十速

TM8722

with LCD Driver

User Manual

Rev 1.5

AMENDMENT HISTORY

Version	Date	Description
1.0	Oct, 2003	New release.
1.1	Nov, 2003	Modify page 5, 8, 14, 16~18, 21, 66, 68, 70, 72, 75~81, 106, 120.
1.2	May, 2009	Add experiment data.
1.3	Nov, 2009	Add circuit diagram description for LCD 1.5V 1/3 Bias.
1.4	Dec, 2011	Add Ordering Information table
1.5	Dec, 2016	Deleted Segment Driver Output Characteristics 'Static'

CONTENTS

1. General Description

1.1 GENERAL DESCRIPTION

The TM8722 is an embedded high-performance 4-bit micro-controller with LCD/LED driver on a chip. It contains all the necessary functions for a micro-controller, such as 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, Resistance to Frequency Converter(RFC), EL panel driver, LCD driver, look-up table, watchdog timer and key matrix scanning circuitry etc. on a single chip.

1.2 FEATURES

1. Low power dissipation.

2. Powerful instruction set (143 instructions).

- Binary addition subtraction, BCD conversion, logical operation in direct and index addressing mode.
- Single-bit manipulation (set, reset, decision for branch).
- Various conditional branches.
- 16 working registers and manipulation.
- Look-up Table.
- LCD driver with data transfer.

3. Memory capacity.

- ROM capacity 2048 x 16 bits.
- RAM capacity 128 x 4 bits.

4. LCD/LED driver output.

- 5 common outputs and 35 segment outputs (drive up to 175 LCD/LED segments).
- 1/2 Duty, 1/3 Duty, 1/4 Duty or 1/5 Duty for both LCD/LED drivers selectable in mask option.
- 1/2 Bias or 1/3 Bias for LCD driver selectable in mask option.
- Single instruction to turn off all segments.
- All segment outputs can be defined as CMOS or P_open drain output type in mask option.

5. Input/output ports.

- Port IOA 4 pins (with internal pull-low), muxed with SEG24~27.
- Port IOB 4 pins (with internal pull-low), muxed with SEG28~31.
- Port IOC 4 pins (with internal pull-low/ low-level-hold), muxed with SEG32~35. IOC port with built-in input signal chattering prevention circuitry.

6. 8 level subroutine nesting.

7. Interrupt function.

- External factors 3 (INT pin, Port IOC & KI input).
- Internal factors 4 (Pre-Divider, Timer1, Timer2 & RFC).

8. Built-in EL panel driver.

- ELC, ELP (Muxed with SEG28, SEG29).

9. Built-in Alarm, clock or single tone melody generator.

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

10. Built-in R to F Converter circuit.

- CX, RR, RT, RH (Muxed with SEG24~SEG27).

11. Built-in key matrix scanning function.

- K1~K16 (Shared with SEG1~SEG16).
- KI1~KI4 (Muxed with SEG32~SEG35).

12. Two 6-bit programmable timers with programmable clock source.

13. Watch dog timer.

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

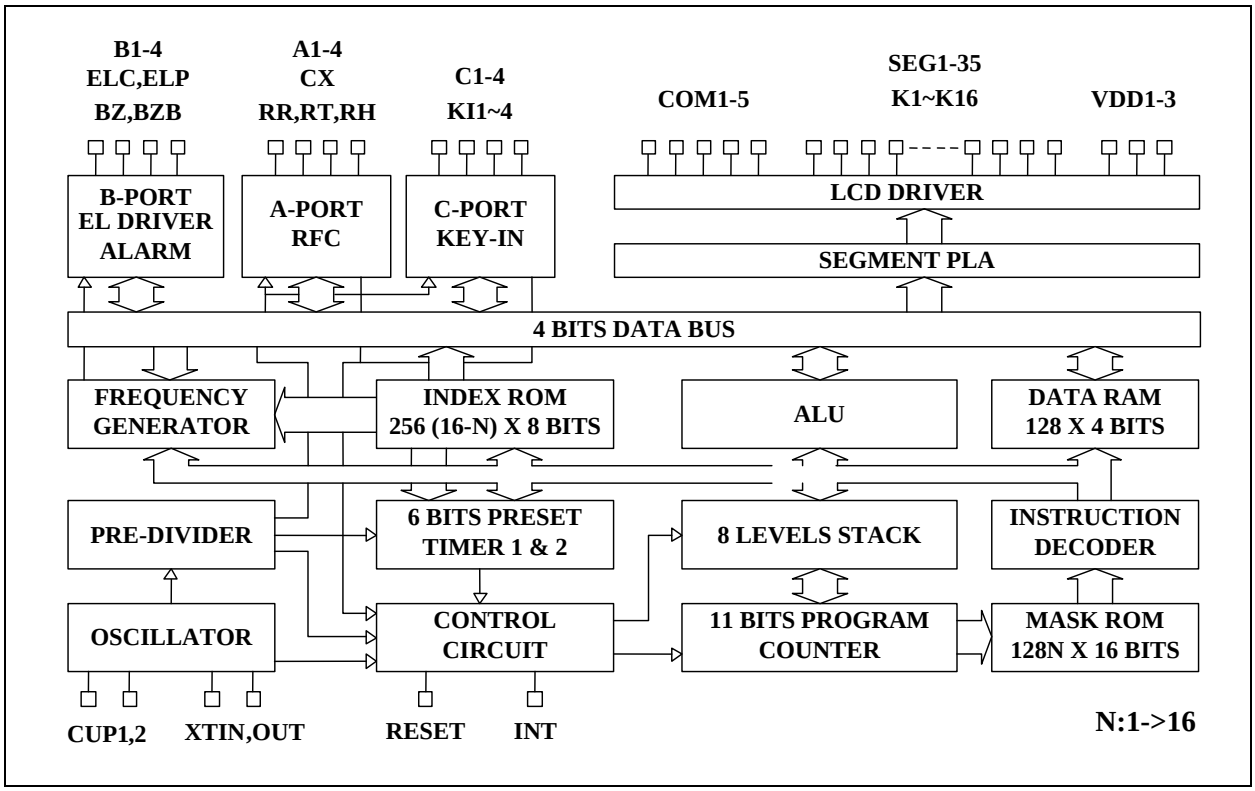
15. Dual clock operation

- Slow clock oscillation can be defined as X'tal or external RC type oscillator in mask option.
- Fast clock oscillation can be defined as internal R or external R type oscillator in mask option.

16. HALT function.

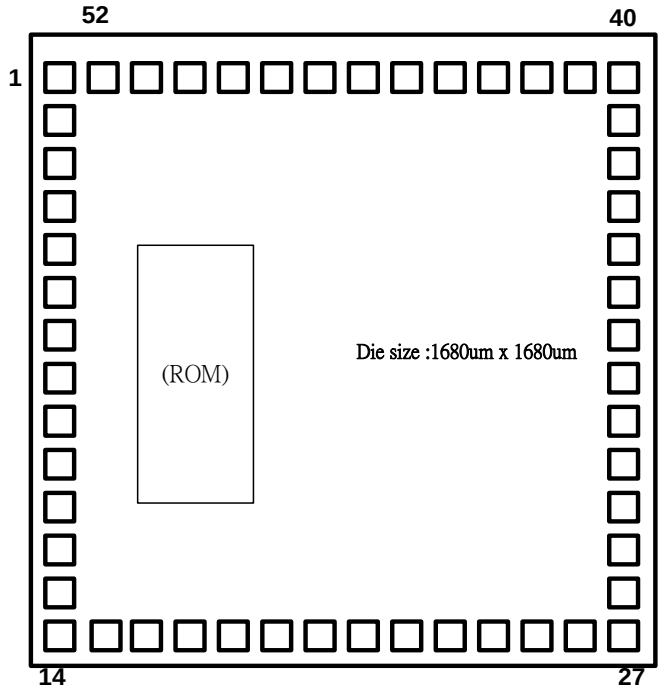
17. STOP function.

1.3 BLOCK DIAGRAM



1.4 PAD DIAGRAM

The chip substrate should be connected to GND.



1.5 PAD COORDINATE

No	Name	X	Y	No	Name	X	Y
1	BAK	77.5	1602.5	27	SEG13/K13	1602.5	77.5
2	XIN	77.5	1472.5	28	SEG14/K14	1602.5	207.5
3	XOUT	77.5	1357.5	29	SEG15/K15	1602.5	322.5
4	GND	77.5	1242.5	30	SEG16/K16	1602.5	437.5
5	VDD1	77.5	1127.5	31	SEG17	1602.5	552.5
6	VDD2	77.5	1012.5	32	SEG18	1602.5	667.5
7	VDD3	77.5	897.5	33	SEG19	1602.5	782.5
8	CUP1	77.5	782.5	34	SEG20	1602.5	897.5
9	CUP2	77.5	667.5	35	SEG21	1602.5	1012.5
10	COM1	77.5	552.5	36	SEG22	1602.5	1127.5
11	COM2	77.5	437.5	37	SEG23	1602.5	1242.5
12	COM3	77.5	322.5	38	SEG24/IOA1/CX	1602.5	1357.5
13	COM4	77.5	207.5	39	SEG25/IOA2/RR	1602.5	1472.5
14	COM5	77.5	77.5	40	SEG26/IOA3/RT	1602.5	1602.5
15	SEG1/K1	207.5	77.5	41	SEG27/IOA4/RH	1472.5	1602.5
16	SEG2/K2	322.5	77.5	42	SEG28/IOB1/ELC	1357.5	1602.5
17	SEG3/K3	437.5	77.5	43	SEG29/IOB2/ELP	1242.5	1602.5
18	SEG4/K4	552.5	77.5	44	SEG30/IOB3/BZB	1127.5	1602.5
19	SEG5/K5	667.5	77.5	45	SEG31/IOB4/BZ	1012.5	1602.5
20	SEG6/K6	782.5	77.5	46	SEG32/IOC1/KI1	897.5	1602.5
21	SEG7/K7	897.5	77.5	47	SEG33/IOC2/KI2	782.5	1602.5
22	SEG8/K8	1012.5	77.5	48	SEG34/IOC3/KI3	667.5	1602.5
23	SEG9/K9	1127.5	77.5	49	SEG35/IOC4/KI4	552.5	1602.5
24	SEG10/K10	1242.5	77.5	50	RESET	437.5	1602.5
25	SEG11/K11	1357.5	77.5	51	INT	322.5	1602.5
26	SEG12/K12	1472.5	77.5	52	TEST	207.5	1602.5

1.6 PIN DESCRIPTION

Name	I/O	Description
BAK	P	Positive Back-up voltage. In Li power Mode, connect a 0.1u capacitor to GND.
VDD1, 2, 3	P	LCD supply voltage, and positive supply voltage. • In Ag Mode, it will connect a positive power to VDD1. • In Li or ExtV power mode, it will connect a positive power to VDD2.
RESET	I	Input pin for external reset request signal. Built-in internal pull-down resistor.
INT	I	Input pin for external INT request signal. • It can be trigger by Falling edge or rising edge and is defined in mask option. • nternal pull-down or pull-up resistor is defined in mask option.
TESTA		Test signal input pin.
CUP1,2	O	Switching pins for supplying the LCD driving voltage to the VDD1, 2, 3 pins. • Connect the CUP1 and CUP2 pins with non-polarized electrolytic capacitor when chip operates in 1/2 or 1/3 bias mode. • In no BIAS mode application, leave these pins open.
XIN XOUT	I O	Input /Output pins for slow clock oscillator. • 32KHz Crystal oscillator. • In FAST mode, connect an external resistor to form a RC oscillator (mask option). • In SLOW mode, connect an external resistor to form a RC oscillator (mask option).
COM1~5	O	Output pins for driving the common pins for the LCD or LED panel.
SEG1-35	O	Output pins for driving the LCD or the LED panel segment.
IOA1-4	I/O	Input/Output port A. (Muxed with SEG24~SEG27)
IOB1-4	I/O	Input/Output port B. (Muxed with SEG28~SEG31)
IOC1-4	I/O	Input/Output port C. (Muxed with SEG32~SEG35)
CX RR/RT/RH	I O	RFC application with 1 input pin and 3 output pins. (Muxed with SEG24~SEG27)
ELC/ELP	O	Output port for the EL panel driver. (Muxed with SEG28,SEG29)
BZB/BZ	O	Output port for alarm, clock or the single melody tone generator. (Muxed with SEG30~SEG31)
K1~K16	O	Output port for the key matrix scanning.(Shared with SEG1~SEG16)
KI1~4	I	Input port for the key matrix scanning.(Muxed with SEG32~SEG35)
GND	P	Negative supply voltage.

