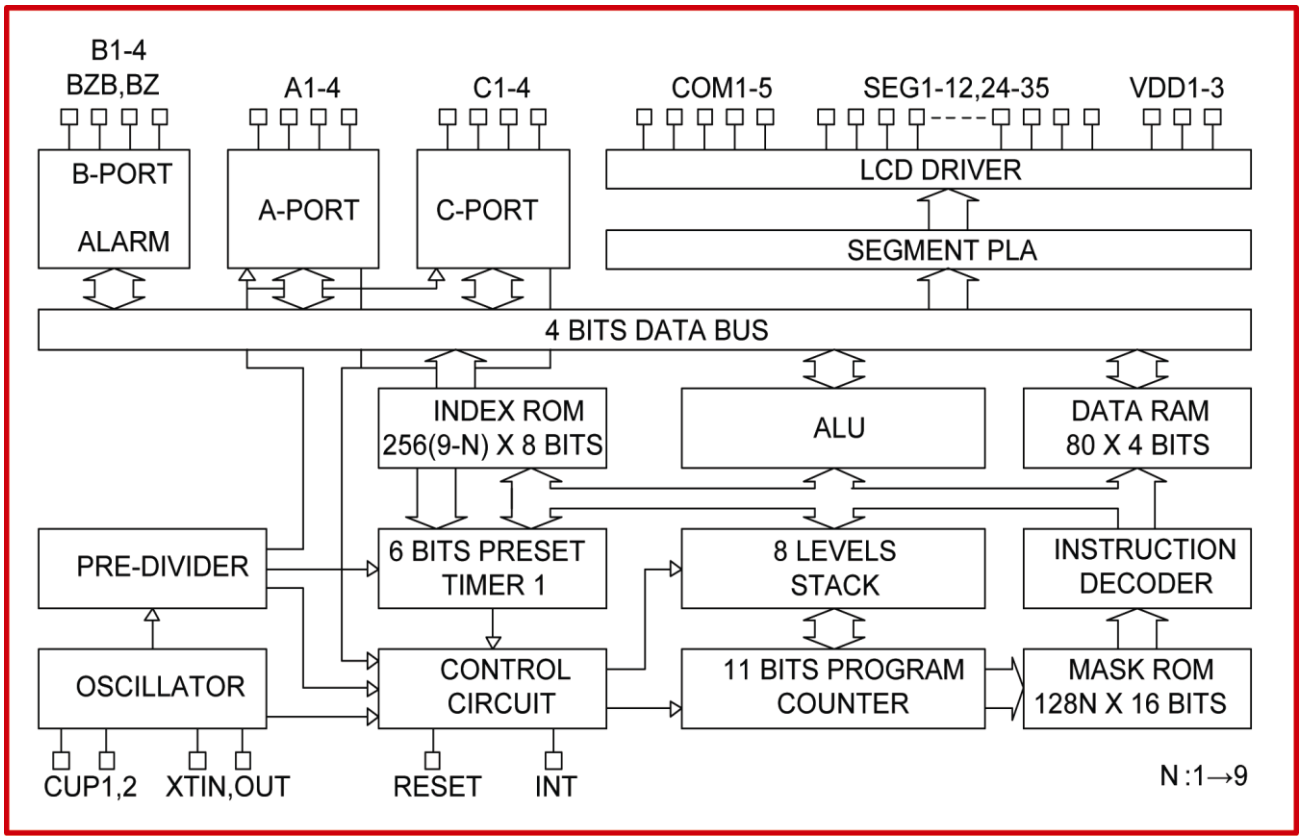
(9). Built-in Voltage doubler, tripler charge pump circuit.

(10). Dual clock operation.

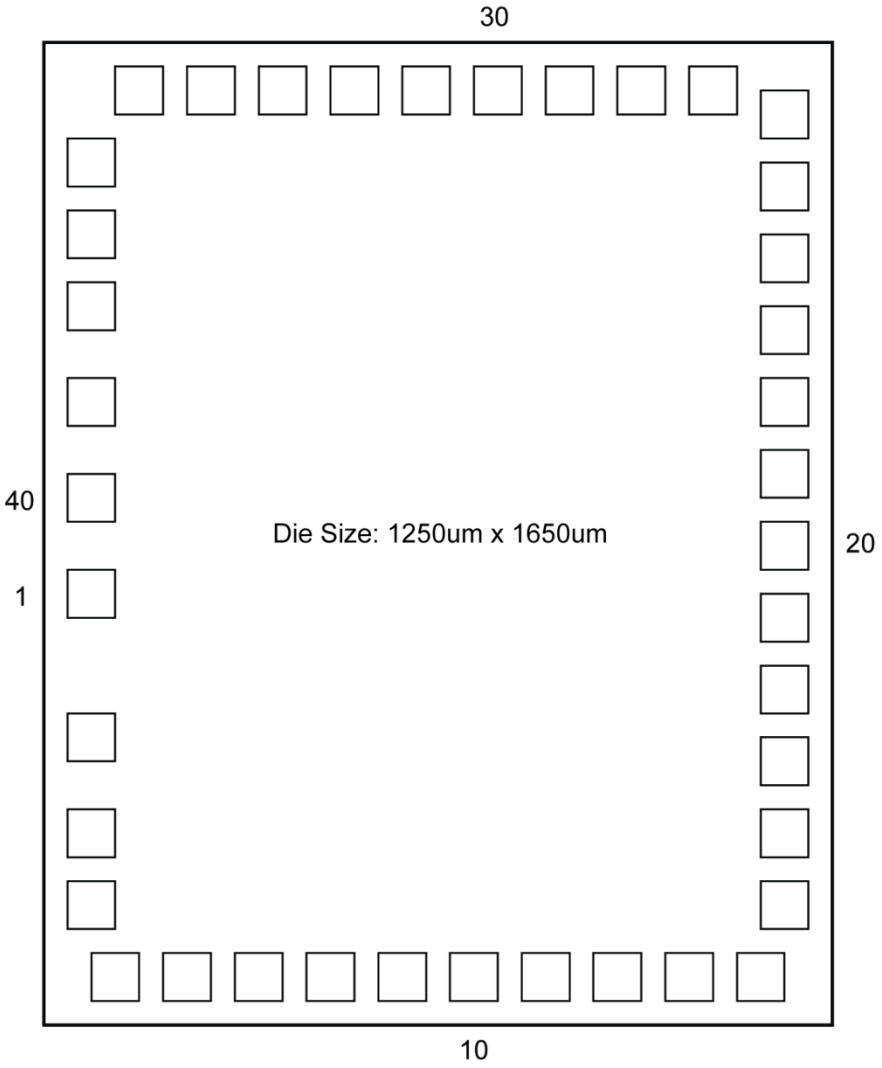
(11). HALT function.

(12). STOP function.

1-3 BLOCK DIAGRAM



1-4 PAD DIAGRAM



The substrate of chip should be connected to GND.

1-5 PAD COORDINATE

No	Name	X	Y	No	Name	X	Y
1	XIN	72.50	712.85	21	SEG8	1176.85	919.50
2	XOUT	72.50	451.35	22	SEG9	1176.85	1041.50
3	GND	75.00	314.05	23	SEG10	1176.85	1163.50
4	VDD1	75.00	199.05	24	SEG11	1176.85	1285.50
5	VDD2	94.95	72.50	25	SEG12	1176.85	1407.50
6	VDD3	209.95	72.50	26	SEG24/IOA1	1176.85	1534.20
7	CUP1	324.95	72.50	27	SEG25/IOA2	1056.10	1571.30
8	CUP2	439.95	72.50	28	SEG26/IOA3	936.10	1571.30
9	COM1	554.95	72.50	29	SEG27/IOA4	816.10	1571.30
10	COM2	669.95	72.50	30	SEG28/IOB1	696.10	1571.30
11	COM3	784.95	72.50	31	SEG29/IOB2	576.10	1571.30
12	COM4	899.95	72.50	32	SEG30/IOB3/BZB	456.10	1571.30
13	COM5	1014.95	72.50	33	SEG31/IOB4/BZ	336.10	1571.30
14	SEG1	1129.95	72.50	34	SEG32/IOC1	216.10	1571.30
15	SEG2	1176.85	187.50	35	SEG33/IOC2	96.10	1571.30
16	SEG3	1176.85	309.50	36	SEG34/IOC3	72.50	1440.65
17	SEG4	1176.85	431.50	37	SEG35/IOC4	72.50	1325.65
18	SEG5	1176.85	553.50	38	RESET	72.50	1188.65
19	SEG6	1176.85	675.50	39	INT	72.50	1029.85
20	SEG7	1176.85	797.50	40	TEST	72.50	871.65

1-6 PIN DESCRIPTION

Name	I/O	Description
VDD1, 2, 3	P	LCD supplies voltage, and positive supply voltage.
RESET	I	Input pin for external reset request signal, built-in internal pull-down resistor.
INT	I	Input pin for external interrupt request signal. ♦ Falling edge or rising edge triggered is defined by mask option. ♦ Internal pull-down or pull-up resistor is defined by mask option.
TEST		Test signal input pin.
CUP1, 2	O	Switching pins for supply the LCD driving voltage to the VDD1, 2, 3 pins. ♦ Connect the CUP1 and CUP2 pins with non-polarized electrolytic capacitor when chip operated in 1/2 or 1/3 bias mode. ♦ In no BIAS mode, leave these pins opened.
XIN XOUT	I O	Time base counter frequency (clock specified. LCD alternating frequency. Alarm signal frequency) or system clock oscillation. ♦ 32KHz Crystal oscillator or External RC for SLOW ONLY or DUAL by mask option. ♦ In FAST ONLY mode option, connect an external resistor could compose a RC oscillator.
COM1~5	O	Output pins for driving the common pins of the LCD panel.
SEG1-12, SEG24-35	O	Output pins for driving the LCD panel segment.
IOA1-4	I/O	Input / Output port A. (muxed with SEG24~27)
IOB1-4	I/O	Input / Output port B. (muxed with SEG28~30)
IOC1-4	I/O	Input / Output port C. (muxed with SEG32~35)
BZB/BZ	O	Output port for alarm generator
GND	P	Negative supply voltage.

1-7 CHARACTERISTIC

(1). ABSOLOUTE MAXIMUM RATINGS

at Ta=-40 to 80℃,GND= 0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD1	-0.3 to 2.0	V
	VDD2	-0.3 to 4.0	
	VDD3	-0.3 to 6.0	
Maximum Input Voltage	Vin	-0.3 to VDD1/2+0.3	
Maximum output Voltage	Vout1	-0.3 to VDD1/2+0.3	
	Vout2	-0.3 to VDD3+0.3	
Maximum Operating Temperature	Topg	-40 to +80	℃
Maximum Storage Temperature	Tstg	-50 to +125	

(2). POWER CONSUMPTION

at Ta=-40 to 80℃,GND= 0V

Name	Sym.	Condition	Min.	Typ.	Max.	Unit
HALT mode	IHALT1	Only 32.768KHz Crystal Oscillator operating, without loading. (BCF = 0) Ag mode, VDD1=1.5V, BCF = 0		2	5	uA
	IHALT2	Only 32.768KHz Crystal Oscillator operating, without loading. (BCF = 0) Li mode, VDD2=3.0V, BCF = 0		5	10	
STOP mode	ISTOP				1	

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

(3). ALLOWABLE OPERATING CONDITIONS

at Ta=-40 to 80℃,GND= 0V

Name	Symb.	Condition	Min.	Max.	Unit
Supply Voltage	VDD1		1.2	1.8	V
	VDD2		2.4	3.6	
	VDD3		2.4	5.4	
Supply Voltage	VDD1	Ag Mode	1.2	1.65	
Supply Voltage	VDD2	EXT-V, Li Mode	2.4	5.25	
Oscillator Start-Up Voltage	VDDB	Crystal Mode	1.3		
Oscillator Sustain Voltage	VDDB	Crystal Mode	1.2		
Input “H” Voltage	Vih1	Ag Battery Mode	VDD1-0.7	VDD1+0.7	
Input “L” Voltage	Vil1		-0.7	0.7	
Input “H” Voltage	Vih2	Li Battery Mode	VDD2-0.7	VDD2+0.7	
Input “L” Voltage	Vil2		-0.7	0.7	
Input “H” Voltage	Vih3	OSCIN at Ag Battery Mode	0.8xVDD1	VDD1	
Input “L” Voltage	Vil3		0	0.2xVDD1	
Input “H” Voltage	Vih4	OSCIN at Li Battery Mode	0.8xVDD2	VDD2	
Input “L” Voltage	Vil4		0	0.2xVDD2	
Operating Freq	Fopg1	Crystal Mode	32		KHz
	Fopg2	RC/CF Mode	10	1000	

(4). INTERNAL RC FREQUENCY RANGE

Option Mode	BAK	Min.	Typ.	Max.
250KHz	1.5V	300KHz	350KHz	400KHz
	3.0V	250KHz	300KHz	350KHz
500KHz	1.5V	550KHz	650KHz	750KHz
	3.0V	450KHz	550KHz	650KHz

(5). ELECTRICAL CHARACTERISTICS

at #1:VDD1=1.5V(Ag);

at #2:VDD2=3.0V(Li);

(6). Input Resistance

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
IOB, C Pull-Down Tr	Rmad1	Vi=VDD1, #1	200	500	1000	KΩ
	Rmad2	Vi=VDD2, #2	200	500	1000	
INT Pull-up Tr	Rintu1	Vi=VDD1, #1	100	250	500	
	Rintu2	Vi=VDD2, #2	100	250	500	
INT Pull-Down Tr	Rintd1	Vi=GND, #1	200	500	1000	
	Rintd2	Vi=GND, #2	200	500	1000	
RES Pull-Down R	Rres1	Vi=VDD1, #1	10	40	100	
	Rres2	Vi=VDD2, #2	10	40	100	

(7). DC Output Characteristics

Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit
Output "H" Voltage	Voh1c	Ioh=-200uA	COM1~5 SEG1~35	0.8	0.9	1.0	V
	Voh2c	Ioh=-1mA		1.5	1.8	2.1	
Output "L" Voltage	Vol1c	Iol=400uA		0.2	0.3	0.4	
	Vol2c	Iol=2mA		0.3	0.6	0.9	

(8). Segment Driver Output Characteristics

(A). Static Display Mode

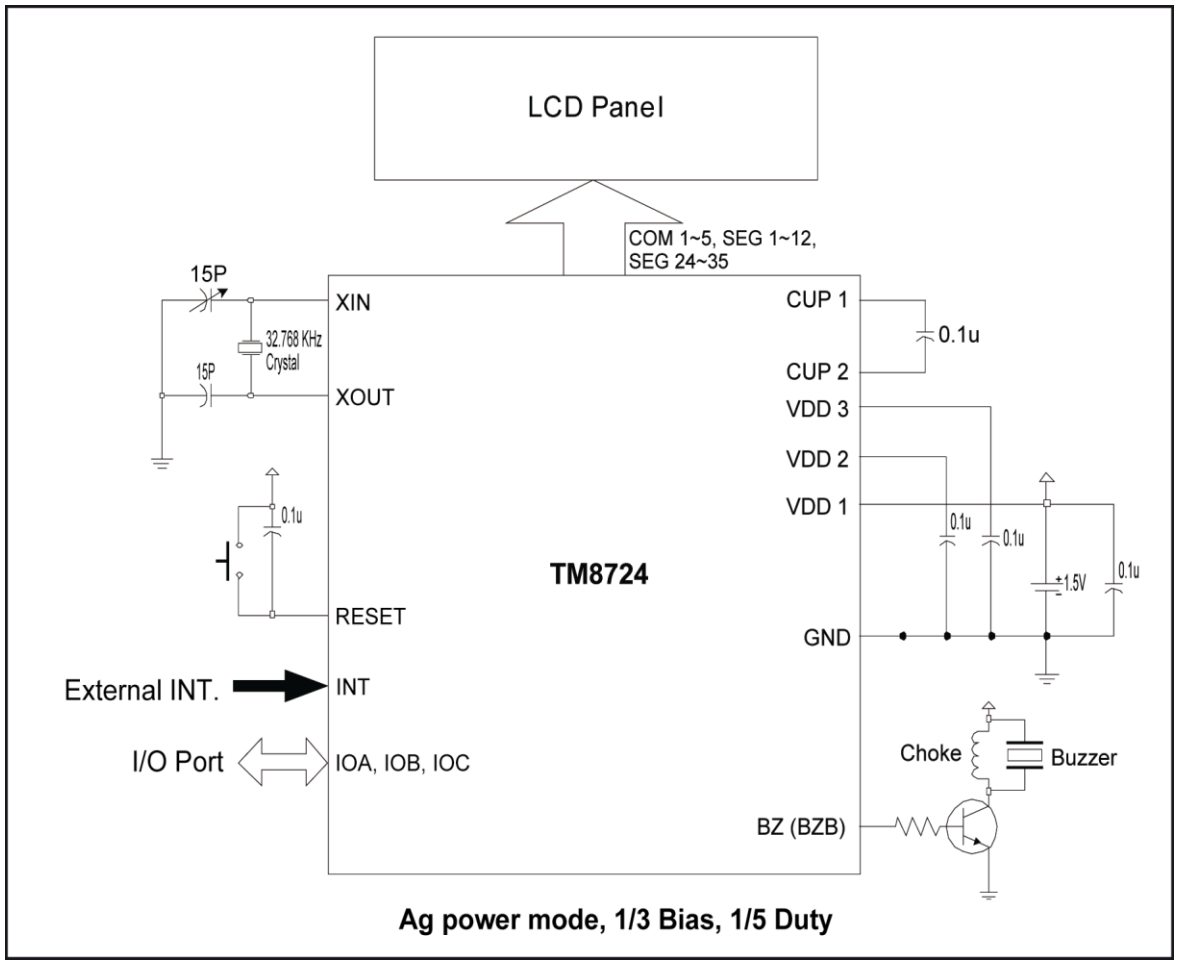
Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
Output "H" Voltage	Voh1d	Ioh=-1uA, #1	SEG-n	1.0			V
	Voh2d	Ioh=-1uA, #2		2.2			
Output "L" Voltage	Vol1d	Iol=1uA, #1				0.2	
	Vol2d	Iol=1uA, #2				0.2	
Output "H" Voltage	Voh1e	Ioh=-10uA, #1	COM-n	1.0			
	Voh2e	Ioh=-10uA, #2		2.2			
Output "L" Voltage	Vol1e	Iol=10uA, #1				0.2	
	Vol2e	Iol=10uA, #2				0.2	

(B). 1/2 Bias Display Mode							
Name	**Symb.**	**Condition**	**For**	**Min.**	**Typ.**	**Max.**	**Unit.**
Output ”H” Voltage	Vohd	Ioh=-1uA, #1, #2	SEG-n	2.2			V
Output ”L” Voltage	Vold	Iol=1uA, #1,#2				0.2	
Output ”H” Voltage	Vohe	Ioh=-10uA, #1, #2	COM-n	2.2			
Output ”M” Voltage	Vom1e	Iol/h=+/-10uA, #1,#2		1.0		1.4	
Output ”L” Voltage	Vole	Iol=10uA, #1, #2				0.2	

(C). 1/3 Bias display Mode							
Name	**Symb.**	**Condition**	**For**	**Min.**	**Typ.**	**Max.**	**Unit.**
Output ”H” Voltage	Vohf	Ioh=-1uA, #1, #2	SEG-n	3.4			V
Output ”M1” Voltage	Vom1f	Iol/h=+/-10uA, #1, #2		1.0		1.4	
Output ”M2” Voltage	Vom2f	Iol/h=+/-10uA, #1, #2		2.2		2.6	
Output ”L” Voltage	Volf	Iol=1uA, #1, #2				0.2	
Output ”H” Voltage	Vohg	Ioh=-10uA, #1, #2	COM-n	3.4			
Output ”M1” Voltage	Vom1g	Iol/h=+/-10uA, #1, #2		1.0		1.4	
Output ”M2” Voltage	Vom2g	Iol/h=+/-10uA, #1, #2		2.2		2.6	
Output ”L” Voltage	Volg	Iol=10uA, #1, #2				0.2	

1-8 Typical Application Circuit

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



Ag power mode, 1/3 Bias, 1/5 Duty

2. Chapter 2 TM8724 Internal System Architecture

2-1 Power Supply

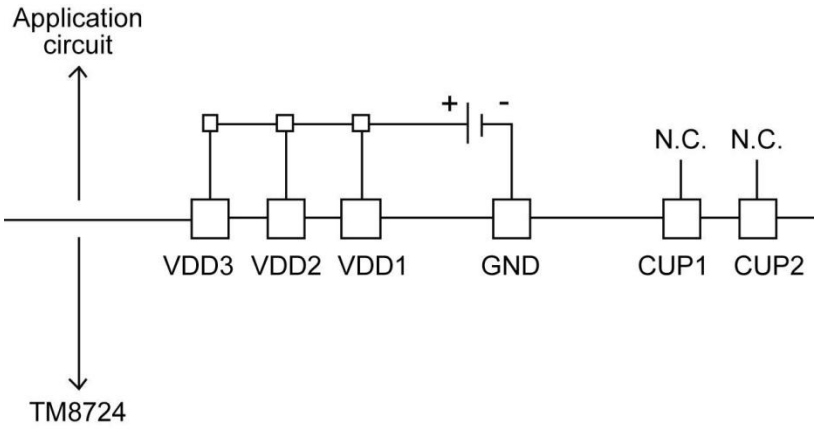
TM8724 could operate at Ag, Li, and EXT-V 3 types of supply voltage; all of these operating types are defined by mask option. The power supply circuitry also generates the necessary voltage level to drive the LCD panel with different bias. Shown below are the connection diagrams for 1/2 bias, 1/3 bias and no bias application.