1.7 CHARACTERISTICS

ABSOLOUTE MAXIMUM RATINGS

GND=0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD1	-0.3 to 5.5	V
	VDD2	-0.3 to 5.5	
	VDD3	-0.3 to 8.5	
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	-20 to +70	°C
Maximum Storage Temperature	Tstg	-25 to +125	

POWER CONSUMPTION

at Ta= -20°C to 70°C, GND=0V

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

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

ALLOWABLE OPERATING CONDITIONS

at Ta= -20°C to 70°C, GND=0V

Name	Symb.	Condition	Min.	Max.	Unit
Supply Voltage	VDD1		1.2	5.25	V
	VDD2		2.4	5.25	
	VDD3		2.4	8.0	
Oscillator Start-Up Voltage	VDDB	Crystal Mode	1.3		
Oscillator Sustain Voltage	VDDB	Crystal Mode	1.2		
Supply Voltage	VDD1	Ag Mode	1.2	1.65	
Supply Voltage	VDD2	EXT-V, Li Mode	2.4	5.25	
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 in Ag Battery Mode	0.8xVDD1	VDD1	
Input “L” Voltage	Vil3		0	0.2xVDD1	
Input “H” Voltage	Vih4	OSCIN in Li Battery Mode	0.8xVDD2	VDD2	
Input “L” Voltage	Vil4		0	0.2xVDD2	
Input “H” Voltage	Vih5	CFIN in Li Battery or EXT-V Mode	0.8xVDD2	VDD2	
Input “L” Voltage	Vil5		0	0.2xVDD2	
Input “H” Voltage	Vih6	RC Mode	0.8xVDDO	VDDO	
Input “L” Voltage	Vil6		0	0.2xVDDO	
Operating Freq	Fopg1	Crystal Mode	32		KHz
	Fopg2	RC Mode	10	1000	

INTERNAL RC FREQUENCY RANGE

Option Mode	BAK	Min.	Typ.	Max.
250 KHz	1.5V	200KHz	300KHz	400KHz
	3.0V	200KHz	250KHz	300KHz
500 KHz	1.5V	450KHz	600KHz	750KHz
	3.0V	400KHz	500KHz	600KHz

ELECTRICAL CHARACTERISTICS

at#1: VDD1=1.2V (Ag) ;

at#2: VDD2=2.4V (Li) :

at#3: VDD2=4V (Ext-V) ;

Input Resistance

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
"L" Level Hold Tr(IOC)	Rllh1	Vi=0.2VDD1,#1	10	40	100	KΩ
	Rllh2	Vi=0.2VDD2,#2	10	40	100	
	Rllh3	Vi=0.2VDD2,#3	5	20	50	
IOA/B/C Pull-Down Tr	Rmad1	Vi=VDD1,#1	200	500	1000	
	Rmad2	Vi=VDD2,#2	200	500	1000	
	Rmad3	Vi=VDD2,#3	100	250	500	
INT Pull-up Tr	Rintu1	Vi=VDD1,#1	200	500	1000	
	Rintu2	Vi=VDD2,#2	200	500	1000	
	Rintu3	Vi=VDD2,#3	100	250	500	
INT Pull-Down Tr	Rintd1	Vi=GND,#1	200	500	1000	
	Rintd2	Vi=GND,#2	200	500	1000	
	Rintd3	Vi=GND,#3	100	250	500	
RES Pull-Down R	Rres1	Vi=GND or VDD1,#1	10	40	100	
	Rres2	Vi=GND or VDD2,#2	10	40	100	
	Rres3	Vi=GND or VDD2,#3	10	40	100	

DC Output Characteristics

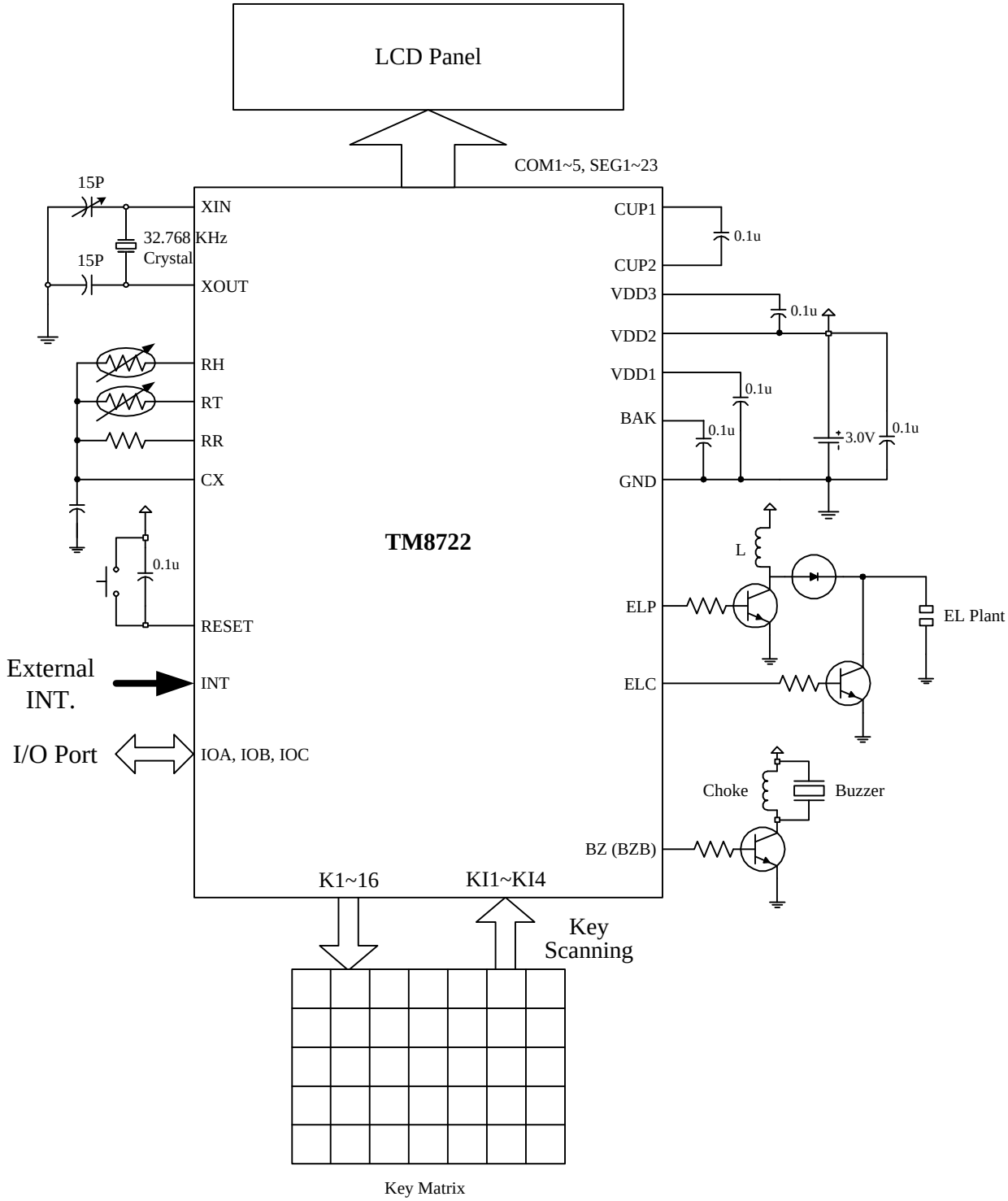
Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit
Output "H" Voltage	Voh1c	Ioh=-200uA,#1	SEG1~35	0.8	0.9	1.0	V
	Voh2c	Ioh=-1mA,#2		1.5	1.8	2.1	
	Voh3c	Ioh=-3mA,#3		2.5	3.0	3.5	
Output "L" Voltage	Vol1c	Iol=400uA,#1		0.2	0.3	0.4	
	Vol2c	Iol=2mA,#2		0.3	0.6	0.9	
	Vol3c	Iol=6mA,#3		0.5	1.0	1.5	

Segment Driver Output Characteristics

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
1/2 Bias Display Mode							
Output ”H” Voltage	Voh12f	Ioh=-1uA,#1,#2	SEG-n	2.2			V
	Voh3f	Ioh=-1uA,#3		3.8			
Output ”L” Voltage	Vol12f	Iol=1uA,#1,#2				0.2	
	Vol3f	Iol=1uA,#3				0.2	
Output ”H” Voltage	Voh12g	Ioh=-10uA,#1,#2	COM-n	2.2			
	Voh3g	Ioh=-10uA,#3		3.8			
Output ”M” Voltage	Vom12g	Iol/h=+/-10uA,#1,#2		1.0		1.4	
	Vom3g	Iol/h=+/-10uA,#3		1.8		2.2	
Output ”L” Voltage	Vol12g	Iol=10uA,#1,#2				0.2	
	Vol3g	Iol=10uA,#3				0.2	
1/3 Bias display Mode							
Output ”H” Voltage	Voh12i	Ioh=-1uA,#1,#2	SEG-n	3.4			V
	Voh3i	Ioh=-1uA,#3		5.8			
Output ”M1” Voltage	Vom12i	Iol/h=+/-10uA,#1,#2		1.0		1.4	
	Vom13i	Iol/h=+/-10uA,#3		1.8		2.2	
Output ”M2” Voltage	Vom22i	Iol/h=+/-10uA,#1,#2		2.2		2.6	
	Vom23i	Iol/h=+/-10uA,#3		3.8		4.2	
Output ”L” Voltage	Vol12i	Iol=1uA,#1,#2				0.2	
	Vol3i	Iol=1uA,#3				0.2	
Output ”H” Voltage	Voh12j	Ioh=-10uA,#1,#2	COM-n	3.4			
	Voh3j	Ioh=-10uA,#3		5.8			
Output ”M1” Voltage	Vom12j	Iol/h=+/-10uA,#1,#2		1.0		1.4	
	Vom13j	Iol/h=+/-10uA,#3		1.8		2.2	
Output ”M2” Voltage	Vom22j	Iol/h=+/-10uA,#1,#2		2.2		2.6	
	Vom23j	Iol/h=+/-10uA,#3		3.8		4.2	
Output ”L” Voltage	Vol12j	Iol=10uA,#1,#2				0.2	
	Vol3j	Iol=10uA,#3				0.2	

1.8 TYPICAL APPLICATION CIRCUIT

This application circuit is only an example, and can not be guaranteed to work.



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

2. TM8722 Internal System Architecture

2.1 Power Supply

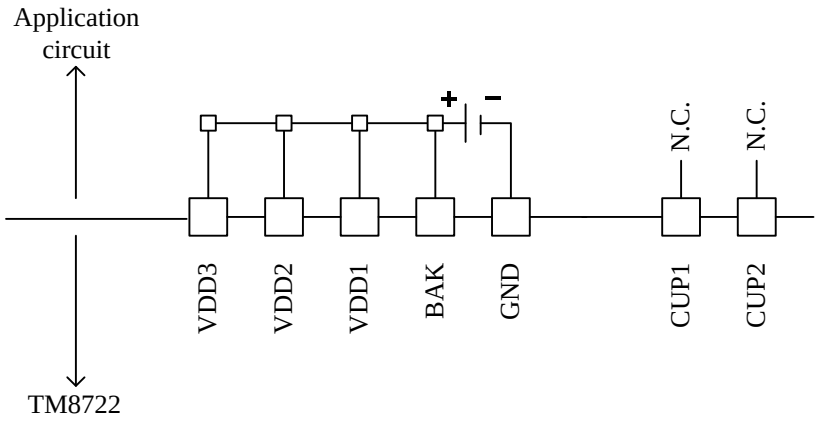
TM8722 can operate using 3 types of supply voltage: Ag, Li, and EXTV, which can be selected in mask option. The power supply circuitry selected will also provide the necessary voltage level to drive the LCD panel with different biases. The connection diagrams for 1/2 bias,1/3 bias and no bias applications are shown below.

2.1.1 Ag BATTERY POWER SUPPLY

Operating voltage range: 1.2V ~ 1.8V.

The connection diagrams for different LCD bias applications are shown below:

2.1.1.1 NO LCD BIAS NEEDED using a Ag BATTERY POWER SUPPLY



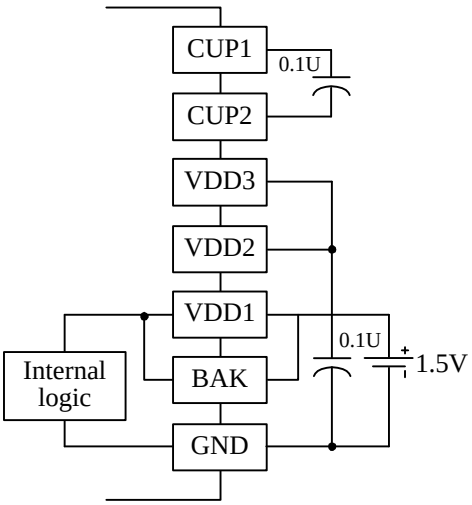
MASK OPTION table:

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

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

Note 2: The backup flag (BCF) is set to 1 in the initial reset cycle. When as the backup flag is set to 1, the driving capability of the oscillator circuit increases, it improves the oscillation conditions but the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 after the initial reset cycle. For the the backup flag, please refer to 3-5.

2.1.1.2 1/2 BIAS & STATIC using a AG BATTERY POWER SUPPLY



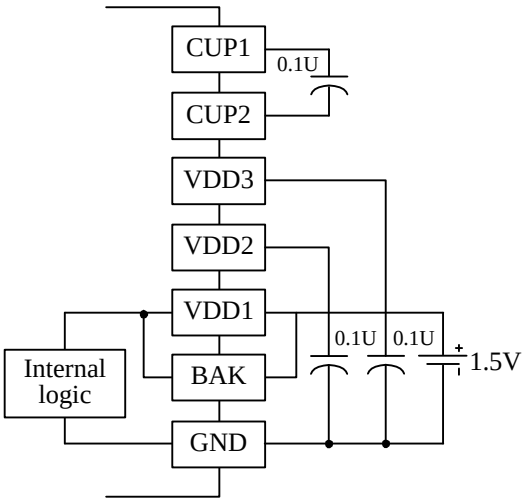
MASK OPTION table:

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

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

Note 2: The backup flag (BCF) is set to 1 in the initial reset cycle. When the backup flag is set to 1, the driving power of the oscillator circuit increases, it improves the oscillation conditions but the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 after the initial reset cycle. For the backup flag, please refer to 3-5.

2.1.1.3 1/3 BIAS using a AG BATTERY POWER SUPPLY



MASK OPTION table:

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

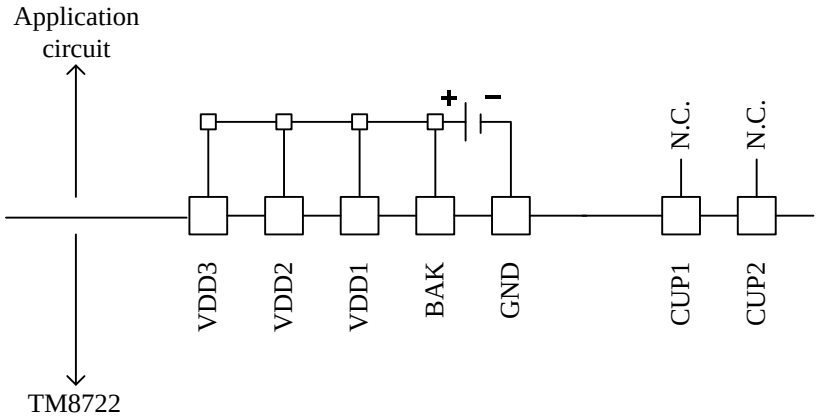
Note 1: The input/output ports operate between GND and VDD1.
Note 2: The backup flag (BCF) is set **to 1** in the initial reset cycle. When the backup flag is set **to 1**, the driving power of the oscillator circuit increases, it improves the oscillation conditions but the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset **to 0** after the initial reset cycle. For the backup flag, please refer to 3-5.

2.1.2 LI BATTERY POWER SUPPLY

Operating voltage range: 2.4V ~ 3.6V.

The connection diagrams for different LCD bias applications are shown below:

2.1.2.1 NO BIAS using a LI BATTERY POWER SUPPLY



MASK OPTION table:

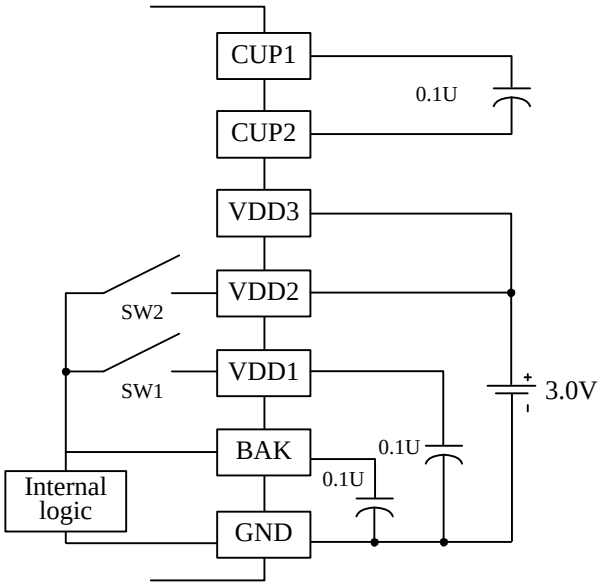
Mask Option name	Selected item
POWER SOURCE	(2) 3V BATTERY OR HIGHER
LCD BIAS	(3) NO BIAS

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

2.1.2.2 1/2 BIAS using a LI BATTERY POWER SUPPLY

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

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



MASK OPTION table:

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

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

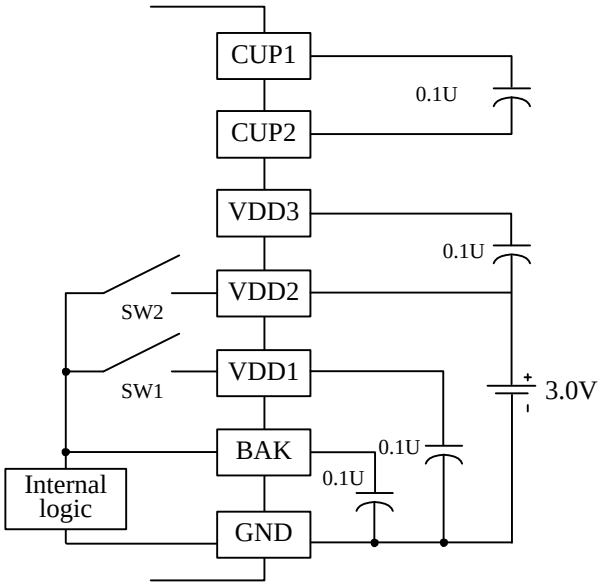
Note 2: The backup flag (BCF) is set to 1 in the initial reset cycle. When the backup flag is set to 1, the internal logic signal operated on VDD2 and the driving power of the oscillator circuit increases and the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 after the initial reset cycle. For the backup flag, please refer to 3-5.

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

2.1.2.3 1/3 BIAS using a LI BATTERY POWER SUPPLY

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

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



MASK OPTION table:

Mask Option name	Selected item
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: The backup flag (BCF) is set to 1 in the initial reset cycle. When the backup flag is set to 1, the internal logic signal operated on VDD2 and the driving power of the oscillator circuit increases and the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 after the initial reset cycle. For the backup flag, please refer to 3-5.

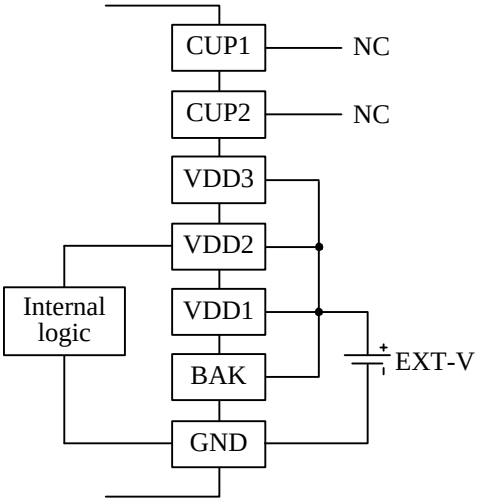
Note 3: The VDD1 level (≈□ 1/2 * VDD)in the off-state of SW1 is used as an intermediate voltage level for LCD driver.

2.1.3 EXTV POWER SUPPLY

Operating voltage range: 3.6V ~ 5.4V.

The connection diagrams for different LCD bias applications are shown below:

2.1.3.1 NO BIAS using an EXT-V BATTERY POWER SUPPLY



MASK OPTION table:

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

Note 1: The input/output ports operate between GND and VDD2.
Note 2: the backup flag (BCF) is reset to 0 in the initial reset cycle.
Note 3: When the backup flag is set to 1, the operating current increases.

2.1.3.2 1/2 BIAS using an EXT-V POWER SUPPLY

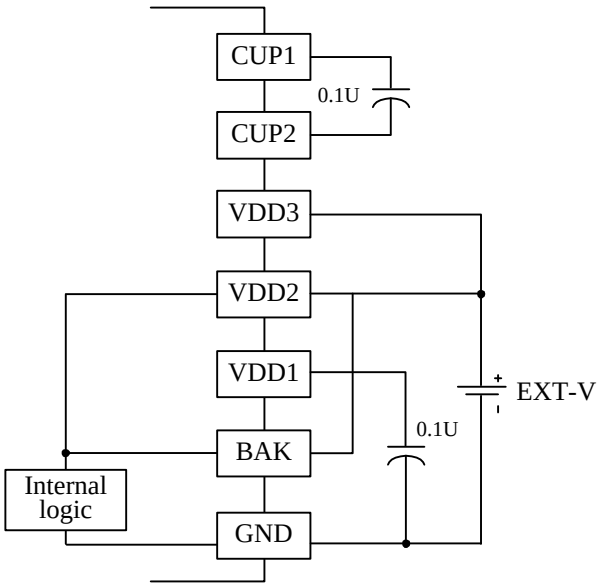
MASK OPTION table:

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

Note：

1. The input/output ports operate between GND and VDD2.
2. the backup flag (BCF) is reset to 0 in the initial reset cycle.
3. When the backup flag is set to 1, the operating current increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 in normal mode.

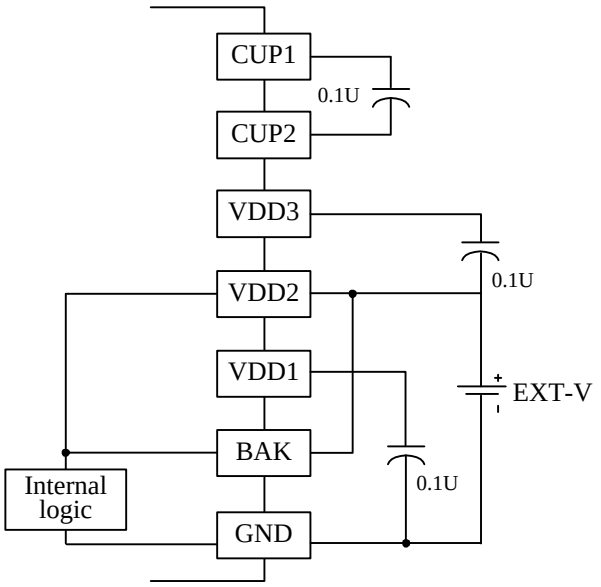
2.1.3.3 1/3 BIAS using an EXT-V POWER SUPPLY



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: the backup flag (BCF) is reset to 0 in the initial reset cycle.
Note 3: When the backup flag is set to 1, the operating current increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 in normal mode.



2.2 SYSTEM CLOCK

The clock oscillation circuitry consists of a XT clock (slow clock) oscillator and a CF clock (fast clock) oscillator. The output from this circuitry forms the clock source for the system clock generator (provides the necessary clock signals for the execution of instructions), the pre-divider (generates several clock signals with different frequencies to be used by the LCD driver),and for the frequency generator etc.

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

	PH0(per-divider)	BCLK(system clock)
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 THE SLOW CLOCK OSCILLATOR (XT CLOCK)

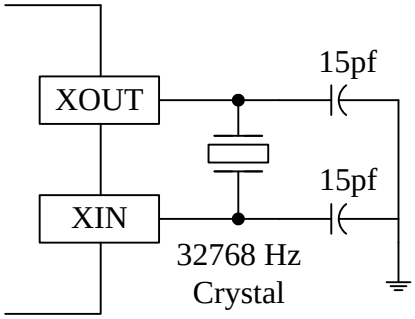
This clock oscillator provides the lower speed clock signals to the system clock generator, the pre-divider, the timer, the chattering prevention of the IO port and the LCD circuitry. This oscillator is disabled when the “fast clock only “option is selected in mask option, otherwise it is active all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

There are 2 types of oscillators that can be used as the slow clock oscillator, which can be selected in mask option :

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



(1) X'tal

When the backup flag (BCF) is set to 1, the oscillator operates in a state with a higher driving capability and the start-up time of the oscillator can be reduced as a result. However it increases the power consumption. Therefore, the backup flag should be reset unless required otherwise.

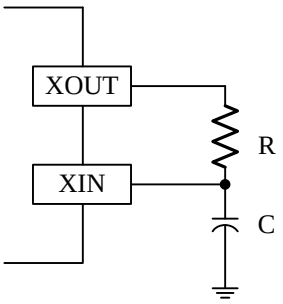
The following table shows the power consumption of Crystal oscillator under different conditions :

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

2.2.1.2 External RC oscillator (XT CLOCK)

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 THE FAST CLOCK OSCILLATOR (CF CLOCK)

The CF clock consists of 3 types of oscillators (selectable in mask option) which provides a faster clock sources to the system. In single clock operation (fast only), this oscillator provides the clock signals to the system clock generator, the pre-divider, the timer, the I/O port chattering prevention clock and the LCD circuitry. In dual clock operation, CF clock provides the clock signals to the system clock generator only.

When the dual clock option is selected in mask option, this oscillator is 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 but the clock source for other functions will still come in from the XT clock. The Halt mode, the stop mode and the execution of the SLOW instruction will stop this oscillator and the system clock (BCLK) will be switched to the XT clock.

There are 3 types of oscillators that can be used as the fast clock oscillator, which can be selected in mask option:

2.2.2.1 RC OSCILLATOR WITH EXTERNAL RESISTOR (CF CLOCK)

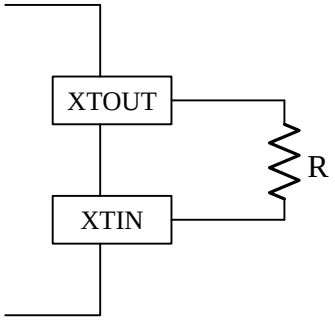
This type of oscillator can only be used in the "FAST only" option, however, this oscillator can not be used as the fast clock source in dual clock mode. When this oscillator is selected in mask option, the RC oscillator with internal RC will be disabled.

MASK OPTION table:

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

MASK OPTION table:

Mask Option name	Selected item
FAST CLOCK OSC TYPE FOR FAST ONLY OR DUAL	(1) or (2), don't care



External Resistor

2.2.2.2 RC OSCILLATOR WITH INTERNAL RESISTOR (CF CLOCK)

Two frequencies can be selected in this type of oscillator**:** 250 KHz and 500KHz. When the oscillator is used, CFOUT and CFIN pins should be left open.

This type of oscillator can be used both in the "FAST only" and the "DUAL clock" options.