2-1-1. Ag BATTERY POWER SUPPLY

Operating voltage range: 1.2V ~ 1.8V.

For different LCD bias application, the connection diagrams are shown below:

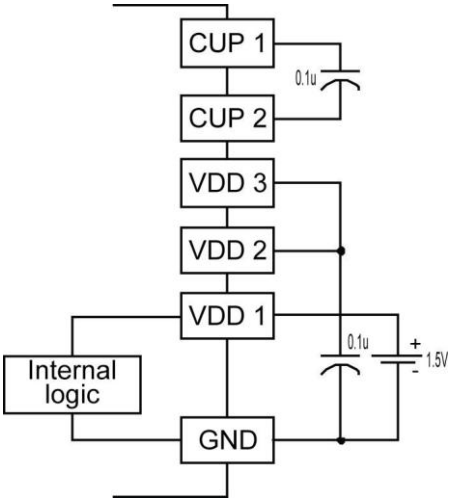
2-1-1-1. NO LCD BIAS NEED AT Ag BATTERY POWER SUPPLY



MASK OPTION table:

Mask Option name	Selected item
POWER SOURCE	(3) 1.5V BATTERY
LCD BIAS	(3) NO BIAS

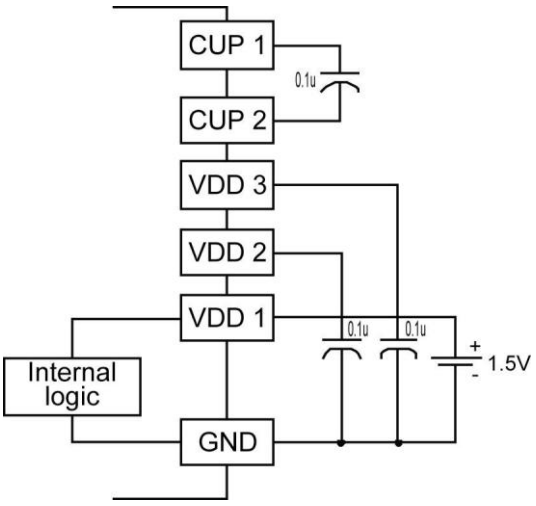
2-1-1-2. 1/2 BIAS & STATIC AT AG BATTERY POWER SUPPLY



MASK OPTION table:

Mask Option name	Selected item
POWER SOURCE	(3) 1.5V BATTERY
LCD BIAS	(2) 1/2 BIAS

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



MASK OPTION table:

Mask Option name	Selected item
POWER SOURCE	(3) 1.5V BATTERY
LCD BIAS	(1) 1/3 BIAS

Note-1: The input/output ports operate between GND and VDD1.

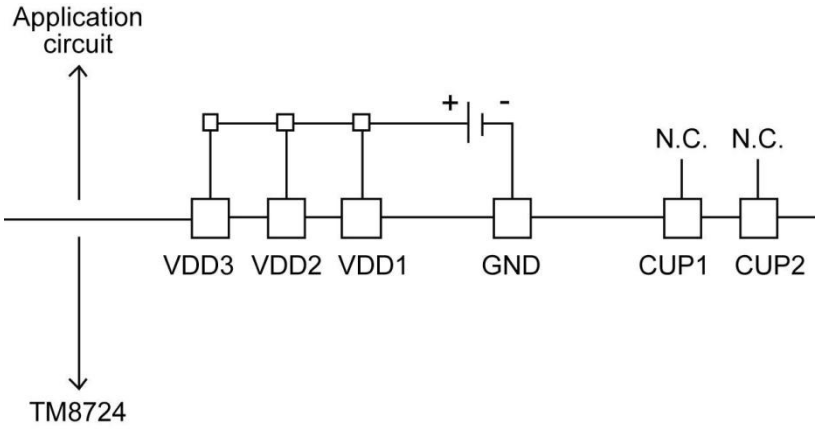
Note-2: At the initial clear mode, the backup flag (BCF) is set. When the backup flag is set, the oscillator circuit's invertor size becomes large and the oscillation conditions are improved, but the operating current is also increased. Therefore, the backup flag must be reset unless otherwise required.

2-1-2. LI BATTERY POWER SUPPLY

Operating voltage range: 2.4V ~ 3.6V.

For different LCD bias application, the connection diagrams are shown below:

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

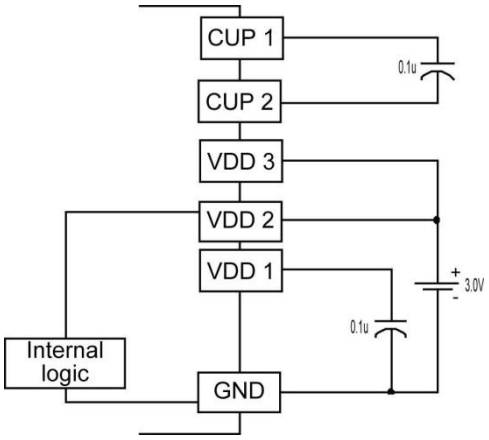


MASK OPTION table:

Mask Option name	Selected item
POWER SOURCE	(2) 3V BATTERY OR HIGHER
LCD BIAS	(3) NO BIAS

2-1-2-2. 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.

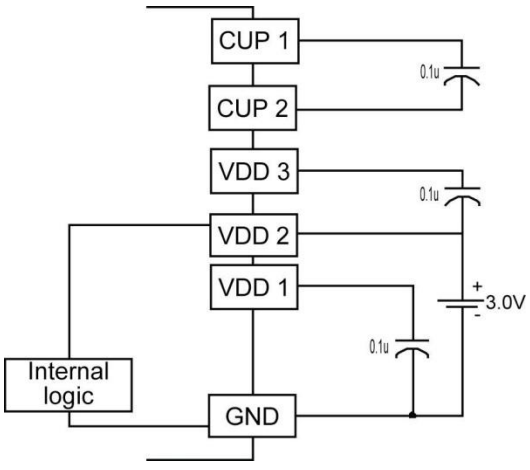


MASK OPTION table :

Mask Option name	Selected item
POWER SOURCE	(2) 3V BATTERY OR HIGHER
LCD BIAS	(2) 1/2 BIAS

2-1-2-3. 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.



MASK OPTION table :

Mask Option name	Selected item
POWER SOURCE	(2) 3V BATTERY OR HIGHER
LCD BIAS	(1) 1/3 BIAS

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

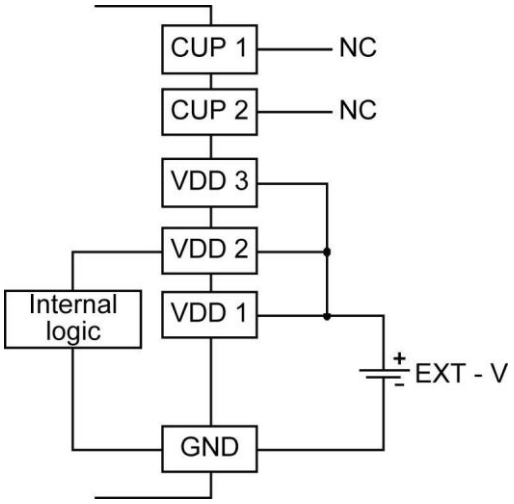
Note-2: At the initial clear mode, the backup flag (BCF) is set. When the backup flag is set, the oscillator circuit's invertor size becomes large and the oscillation conditions are improved, but the operating current is also increased. Therefore, the backup flag must be reset unless otherwise required.

2-1-3. EXTV POWER SUPPLY

Operating voltage range: 3.6V ~ 5.4V.

For different LCD bias application, the connection diagrams are shown below:

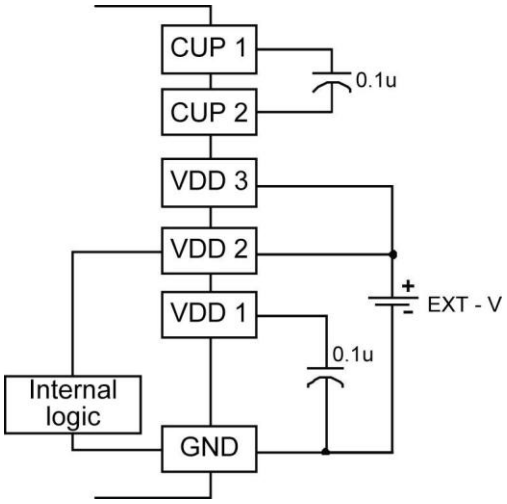
2-1-3-1. NO BIAS AT EXT-V BATTERY POWER SUPPLY



MASK OPTION table :

Mask Option name	Selected item
POWER SOURCE	(1) EXT-V
LCD BIAS	(3) NO BIAS

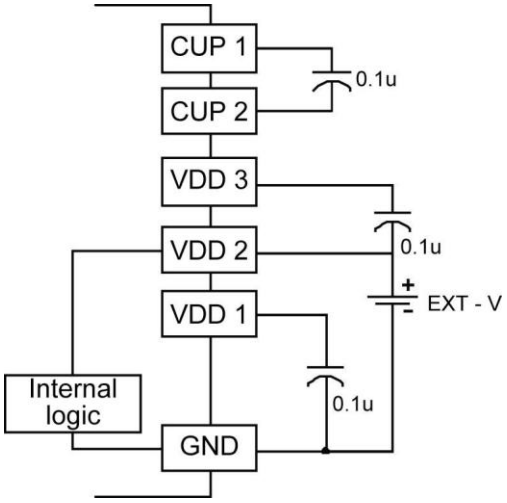
2-1-3-2. 1/2 BIAS AT EXT-V POWER SUPPLY



MASK OPTION table :

Mask Option name	Selected item
POWER SOURCE	(1) EXT-V
LCD BIAS	(2) 1/2 BIAS

2-1-3-3. 1/3 BIAS AT EXT-V POWER SUPPLY



MASK OPTION table :

Mask Option name	Selected item
POWER SOURCE	(1) EXT-V
LCD BIAS	(1) 1/3 BIAS

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

Note-2: At the initial clear mode the backup flag (BCF) is reset.

Note-3: At the backup flag set mode the operating current is increased. Therefore, the backup flag must be reset unless otherwise required.

2-2 SYSTEM CLOCK

XT clock (slow clock oscillator) and CF clock (fast clock oscillator) compose the clock oscillation circuitry 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 usage of LCD driver, frequency generator … etc.

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

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

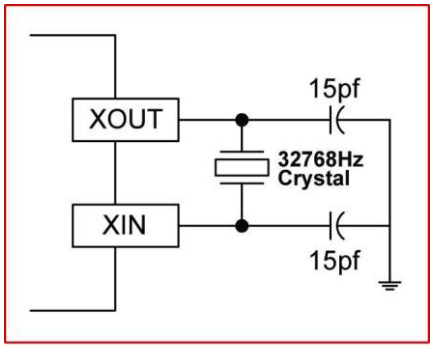
2-2-1. CONNECTION DIAGRAM OF SLOW CLOCK OSCILLATOR (XT CLOCK)

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, or it will be active all the time after the initial reset. In stop mode, this oscillator will be stopped.

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

MASK OPTION table :

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



X'tal

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

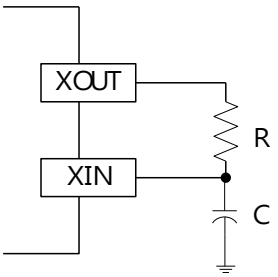
The following table shows the power consumption of Crystal oscillator in different conditions:

	Ag power option	Li power option	EXT-V option
BCF=1	Increased	Increased	Increased
BCF=0	Normal	Normal	Increased
Initial reset	Increased	Increased	Increased
After reset	Increased	Increased	Increased

2-2-1-2. External RC Oscillator

MASK OPTION table:

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



(2) RC

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

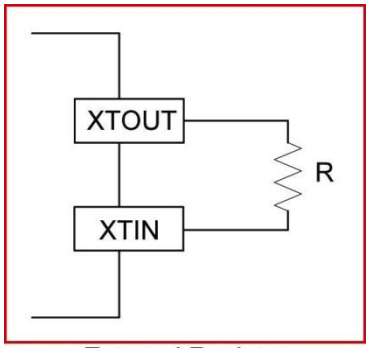
The CF clock is a multiple type oscillator (mask option), which provides a faster clock source to system. In single clock operation (fast only), this oscillator will provide the clock to the system clock generator, pre-divider, timer, I/O port chattering prevention clock and LCD circuitry. In dual clock operation, CF clock provides the clock to 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 still come from XT clock. Halt mode, stop mode or SLOW instruction execution will stop this oscillator and the system clock (BCLK) will be switched to XT clock.

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

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

This kind of oscillator could only be used in "FAST only" option, the fast clock source of dual clock mode can't use this oscillator. When this oscillator is used, the frequency option of the RC oscillator with internal RC is not cared.



External Resistor

MASK OPTION table :

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

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

Two kinds of the frequencies could be selected in this oscillation mode, one is 250KHz and the other is 500KHz. This kind of oscillator could be used in "FAST only" or "DUAL clock" options.

MASK OPTION table:

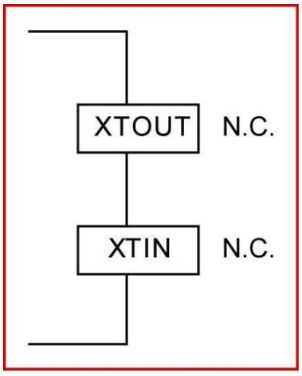
Mask Option name	Selected item
CLOCK SOURCE	(1) FAST ONLY & USE INTERNAL RESISTOR or (4)DUAL

For 250KHz output frequency:

Mask Option name	Selected item
FAST CLOCK OSC TYPE FOR FAST ONLY OR DUAL	(1) INTERNAL RESISTOR FOR 250KHz

For 500KHz output frequency:

Mask Option name	Selected item
FAST CLOCK OSC TYPE FOR FAST ONLY OR DUAL	(2) INTERNAL RESISTOR FOR 500KHz



Internal RC

FREQUENCY RANGE OF INTERNAL RC OSCILLATOR:

Option Mode	BAK	Min.	Typ.	Max.
250KHz	1.5V	300KHz	350KHz	400KHz
	3.0V	250KHz	300KHz	350KHz
500KHz	1.5V	550KHz	650KHz	750KHz
	3.0V	450KHz	550KHz	650KHz

2-2-3. COMBINATION OF THE CLOCK SOURCES

There are three types of combination of the clock sources that can be selected by mask option:

2-2-3-1. DUAL CLOCK

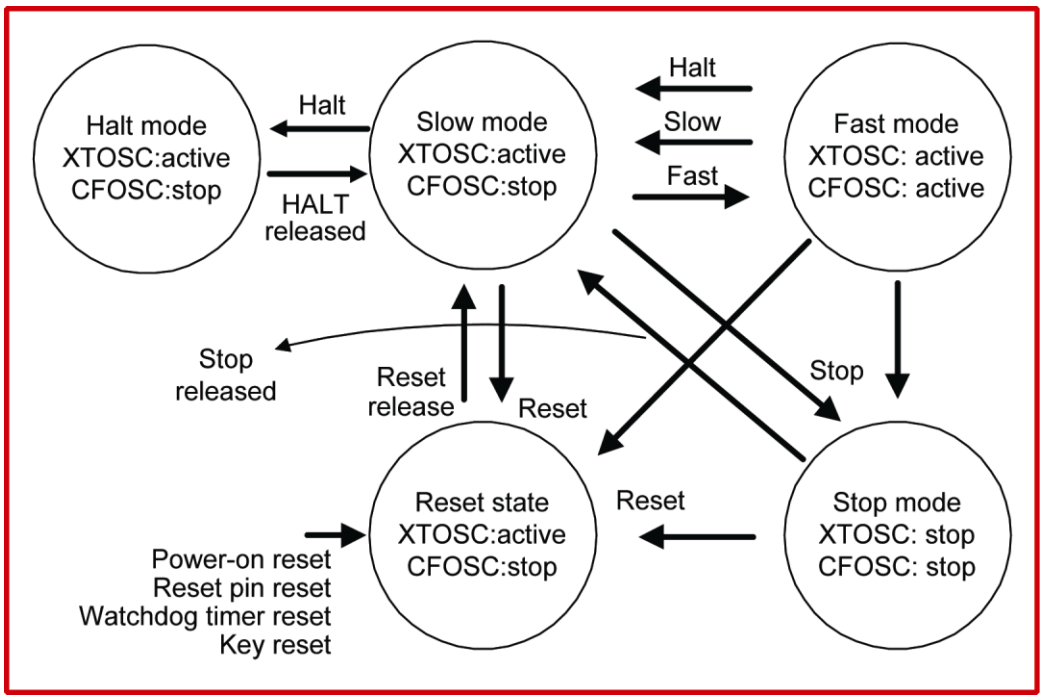
MASK OPTION table:

Mask Option name	Selected item
CLOCK SOURCE	(4) DUAL

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

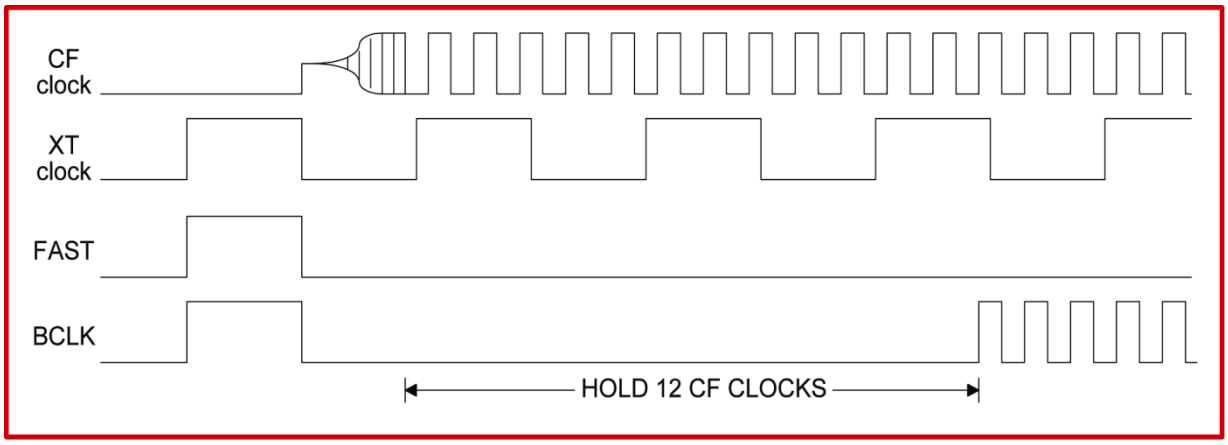
When this option is selected by mask option, the clock source (BCLK) of system clock generator will switch between XT clock and CF clock according to the user's program. When the halt and stop instructions are executed, the clock source (BCLK) will switch to XT clock automatically.

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



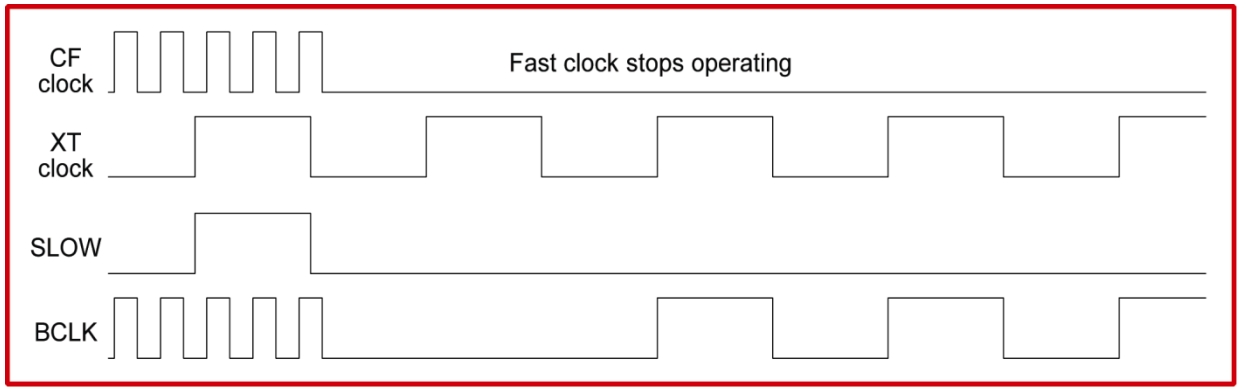
State Diagram of Dual Clock Option was shown on above figure 2-2-3-1-(1).

After executing FAST instruction, the system clock generator will hold 12 CF clocks after the CF clock oscillator starts up and then switches CF clock to BCLK. This will prevent the incorrect clock from delivering to the system clock in the start-up duration of the fast clock oscillator.



The figure 2-2-3-1-(2) Shows the System Clock Switches from Slow to Fast

After executing SLOW instruction, the system clock generator will hold 2 XT clocks and then switches XT clock to BCLK.