MASK OPTION table:

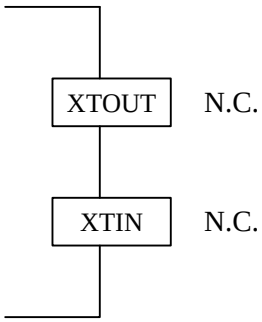
Mask Option name	Selected item
CLOCK SOURCE	(1) FAST ONLY & USE EXTERNAL RESISTOR or (4) DUAL

For 250 KHz output frequency:

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

For 500 KHz output frequency:

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



Internal RC

FREQUENCY RANGE OF INTERNAL RC OSCILLATOR

Option Mode	BAK	Min.	Typ.	Max.
250 KHz	1.5V	200 KHz	300 KHz	400 KHz
	3.0V	200 KHz	250 KHz	300 KHz
500KHz	1.5V	450 KHz	600 KHz	750 KHz
	3.0V	400 KHz	500 KHz	600 KHz

2.2.3 COMBINATION OF THE CLOCK SOURCES (CLCOK SWITCH CIRCUIT)

The clock switch circuit allows the selection of different clock inputs from XTOSC and CFOSC. There are three combinations of the clock sources that can be selected in mask option:

2.2.3.1 1 DUAL CLOCK

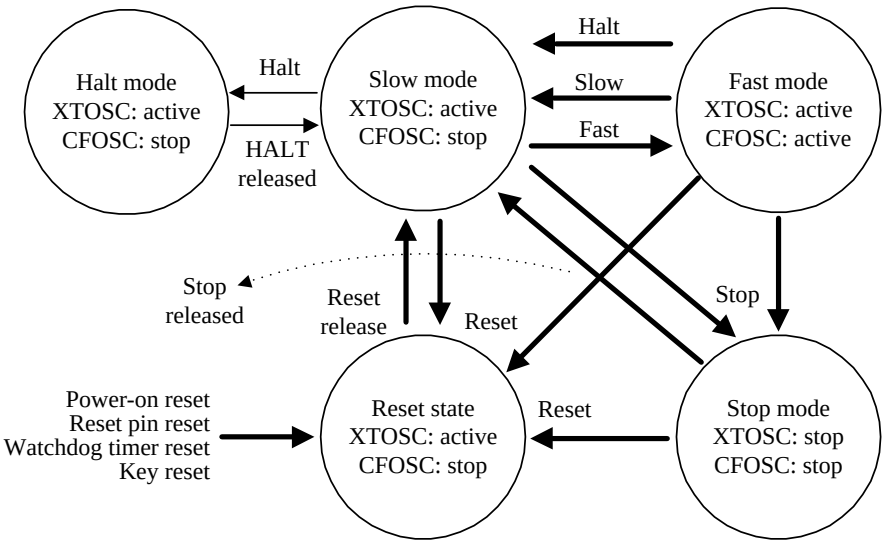
MASK OPTION table:

Mask Option name	Selected item
CLOCK SOURCE	(4) DUAL

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

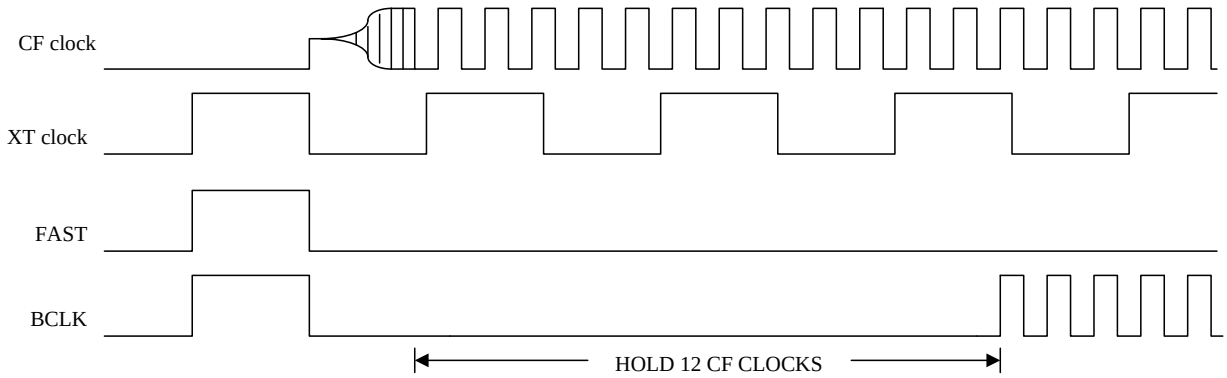
When this mode is selected in mask option, the clock source (BCLK) of the system clock generator will switch between XT clock and CF clock according to the user's program. When the HALT and STOP instructions are executed, the clock source (BCLK) will switch to the XT clock automatically.

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



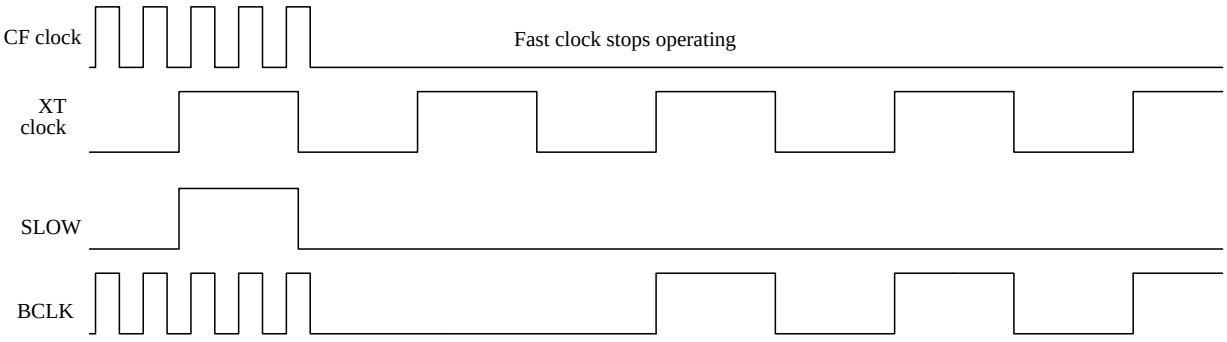
The state diagram of the dual clock mode is shown in the above figure.

After the execution of the FAST instruction, the system clock generator will hold for 12 CF clock cycles after the CF clock oscillator starts up and then BCLK will switch to the CF clock. It prevents the delivery of incorrect clock signals to the system clock in the start-up duration of the fast clock oscillator.



This figure shows the System Clock Switches from Slow to Fast

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



This figure 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 INTERNAL RESISTOR or (2) FAST ONLY & USE EXTERNAL RESISTOR

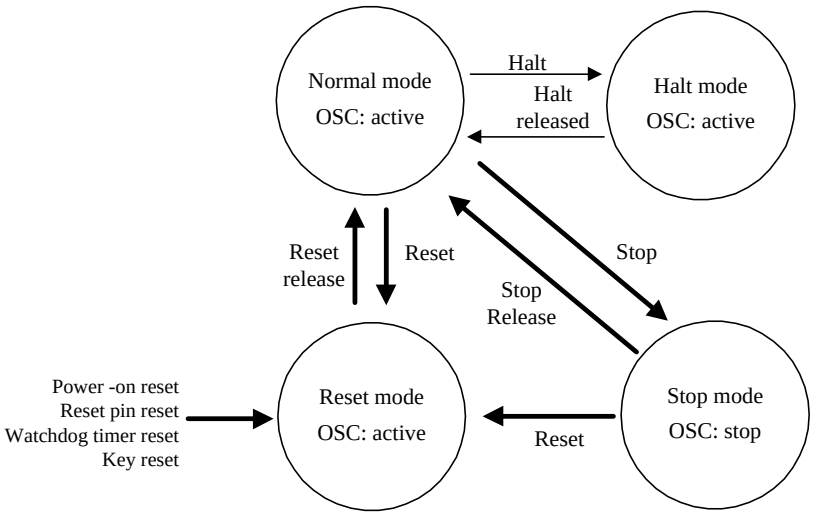
For slow clock oscillator only

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 can be selected in mask option in this mode. The FAST and SLOW instructions will be treated as the NOP instruction in this mode.

The backup flag (BCF) will be set to 1 automatically before the program enters the stop mode. It can ensure that the Crystal oscillator starts up in a favorable condition.

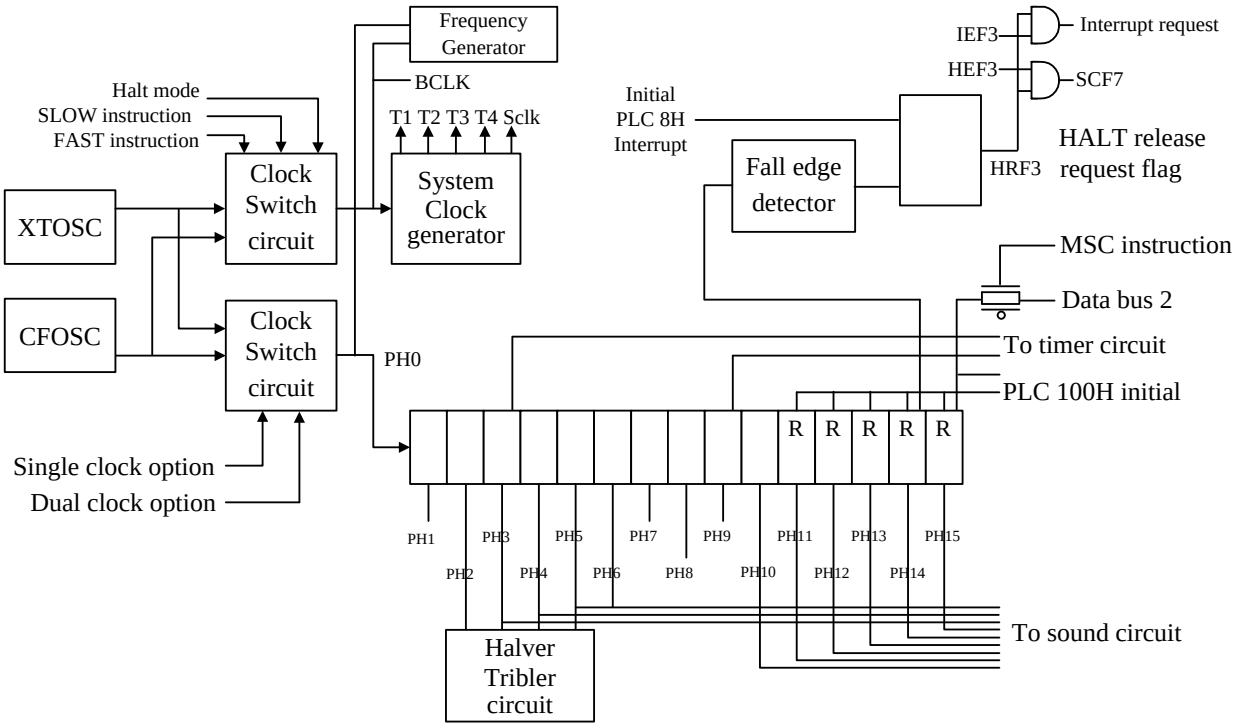


This figure shows the State Diagram of Single Clock Option

2.2.4 PREDIVIDER

The pre-divider is a 15-stage counter that receives the clock signals from the output of the clock switch circuitry (PH0). When PH0 changes from "H" level to "L" level, the content of this counter changes accordingly. The PH11 to PH15 of the pre-divider are reset to "0" when the PLC 100H instruction is executed or in the initial reset cycle.

The pre-divider delivers the signals to the halver/tripler circuit, the LCD driver, the sound generator and the I/O port chattering prevention function.



This figure shows the pre-divider and its peripherals

The falling edge of PH14 will set the halt mode release request flag (HRF3) to 1, and if the pre-divider interrupt enable mode (IEF3) was also set in advance, the interrupt comes from predivider will be accepted; and if the halt release enable mode (HEF3) was also set in advance, then the halt release request signal will be delivered and the start condition flag 7 (SCF7) in status register 3 (STS3) will be set.

The clock source of the pre-divider is PH0, there are 4 kinds of frequencies of PH0 that can be selected in mask option:

MASK OPTION table:

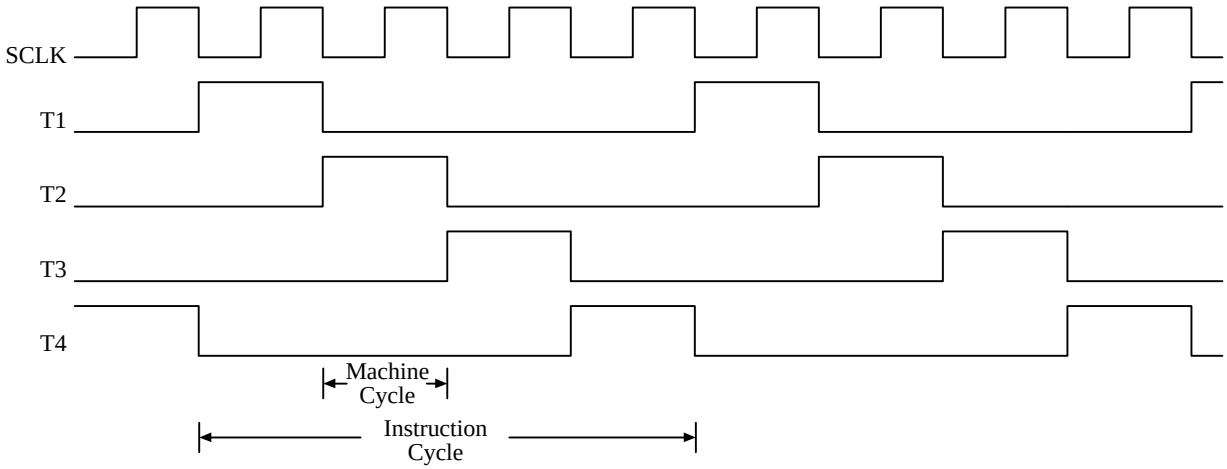
Mask Option name	Selected item
PH0 <-> BCLK FOR FAST ONLY	(1) PH0 = BCLK
PH0 <-> BCLK FOR FAST ONLY	(2) PH0 = BCLK/4
PH0 <-> BCLK FOR FAST ONLY	(3) PH0 = BCLK/8
PH0 <-> BCLK FOR FAST ONLY	(4) PH0 = BCLK/16

2.2.5 SYSTEM CLOCK GENERATOR

The system clock generator provides the necessary clock signals to control the execution of instructions.

The FAST and SLOW instructions can also be used to switch the clock input of the system clock generator.

The basic system clock is shown below:



2.3 PROGRAM COUNTER (PC)

The program counter is a 11-bit counter, which addresses the program memory (ROM) up to 2048 addresses.

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

  PC ← PC + 1

- When executing JMP instruction, subroutine call instruction (CALL), interrupt service routine or when reset occurs, the program counter (PC) will be loaded with the corresponding address in table 2-1.

  PC ← corresponding address shown in Table 2- 1

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

  PC ← current page (PC11) + specified address in the operand

- Return instruction (RTS)

  PC ← content of stack specified by the stack pointer

  Stack pointer ← stack pointer – 1

Table 2- 1

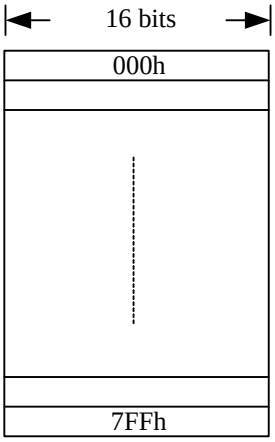
	PC10	PC9	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial reset	0	0	0	0	0	0	0	0	0	0	0
Interrupt 2 (INT pin)	0	0	0	0	0	0	1	0	0	0	0
Interrupt 0 (input port C)	0	0	0	0	0	0	1	0	1	0	0
Interrupt 1 (timer 1 interrupt)	0	0	0	0	0	0	1	1	0	0	0
Interrupt 3 (pre-divider interrupt)	0	0	0	0	0	0	1	1	1	0	0
Interrupt 4 (timer 2 interrupt)	0	0	0	0	0	1	0	0	0	0	0
Interrupt 5 (Key Scanning interrupt)	0	0	0	0	0	1	0	0	1	0	0
Interrupt 6 (RFC counter interrupt)	0	0	0	0	0	1	0	1	0	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: the **11** Low-order bits of instruction operand.

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

2.4 PROGRAM/TABLE MEMORY

The built-in mask ROM is organized into 2048 x 16 bits.



Both instruction ROM (PROM) and table ROM (TROM) share 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(16 - N) bytes (N = 1 ~ 16).

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

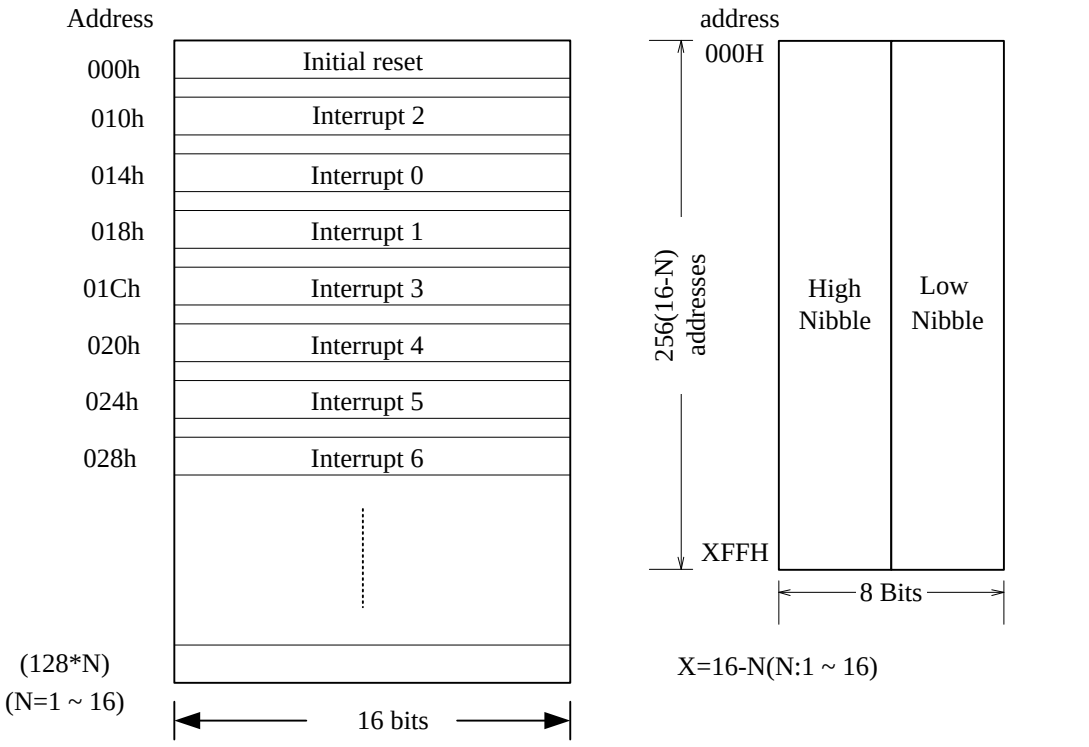
The partition of memory space is defined in mask option, the table is shown below :

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	3840
INSTRUCTION ROM <-> TABLE ROM	2 (N=2)	256	3584
INSTRUCTION ROM <-> TABLE ROM	3 (N=3)	384	3328
INSTRUCTION ROM <-> TABLE ROM	4 (N=4)	512	3072
INSTRUCTION ROM <-> TABLE ROM	5 (N=5)	640	2816
INSTRUCTION ROM <-> TABLE ROM	6 (N=6)	768	2560
INSTRUCTION ROM <-> TABLE ROM	7 (N=7)	896	2304
INSTRUCTION ROM <-> TABLE ROM	8 (N=8)	1024	2048
INSTRUCTION ROM <-> TABLE ROM	9 (N=9)	1152	1792
INSTRUCTION ROM <-> TABLE ROM	A (N=10)	1280	1536
INSTRUCTION ROM <-> TABLE ROM	B (N=11)	1408	1280
INSTRUCTION ROM <-> TABLE ROM	C (N=12)	1536	1024
INSTRUCTION ROM <-> TABLE ROM	D (N=13)	1664	768
INSTRUCTION ROM <-> TABLE ROM	E (N=14)	1792	512
INSTRUCTION ROM <-> TABLE ROM	F (N=15)	1920	256
INSTRUCTION ROM <-> TABLE ROM	G (N=16)	2048	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), interrupt 4 address (020H), interrupt 5 address (024H), and interrupt 6 address (028H) in the program memory.



This figure shows the Organization of ROM

2.4.2 TABLE ROM (TROM)

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

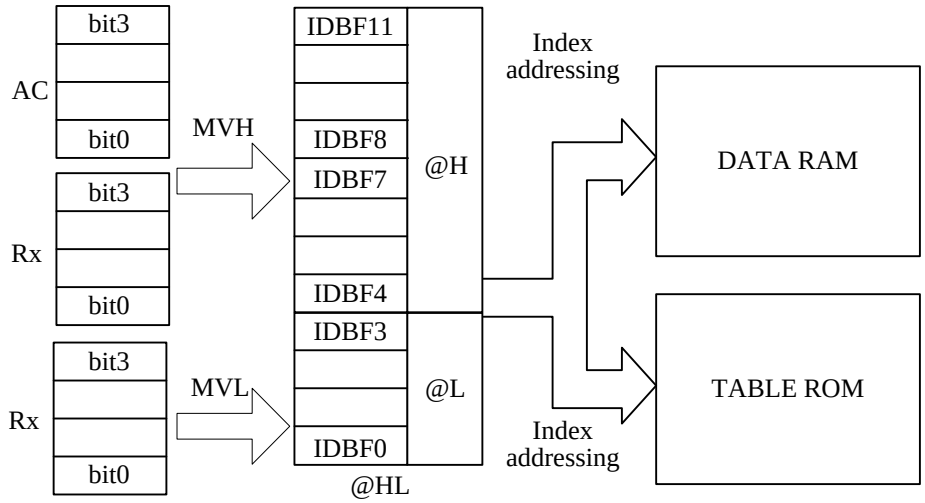
2.5 INDEX ADDRESS REGISTER (@HL)

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. The execution of the MVL instruction will load the content of the specified data memory to the lower nibble of the index register (@L). In the same manner, the execution of the MVH instructions will 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				@H register				@L register			
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 address the whole range of the table ROM and data memory.



This figure shows the diagram of the index address register

2.6 STACK REGISTER (STACK)

Stack is a special data structure that follows the First-In-Last-Out rule. It is used to save the contents of the program counter sequentially during subroutine calls or the execution of the interrupt service routines.

The contents of the stack registers are returned sequentially to the program counter (PC) when return instruction (RTS) is executed.

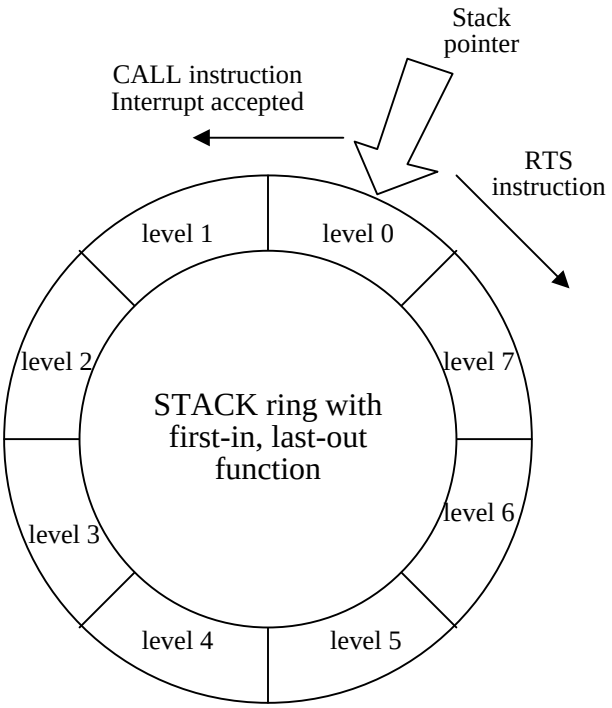
The stack registers are organized into 11 bits by 8 levels with no overflow/underflow flag; Therefore only 8 levels of subroutine call or interrupt are allowed (If stack is full, and either interrupt occurs or subroutine call executes, the first level will be overwritten).

Once a subroutine call or interrupt causes the stack registers (STACK) to 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 registers (STACK) are returned sequentially to the program counter (PC) during execution of the RTS instruction.

Once a RTS instruction causes the stack register (STACK) to underflow, the stack pointer will return to level 7 and the content of the level 7 stack will be restored to the program counter.

The following figure shows the diagram of the stack.



2.7 DATA MEMORY (RAM)

The static RAM is organized into 128 addresses x 4 bits and is used to store data.

The data memory may be accessed by two methods:

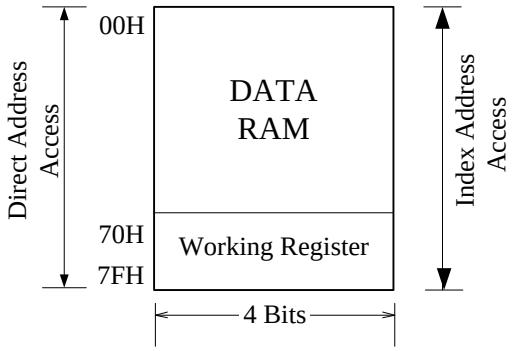
1. Direct addressing mode

The address of the data memory is specified by the instruction and the addressing range is from 00H to 7FH.

2. Index addressing mode

The index address register (@HL) can address the data memory from 00H to 07FH.

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



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

2.8 WORKING REGISTER (WR)

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

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

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

   MWR Rx, Ry; MRW Ry, Rx

3. To decode (or directly transfer) the contents a 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 is a circuitry that performs arithmetic and logic operations. The ALU provides the following functions:

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

2.11 BINARY CONVERT TO DECIMAL (BCD)

Decimal format is another number format supported by TM8722. When the content of the data memory is assigned as decimal format, it is necessary to convert the results into decimal format after the execution of ALU instructions. When the decimal converting operation is in execution, all the operands (including the content of the data memory (RAM), accumulator (AC), immediate data, and look-up table) should be in the decimal format, otherwise the results of conversion will be incorrect.