The figure 2-2-3-1-(3) Shows the System Clock Switches 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 & USE INTERANL RESISTOR or (2) FAST ONLY & USE EXTERANL RESISTOR

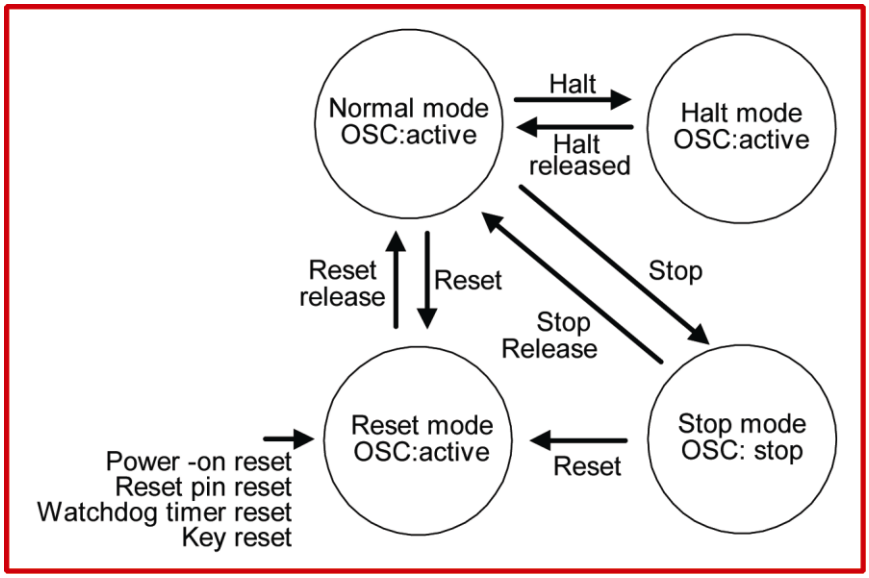
- For slow clock oscillator only

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

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

Either XT or CF clock may be selected by mask option in this mode. The FAST and SLOW 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 could ensure the Crystal oscillator would start up in a better condition.

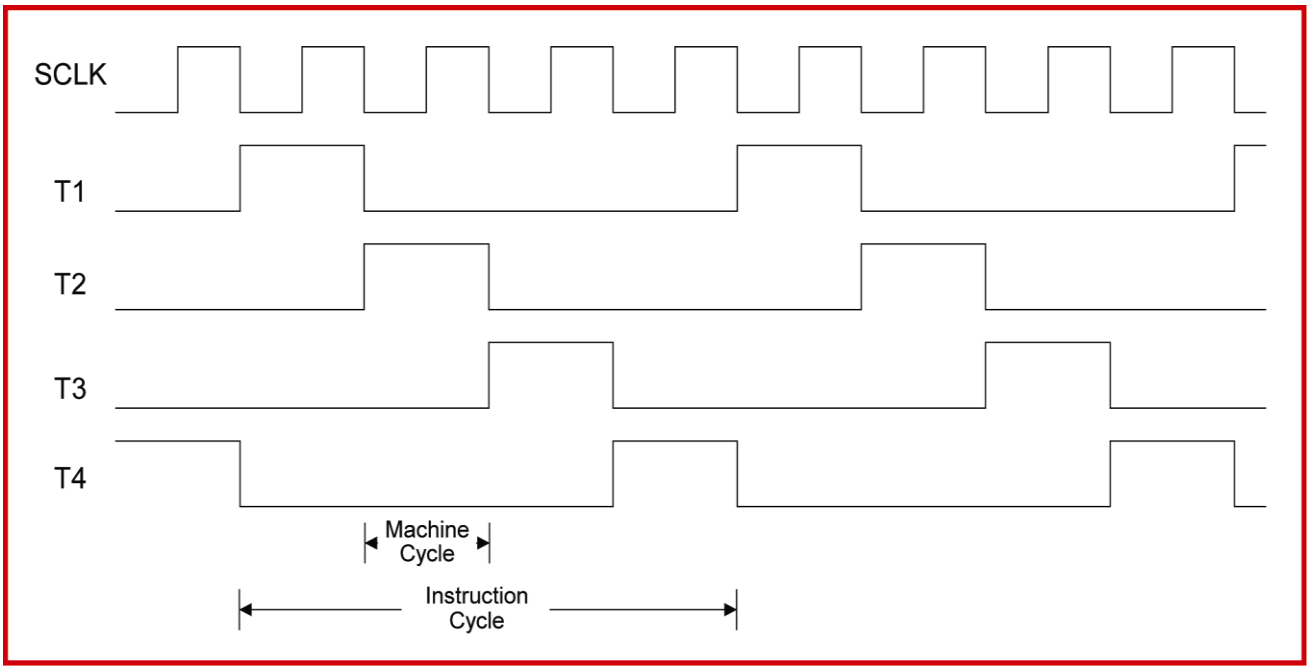


The figure 2-2-3-2-(1) shows the State Diagram of Single Clock Option

2-2-4. SYSTEM CLOCK GENERATOR

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

The basic system clock is shown below:



2-3 PROGRAM COUNTER (PC)

This is an 11-bit counter, which addresses the program memory (ROM) up to 1152 addresses.

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

  PC ← PC + 1

- When executing JMP instruction, subroutine call instruction (CALL), interrupt service routine or reset occurs, the program counter (PC) loads the specified address corresponding to ***Table-2-3***.

  *PC ← specified address shows in **Table-2-3**.*

- When executing a jump instruction except JMP and CALL, the program counter (PC) loads the specified address in the operand of instruction.

  PC ← specified address in operand

- Return instruction (RTS)

  PC ← content of stack specified by the stack pointer

  Stack pointer ← stack pointer - 1

Table-2-3

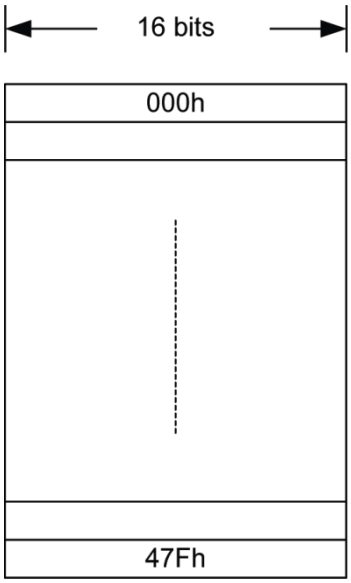
	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	0	0	0	0	0	0	0	0	0	0	0
Interrupt 2 (INT pin)	0	0	0	0	0	0	1	0	0	0	0
Interrupt 0 (input port C)	0	0	0	0	0	0	1	0	1	0	0
Interrupt 1 (timer 1 interrupt)	0	0	0	0	0	0	1	1	0	0	0
Interrupt 3 (pre-divider interrupt)	0	0	0	0	0	0	1	1	1	0	0
Jump instruction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

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

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

2-4 PROGRAM/TABLE MEMORY

The built-in mask ROM is organized with 1152 x 16 bits.



Both instruction ROM (PROM) and table ROM (TROM) shares this memory space together. The partition formula for PROM and TROM is shown below:

Instruction ROM memory space = (128 * N) words,

Table ROM memory space = 256(9 - N) bytes (N = 1 ~ 9).

Note:

The data width of table ROM is 8-bit

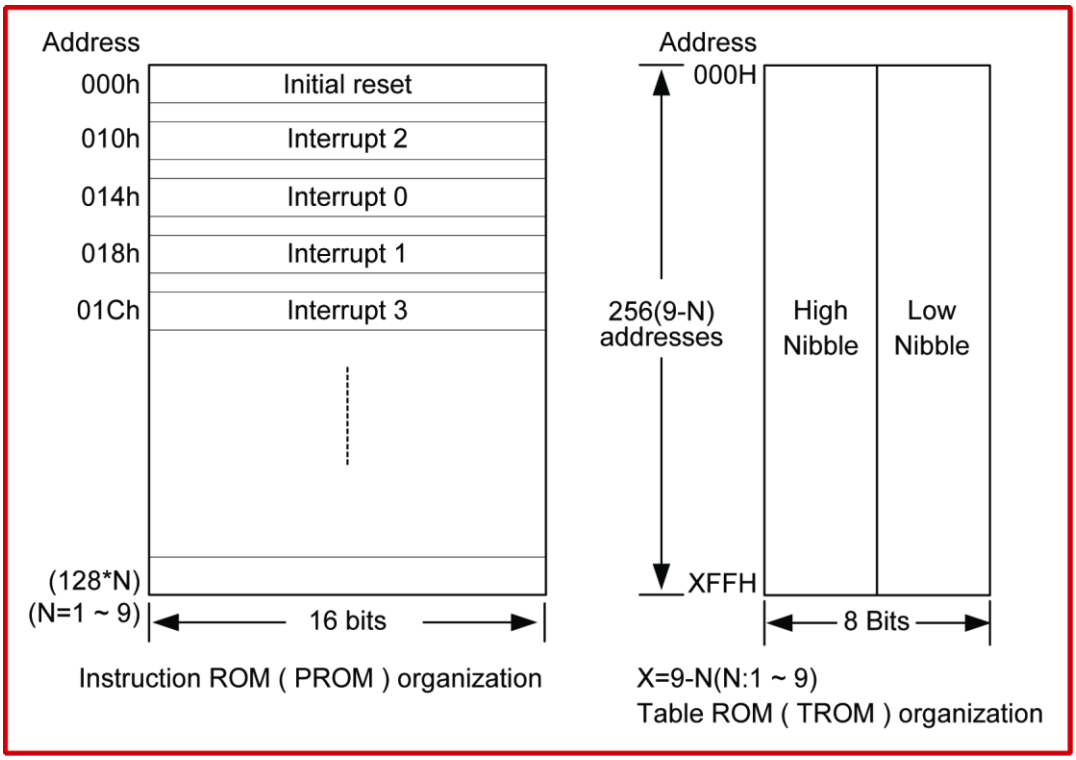
The partition of memory space is defined by mask option. The table is shown below:

MASK OPTION table:

Mask Option name	Selected item	Instruction ROM memory space (Words)	Table ROM memory space (Bytes)
INSTRUCTION ROM <-> TABLE ROM	1 (N=1)	128	2048
INSTRUCTION ROM <-> TABLE ROM	2 (N=2)	256	1792
INSTRUCTION ROM <-> TABLE ROM	3 (N=3)	384	1536
INSTRUCTION ROM <-> TABLE ROM	4 (N=4)	512	1280
INSTRUCTION ROM <-> TABLE ROM	5 (N=5)	640	1024
INSTRUCTION ROM <-> TABLE ROM	6 (N=6)	768	768
INSTRUCTION ROM <-> TABLE ROM	7 (N=7)	896	512
INSTRUCTION ROM <-> TABLE ROM	8 (N=8)	1024	256
INSTRUCTION ROM <-> TABLE ROM	9 (N=9)	1152	0

2-4-1. INSTRUCTION ROM (PROM)

There are some special locations that serve as the interrupt service routines, such as reset address (000H), interrupt 0 address (014H), interrupt 1 address (018H), interrupt 2 address (010H), interrupt 3 address (01CH), in the program memory.



The figure 2-4-1 shows the Organization of ROM

2-4-2. TABLE ROM (TROM)

The table ROM is organized with 256(9-N) x 8 bits that shared the memory space with 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 could be 8 bits (256(9-N) x 8 bits) or 4 bits (512(9-N) x 4 bits), which depends on the different usage. Refer to the explanation of instruction chapter.

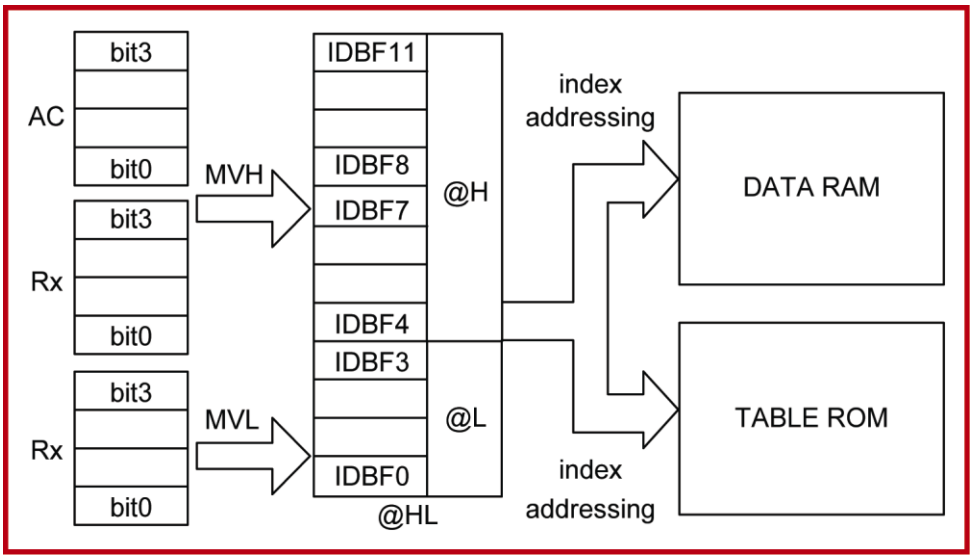
2-5 INDEX ADDRESS REGISTER (@HL)

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 and MVL instructions. Executed MVL instruction will load the content of specified data memory to the lower nibble of the index register (@L). In the same manner, executed MVH instructions may load the contents of the data RAM (Rx) and AC into the higher nibble of the register @H.

@L is a 4-bit register and @H is an 8-bit register.

@H register (8-bit)								@L register (4-bit)			
Bit7	Bit6	Bit5	Bit4	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.



The figure 2-5 shows the diagram of the index address register

2-6 STACK REGISTER (STACK)

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

The stack register is organized using 11 bits by 8 levels but with no overflow flag; hence only 8 levels of subroutine call or interrupt are allowed (If the stacks are full, and either interrupt occurs or subroutine call executes, the first level will be overwritten). Once the subroutine call or interrupt causes the stack register (STACK) overflow, the stack pointer will return to 0 and the content of the level 0 stack will be overwritten by the PC value.

The contents of the stack register (STACK) are returned sequentially to the program counter (PC) during execution of the RTS instruction. Once the RTS instruction causes the stack register (STACK) underflow, the stack pointer will return to level 7 and the content of the level 7 stack will be restored to the program counter.

The following *figure 2-6* shows the diagram of the stack.

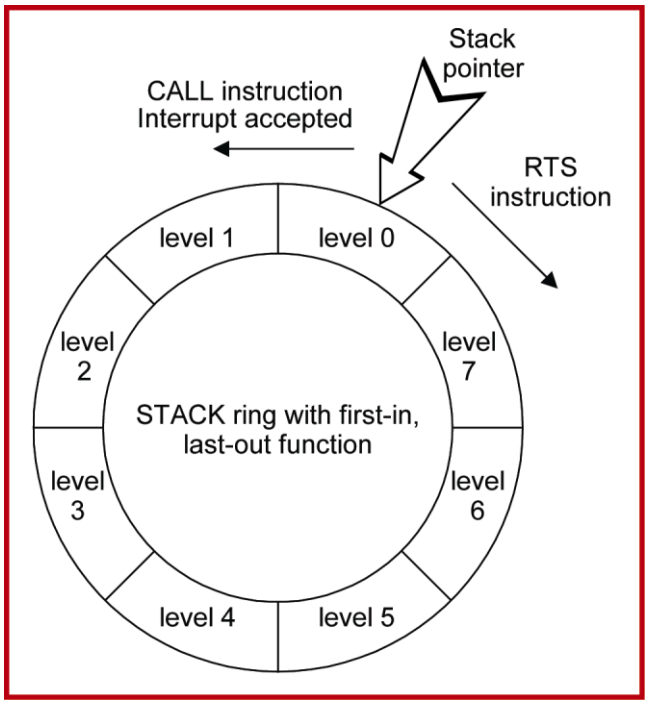


Figure 2-6

2-7 DATA MEMORY (RAM)

The static RAM is organized with 96 addresses x 4 bits and is used to store data. The address range of data memory is from 00h to 7Fh, but addresses between 40h to 6Fh are not reachable.

The data memory may be accessed using two methods:

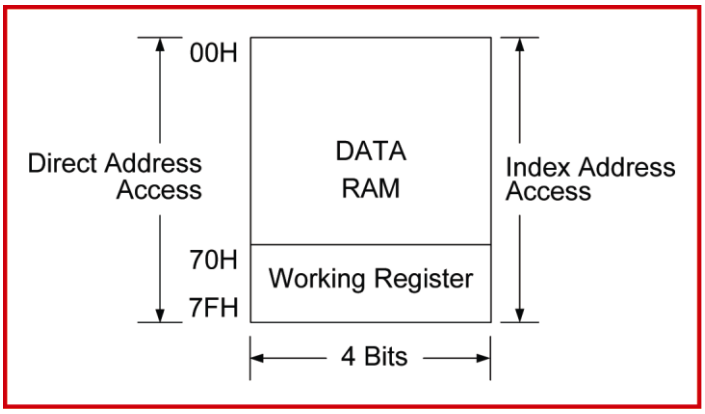
1. Direct addressing mode

The address of the data memory is specified by the instruction and the addressing range is from 00H to 7FH. (Addresses between 40h to 6Fh are not reachable)

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. (Addresses between 40h to 6Fh are not reachable)

The 16 specified addresses (70H to 7FH) in the direct addressing memory are also used as 16 working registers. *The function of working register will be described in detail in section 2-6.*



The figure 2-7 shows the Data Memory (RAM) and Working Register Organization

2-8 WORKING REGISTER (WR)

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

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

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

MWR Rx, Ry; MRW Ry, Rx

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

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 can be performed.

2-10 ALU (Arithmetic and Logic Unit)

This circuitry performs arithmetic and logic operation. 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 TM8724. When the content 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 processing, 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 the data from hexadecimal to decimal format after any addition operation. The conversion rules are shown in the following *table 2-11-(1)* and illustrated in *example 1*.

Table 2-11-(1)

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

Example 1:

```
LDS     10h, 9   ; Load immediate data”9”to data memory address 10H.
LDS     11h, 1   ; Load 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      ; Convert 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 2-11-(2)* and illustrated in *Example 2*.

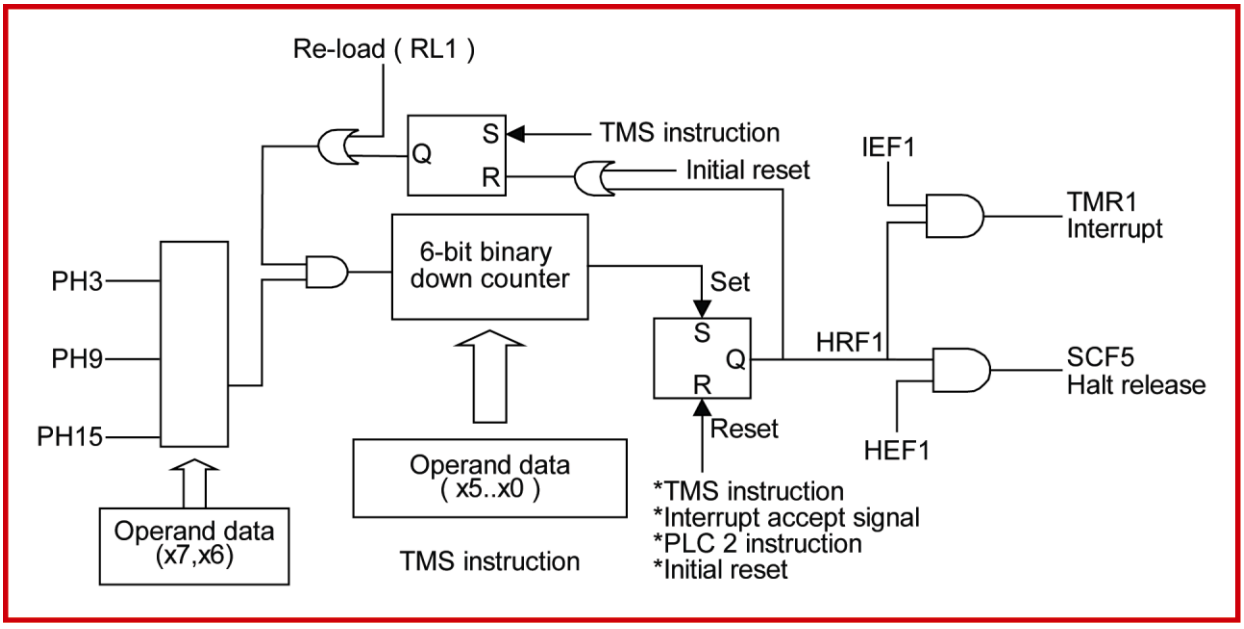
Table 2-11-(2)

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

Example 2:

```
LDS    10h, 1        ; Load immediate data"1"to the data memory address 10H.
LDS    11h, 2        ; Load immediate data"2"to the data memory address 11H and AC.
SF     1h            ; Set 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           ; Convert 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)



The figure 2-12 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 TMS or TMSX instruction.

Once the TMR1 counts down to 3Fh, it generates an underflow signal to set the halt release request flag1 (HRF1) to 1 and then stop to count down.

When HRF1 = 1, and the TMR1 interrupt enable flag (IEF1) = 1, the interrupt is generated.
When HRF1 = 1, if the IEF1 = 0 and the TMR1 halt release enable (HEF1) = 1, program will escapes 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 occurred, the clock source of TMR1 will still keep the previous selection.

The following *table 2-12-1-(1)* shows the definition of each bit in TMR1 instructions

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

The following *table 2-12-1-(2)* shows the clock source setting for TMR1

X7	X6	clock source
0	0	PH9
0	1	PH3
1	0	PH15
1	1	FREQ

Notes:

1. When the TMR1 clock is PH3
   TMR1 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the TMR1 clock is PH9
   TMR1 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the TMR1 clock is PH15
   TMR1 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
   Set value: Decimal number of timer set value
   error: The tolerance of set value, 0 < error <1.
   fosc: Input of the predivider
   PH3: The 3rd stage output of the predivider
   PH9: The 9th stage output of the predivider
   PH15: The 15th stage output of the predivider
4. When the TMR1 clock is FREQ
   TMR1 set time = (Set value + error) * 1/FREQ (KHz) (ms).

2-12-2. RE-LOAD OPERATION

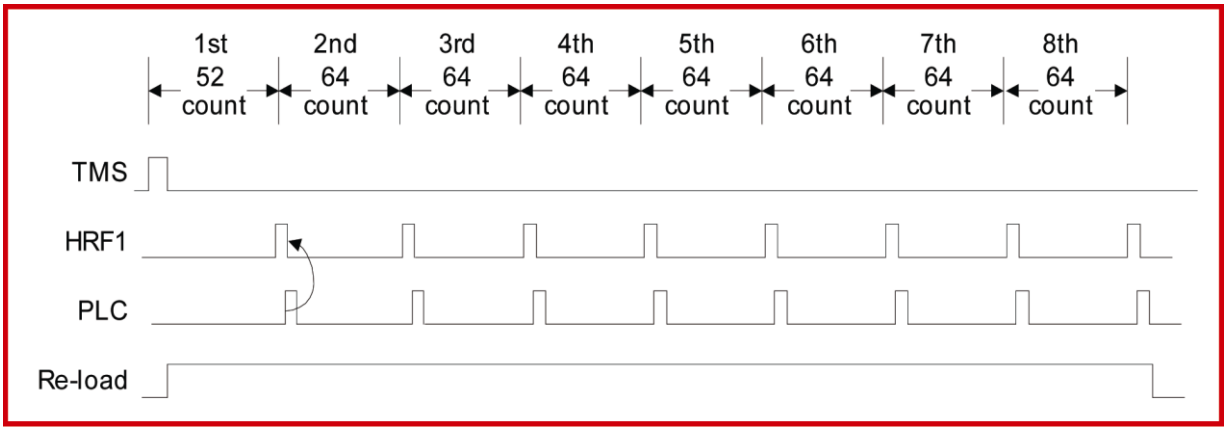
TMR1 provides the re-load function which can extend any time interval greater than 3Fh. The SF 80h instruction enables the re-load function and RF 80h instruction disables it.

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 the TMS or TMSX instruction 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 by PLC instruction. After halt release or interrupt occurs 8 times, disable the re-load function and the counting is completed.



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

```
        LDS    0, 0          ;initiate the underflow counting register
        PLC    2
        SHE    2             ;enable the HALT release caused by TMR1
        TMSX   34h           ;initiate the TMR1 value (52) and clock source is ϕ9
        SF     80h           ;enable the re-load function
RE_LOAD:
        HALT
        INC*   0             ;increase the underflow counter
```

```
        PLC     2           ;clear HRF1
        JB3     END_TM1     ;if the TMR1 underflow counter is equal to 8, exit subroutine
        JMP     RE_LOAD
END_TM1:
        RF      80h         ;disable the re-load function
```

2-13 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 2-13* shows the configuration of the start condition flags for TM8724.

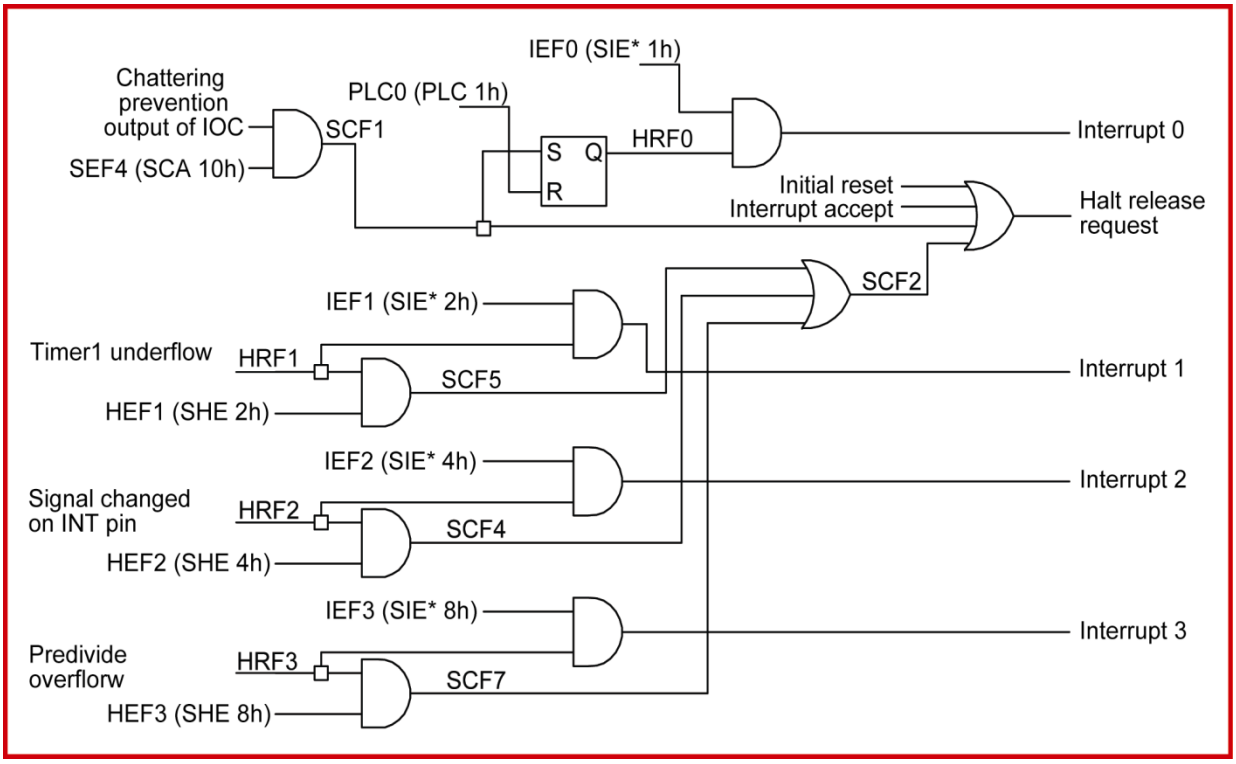


Figure 2-13

2-13-1. STATUS REGISTER 1 (STS1)

Status register 1 (STS1) consists of 2 flags:

1. Carry flag (CF)

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

2. Zero flag (Z)

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

2-13-2. 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 of 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 2-13-2* shows the bit pattern of each flag in status register 2 (STS2).

Bit 3	Bit 2	Bit 1	Bit 0
NA	Start condition flag 2(SCF2)	Start condition flag 1(SCF1)	Backup flag (BCF)
NA	Halt release caused by SCF4,5,7	Halt release caused by the IOC port	The back up mode status
NA	Read only	Read only	Read only

Table 2-13-2

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 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, 7 are set to 1, SCF2 will also be set to 1 simultaneously. When all of the flags in SCF4, 5, 7 are clear, start condition flag 2 (SCF2) is reset to 0.

Note: If start condition flag is set to 1, the program will not be able to enter halt mode.

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

2-13-3. STATUS REGISTER 3 (STS3)

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

Status register 3 (STS3) consists of 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) or 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) or 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) or the SHE instruction must be used to reset the halt release enable flag 3 (HEF3).

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

   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 2-13-3 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

Table 2-13-3

2-13-4. STATUS REGISTER 4 (STS4)

Status register 4 (STS4) consists of 2 flags:

1. System clock selection flag (CSF)

   The system clock selection flag (CSF) indicates which clock source of the system clock generator (SCG) is used. Executing SLOW instruction will change the clock source (BCLK) of the system clock generator (SCG) to the slow speed oscillator (XT clock), and the system clock selection flag (CSF) is reset to 0. Executing FAST instruction will change the clock source (BCLK) of the system clock generator (SCG) to the fast speed oscillator (CF clock), and the system clock selection flag (CSF) is set 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.

   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 2-13-4 shows the Bit Pattern of Status Register 4 (STS4)

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Watchdog timer Enable flag (WDF)	System clock selection flag (CSF)
Reserved	Reserved	Read only	Read only

Table 2-13-4

2-13-5. 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, and stop release enable flag 4 (SRF4) is set beforehand.
- The signal change from the INT pin causes the halt release flag 2 (HRF2) to output and the stop release enable flag 5 (SRF5) is set beforehand.

The following *figure 2-13-5* shows the organization of start condition flag 11 (SCF 11).

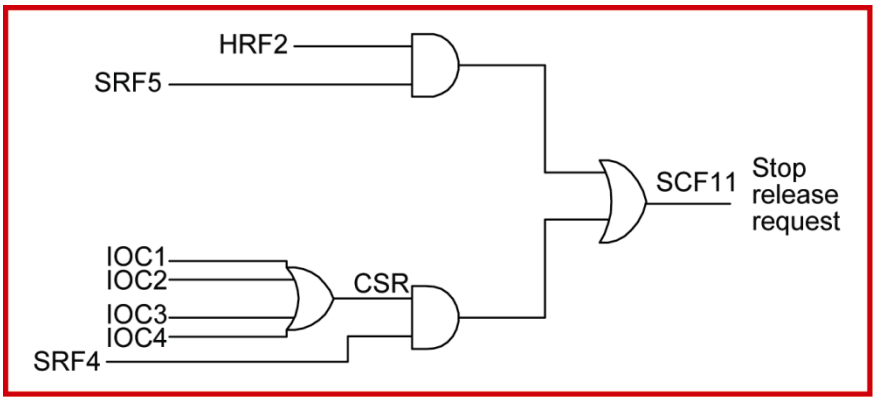


Figure 2-13-5

The stop release flags (CSR, HRF2) were specified by the stop release enable flags (SRFx) and these flags should be clear before the chip enters the stop mode. All of the pins in IOC port had to be defined as input mode and keep in 0 state before the chip enters the STOP mode, or the program can't enter the STOP mode.

Instruction SRE is used to set or reset the stop release enable flags (SRF4, 5).

The following table 2-13-5 shows the stop release request flags

	The OR-ed input mode pins of IOC port	The rising or falling edge on INT pin
Stop release request flag	CSR	HRF2
Stop release enable flag	SRF4	SRF5

Table 2-13-5

2-14 CONTROL REGISTER (CTL)

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

2-14-1. 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 defined as input mode that causes the halt mode or stop mode to be released.

Executed SCA instruction may set or reset these flags.

The following table 2-14-1 shows Bit Pattern of Control Register 1 (CTL1)

Bit 4

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

Table 2-14-1

The following *figure 2-14-1* shows the organization of control register 1 (CTL1).

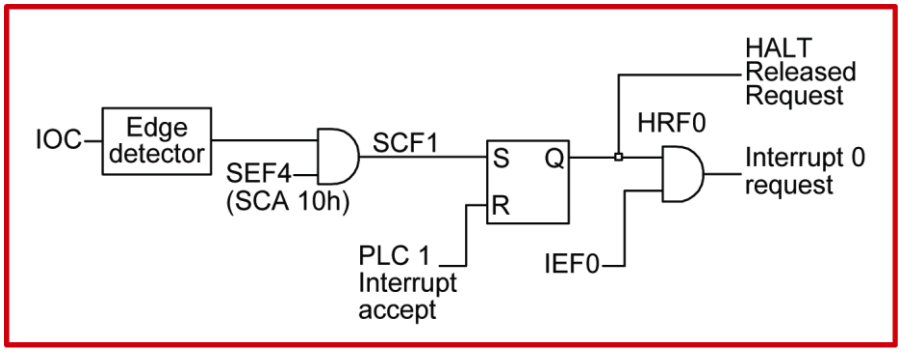


Figure 2-14-1

2-14-1-1. The Setting for Halt Mode

If the SEF4 is set to 1, the signal changed on IOC port will cause the halt mode to be released, and set SCF1 to 1. Because the input signal of IOC port were ORed, so it is necessary to keep the unchanged input signals at “0” state and only one of the input signal could change state.

2-14-1-2. The Setting for Stop Mode

If SRF4 and SEF4 are set, the stop mode will be released to set the SCF1 when a high level signal is applied to one of the input mode pins of IOC port and the other pins stay in ”0” state. After the stop mode is released, TM8724 enters the halt condition. The high level signal must hold for a while to cause the chattering prevention circuitry of IOC port to detect this signal and then set SCF1 to release the halt mode, or the chip will return to the stop mode again.

2-14-1-3. 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 port will deliver the SCF1 when SEF4 has been set to 1 by executing SCA instruction. Once the SCF1 is delivered, the halt release request flag (HRF0) will be set to 1. In this case, if the interrupt enable flag 0 (IEF0) is set to 1 by executing SIE instruction, the interrupt request flag 0 (interrupt 0) will be delivered to interrupt the program.

If the interrupt 0 is accepted by SEF4 and IEF0, the interrupt 0 request to the next signal change at IOC will be inhibited. To release this mode, SCA instruction must be executed again.

2-14-2. CONTROL REGISTER 2 (CTL2)

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

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 3 (HEF3) is set, an overflow signal from the pre-divider causes the halt mode to be released. In the same manner, when HEF1 to HEF2 are set to 1, the following conditions will cause the halt mode to be released respectively: an underflow signal from TMR1 and the signal change at the INT pin.

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.

2-14-3. CONTROL REGISTER 3 (CTL3)

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

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

Interrupt enable flag	IEF3	IEF2	IEF1	IEF0
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)	Enable the interrupt request caused by IOC port signal to be changed (HRF0)
Interrupt flag	Interrupt 3	Interrupt 2	Interrupt 1	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.

2-14-4. CONTROL REGISTER 4 (CTL4)

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

The following table 2-14-4 shows the Bit Pattern of Control Register 4 (CTL4)

Stop release enable flag	SRF5	SRF4
Stop release request flag	Enable the stop release request caused by signal change on INT pin (HRF2)	Enable the stop release request caused by signal change on IOC

Table 2-14-4

When the stop release enable flag 5 (SRF5) is set to 1, the signal changed on INT pin causes the stop mode to be released. In the same manner, when SRF4 is set to 1, the input signal change at the input mode pins of IOC port and causes the stop mode to be released.

Example:

This example illustrates the stop mode released by port IOC and INT pin. Assume all of the pins in IOD and IOC have been defined as input mode.

```
PLC     25h         ; Reset the HRF0 and HRF2.
SHE     24h         ; HEF2 is set so that the signal change at INT
                    ; causes start condition flag 4 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                ; Enter the stop mode.
  ..............    ;STOP release
MSC     10h         ; Check 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.
```

2-15 HALT FUNCTION

The halt function is provided to minimize the current dissipation of the TM8724 when LCD is operating. During the halt mode, the program memory (ROM) is not in operation and only the oscillator circuit, pre-divider circuit, sound circuit, I/O port chattering prevention circuit, and LCD driver output circuit are in operation. (If the timer has started operating, the timer counter still operates in the halt mode). After the HALT instruction is executed and no halt release signal (SCF1, HRF1 ~ 3) is delivered, the CPU enters the halt mode.

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

(1). An interrupt is accepted.

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

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

(2). The signal change specified by the SCA instruction is applied to port IOC (SCF1).

(3). The halt release condition specified by the SHE instruction is met (HRF1 ~ HRF3).

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

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

2-16 HEAVY LOAD FUNCTION

When heavy loading (lamp light-up, motor start, etc.) causes a temporary voltage drop on supply voltage, the heavy loading function (set BCF=1) prevents TM8724 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. This will improve TM8724's noise immunity. Please pay attention to that when TM8724's power source mask option is set as EXT-V or 3V, internal logic circuit's power supply won't be switched to VDD1 when BCF is reset as 0.

For shorten the start-up time of 32.768KHz Crystal oscillator, TM8724 will set the BCF to 1 during reset cycle and reset BCF to 0 after reset cycle automatically in Ag mode option.

Table 2-16-(1) The back-up flag status in different conditions

	Ag 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	

For low power consumption application, reset BCF to 0 is necessary; the 32.768KHz Crystal oscillator operates with a normal buffer only. When the heavy load function is performed, the current dissipation will increase.

Table 2-16-(2) Ag power option:

	Initial reset	After reset	STOP mode	SF 2	RF 2
BCF	1	1	1*	1	0
Internal logic	VDD1	VDD1	VDD1	VDD1	VDD1
Peripheral logic	VDD1	VDD1	VDD1	VDD1	VDD1

Table 2-16-(3) Li power option:

	Initial reset	After reset	STOP mode	SF 2	RF 2
BCF	1	1	1*	1	0
Internal	VDD2	VDD2	VDD2	VDD2	VDD2
Peripheral	VDD2	VDD2	VDD2	VDD2	VDD2

Table 2-16-(4) EXT-V power option:

	Initial reset	After reset	STOP mode	SF 2	RF 2
BCF	0	0	1*	1	1
Internal	VDD2	VDD2	VDD2	VDD2	VDD2
Peripheral	VDD2	VDD2	VDD2	VDD2	VDD2

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.

2-17 STOP FUNCTION (STOP)

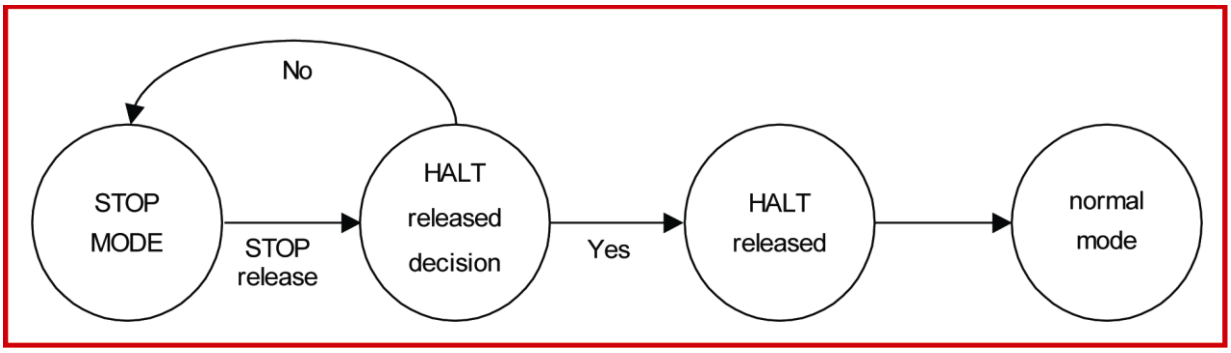
The stop function is another solution to minimize the current dissipation for TM8724. In stop mode, all of functions in TM8724 are held including oscillators. All of the LCD corresponding signals (COM and Segment) will output "L" level. In this mode, TM8724 does not dissipate any power in the stop mode. Because the stop mode will set the BCF flag to 1 automatically, it is recommended to reset the BCF flag after releasing the stop mode in order to reduce power consumption. Before the stop instruction is executed, all of the signals on the pins defined as input mode of IOC port must be in the "L" state, and no stop release signal (SRFn) should be delivered. The CPU will then enter the stop mode.

The following conditions cause the stop mode to be released.

- One of the signals on the input mode pin of 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.(INT pin is exclusive)

When the TM8724 is released from the stop mode, the TM8724 enters the halt mode immediately and will process the halt release procedure. If the "H" signal on the IOC port does not hold long enough to set the SCF1, once the signal on the IOC port returns to "L", the TM8724 will enter the stop mode immediately. The backup flag (BCF) will be set to 1 automatically after the program enters the stop mode.

The following *figure 2-17* shows the stop release procedure:



The Figure 2-17 shows the stop release state machine

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

- Specify the stop release conditions by the SRE instruction.
- Specify the halt release conditions corresponding to the stop release conditions if needed.
- Specify the interrupt conditions corresponding to the stop release conditions if needed.

When the stop mode is released by an interrupt request, the TM8724 will enter the halt mode immediately. While the interrupt is accepted, the halt mode will be released by the interrupt

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

2-18 BACK UP FUNCTION

TM8724 provide a back up mode to avoid system malfunction when heavy loading occurred, such as buzzer is active, LED is lighting… etc. Since the heavy loading will cause a large voltage drop on the supply voltage, and the system will be malfunction in this condition. Once the program enter back up mode (BCF = 1), 32.768KHz Crystal oscillator will operate in a large driver condition and internal logic function operates with higher supply voltage. TM8724 will get more power supply noise margin while back up mode is active but also increases more power consumption. The back up flag (BCF) indicated the status of back up function. BCF flag could be set or reset by executing SF or RF instruction respectively. The back up function has different performance corresponding to different power mode option, shown in the following *table 2-18*.

Table 2-18.

(1) 1.5V battery mode:

TM8724 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)	
TM8724 status	BCF = 0	BCF = 1
32.768KHz Crystal Oscillator	Small driver	Large driver
Internal operating voltage	VDD1	VDD1

(2) 3V battery or highter mode:

TM8724 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)

TM8724 status	BCF = 0	BCF = 1
32.768KHz Crystal Oscillator	Small driver	Large driver
Internal operating voltage	VDD2	VDD2

(3) Ext-V power mode:

TM8724 status	BCF flag status
Initial reset cycle	BCF = 0 (hardware controlled)
After initial reset cycle	BCF = 0 (hardware controlled)
Executing SF 2h instruction	BCF = 1
Executing RF 2h instruction	BCF = 0
HALT mode	Previous state
STOP mode	BCF = 1 (hardware controlled)

TM8724 status	BCF = 0	BCF = 1
32.768KHz Crystal Oscillator	Large driver	Large driver
Internal operating voltage	VDD2	VDD2

Note:

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

3. Chapter 3 Control Function

3-1 INTERRUPT FUNCTION

There are 4 interrupt resources: 2 external interrupt factors and 2 internal interrupt factors. When an interrupt is accepted, the program in execution is suspended temporarily and the corresponding interrupt service routine specified by a fix address in the program memory (ROM) is called.