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

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

Example 1:

```
LDS     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             ; The content of the data memory at addresses 10H and AC are
                        ; binary-added
                        ; the result is loaded into AC & the data memory address
                        ; 10H. (R10 = AC = A(binary), CF = 0)
DAA*    10h             ; Convert the content of AC into decimal format.
                        ; The result in the data memory at address 10H is”0”and in
                        ; the CF is “1”. This represents the decimal number”10”.
```

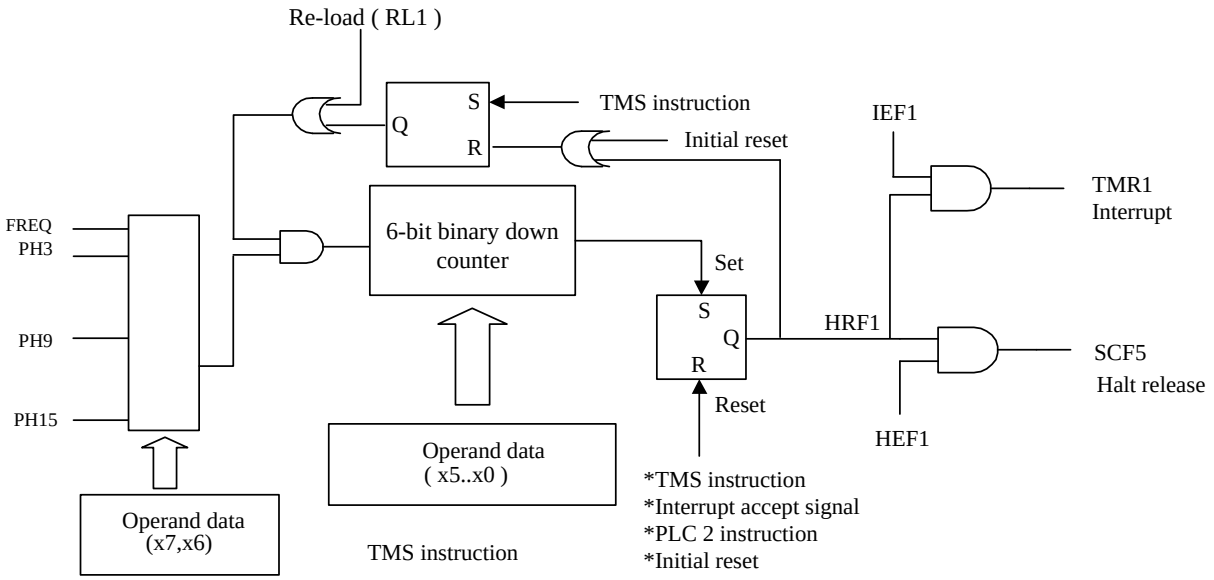
Instructions DAS, DAS*, DAS @HL can convert the data from binary format to decimal format after any subtraction operation. The conversion rules are shown in the following table and illustrated in Example 2.

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

Example 2:

```
LDS     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 occurrs.
SUB* 10h        ; The Content of the data memory at address 10H is binary-subtracted;
                ; the result is loaded into data memory address
                ; 10H. (R10 = AC = F(binary), CF = 0)
DAS* 10h        ; Convert the content of the data memory address at 10H to decimal format.
                ; The result in the data memory address at 10H is”9”and in
                ;  the CF is “0”. This represents the decimal number”–1”.
```

2.12 TIMER 1 (TMR1)



This figure shows the TMR1 organization.

2.12.1 NORMAL OPERATION

TMR1 consists of a programmable 6-bit binary down counter, which can be loaded and enabled by executing the TMS and the TMSX instruction.

Once the TMR1 counts down to 00h, it will generate 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, an interrupt is generated.

When HRF1 = 1, if the IEF1 = 0 and the TMR1 halt release enabled (HEF1) = 1, the program will exit from the halt mode (if CPU is in the 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 a watchdog reset occurs, the clock source of TMR1 will stay with the former selection.

The following table shows the definition of each operand bit in TMR1's 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 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 clock source of TMR1 is PH3
   TMR1 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the clock source of TMR1 is PH9
   TMR1 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the clock source of TMR1 is PH15
   TMR1 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
   Set value: Decimal number of the 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 clock source of TMR1 is FREQ
   TMR1 set time = (Set value + error) * 1/FREQ (KHz) (ms).
   FREQ: Please refer to section 3-3-4.

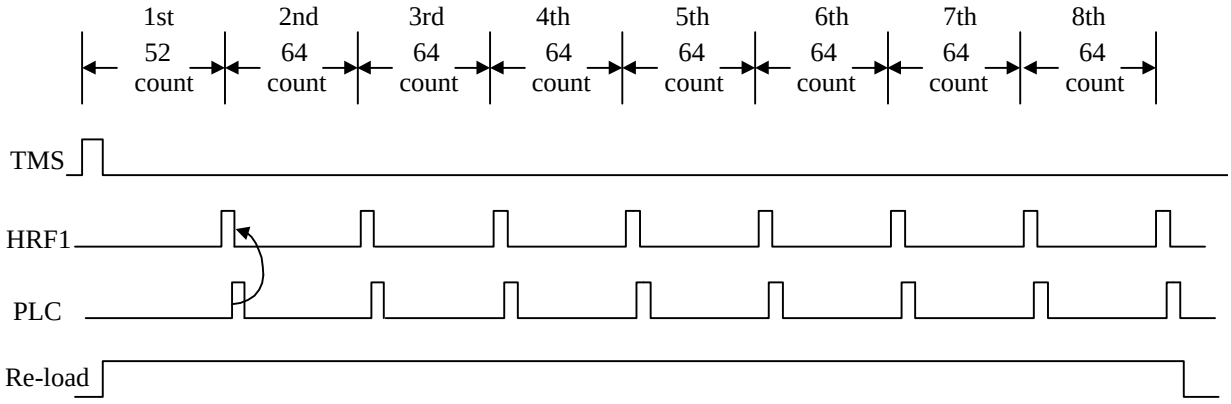
2.12.2 RE-LOAD OPERATION

TMR1 provides a re-load function which can last for a time interval longer than 3Fh. The SF 80h instruction enables the re-load function and RF 80h instruction disables it.

When the re-load function is enabled, TMR1 will count down with an initial data 3Fh automatically if TMR1's underflow occurs. Once the re-load function was disabled, the occurrence of TMR1's underflow will stop TMR1 immediately. During this operation, the program must use the halt release request flag or an interrupt to calculate the desired counting value.

- It is necessary to execute the TMS or the TMSX instruction to initiate the count value before the re-load function is enabled, otherwise TMR1 will count down with an unknown value.
- Do not disable the re-load function before the last expected halt release or interrupt occurs. If the TMS related instructions are executed before a halt release or an interrupt occurs, 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 initial count down value of TMR1 to 52 and start counting, then enable the TMR1 halt release or interrupt function. Before the first underflow occurs, enable the re-load function. TMR1 will continue operating even though TMR1 underflow occurs. When a halt release or an interrupt occurs, clear the HRF1 flag by executing the PLC instruction. After a halt release or an interrupt occurs 8 times, disable the re-load function, and then the counting is completed.