The following *table 3-1* shows the flag and service of each interrupt:

Table 3-1. Interrupt information

Interrupt source	INT pin	IOC port	TMR1 underflow	Pre-divider Overflow
Interrupt vector	010H	014H	018H	01CH
Interrupt enable flag	IEF2	IEF0	IEF1	IEF3
Interrupt priority	6^{th}	5^{th}	2^{nd}	1^{st}
Interrupt request flag	Interrupt 2	Interrupt 0	Interrupt 1	Interrupt 3

The following *figure 3-1* shows the Interrupt Control Circuit

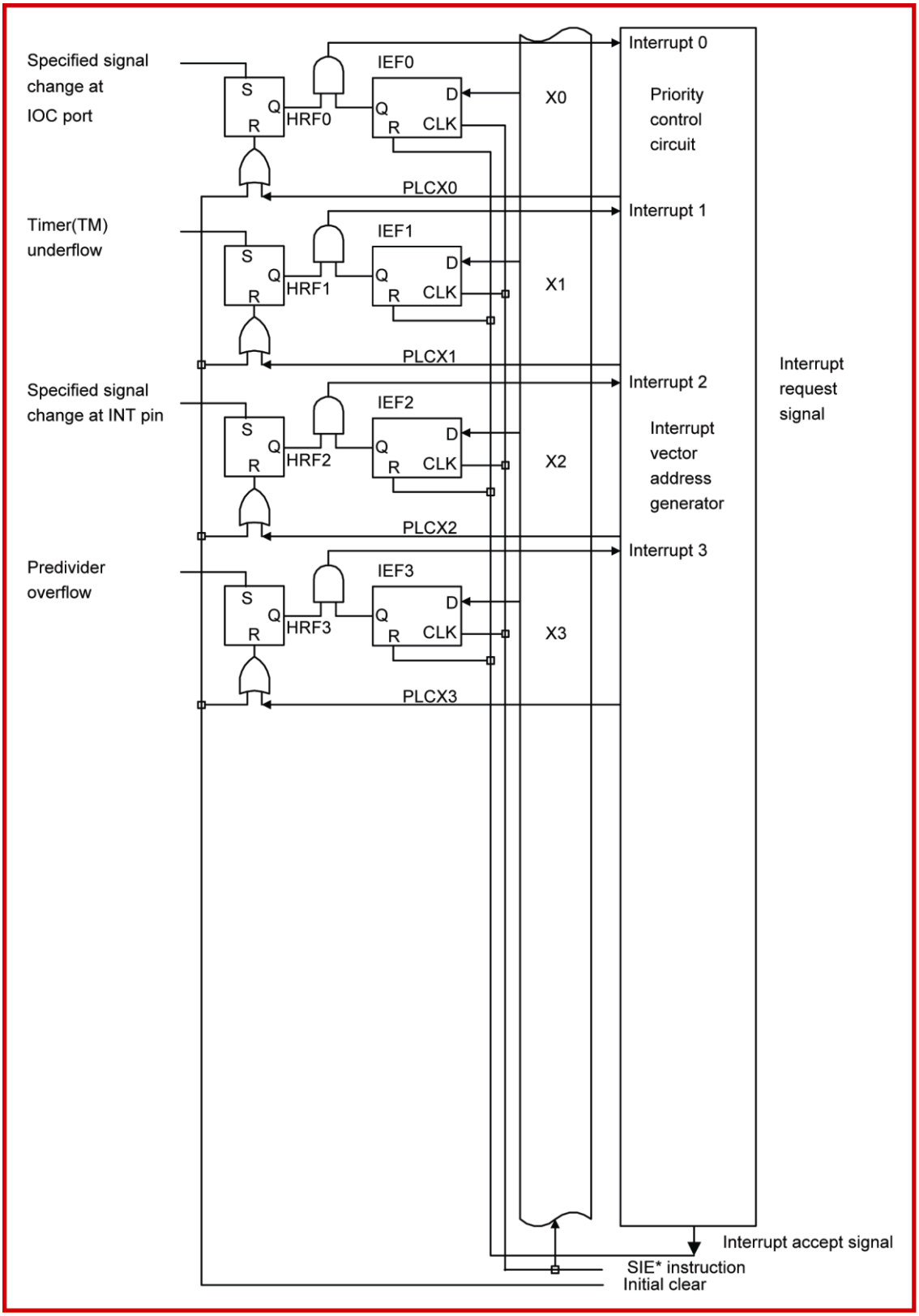


Figure 3-1

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 and IOC ports.

3-1-1-1-1. External INT pin interrupt request

By using 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 signal on the INT pin change that matches the mask option 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.

3-1-1-1-2. I/O port IOC interrupt request.

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

3-1-1-2. Internal interrupt factor

The internal interrupt factor involves the use of timer 1 (TMR1) and the pre-divider.

3-1-1-2-1. Timer1 (TMR1) interrupt request

An interrupt request signal (HRF1) is delivered when timer1 (TMR1) underflows. In this case, if the interrupt enable flag 1 (IEF1) is set, interrupt 1 is accepted and the instruction at address 18H is executed automatically.

3-1-1-2-2. Pre-divider interrupt request

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

3-1-2. INTERRUPT PRIORITY

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

Example:

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

```
PLC 0Fh          ;Clear all of the HRF flags
SCA 10h          ;enable the interrupt request of IOC
SIE*0Fh          ;enable all interrupt requests
                 ;all interrupts are requested simultaneously.
                 ;Interrupt caused by the predivider overflow occurs, and interrupt  service is concluded.
SIE*07h          ;Enable the interrupt request (except  the predivider).
                 ;Interrupt caused by the TM1 underflow occurs, and interrupt service is concluded.
SIE*05h          ;Enable the interrupt request (except  the predivider and TMR1).
                 ;Interrupt caused by the IOC port, and interrupt service is concluded.
SIE*04h          ;Enable the interrupt request (except  the predivider, TMR1, and IOC port)
                 ;Interrupt caused by the INT pin, 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 on Section 3-1*). In this case, the CPU performs the following services automatically.

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

(2). The corresponding interrupt service routine address is loaded in the program counter (PC).

The interrupt request flag corresponding to the interrupt accepted is reset and the interrupt enable flags are all reset.

When the interrupt occurs, the TM8724 will follow the procedure below:

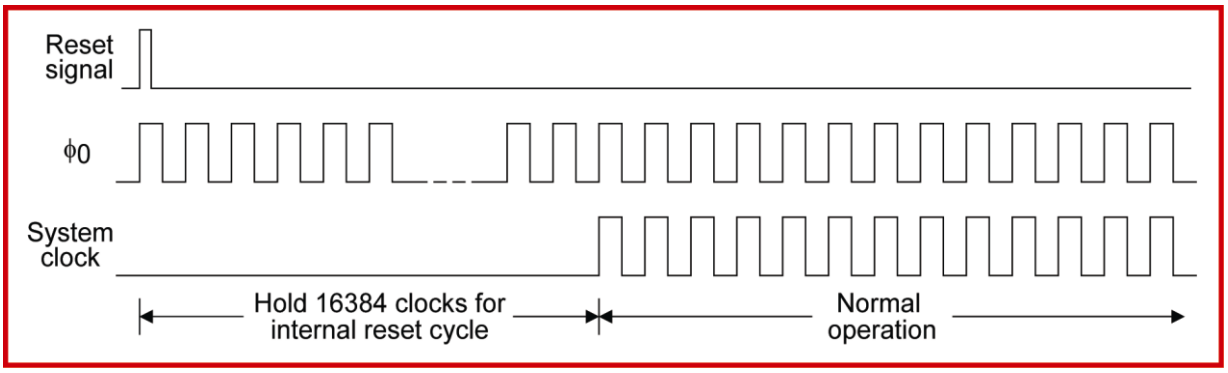
```
Instruction 1    ;In this instruction, interrupt is accepted.
NOP              ;TM8724 stores the program counter data into the STACK. At this time,
                 ;no instruction will be executed, as with NOP instruction.
Instruction A    ;The program jumps to the interrupt service routine.
Instruction B
Instruction C
.....
```

```
RTS                 ;Finishes the interrupt service routine
Instruction 1*      ;re-executes the instruction which was interrupted.
Instruction 2
```

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

3-2 RESET FUNCTION

TM8724 contains four reset sources: power-on reset, RESET pin reset, IOC port reset and watchdog timer reset. When reset signal is accepted, TM8724 will generate a time period (PH15/2) for internal reset cycle.



3-2-1. POWER ON RESET

TM8724 provides a power on reset function. If the power (VDD) is turned on or power supply drops below 0.6V, it will generate a power-on reset signal.

Power-on reset function can be disabled by mask option.

MASK OPTION table:

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

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

TM8724 also provides reset time and reset type's mask option.

Mask Option name	Selected item
RESET TIME	(1) PH15/2
RESET TIME	(2) PH12/2

Mask Option name	Selected item
RESET TYPE	(1) LEVEL
RESET TYPE	(2) PULSE

3-2-2. RESET PIN RESET

When "H" level is applied to the reset pin, the reset signal will be issued. There is a built-in pull down resistor on this pin. It is recommended to connect a capacitor (0.1uf) between RESET pin and VDD. This connection will prevent the bounce signal on RESET pin. Once a "1" signal applied on the RESET pin, TM8724 will not release the reset cycle until the signal on RESET pin returned to "0". After the signal on reset pin is cleared to 0, TM8724 begins the internal reset cycle and then release the reset status automatically.

The following *table 3-2-2* shows the initial condition of TM8724 in reset cycle.

Program counter	(PC)	Address 000H
Start condition flags 1,4,5,7	(SCF1,4,5,7)	0
Stop release enable flags 4,5	(SRF4,5)	0
Switch enable flags 4	(SEF4)	0
Halt release request flag	(HRF 0~3)	0
Halt release enable flags 1 to 3	(HEF1-3)	0
Interrupt enable flags 0 to 3	(IEF0-3)	0
Alarm output	(ALARM)	DC 0
Pull-down flags in I/OC port		1(with pull-down resistor)
Input/output ports I/OA, I/OB, I/OC	(PORT I/OA, I/OB, I/OC)	Input mode
I/OC port chattering clock	Cch	PH10*
LCD driver output		All lighted (mask option)*
Timer 1		Inactive
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 RESET

Key reset function is selected by mask option. When IOC port is in used, the '0' signal applied to all these pins that had be set as input mode in the same time, reset signal is delivered.

MASK OPTION table:

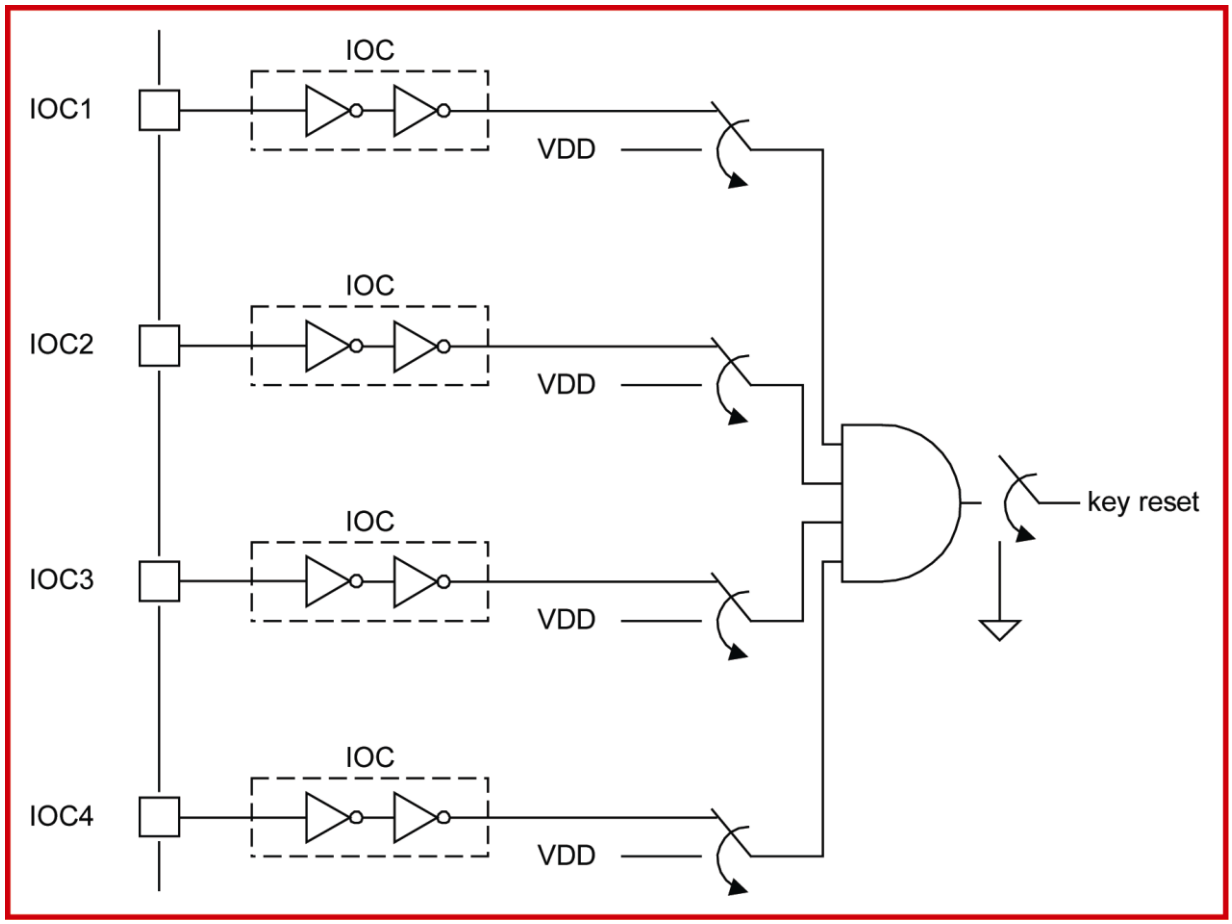
IOC pins are used as key reset:

Mask Option name	Selected item
IOC1 FOR KEY RESET	(1) USE
IOC2 FOR KEY RESET	(1) USE
IOC3 FOR KEY RESET	(1) USE
IOC4 FOR KEY RESET	(1) USE

IOC pins aren't used as key reset:

Mask Option name	Selected item
IOC1 FOR KEY RESET	(2) NO USE
IOC2 FOR KEY RESET	(2) NO USE
IOC3 FOR KEY RESET	(2) NO USE
IOC4 FOR KEY RESET	(2) NO USE

The following *figure 3-2-3* shows the key reset organization.



3-3 CLOCK GENERATOR

3-3-1. Doubler / Tripler

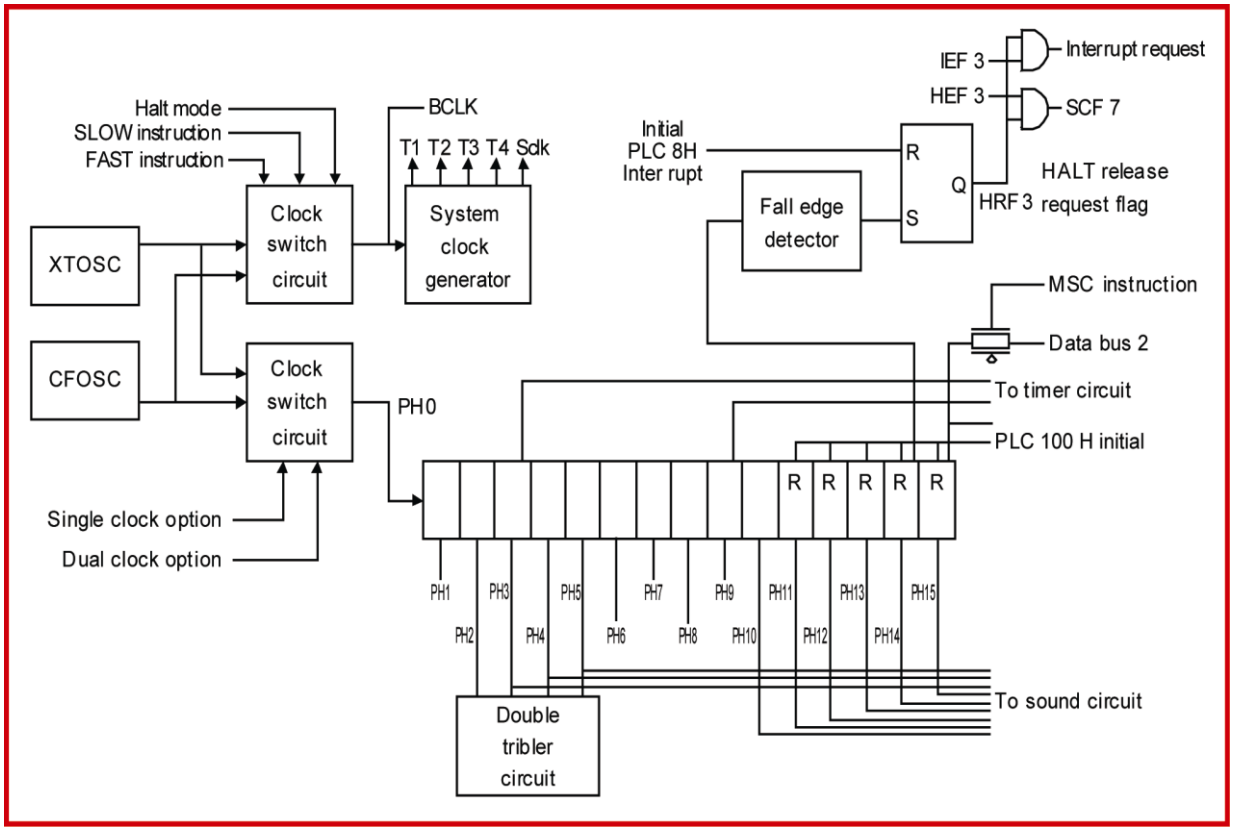
The doubler / tripler circuits are used to generate the bias voltage for LCD and are composed of a combination of PH2, PH3, PH4 and PH5.

3-3-2. Alternating Frequency for LCD

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

3-3-3. PREDIVIDER

The pre-divider is a 15-stage counter that receives the clock from the output of clock switch circuitry (PH0) as input. When PH0 is changed from "H" level to "L" level, the content of this counter changes. The PH11 to PH15 of the pre-divider are reset to "0" when the PLC 100H instruction is executed or at the initial reset mode. The pre-divider delivers the signal to the 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 3-3-3 shows the Pre-divider and its Peripherals

The PH14 delivers the halt mode release request signal, setting the halt mode release request flag (HRF3). In this case, if the pre-divider interrupt enable mode (IEF3) is provided, the interrupt is accepted; and if the halt release enable mode (HEF3) is provided, the halt release request signal

is delivered, setting the start condition flag 7 (SCF7) in status register 3 (STS3). The clock source of pre-divider is PH0, and 4 kinds of frequency of PH0 could be selected by mask option:

MASK OPTION table:

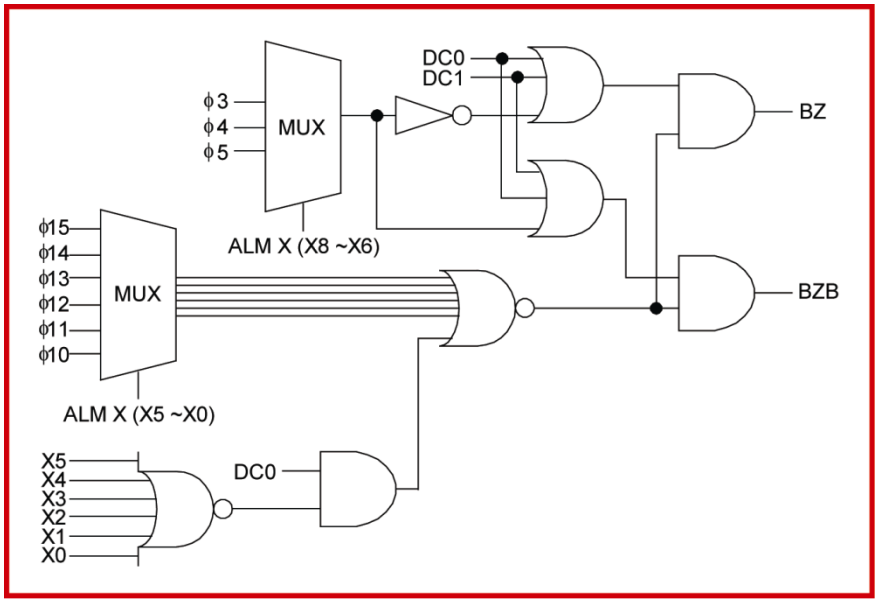
Mask Option name	Selected item
PH0 <-> BCLK FOR FAST ONLY	(1) PH0 = BCLK
PH0 <-> BCLK FOR FAST ONLY	(2) PH0 = BCLK/4
PH0 <-> BCLK FOR FAST ONLY	(3) PH0 = BCLK/8
PH0 <-> BCLK FOR FAST ONLY	(4) PH0 = BCLK/16

3-4 BUZZER OUTPUT PINS

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

MASK OPTION table:

Mask Option name	Selected item
SEG30/IOB3/BZB	(3) BZB
SEG31/IOB4/BZ	(3) BZ



This figure 3-4 shows the organization of the buzzer output.

3-4-1. BASIC BUZZER OUTPUT

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

Note:

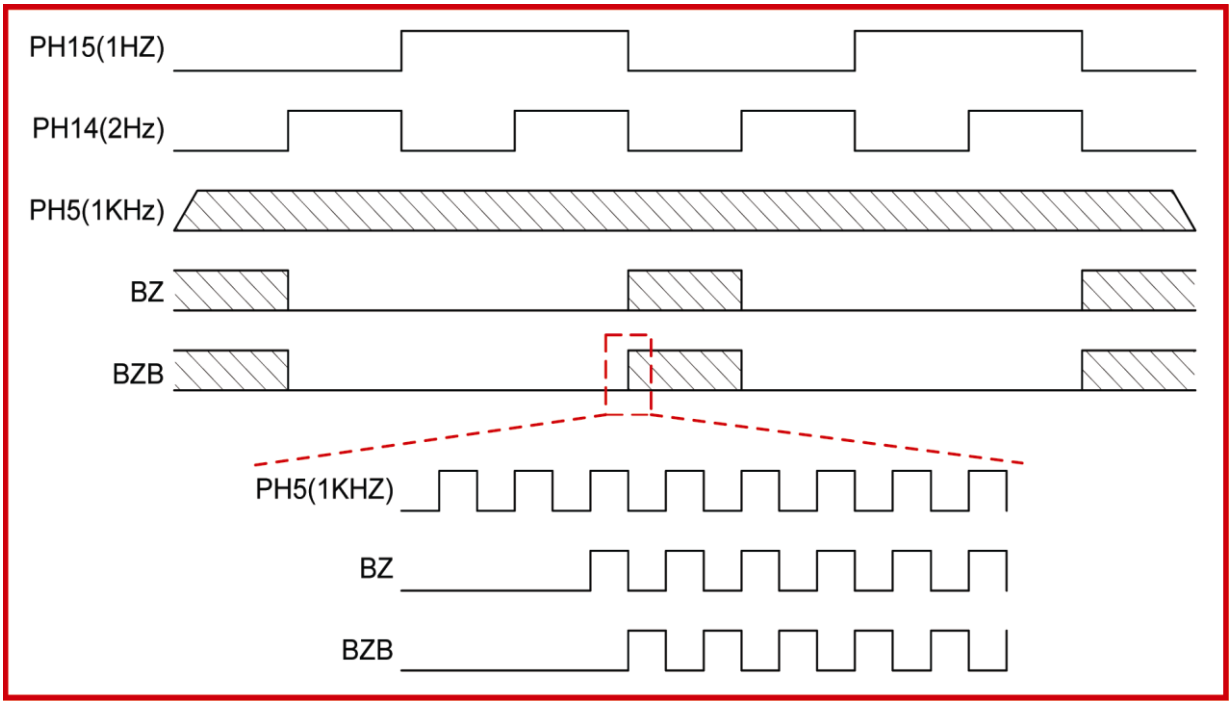
1. *The high frequency clock source should only be one of PH3, PH4, PH5 or FREQ, and the lower frequency may be any/all of the combinations from PH10 ~ PH15.*
2. *The frequencies in () corresponding to the input clock of the pre-divider (PH0) is 32768Hz.*
3. *The BZ and BZB pins will output DC0 after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

Example:

Buzzer output generates a waveform with 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 3-4-1*:



3-5 INPUT / OUTPUT PORTS

Three I/O ports are available in TM8724: IOA, IOB and IOC. 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 function by mask option, the input / output function of the pins will be disabled.

3-5-1. IOA PORT

IOA1 ~ IOA4 pins are MUX with SEG24, SEG25, SEG26 and SEG27 pins respectively by mask option.

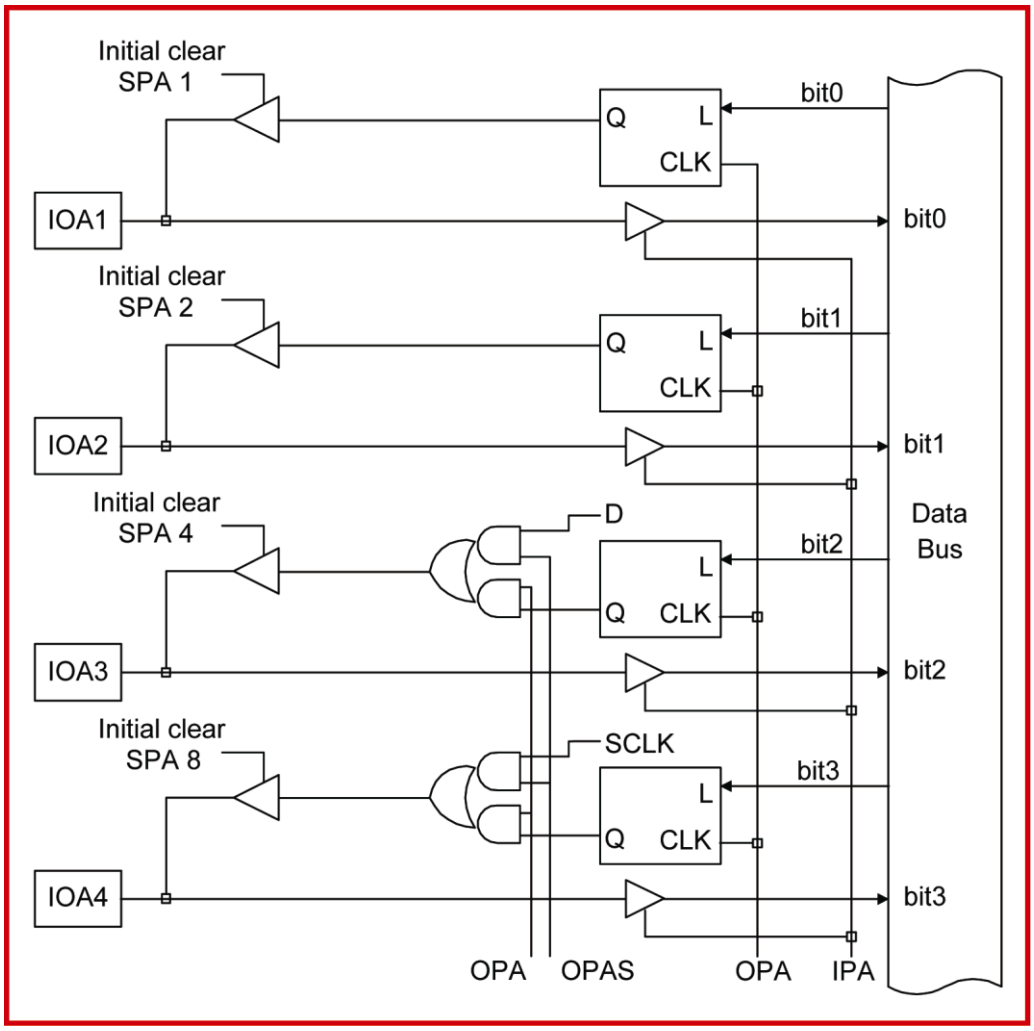
MASK OPTION table:

Mask Option name	Selected item
SEG24/IOA1	(2) IOA1
SEG25/IOA2	(2) IOA2
SEG26/IOA3	(2) IOA3
SEG27/IOA4	(2) IOA4

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

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

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



This figure 3-5-1 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.

3-5-1-1. Pseudo Serial Output

IOA port may operate as a pseudo serial output port by executing OPAS instruction. IOA port must be defined as the output mode before executing OPAS instruction.

(1). BIT0 and BIT1 of the port deliver RAM data.

(2). BIT2 of the port delivers the constant value of the OPAS.

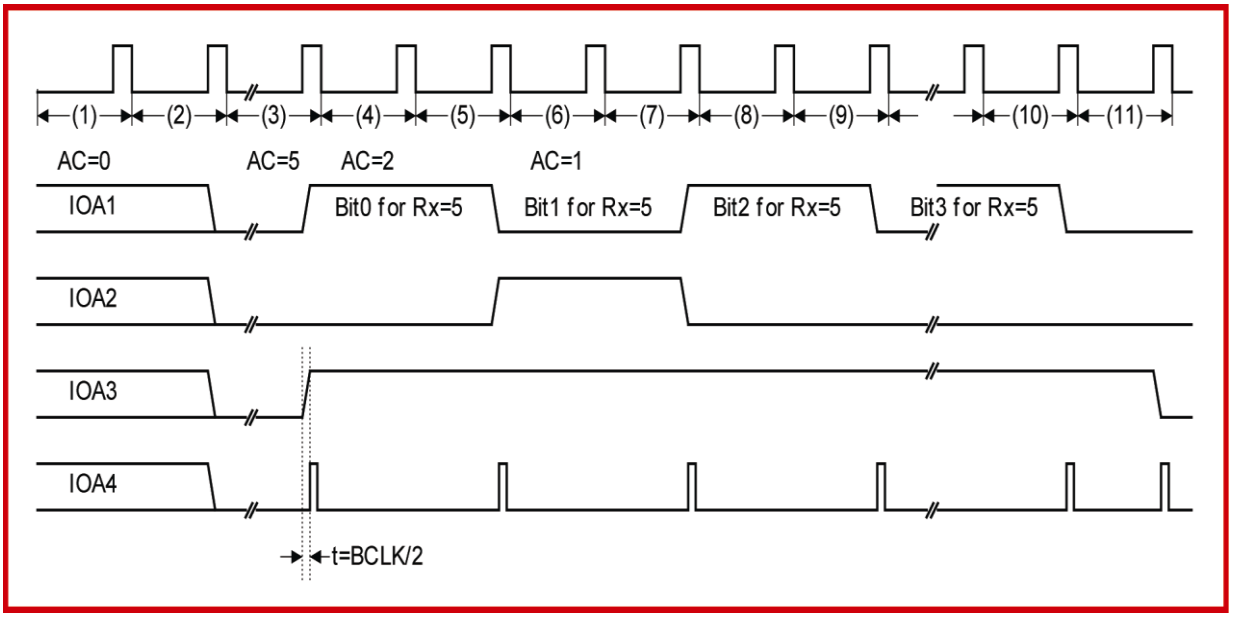
(3). BIT3 of the port delivers pulses.

Shown below is a sample program using the OPAS instruction.

(1) LDS 0AH, 0