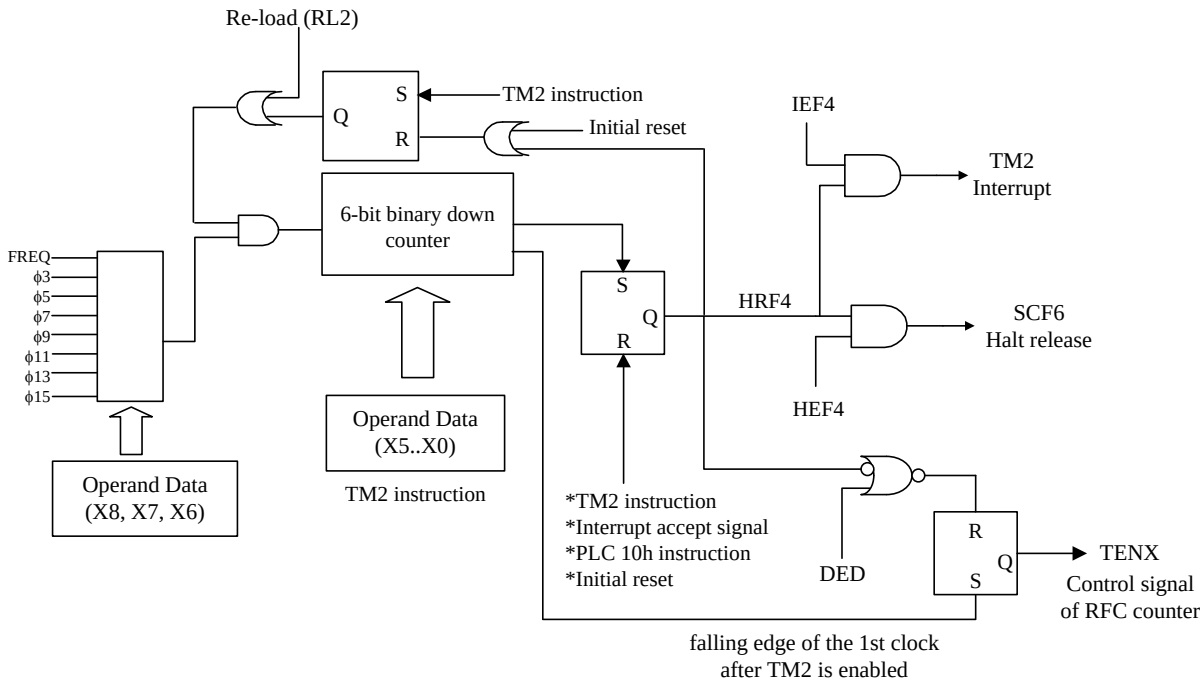


In the following example, S/W enters the halt mode to wait for the underflow of TMR1.

```
        LDS     0, 0            ; initiate the underflow counting register
        PLC     2
        SHE     2               ; enable the HALT release caused by TMR1
        TMSX    34h             ; initiate TMR1 value (52) and the clock source is ϕ9
        SF      80h             ; enable the re-load function
RE_LOAD:
        HALT
        INC*    0               ; increment 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 TIMER 2 (TMR2)

The following figure shows the TMR2 organization.



2.13.1 NORMAL OPERATION

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

Once TMR2 counts down to 00h, it stops counting, and then generates an underflow signal and the halt release request flag 4 (HRF4) will be set to 1.

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

After power on reset, the default clock source of TMR2 is PH7.

If a watchdog reset occurrs, the clock source of TMR2 will stay with the former selection.

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

OPCODE	Select clock			Initiate value of timer					
TM2X X	X8	X7	X6	X5	X4	X3	X2	X1	X0
TM2 Rx	0	AC3	AC2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
TM2 @HL	0	bit7	bit6	bit5	Bit4	bit3	bit2	bit1	bit0

The following table shows the clock source setting for TMR2

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

Notes:

1. When the clock source of TMR2 is PH3
TMR2 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)

2. When the clock source of TMR2 is PH9
TMR2 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)

3. When the clock source of TMR2 is PH15
TMR2 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)

4. When the clock source of TMR2 is PH5
TMR2 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)

5. When the clock source of timer is PH7
TMR2 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)

6. When the clock source of TMR2 is PH11
TMR2 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)

7. When the clock source of TMR2 is PH13
TMR2 set time = (Set value + error) * 8192 * 1/fosc (KHz) (ms)
Set value: Decimal number of the timer set value
error: the tolerance of set value, 0 < error <1.
fosc: Input of the predivider
PH3: The 3rd stage output of the predivider
PH5: The 5th stage output of the predivider
PH7: The 7th stage output of the predivider
PH9: The 9th stage output of the predivider
PH11: The 11th stage output of the predivider
PH13: The 13th stage output of the predivider
PH15: The 15th stage output of the predivider

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

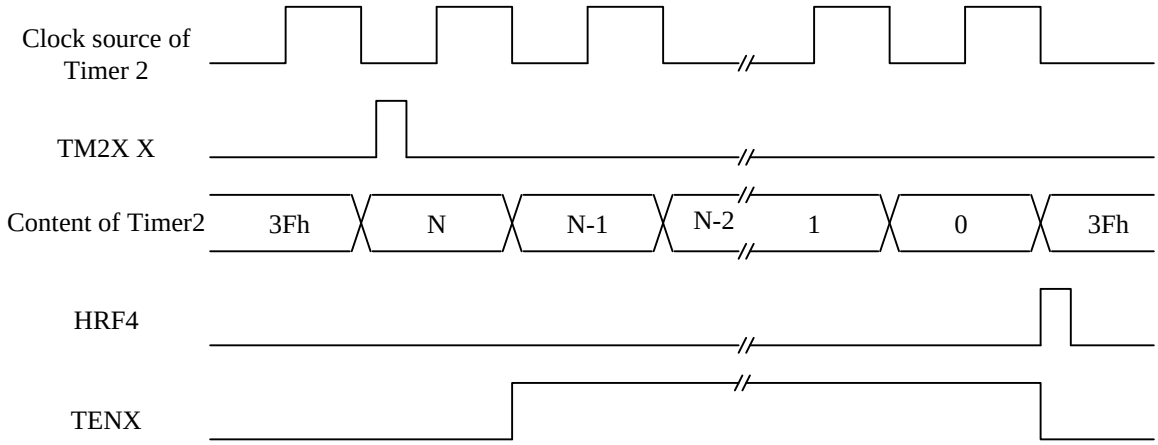
2.13.2 RE-LOAD OPERATION

TMR2 also provides a re-load function which works in the same fashion as TMR1. The instruction SF2 1 enables the re-load function; the instruction RF2 1 disables it.

2.13.3 TIMER 2 (TMR2) IN RESISTOR TO FREQUENCY CONVERTER (RFC)

TMR2 also controls the operation of the RFC function.

TMR2 sets TENX flag to 1 to enable the RFC counter. Once TMR2 underflows, the TENX flag will be reset to 0 automatically. In behaving this way, Timer 2 can set an accurate time period without setting a value error like the other operations of TMR1 and TMR2. Please refer to section 2-16 for detail information on controlling the RFC counter. The following figure shows the operating timing of TMR 2 in RFC mode.



TMR2 can also controll the RFC function through the re-load function.

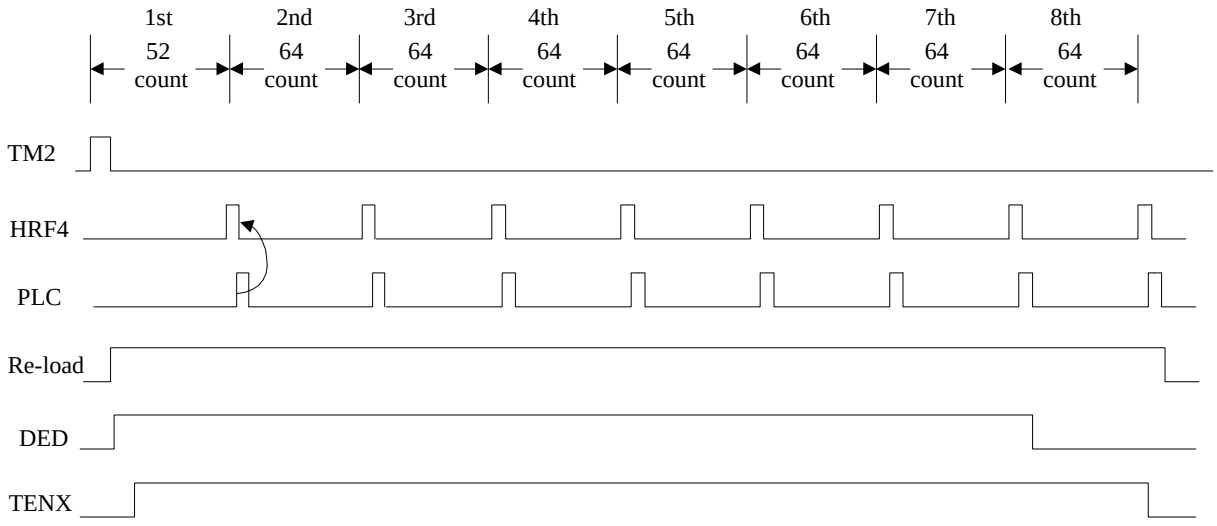
The SF2 1h instruction enables the re-load function, and the DED flag should be set to 1 by the SF2 2h instruction. Once the DED flag is set to 1, the TENX flag will not be cleared to 0 when TMR2 underflows (but HRF4 will be set to1). The DED flag must be cleared to 0 by executing RF2 2h instruction before the last HRF4 occurs; thus, the TENX flag will be reset to 0 when the last HRF4 flag signal is delivered. After the last underflow (HRF4) of TMR2 occurs, disable the re-load function by the executing RF2 1h instruction.

For example, if the expected count value is 500, it will be divided as 52+7*64.

1. Set the initial value of TMR2 to 52 and start counting.
2. Enable the TMR2 halt release or the interrupt function.
3. Before the first underflow occurs, enable the re-load function and set the DED flag. The TMR2 will continue counting even if TMR2 underflows.
4. When a halt release or an interrupt occurs, clear the HRF4 flag by PLC instruction and increment the counting value to count the underflow times.
5. When a halt release or an interrupt occurs at the 7^{th} time, reset the DED flag.
6. When a halt release or an interrupt occurs at the 8^{th} time, disable the re-load function, and then the counting is completed.

In the following example, S/W enters the halt mode to wait for the underflow of TM2

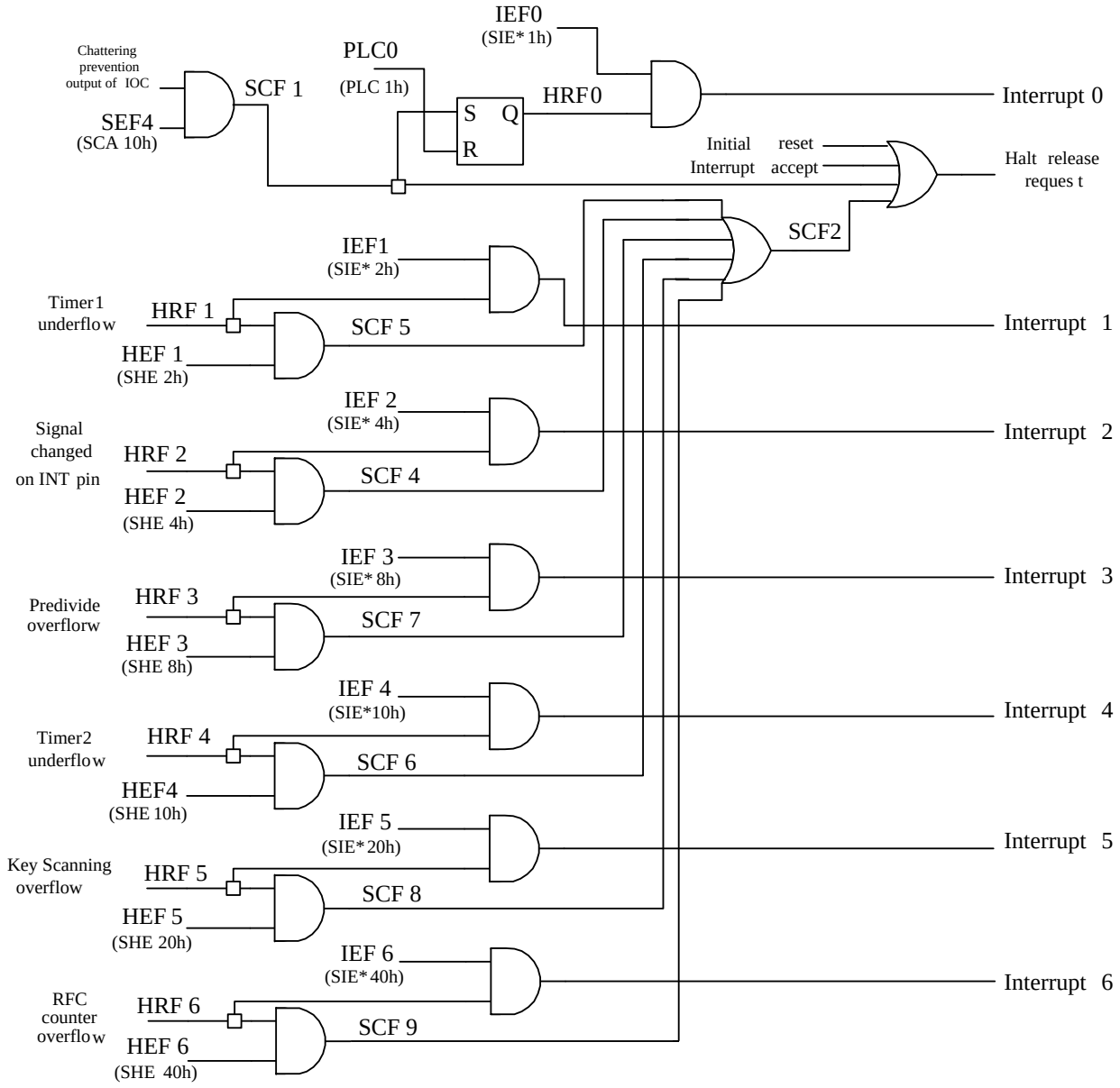
```
        LDS     0,0             ;initiate the underflow counting register
        PLC     10h
        SHE     10h             ;enable the halt release caused by TM2
        SRF     19h             ;enable RFC, and controlled by TM2
        TM2X    34h             ;initiate the TM value(52) and the clock source is ϕ9
        SF2     3h              ;enable the re-load function and set the DED flag to 1
RE_LOAD:
        HALT
        INC*    0               ;increment the underflow counter
        PLC     10h             ;clear HRF4
        LDS     20h, 7
        SUB     0               ;when halt is released at the 7th time, reset the DED flag
        JNZ     NOT_RESET_DED
        RF2     2               ;reset the DED flag
NOT_RESET_DED:
        LDA     0               ;restore the underflow counter to AC
        JB3     END_TM1         ;if the TM2 underflow counter is equal to 8, exit this subroutine
        JMP     RE_LOAD
END_TM1:
        RF2     1               ;disable the re-load function
```



This figure shows the operating timing of TMR2 re-load function for RFC

2.14 STATUS REGISTER (STS)

The status register (STS) is a 4 bit register and contains 4 types: status register 1 (STS1) to status register 4 (STS4). The following figure shows the configuration of the start condition flags for TM8722.



2.14.1 STATUS REGISTER 1 (STS1)

Status register 1 (STS1) consists of 2 flags:

1. Carry flag (CF)
   The carry flag is used to save the 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 transfers the data of the status register 1 (STS1) to the accumulator (AC) and the data memory (RAM).

4. The MRA instruction transfers the data of the data memory (RAM) to the status register 1 (STS1)

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

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

2.14.2 STATUS REGISTER 2 (STS2)

Status register 2 (STS2) consists of the start condition flag 1, 2 (SCF1, SCF2) and the backup flag.

The MSB instruction transfers the data of the status register 2 (STS2) to the accumulator (AC) and the data memory (RAM). The status register 2 (STS2) is read-only.

The following table shows the bit pattern of each flag in the 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 SCF4,5,6,7,9	Halt release caused by the IOC port	The back up mode status
NA	Read only	Read only	Read only

Start condition flag 1 (SCF1)
When a signal change occurs on port IOC due to that the execution of SCA instruction and the halt mode is released as a result, SCF1 will be set. Executing the SCA instruction will cause SCF1 to be reset to 0.

Start condition flag 2 (SCF2)
When factors other than port IOC cause the halt mode to be released, SCF2 will be set to1. Also, if one or more start condition flags in SCF4, 5, 6, 7, 9 are set to 1, SCF2 will be set to 1 synchronously. When all the flags in SCF4, 5, 6, 7, 9 are cleared, the start condition flag 2 (SCF2) is reset to 0.

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

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

2.14.3 STATUS REGISTER 3 (STS3)

When the halt mode is released by start condition flag 2 (SCF2), the status register 3 (STS3) will update in the corresponding status flag wherein the cause for the release of the halt mode.

Status register 3 (STS3) consists of 4 flags:

1. The Start condition flag 4 (SCF4)

If the halt release enable flag 2 (HEF2) is set, the start condition flag 4 (SCF4) will be set to 1 when the signal change on the INT pin causes the halt release request flag 2 (HRF2) to be outputted.

There are two methods to reset the start condition flag 4 (SCF4), the first method is to execute the PLC instruction to reset the halt release request flag 2 (HRF2), the other method is to execute the SHE instruction to reset the halt release enable flag 2 (HEF2).

2. The Start condition flag 5 (SCF5)

If the halt release enable flag 1 (HEF1) is set, the Start condition flag 5 (SCF5) will be set when an underflow signal from Timer 1 (TMR1) causes the halt release request flag 1 (HRF1) to be outputted.

There are two methods to reset the start condition flag 5 (SCF5), the first method is to execute the PLC instruction to reset the halt release request flag 1 (HRF1), the other method is to execute the SHE instruction to reset the halt release enable flag 1 (HEF1).

3. The Start condition flag 7 (SCF7)

If the halt release enable flag 3 (HEF3) was set beforehand, the Start condition flag 7 (SCF7) will be set when an overflow signal from the pre-divider causes the halt release request flag 3 (HRF3) to be outputted.

There are two methods to reset the start condition flag 7 (SCF7), the first method is to execute the PLC instruction to reset the halt release request flag 3 (HRF3), the other method is to execute the SHE instruction 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 the status register 3 (STS3) to the accumulator (AC) and the data memory (RAM).

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

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 7 (SCF7)	15th stage of the pre-divider	Start condition flag 5 (SCF5)	Start condition flag 4 (SCF4)
Halt release caused by pre-divider overflow		Halt release caused by TMR1 underflow	Halt release caused by INT pin
Read only	Read only	Read only	Read only

2.14.4 STATUS REGISTER 3X (STS3X)

When the halt mode is released by the start condition flag 2 (SCF2), the status register 3X (STS3X) will update in the corresponding status flag wherein the cause for the release of the halt mode.

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

1. The Start condition flag 8 (SCF8)

If the halt release enable flag 5 (HEF5) was set beforehand, the SCF8 flag will be set to 1 when a signal change on KI1~4 pins (KI1~4=1 in LED mode / KI1~4=0 in LCD mode) causes the halt release request flag 5 (HRF5) to be outputted.

There are two methods to reset the start condition flag 8 (SCF8), the first method is to execute the PLC instruction to reset the halt release request flag 5 (HRF5), the other method is to executie the SHE instruction to reset the halt release enable flag 5 (HEF5).

2. The Start condition flag 6 (SCF6)

If the halt release enable flag 4 (HEF4) was set beforehand, the SCF6 flag will be set to 1 when an underflow signal from timer 2 (TMR2) causes the halt release request flag 4 (HRF4) to be outputted.

There are two methods to reset the start condition flag 6 (SCF6), the first method is to execute the PLC instruction to reset the halt release request flag 4 (HRF4), the other method is to execute the SHE instruction to reset the halt release enable flag 4 (HEF4).

3. The Start condition flag 9 (SCF9)

If the halt release enable flag 9 (HEF9) was set beforehand, the SCF9 flag will be set to 1 when a finish signal from mode 3 of the RFC function causes the halt release request flag 6 (HRF6) to be outputted. In this case, the 16-bit counter of the RFC function will be controlled by CX pin; please refer to 2-16-9.

There are two methods to reset the start condition flag 9 (SCF9), the first method is to execute the PLC instruction to reset the halt release request flag 6 (HRF6), the other method is to execute the SHE instruction to reset the halt release enable flag 6 (HEF6).

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

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

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 9 (SCF9)	NA	Start condition flag 6 (SCF6)	Start condition flag 8 (SCF8)
Halt release caused by RFC counter finish		Halt release caused by TMR2 underflow	Halt release caused by SKI underflow
Read only	Read only	Read only	Read only

2.14.5 STATUS REGISTER 4 (STS4)

The Status register 4 (STS4) consists of 3 flags:

1. The System clock selection flag (CSF)

The system clock selection flag (CSF) shows which clock source of the system clock generator is selected by the system. Executing SLOW instruction will change the clock source (BCLK) of the system clock generator to the slow speed oscillator (XT clock) and the system clock selection flag (CSF) will be reset to 0. Executing FAST instruction will change the clock source (BCLK) of the system clock generator to the fast speed oscillator (CF clock) and the system clock selection flag (CSF) will be set to 1. For the operation of the system clock generator, refer to 3-3.

2. The Watchdog timer enable flag (WTEF)

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

3. The Overflow flag of the 16-bit counter of RFC (RFOVF)

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

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

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

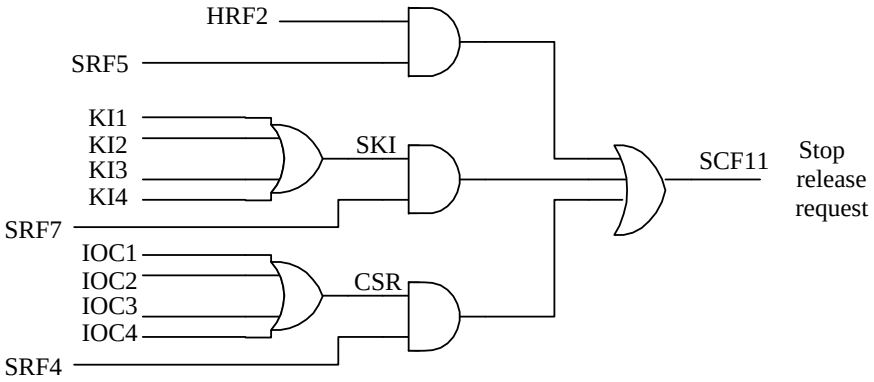
Bit 3	Bit 2	Bit 1	Bit 0
Reserved	The overflow flag of 16-bit counter of RFC (RFVOF)	Watchdog timer Enable flag (WTEF)	System clock selection flag (CSF)
Read only	Read only	Read only	Read only

2.14.6 START CONDITION FLAG 11 (SCF11)

The 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 via the pins which are defined as input mode in the IOC port circuit. It causes the stop release flag of the IOC port circuit (CSR) to output, the stop release enable flag 4 (SRF4) has to be set beforehand.
- A high level signal comes from the OR-ed output of the signal latch via KI1~4 pins. It causes the stop release flag of the Key Scanning circuit (SKI) to output, the stop release enable flag 4 (SRF7) has to be set beforehand.
- The signal change from the INT pin causes the halt release flag 2 (HRF2) to output, the stop release enable flag 5 (SRF5) has to be set beforehand.

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



The stop release flags (SKI, CSR, HRF2) can be set by the stop release enable flags (SRFx) and these flags should be cleared before the MCU enters into the stop mode. All of the IOC port pins have to be defined as the input mode and remain in 0 state before the MCU can enter tinto the STOP mode, otherwise the program can not enter into the STOP mode.

The SRE instruction is used to set or reset the stop release enable flags (SRF4,5,7).

The following table shows the stop release request flags

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

2.15 CONTROL REGISTER (CTL)

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

2.15.1 CONTROL REGISTER 1 (CTL1)

The control register 1 (CTL1) is a 1-bit register:

1. Switch enable flag 4 (SEF4)

It stores the status of the input signal change on IOC pins which is defined as input mode that causes the halt mode or the stop mode to be released.

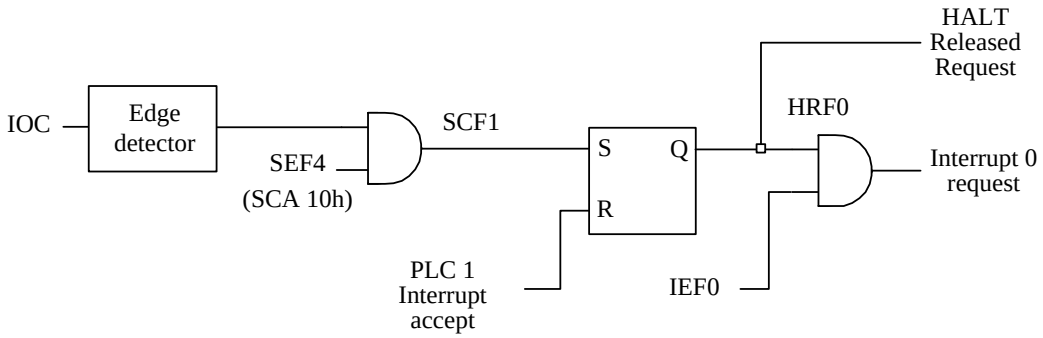
Executing a SCA instruction can set or reset these flags.

The following table 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 the IOC port
Write only

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



2.15.1.1 The Setting for releasing the Halt Mode

If SEF4 is set to 1, a signal change on IOC port will cause the halt mode to be released and set SCF1 to 1. Because the signal change on the IOC port indicates an ORed output of the IOC1~4, it is necessary to keep the rest of input pins in " 0 " state when one of the input pin of IOC port is changing.

2.15.1.2 The Setting for releasing the Stop Mode

If SRF4 and SEF4 are set, the stop mode will be released and set the SCF1 when a high level signal is applied to one of the input mode pins of the IOC port and the other pins stay in "0" state.

After the stop mode is released, TM8722 enters the halt mode.

The high level signal must hold for a period long enough to cause the chattering prevention circuitry of the 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.

2.15.1.3 Interrupt for CTL1

The control register 1 (CTL1) performs the following functions by the execution of the SIE instruction to enable the interrupt function.

An input signal change on the input pins of IOC port will cause MCU to deliver the SCF1 flag when SEF4 has been set to 1 by executing the SCA instruction. After the SCF1 flag is delivered, the halt release request flag (HRF0) will be set to 1.

In this case, if the interrupt enable flag 0 (IEF0) was set to 1 by executing the SIE instruction beforehand, it will also deliver the interrupt request flag 0 (interrupt 0) to interrupt the program. Once the interrupt 0 is accepted by MCU, more interrupt0 requests come from at later time will not be accepted until the inhibition is released by executing the SCA instruction. Please refer to 2-16-1-1

2.15.2 CONTROL REGISTER 2 (CTL2)

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

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

When the halt release enable flag 6 (HEF6) is set, the stop counting signal from the 16-bit counter of RFC will cause the halt mode to be released. In the same manner, when HEF1 to HEF4 are set to 1, the following conditions will cause the halt mode to be released: for example, an underflow signal from TMR1, the signal change at the INT pin, an overflow signal from the pre-divider and an underflow signal from TMR2, and a 'H' signal from OR-ed output of KI1~4 latch signals.

When the stop release enable flag 5 (SRF5) and the HEF2 are set, a signal change on the INT pin can cause the stop mode to be released.

When the stop release enable flag 7 (SRF7) and the HEF5 are set, the 'H' signal from OR-ed output of K1~4 latch signals can cause the stop mode to be released.

2.15.3 CONTROL REGISTER 3 (CTL3)

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

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

Interrupt enable flag	IEF6	IEF5	IEF4
Interrupt request flag	Enable the interrupt request caused by RFC function (HRF6)	Enable the interrupt request caused by Key Scanning (HRF5)	Enable the interrupt request caused by TMR2 underflow (HRF4)
Interrupt flag	Interrupt 6	Interrupt 4	Interrupt 4
Interrupt enable flag	IEF3	IEF2	IEF1
Interrupt request flag	Enable the interrupt request caused by predivider overflow (HRF3)	Enable the interrupt request caused by INT pin (HRF2)	Enable the interrupt request caused by TM1 underflow (HRF1)
Interrupt flag	Interrupt 3	Interrupt 2	Interrupt 1
Interrupt enable flag	IEF0		
Interrupt request flag	Enable the interrupt request caused by IOC port signal to be changed (HRF0)		
Interrupt flag	Interrupt 0		

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

2.15.4 CONTROL REGISTER 4 (CTL4)

The Control register 4 (CTL4) is a a 3-bit register. It is set/reset by the SRE instruction.

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

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

When the stop release enable flag 7 (SRF7) is set to 1, an input signal change on pin KI1~4 will cause the stop mode to be released. In the same manner, when SRF4 (SRF3) and SRF5 are set to 1, an input signal change on the input mode pins of the IOC port and a signal changed on the INT pin will cause the stop mode to be released respectively.

Example:

This example illustrates the stop mode released by port IOC, KI1~4 and INT pins. Assuming all the pins in IOD port and IOC port have been defined as input mode.