```
(2) OPA 0AH
    SPA  0FH
    :
    :
    LDS  1,5
(3) OPAS      1,1     ;Bit 0 output, shift gate open
(4) SR0 1     ;Shifts bit 1 to bit 0
(5) OPAS      1,1     ;Bit 1 output
(6) SR0 1     ;Shifts bit2 to bit 0
(7) OPAS      1,1     ;Bit 2 output
(8) SR0 1     ;Shifts bit 3 to bit 0
(9) OPAS      1,1     ;Bit 3 output
    :
    :
(10) OPAS     1,1     ;Last data
(11) OPAS     1,0     ;Shift gate closes
```

The timing chart below illustrates the above program.



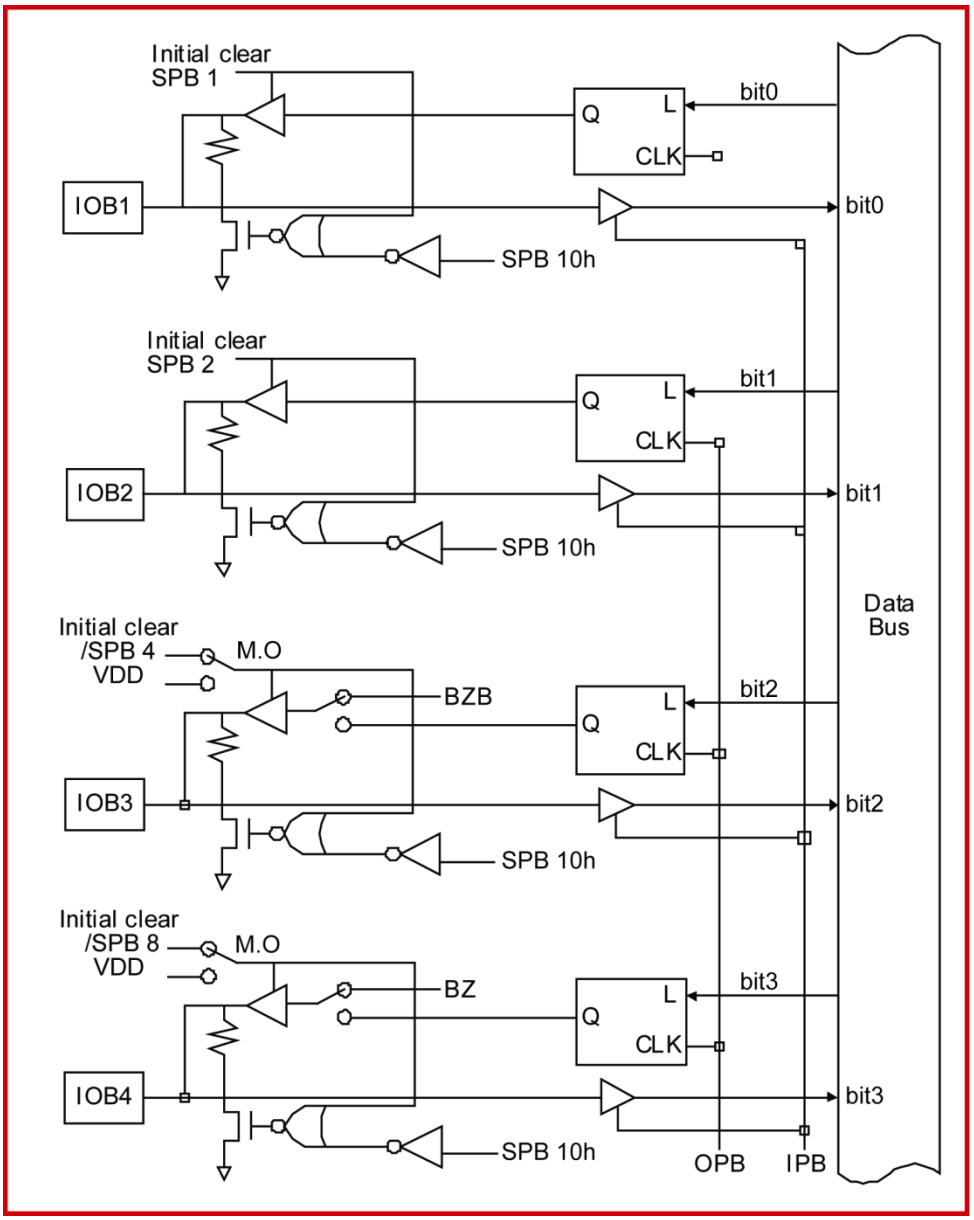
3-5-2. IOB PORT

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

The following *figure 3-5-2* shows the organization of IOB port.



Note:

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

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

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

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

3-5-3. IOC PORT

IOC1 ~ IOC4 pins are MUXed with SEG32, SEG33, SEG34 and SEG35 pins respectively by mask option.

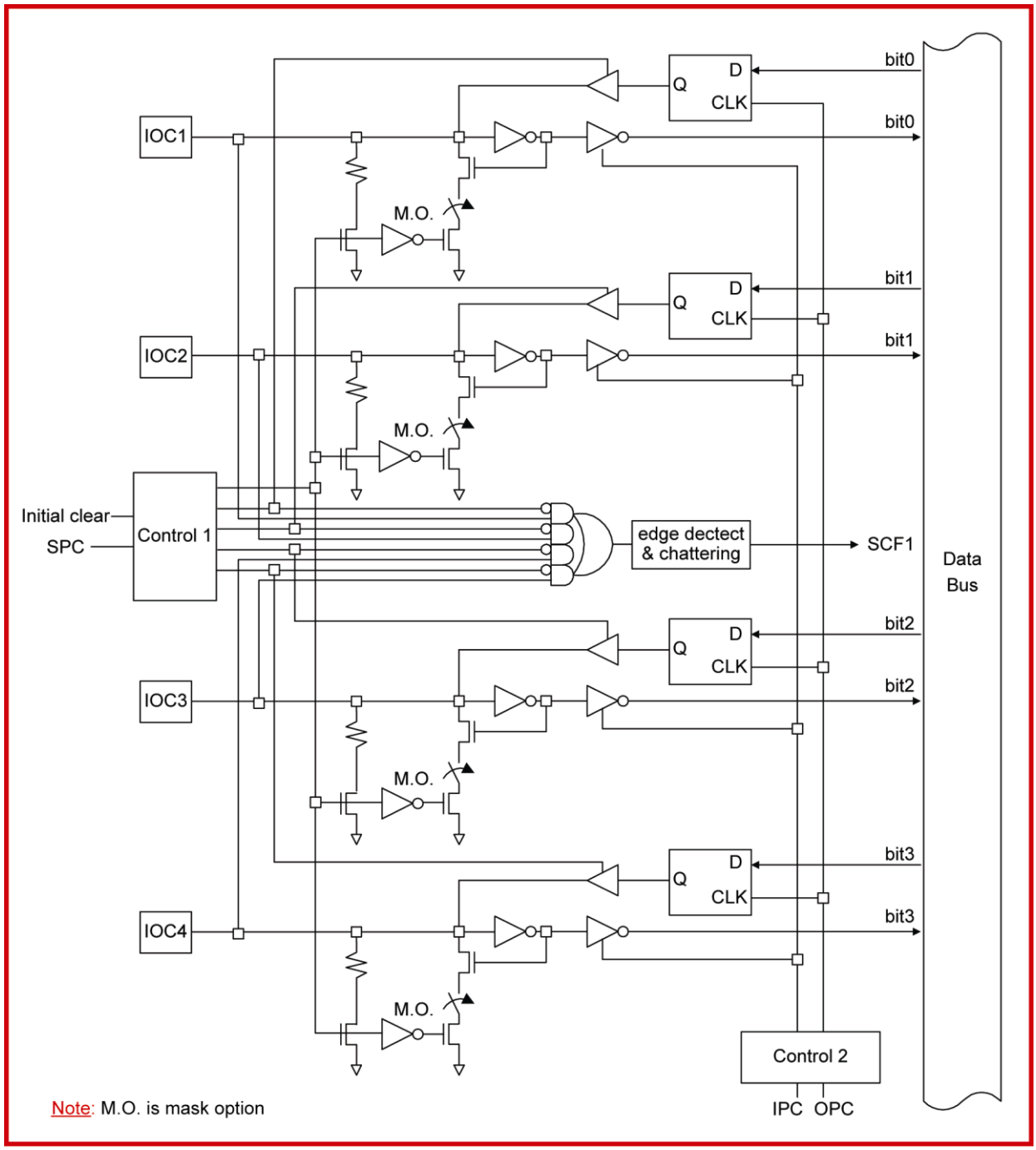
MASK OPTION table :

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

After the reset cycle, the IOC port is set as input mode and each bit of port can be defined as input mode or output mode individually by executing SPC instruction. Executed OPC instruction may output the content of specified data memory to the pins defined as output; the other pins, which are defined as the input will still remain the input mode.

Executed IPC instructions may store the signals applied to the IOC pins in the specified data memory. When the IOC pins are defined as the output, executing IPC instruction will save the data stored in the output latches in the specified data memory.

Before executing SPC instruction to define the IOC pins as output, the OPC instruction must be executed to output the data to those output latches. This will prevent the chattering signal when the IOC pins change to output mode.



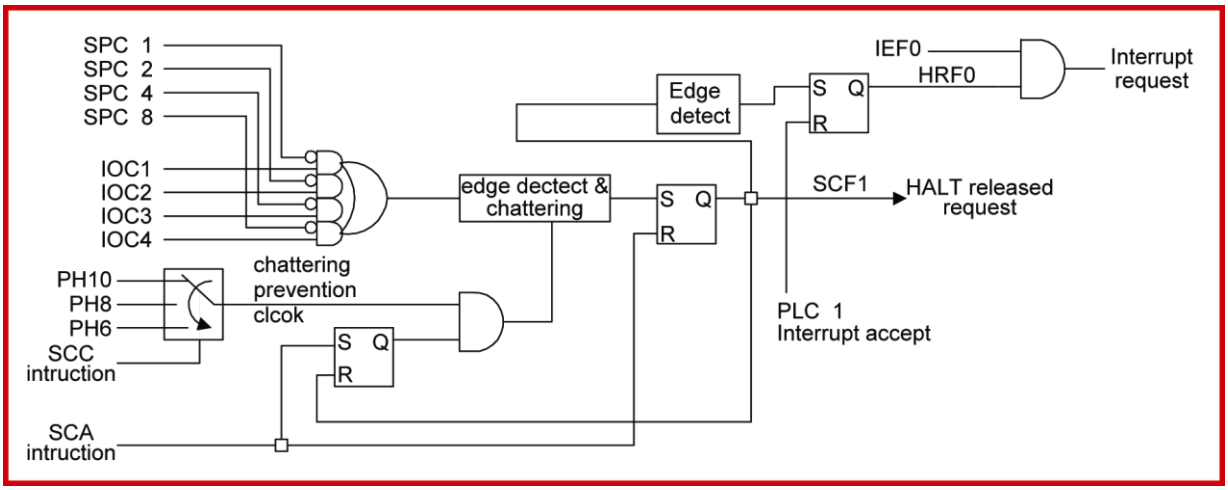
This figure 3-5-3 shows the organization of IOC port.

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. IOC port had built-in pull-down resistor and executing SPC instruction to enable / disable this device. IOC port also built in the pull-low device for each pin, but these devices are enabled by mask option. Executing SPC 10h instruction enables the pull-low device. Executing SPC 0h disables the pull-low device.

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 SCC instruction, and the default selection is PH10 after the reset cycle. When the pins of the IOC port are defined as output, the signals applied to the output pins will be inhibited for the chattering prevention function. The following figure shows the organization of chattering prevention circuitry.



Note: The default prevention clock is PH10

This chattering prevention function works when the signal at the applicable pin (ex. 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 IOC 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 (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 SCA instruction and 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.

3-6 EXTERNAL INT PIN

The INT pin can be selected as pull-up or 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.

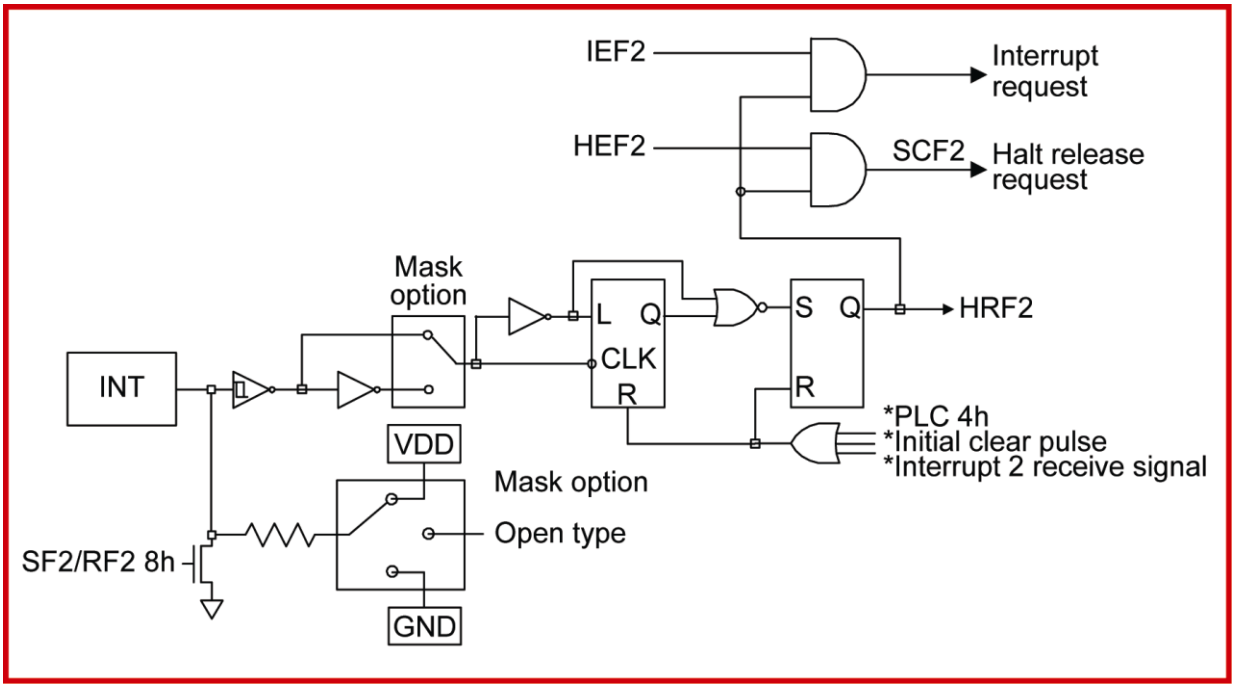
MASK OPTION table:

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 3-6 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.

Chapter 4 LCD DRIVER OUTPUT

4-1 LCD DRIVER OUTPUT

The number of the LCD driver outputs in TM8724 is 24 segment pins with 5 common pins. All SEG pins could also be used as DC output ports (mask option). If more than one of LCD driver output pin was defined as LCD driver output, the following mask option must be selected.

MASK OPTION table:

When more than one of SEG and COM pins have been used to drive LCD panel

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

When all 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, all of LCD's lighting system may be lighted or unlighted by mask option. All of the LCD output will keep the initial setting until the LCD relative instructions 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

4-1-1. LCD LIGHTING SYSTEM IN TM8724

There are several LCD lighting systems could be selected by mask option in TM8724, they are:

- 1/2 bias 1/2 duty, 1/2 bias 1/3 duty, 1/2 bias 1/4 duty, 1/2bias 1/5duty,
- 1/3 bias 1/3 duty, 1/3 bias 1/4 duty, 1/3 bias 1/5duty,
- 1/1 duty

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

MASK OPTION table :

LCD duty cycle option

Mask Option Name	Selected Item
LCD DUTY CYCLE	(1) O/P or STATIC
LCD DUTY CYCLE	(2) DUPLEX (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 bias option

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

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

The LCD alternating frequency in STATIC type

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

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

Mask Option name	Selected item	Remark (alternating frequency)
LCD frame frequency	(1) SLOW	16Hz
	(2) TYPICAL	32Hz
	(2) FAST	64Hz
	(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
	(2) TYPICAL	42Hz
	(2) FAST	85Hz
	(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
	(2) TYPICAL	32Hz
	(2) FAST	64Hz
	(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

	(2) TYPICAL	51Hz
	(2) FAST	102Hz
	(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
Static	24	Connect VDD3 to VDD2
Duplex	48	
1/2bias 1/3duty	72	
1/2bias 1/4duty	96	
1/2bias 1/5duty	120	
1/3 bias 1/3 duty	72	
1/3 bias 1/4 duty	96	
1/3 bias 1/5 duty	120	

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

4-1-2. DC OUTPUT

TM8724 permits LCD driver output pins (SEG1 ~ SEG12) to be defined as CMOS type DC output or P open-drain DC output ports by mask option. In this case, it is possible to use some LCD driver output pins for DC output and the rest LCD driver output pins for LCD driver.

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, the output data on this port will not be affected while the program entered stop mode or LCD turn-off mode.

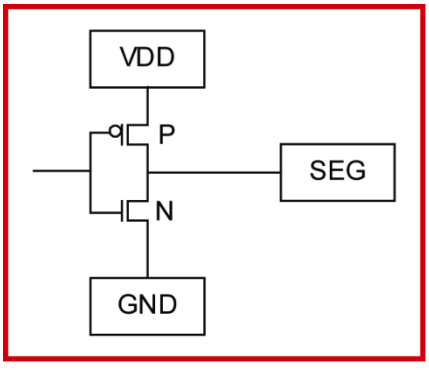


Figure 4-1-2-(1) CMOS Output Type

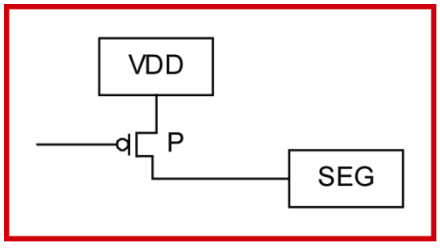


Figure 4-1-2-(2) P Open-Drain Output Type

4-1-3. SEGMENT PLA CIRCUIT FOR LCD DISPLAY

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

Explained below is how the LCD driver section operates when the instructions are executed.

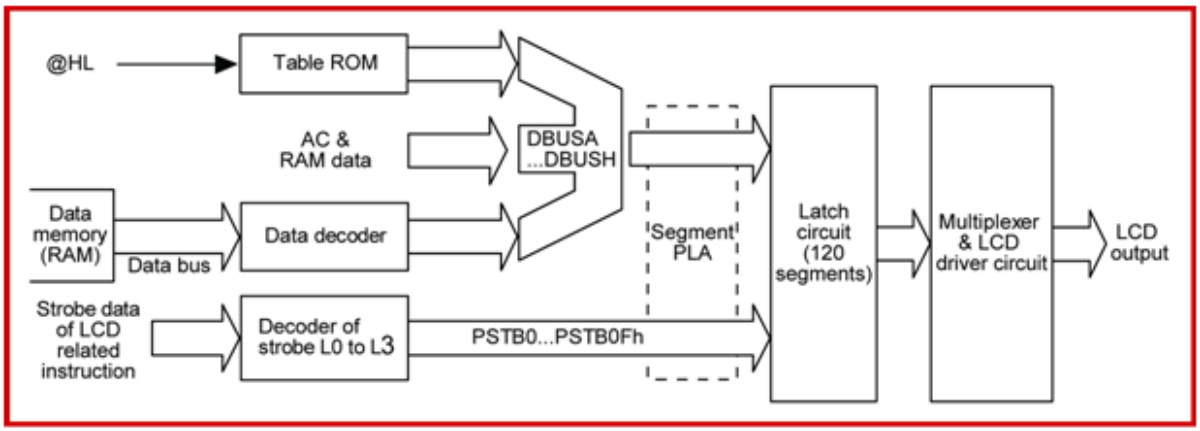


Figure 4-1-3-1 Principal Drawing of LCD Driver Section

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 L3 decoder to decode the Lz-specified data in the LCD-related instructions which specifies the strobe of the latch circuit
- Multiplexer to select 1/1duty, 1/2duty, 1/3duty, 1/4duty, 1/5duty
- LCD driver circuitry
- Segment PLA circuit connected between data decoder, L0 to L3 decoder and latch circuit.

The data decoder is used for decoding the content of the working register specified in LCD-related instructions as 7-segment pattern on LCD panel.

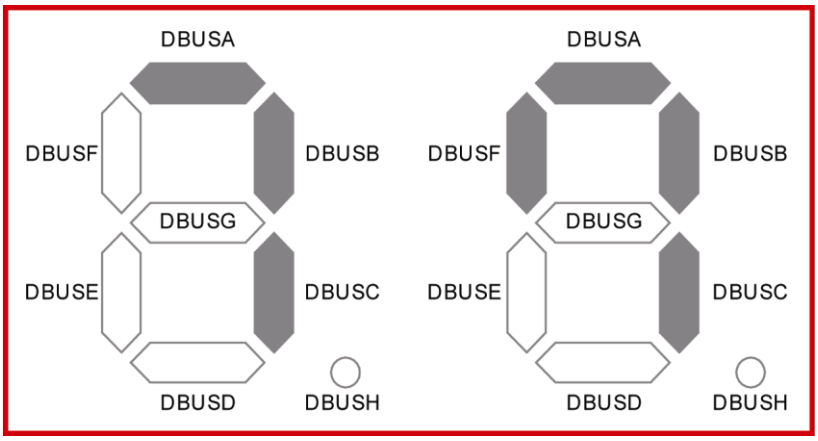
The decoding *table 4-1-3-1-(1)* is shown below:

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 4-1-3-1-(2)* shows the option table for displaying 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 data memory that specified. When the content of data memory that specified by LCB instruction is "0", the decoded output of DBUSA ~ DBUSH are all "0". (This is used for blanking the leading digit "0" on LCD panel).

The LCP instruction transferred the data of 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 4-1-3-1-(3) 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 of "DBUSA" to "DBUSH" and 16 L0 to L3 decoder outputs of PSTB 0h to PSTB 0Fh. The input data and clock signal of the latch circuit are "DBUSA"

to ”DBUSH” and PSTB 0h to PSTB 0Fh, respectively. Each segment pin has 4 latches corresponding to COM1-5.

The segment PLA performs the function of combining ”DBUSA” to ”DBUSH” inputs to each latch and strobe PSTB 0h to PSTB 0Fh is selected freely by mask option.

Of 128 signals obtainable by combining ”DBUSA” to ”DBUSH” and PSTB 0h to PSTB 0Fh, any 120 (corresponding to the number of latch circuits incorporated in the hardware) signals can be selected by programming and the above-mentioned segment PLA.

Table 4-1-3-1-(4) shows the PSTB 0h to PSTB 0Fh signals concretely.

Table 4-1-3-1-(4) 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
……	…….
PSTB0Ah	0AH
PSTB0Bh	0BH
PSTB0Ch	0CH
PSTB0Dh	0DH
PSTB0Eh	0EH
PSTB0Fh	0FH

Note:

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.

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

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

4-1-3-2. Relative Instructions

(1). LCT Lz, Ry

Decodes the content specified in Ry with the data decoder and transfers the DBUSA ~ H to 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 LCD latch specified by Lz. The "DBUSA" to "DBUSH" are all 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" to "DBUSH" without passing through the data decoder. The mapping table is shown in table 2-32.

(4). LCP Lz, Ry

The data of the RAM and accumulator (AC) are transferred directly to "DBUSA" to "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 the DBUSA ~ H to 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 LCD latch specified by Lz. The "DBUSA" to "DBUSH" are all 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" to "DBUSH" without passing through the data decoder. The mapping table is shown below:

(8). SF2 4h

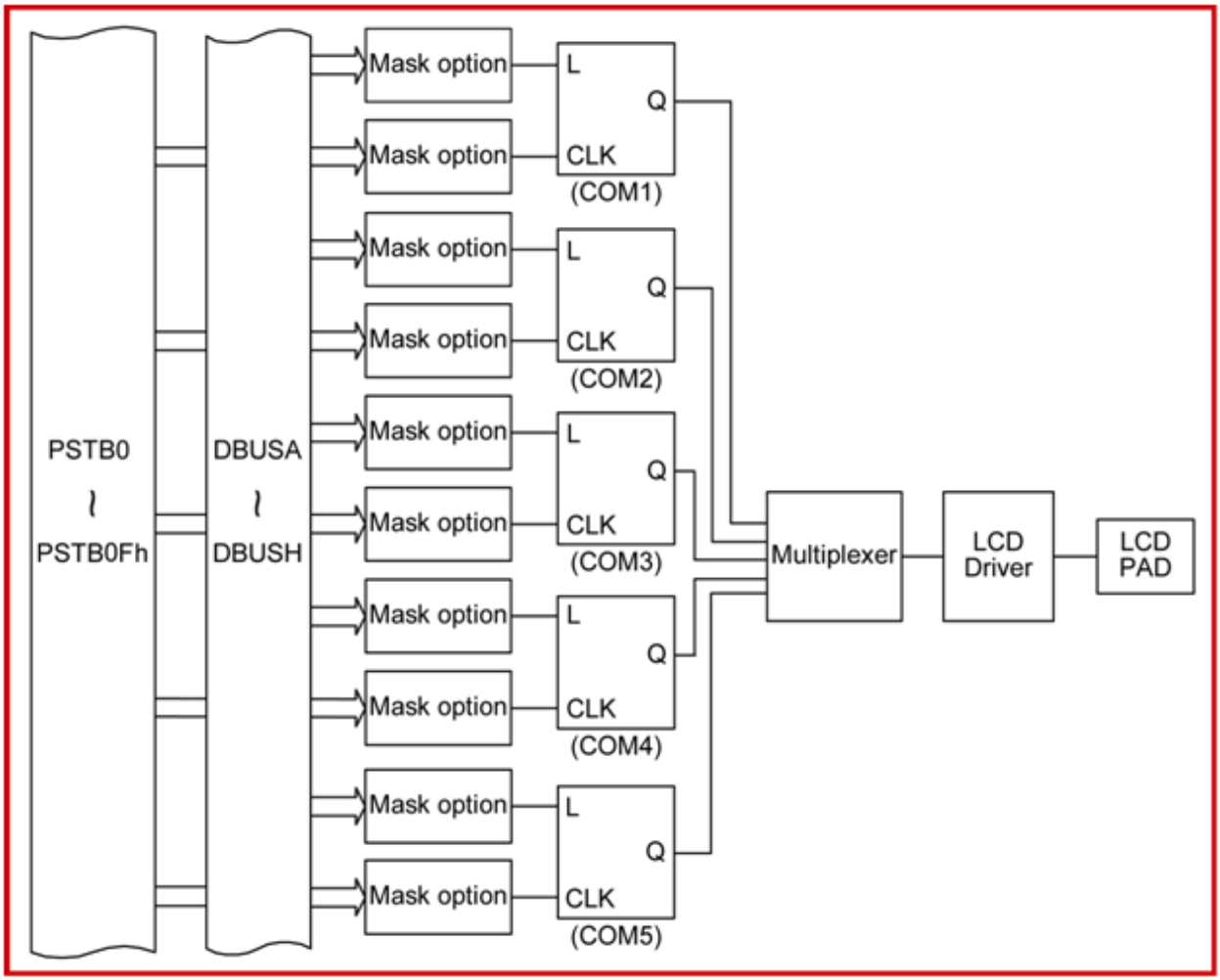
Turn off the LCD display.

(9). RF2 4h

Turn on the LCD display.

4-1-3-3. CONCRETE EXPLANATION

Each LCD driver output corresponds to the LCD 1/5 duty panel and has 5 latches (refer to ***The Figure 4-1-3-3 Shows 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 the segments in the LCD driver outputs is flexible. In other words, one of the data decoder outputs "DBUSA" to "DBUSH" is applied to the latch input L, and one of the PSTB 0h to PSTB 0Fh outputs are applied to clock CLK.



The Figure 4-1-3-3 Shows Sample Organization of Segment PLA Option

4-1-3-4. 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 *.cfg file is as follows:

SEG COM PSTB DBUS

SEG: Specifies the segment pin

COM: Specifies the corresponding latch in each segment pin. Only 0, 1, 2, 3, 4, 5 and 9 can be entered into this field.”0”is for CMOS type DC output option and ”9” 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 Detail Explanation of TM8724 Instructions

- Before using the data memory, it is necessary to initiate the content of data memory because the initial value is unknown.

- The working registers are part of the data memory (RAM), and the relationship between them can be shown as follows:

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

  Note:

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

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

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

- @HL is an 8-bit index address buffer. This buffer may address all data memory and table ROM. The contents of the index address buffer may be specified by two instructions: MVH and MVL. MVH transfers the contents of data memory Rx to the higher nibble (4-bits) and MVL to the lower nibble (4-bits).

  The organization of the index address buffer (@HL) is shown below.

Index Address Buffer	
Higher nibble @H	Lower nibble @L
H7~0	L3~L0
Transferred by MVH	Transferred by MVL

- Lz represents the address of the latch of LCD PLA; the address range specified by Lz is from 00H to 0FH.

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.

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 outputs of the data decoder are all "0".

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.

LCD Lz, @HL

Function: LCD latch Lz ← (T@HL)

Description: @HL indicates an index address of table ROM.
The contents of table ROM specified by @HL is loaded to the LCD latch specified by Lz directly.

LCT Lz, @HL

Function: LCD latch Lz ← data decoder ← (R@HL)

Description: The contents of index RAM specified by @HL is loaded to the LCD latch specified by Lz through the data decoder.

LCB Lz, @HL

Function: LCD latch Lz ← data decoder ← (R@HL)

Description: The contents of index RAM specified by @HL is loaded to the LCD latch specified by Lz through the data decoder.
If the content of @HL is "0", the outputs of the data decoder are all "0".

LCP Lz, @HL

Function: LCD latch Lz ← (R@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.

SPA X

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

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

Bit pattern	Setting	Bit pattern	Setting
X3=1	IOA4 as output mode	X3=0	IOA4 as input mode
X2=1	IOA3 as output mode	X2=0	IOA3 as input mode
X1=1	IOA2 as output mode	X1=0	IOA2 as input mode
X0=1	IOA1 as output mode	X0=0	IOA1 as input mode

OPA Rx

Function: I/OA ← (Rx)

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

OPAS Rx, D

Function: IOA1,2 ← (Rx), IOA3 ← D, IOA4 ← pulse

Description: Content of Rx is outputted to IOA port. D is outputted to IOA3, pulse is outputted to IOA4.
D = 0 or 1

IPA Rx

Function: Rx, AC ← (IOA)

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

SPB X

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

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

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable IOB pull low R	X4=0	Disable IOB pull low R
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 ← (IOB)

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 / disables the pull-low device or low-level-hold device.

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

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

ALM X

Function: Sets buzzer output frequency.

Description: The waveform specified by X (X8 ~ X0) is delivered to the BZ and BZB pins.

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	0	0	DC1
0	1	1	ϕ3(4KHz)
0	1	0	ϕ4(2KHz)
0	0	1	ϕ5(1KHz)
0	0	0	DC0

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

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

Notes:

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

5-2 ACCUMULATOR MANIPULATION INSTRUCTIONS AND MEMORY MANIPULATION INSTRUCTIONS

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

MRW @HL, Rx
Function: AC, R@HL ← (Rx)
Description: The content of data memory specified by Rx is loaded to AC and data memory specified by @HL.

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

MWR Rx, @HL
Function: AC, Rx ← (R@HL)
Description: The content of data memory specified by @HL is loaded to AC and data memory specified by Rx.

SR0 Rx
Function: Rxn, ACn ← Rx(n+1),AC(n+1)
Rx3, AC3 ← 0
Description: The Rx content is shifted right and 0 is loaded to the MSB.
The result is loaded to the AC.
0 → Rx3 → Rx2 → Rx1 → Rx0 →

SR1 Rx

Function: Rxn, ACn ← Rx(n+1),AC(n+1)
Rx3, AC3 ← 1

Description: The Rx content is shifted right and 1 is loaded to the MSB. The result is loaded to the AC.
1 → Rx3 → Rx2 → Rx1 → Rx0 →

SL0 Rx

Function: Rxn, ACn ← Rx(n-1),AC(n-1)
Rx0, AC0 ← 0

Description: The Rx content is shifted left and 0 is loaded to the LSB. The results are loaded to the AC.
← Rx3 ← Rx2 ← Rx1 ← Rx0 ← 0

SL1 Rx

Function: Rxn, ACn ← Rx(n-1),AC(n-1)
Rx0, AC0 ← 1

Description: The Rx content is shifted left and 1 is loaded to the LSB. The results are loaded to the AC.
← Rx3 ← Rx2 ← Rx1 ← Rx0 ← 1

MRA Rx

Function: CF ← (Rx)3

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

MAF Rx

Function: AC,Rx ← CF

Description: The content of CF is loaded to AC and Rx. The content of AC and meaning of bit after execution of this instruction are as follows:
Bit 3 .... CF
Bit 2 .... (AC)=0, zero flag
Bit 1 .... (No Use)
Bit 0 .... (No Use)

5-3 OPERATION INSTRUCTIONS

INC* Rx

Function: Rx,AC ← (Rx)+1

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

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

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

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

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

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

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

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

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

. Carry flag (CF) will be affected.

SBC @HL

Function: AC ← (R@HL)+ (AC)B+CF

Description: The contents of AC and CF are binary-subtracted from content of data memory specified by @HL; the result is loaded to AC.

* Carry flag (CF) will be affected.

SBC* Rx

Function: AC, Rx ← (Rx)+(AC)B+CF

Description: The contents of AC and CF are binary-subtracted from content of Rx; the result is loaded to AC and data memory Rx.

. Carry flag (CF) will be affected.

SBC* @HL

Function: AC,R@HL ← (R@HL)+ (AC)B+CF

Description: The contents of AC and CF are binary-subtracted from content of data memory specified by @HL; the result is loaded to AC and data memory specified by @HL.

* Carry flag (CF) will be affected.

ADD Rx

Function: AC ← (Rx)+(AC)

Description: The contents of Rx and AC are binary-added; the result is loaded to AC.

. Carry flag (CF) will be affected.

ADD @HL

Function: AC ← (R@HL)+(AC)

Description: The contents of data memory specified by @HL and AC are binary-added; the result is loaded to AC.

* Carry flag (CF) will be affected.

ADD* Rx

Function: AC, Rx ← (Rx)+(AC)

Description: The contents of Rx and AC are binary-added; the result is loaded to AC and data memory Rx.

. Carry flag (CF) will be affected.

ADD* @HL

Function: AC,R@HL ← (R@HL)+(AC)

Description: The contents of data memory specified by @HL and AC are binary-added; the result is loaded to AC and data memory specified by @HL.

* Carry flag (CF) will be affected.

SUB Rx

Function: AC ← (Rx)+ (AC)B+1

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

SUB @HL
Function: AC ← (R@HL)+ (AC)B+1
Description: The content of AC is binary-subtracted from content of data memory specified by @HL; the result is loaded to AC.
* Carry flag (CF) will be affected.

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

SUB* @HL
Function: AC, R@HL ← (R@HL)+ (AC)B+1
Description: The content of AC is binary-subtracted from content of data memory specified by @HL; the result is loaded to AC and data memory specified by @HL.
* Carry flag (CF) will be affected.

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

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

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

ADN* @HL
Function: AC, R@HL ← (R@HL)+(AC)

Description: The contents of data memory specified by @HL and AC are binary-added; the result is loaded to AC and data memory specified by @HL.
* The result will not affect the carry flag (CF).

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

AND @HL
Function: AC ← (R@HL) & (AC)
Description: The contents of data memory specified by @HL and AC are binary-ANDed; the result is loaded to AC.

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

AND* @HL
Function: AC, R@HL ← (R@HL) & (AC)
Description: The contents of data memory specified by @HL and AC are binary-ANDed; the result is loaded to AC and data memory specified by @HL.

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

EOR @HL
Function: AC ← (R@HL) ⊕ (AC)
Description: The contents of data memory specified by @HL and AC are exclusive-Ored; the result is loaded to AC.

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

EOR* @HL
Function: AC, R@HL ← (R@HL) ⊕ (AC)
Description: The contents of data memory specified by @HL and AC are exclusive-Ored; the result is loaded to AC and data memory data memory specified by @HL.

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

OR @HL
Function: AC ← (R@HL) | (AC)
Description: The contents of @HL and AC are binary-Ored; the result is loaded to AC.
. @HL indicates an index address of data memory.

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

OR* @HL
Function: AC,R@HL ← (R@HL) | (AC)
Description: The contents of data memory specified by @HL and AC are binary-Ored; the result is loaded to AC and data memory specified by @HL.

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

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

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

D = 0H ~ FH

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

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

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

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

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

ADNI Ry, D
Function: AC ← (Ry)+D
Description: D represents an immediate data.

The contents of Ry and D are binary-ADDed; the result is loaded to AC.
* The result will not affect the carry flag (CF).

D = 0H ~ FH

ADNI* Ry, D

Function: AC, Ry ← (Ry)+D

Description: D represents an immediate data.
The contents of Ry and D are binary-ADDed; the result is loaded to AC and working register Ry.
* The result will not affect the carry flag (CF).

D = 0H ~ FH

ANDI Ry, D

Function: AC ← (Ry) & D

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

ANDI* Ry, D

Function: AC, Ry ← (Ry) & D

Description: D represents an immediate data.
The contents of Ry and D are binary-ANDed; the result is loaded to AC and working register Ry.

D = 0H ~ FH

EORI Ry, D

Function: AC ← (Ry) ⊕ D

Description: D represents an immediate data.
The contents of Ry and D are exclusive-OREd; the result is loaded to AC.

D = 0H ~ FH

EORI* Ry, D

Function: AC, Ry ← (Ry) ⊕ D

Description: D represents an immediate data.
The contents of Ry and D are exclusive-OREd; the result is loaded to AC and working register Ry.

D = 0H ~ FH

ORI Ry, D

Function: AC ← (Ry) | D

Description: D represents an immediate data.
The contents of Ry and D are binary-OREd; the result is loaded to AC.

D = 0H ~ FH

ORI* Ry, D

Function: AC, Ry ← (Ry) | D

Description: D represents an immediate data.
The contents of Ry and D are binary-OREd; the result is loaded to AC and working register Ry.

D = 0H ~ FH

5-4 LOAD/STORE INSTRUCTIONS

STA Rx

Function: Rx ← (AC)

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

STA @HL

Function: R@HL ← (AC)

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

LDS Rx, D

Function: AC,Rx ← D

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

LDA Rx

Function: AC ← (Rx)

Description: The content of Rx is loaded to AC.

LDA @HL

Function: AC ← (R@HL)

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

LDH Rx, @HL

Function: Rx , AC ← H(T@HL)

Description: The higher nibble data of Table ROM specified by @HL is loaded to data memory specified by Rx.

LDH* Rx, @HL

Function: Rx , AC ← H(T@HL), @HL←(@HL)+1

Description: The higher nibble data of Table ROM specified by @HL is loaded to data memory specified by Rx and then is increased in @HL.

LDL Rx, @HL

Function: Rx , AC ← L(T@HL)

Description: The lower nibble data of Table ROM specified by @HL is loaded to the data memory specified by Rx.

LDL* Rx, @HL

Function: Rx, AC ← L(T@HL), @HL ← (@HL)+1

Description: The lower nibble data of Table ROM specified by @HL is loaded to the data memory specified by Rx and then incremented the content of @HL.

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 IOC.

3) The halt release condition specified by SHE instruction is met.

When an interrupt is accepted to release the halt mode, the halt mode returns by executing the RTS instruction after completion of interrupt service.

STOP

Function: Enters stop mode and stops all oscillators

Description: Before executing this instruction, all signals on IOC port must be set to low.

The following 2 conditions cause the stop mode to be released.

1) A signal change in the INT pin.

2) One of the signals on IOC port is "H".

SCA X

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

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

Bit pattern	Description
X4=1	Halt mode is released when signal applied to IOC

X7~5,X3~0 is reserved

SIE* X

Function: Set/Reset interrupt enable flag

Description:

X0=1	The IEF0 is set so that interrupt 0(Signal change at port IOC 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.

X7, 6, 5, 4 is reserved

SHE X

Function: Set/Reset halt release enables 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.

X7, 6, 5, 4 is reserved

SRE X

Function: Set/Reset stop release enable flag

Description:

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, 6, X3~0 is reserved

FAST

Function: Switches the system clock to CFOSC clock.

Description: Starts up the CFOSC (high-speed osc.) and then switches the system clock to 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 ← SCF1,SCF2,BCF

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

Bit 3	Bit 2	Bit 1	Bit 0
NA	Start condition flag 2 (SCF2)	Start condition flag 1 (SCF1)	Backup flag (BCF)
	Halt release caused by SCF4,5,7	Halt release caused by the IOC port	The backup mode status in TM8724

MSC Rx

Function: AC, Rx ← SCF4..7

Description: The SCF4 to SCF7 contents are loaded to AC and the data memory specified by Rx.
The content of AC and meaning of bit after execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 7 (SCF7)	The content of 15th stage of the predivider	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

MSD Rx

Function: Rx, AC ← CSF

Description: The system clock status is loaded to data memory specified by Rx and AC.

The content of AC and meaning of bit after execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Reserved	Reserved	Reserved	System clock selection flag (CSF)

5-6 INDEX ADDRESS INSTRUCTIONS

MVH Rx

Function: (@H) ← (Rx),(AC)

Description: Loads content of Rx to higher nibble of index address buffer @H.
@H7=AC3, @H6=AC2, @H5=AC1, @H4=AC0,
@H3=Rx3, @H2=Rx2, @H1=Rx1, @H0=Rx0,

MVL Rx

Function: (@L) ← (Rx)

Description: Loads content of Rx to lower nibble of index address buffer @L.
@L3=Rx3, @L2=Rx2, @L1=Rx1, @L0=Rx0

5-7 DECIMAL ARITHMETIC INSTRUCTIONS

DAA

Function: AC ← BCD(AC)

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

DAA* Rx

Function: AC, Rx ← BCD(AC)

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

DAA* @HL

Function: AC,R@HL ← BCD(AC)

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

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

DAS

Function: AC ← BCD(AC)

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

DAS* Rx

Function: AC, Rx ← BCD(AC)

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

DAS* @HL

Function: AC, @HL ← BCD(AC)

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

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 if AC0=1.

Description: If bit0 of AC is 1, jump occurs.
If 0, the PC increases by 1.

The range of X is from 000H to 47FH.

JB1 X

Function: Program counter jumps to X if AC1=1.
Description: If bit1 of AC is 1, jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 47FH.

JB2 X

Function: Program counter jumps to X if AC2=1.
Description: If bit2 of AC is 1, jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 47FH.

JB3 X

Function: Program counter jumps to X if AC3=1.
Description: If bit3 of AC is 1, jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 47FH.

JNZ X

Function: Program counter jumps to X if (AC)! = 0.
Description: If the content of AC is not 0, jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 47FH.

JNC X

Function: Program counter jumps to X if CF=0.
Description: If the content of CF is 0, jump occurs.
If 1, the PC increases by 1.
The range of X is from 000H to 47FH.

JZ X

Function: Program counter jumps to X if (AC)=0.
Description: If the content of AC is 0, jump occurs.
If 1, the PC increases by 1.
The range of X is from 000H to 47FH.

JC X

Function: Program counter jumps to X if CF=1.
Description: If the content of CF is 1, jump occurs.
If 0, the PC increases by 1.
The range of X is from 000H to 47FH.

JMP X

Function: Program counter jumps to X.
Description: Unconditional jump.
The range of X is from 000H to 47FH.

CALL X

Function: STACK ← (PC)+1
Program counter jumps to X.
Description: A subroutine is called.
The range of X is from 000H to 47FH.

RTS

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

5-9 MISCELLANEOUS INSTRUCTIONS

SCC X

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

Bit pattern	Clock source setting	Bit pattern	Clock source setting
(X2,X1,X0)= 001	Chattering prevention clock = ϕ10	(X2,X1,X0)= 010	Chattering prevention clock = ϕ8
(X2,X1,X0)= 100	Chattering clock = ϕ6		

X7, 5,4,3 is reserved

TMS Rx

Function: Select timer 1 clock source and preset timer 1.
Description: The content of 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		Setting value					
TMS Rx	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0

The clock source option for timer 1

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

TMS @HL

Function: Select timer 1 clock source and preset timer 1.
Description: The content of 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		Setting value					
TMS @HL	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

The clock source option for timer 1

Bit7	Bit6	Clock source
0	0	ϕ9
0	1	ϕ3
1	0	ϕ15

TMSX X

Function: Selects 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:

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

The clock source option for timer 1

X7	X6	Clock source
0	0	ϕ9
0	1	ϕ3
1	0	ϕ15

SF X

Function: Sets flag

Description: Description of each flag

X0 : "1" The CF is set to 1.

X1 : "1" The chip enters backup mode and BCF is set to 1.

X7 : "1" Enables the re-load function of timer 1.

X6, 5, 4, 3, 2 is reserved

RF X

Function: Resets flag

Description: Description of each flag

X0 : "1" The CF is reset to 0.

X1 : "1" The chip is out of backup mode and BCF is reset to 0.

X7 : "1" Disables the re-load function of timer 1.

X6, 5, 4, 3, 2 is reserved

SF2 X

Function: Sets flag

Description: Description of each flag

X2 : "1"Disables the LCD segment output.
X7~3,1,0 is reserved

RF2 X

Function: Resets flag

Description: Description of each flag
X2 : "1" Enables the LCD segment output.
X7~3,1,0 is reserved

PLC X

Function: Pulse control

Description: The pulse corresponding to the data specified by X is generated.
X0 : "1" Halt release request flag HRF0 caused by the signal at I/O port 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 be reset.
X8 : "1" The last 5 bits of the predivider (15 bits) are reset. When executing this instruction, X3 must be set to "1".
X7 ~ 4 are reserved.

ORDERING INFORMATION

The ordering information:

Ordering number	Package
TM8724-COD	Wafer / Dice with code

Appendix A - TM8724 Instruction Table

Instruction		Machine Code	Function		Flag/Remark
NOP		0000 0000 0000 0000	No Operation		
LCT	Lz,Ry	0000 0010 ZZZZ YYYY	(Lz)	←7SEG ← (Ry)	
LCB	Lz,Ry	0000 0100 ZZZZ YYYY	(Lz)	←7SEG ← (Ry)	Blank Zero
LCP	Lz,Ry	0000 0110 ZZZZ YYYY	(Lz)	← (Ry) , (AC)	
LCD	Lz,@HL	0000 1000 ZZZZ 0000	(Lz)	← (R@HL)	
LCT	Lz,@HL	0000 1000 ZZZZ 0001	(Lz)	←7SEG ←(R@HL)	
LCB	Lz,@HL	0000 1000 ZZZZ 0010	(Lz)	←7SEG ←(R@HL)	Blank Zero
LCP	Lz,@HL	0000 1000 ZZZZ 0011	(Lz)	←(R@HL) , (AC)	
OPA	Rx	0000 1010 0XXX XXXX	(IOA)	← (Rx)	
OPAS	Rx,D	0000 1011 DXXX XXXX	IOA1,2,3,4	← (Rx)0,(Rx)1,D,Pulse	
OPB	Rx	0000 1100 0XXX XXXX	(IOB)	←(Rx)	
OPC	Rx	0000 1101 0XXX XXXX	(IOC)	←(Rx)	
MVL	Rx	0001 1100 0XXX XXXX	(@L)	← (Rx)	
MVH	Rx	0001 1101 0XXX XXXX	(@H)	← (Rx) , (AC)	
ADC	Rx	0010 0000 0XXX XXXX	(AC)	← (Rx) + (AC) + CF	CF
ADC	@HL	0010 0000 1000 0000	(AC)	← (R@HL) + (AC) + CF	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
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*	Rx	0010 0011 0XXX XXXX	(AC),(Rx)	← (Rx) + (AC)B + CF	CF
SBC*	@HL	0010 0011 1000 0000	(AC),(R@HL)	← (R@HL) + (AC)B + CF	CF
ADD	Rx	0010 0100 0XXX XXXX	(AC)	← (Rx) + (AC)	CF
ADD	@HL	0010 0100 1000 0000	(AC)	← (R@HL) + (AC)	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
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*	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
ADN	Rx	0010 1000 0XXX XXXX	(AC)	← (Rx) + (AC)	
ADN	@HL	0010 1000 1000 0000	(AC)	← (R@HL) + (AC)	
ADN*	Rx	0010 1001 0XXX XXXX	(AC),(Rx)	← (Rx) + (AC)	
ADN*	@HL	0010 1001 1000 0000	(AC),(R@HL)	← (R@HL) + (AC)	
AND	Rx	0010 1010 0XXX XXXX	(AC)	← (Rx) AND (AC)	

Instruction		Machine Code	Function		Flag/Remark
AND	@HL	0010 1010 1000 0000	(AC)	← (R@HL) AND (AC)	
AND*	Rx	0010 1011 0XXX XXXX	(AC),(Rx)	← (Rx) AND (AC)	
AND*	@HL	0010 1011 1000 0000	(AC),(R@HL)	← (R@HL) AND (AC)	
EOR	Rx	0010 1100 0XXX XXXX	(AC)	← (Rx) EOR (AC)	
EOR	@HL	0010 1100 1000 0000	(AC)	← (R@HL) EOR (AC)	
EOR*	Rx	0010 1101 0XXX XXXX	(AC),(Rx)	← (Rx) EOR (AC)	
EOR*	@HL	0010 1101 1000 0000	(AC),(R@HL)	← (R@HL) EOR (AC)	
OR	Rx	0010 1110 0XXX XXXX	(AC)	← (Rx) OR (AC)	
OR	@HL	0010 1110 1000 0000	(AC)	← (R@HL) OR (AC)	
OR*	Rx	0010 1111 0XXX XXXX	(AC),(Rx)	← (Rx) OR (AC)	
OR*	@HL	0010 1111 1000 0000	(AC),(R@HL)	← (R@HL) OR (AC)	
ADCI	Ry,D	0011 0000 DDDD YYYY	(AC)	← (Ry) + D + CF	CF
ADCI*	Ry,D	0011 0001 DDDD YYYY	(AC),(Ry)	← (Ry) + D + CF	CF
SBCI	Ry,D	0011 0010 DDDD YYYY	(AC)	← (Ry) + DB + CF	CF
SBCI*	Ry,D	0011 0011 DDDD YYYY	(AC),(Ry)	← (Ry) + DB + CF	CF
ADDI	Ry,D	0011 0100 DDDD YYYY	(AC)	← (Ry) + D	CF
ADDI*	Ry,D	0011 0101 DDDD YYYY	(AC),(Ry)	← (Ry) + D	CF
SUBI	Ry,D	0011 0110 DDDD YYYY	(AC)	← (Ry) + DB + 1	CF
SUBI*	Ry,D	0011 0111 DDDD YYYY	(AC),(Ry)	← (Ry) + DB + 1	CF
ADNI	Ry,D	0011 1000 DDDD YYYY	(AC)	← (Ry) + D	
ADNI*	Ry,D	0011 1001 DDDD YYYY	(AC),(Ry)	← (Ry) + D	
ANDI	Ry,D	0011 1010 DDDD YYYY	(AC)	← (Ry) AND D	
ANDI*	Ry,D	0011 1011 DDDD YYYY	(AC),(Ry)	← (Ry) AND D	
EORI	Ry,D	0011 1100 DDDD YYYY	(AC)	← (Ry) EOR D	
EORI*	Ry,D	0011 1101 DDDD YYYY	(AC),(Ry)	← (Ry) EOR D	
ORI	Ry,D	0011 1110 DDDD YYYY	(AC)	← (Ry) OR D	
ORI*	Ry,D	0011 1111 DDDD YYYY	(AC),(Ry)	← (Ry) OR D	
INC*	Rx	0100 0000 0XXX XXXX	(AC),(Rx)	← (Rx) + 1	CF
INC*	@HL	0100 0000 1000 0000	(AC),(R@HL)	← (R@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
IPA	Rx	0100 0010 0XXX XXXX	(AC),(Rx)	← (IOA)	
IPB	Rx	0100 0100 0XXX XXXX	(AC),(Rx)	← (IOB)	
IPC	Rx	0100 0111 0XXX XXXX	(AC),(Rx)	← (IOC)	
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 : (No use)

Instruction		Machine Code	Function		Flag/Remark
					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)
MSD	Rx	0100 1110 0XXX XXXX	(AC),(Rx)	← STS4	B3 : (No use) B2 : (No use) B1 : (No use) 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
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
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	
STA	Rx	0110 1000 0XXX XXXX	(Rx)	← (AC)	
STA	@HL	0110 1000 1000 0000	(R@HL)	← (AC)	
LDA	Rx	0110 1100 0XXX XXXX	(AC)	← (Rx)	
LDA	@HL	0100 1100 1000 0000	(AC)	← (R@HL)	
MRA	Rx	0110 1101 0XXX XXXX	CF	← (Rx)3	
MRW	@HL,Rx	0110 1110 0XXX XXXX	(AC),(R@HL)	← (Rx)	
MWR	Rx,@HL	0110 1111 0XXX XXXX	(AC),(Rx)	← (R@HL)	
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

Instruction		Machine Code	Function		Flag/Remark
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 0XXX XXXX XXXX	STACK (PC)	← (PC) + 1 ← X	
JMP	X	1101 0XXX XXXX XXXX	(PC)	← X	
RTS		1101 1000 0000 0000	(PC)	← STACK	CALL Return
SCC	X	1101 1001 0000 0XXX	X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1101 1010 000X 0000	X4	: Enable SEF4	C1-4
SPA	X	1101 1100 0000 XXXX	X3~0	: Set IOA4-1 Output enable	
SPB	X	1101 1101 000X XXXX	X4 X3~0	: Set IOC4-1 Pull-Low : Set IOB4-1 Output enable	
SPC	X	1101 1110 000X XXXX	X4 X3-0	: Set IOC4-1 Pull-Low / Low-Level-Hold : Set IOC4-1 Output enable	
TMS	Rx	1110 0000 0XXX XXXX	Timer1	← (Rx) & (AC)	
TMS	@HL	1110 0001 0000 0000	Timer1	← (T@HL)	
TMSX	X	1110 0010 XXXX XXXX	X7,6 = 10 X7,6 = 01 X7,6 = 00 X5~0	: Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
SHE	X	1110 1000 0XXX XXX0	X3 X2 X1	: Enable HEF3 : Enable HEF2 : Enable HEF1	PDV INT TMR1
SIE*	X	1110 1001 0XXX XXXX	X3 X2 X1 X0	: Enable IEF3 : Enable IEF2 : Enable IEF1 : Enable IEF0	PDV INT TMR1 CPT
PLC	X	1110 101X 0XXX XXXX	X8 X3-0	: Reset PH15~11 : Reset HRF3-0	
SRE	X	1110 1101 X0XX 0000	X5 X4	: Enable SRF5 : Enable SRF4	SRF5 (INT) SRF4 (C Port)
FAST		1110 1110 0000 0000	SCLK	: High Speed Clock	
SLOW		1110 1111 0000 0000	SCLK	: Low Speed Clock	
SF	X	1111 0000 X00X XXXX	X7 X1 X0	: Reload 1 Set : BCF Set : CF Set	RL1 BCF CF

Instruction		Machine Code	Function		Flag/Remark
RF	X	1111 0100 X00X 0XXX	X7 X1 X0	:Reload 1 Reset : BCF Reset : CF Reset	RL1 BCF CF
SF2	X	1111 1000 0000 XXXX	X3 X2	: Enable INT strong Pull-low : Close all Segments	INTPL RSOFF
RF2	X	1111 1001 0000 XXXX	X3 X2	: Disable INT strong Pull-low : Release Segments	INTPL RSOFF
ALM	X	1111 101X XXXX XXXX	X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: DC1 : PH3 : PH4 : PH5 : DC0 ← PH15~10	
HALT		1111 1110 0000 0000	Halt Operation		
STOP		1111 1111 0000 0000	Stop Operation		

Appendix B -Symbol Description

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