```
PLC         25h     ; Reset the HRF0, HRF2 and HRF5.
SHE   24h           ; Set HEF2 and HEF5, a signal change on INT or KI1~4 pins
                    ; will cause the start condition flag 4 or 8 to be set.
SCA   10h           ; Set SEF4, the signal changes on port IOC
                    ; will cause the start conditions SCF1 to be set.
SRE   0b0h          ; SRF7,5,4 are set so that the signal changes on KI1~4 pins, port
                    ; IOC and INT pin will cause the stop mode to be released.
STOP                ; Enter the stop mode.

……                  ; STOP release

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

2.16 HALT FUNCTION

The halt function can minimize the current dissipation of TM8722 when LCD is still 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 executing the HALT instruction and no halt release signals (SCF1, SCF3, HRF1 ~ 6) are 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 the halt mode again by executing the RTS instruction after the completion of the interrupt service.

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

(2) A signal change on IOC port which is specified by the SCA instruction (SCF1).

(3) The halt release condition specified by the SHE instruction is met (HRF1 ~ HRF6).

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

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

2.17 STOP FUNCTION (STOP)

The stop function is another way to minimize the current dissipation for TM8722. In stop mode, all the functions in TM8722 are put into hold state including oscillators. All of the LCD corresponding signals (COM and Segment) output "L" level. TM8722 will 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 the IOC port must be in the "L" state, and no stop release signals (SRFn) will be delivered. The MCU will then enter into the stop mode by executing STOP instruction.

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

- One of the signals on the input mode pin of the IOC port is in "H" state and holds long enough to cause the CPU to be released from the halt mode.
- A signal change on the INT pin.
- The stop release condition specified by the SRE instruction is met. (INT pin is exclusive)

When TM8722 is released from the stop mode, TM8722 will enter the halt mode immediately and 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", TM8722 will enter into the stop mode immediately. The backup flag (BCF) will be set to 1 automatically after the MCU enters into the stop mode.

The following diagram shows the stop release procedure:

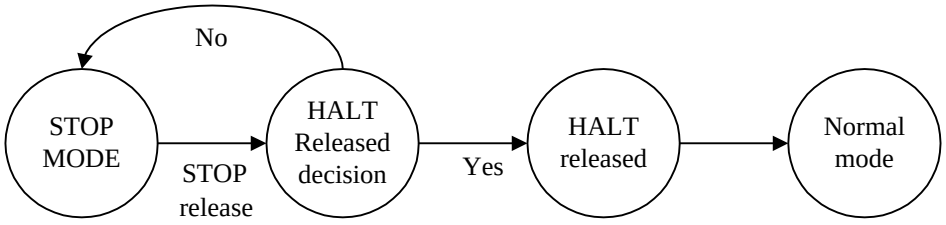


Figure 3- 16 The stop release state machine

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

- Set the stop release conditions by the execution of the SRE instruction.
- Set the halt release conditions corresponding to the stop release conditions if needed.
- Set the interrupt conditions corresponding to the stop release conditions if needed.

When the stop mode is released by an interrupt request, TM8722 will enter the halt mode immediately. Once the interrupt request is accepted, the halt mode will be released and then enters the interrupt service routine. After the interrupt service is completed, the MCU will return to stop mode again by executing the RTS instruction.

Once the MCU comes out of the stop mode, executing any one of the MSB, MSC or MCX instruction will test the halt release signals and executing the PLC instruction will reset the halt release signals. If the STOP instruction is executed when the stop release signal (SRF) is delivered, the CPU will not enter into the stop mode but will enter into 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

TM8722 provides a back up mode to avoid system malfunctioning under heavy loading, such as active buzzer, lighting LED…,etc, since the heavy loading will cause a large voltage drop on the supply voltage, the system will be malfunction under this condition.

In the back up mode, the driving capability for the 32.768KHz Crystal oscillator will be inhanced and its internal power (BAK pin) will be switched from VDD1 to VDD2 (Li power option only). Under this condition, all functions in TM8722 will work under VDD2 voltage level. This will improve the power noise immunity of TM8722 but it also increases the power consumption.

When not in the back up mode, the 32.768KHz Crystal oscillator operates with a normal driving capability and its internal power (BAK pin) switches from VDD2 to VDD1 when BCF flag is cleared.

Under this condition, only peripheral circuitry operates under VDD2 voltage level; the other functions will operate under VDD1 voltage level. It is necessary to connect a 0.1uf capacitor between BAK and GND pins to stabilize the internal power voltage.

It is recommended to exit out of the back up mode anytime unless it is necessary to reduce the current consumption for low power applications.

The back up flag (BCF) indicates the status of the back up function. When the BCF flag is set to 1, the MCU will enter into the backup mode. The BCF flag can be set or reset by executing either the SF or the RF instruction.

In order to shorten the start-up time of the 32.768KHz Crystal oscillator, TM8722 sets the BCF to 1 during the initial reset cycle and reset BCF back to 0 by executing the RF 2 instruction in Ag and Li power mode options. In EXT-V power mode option, BCF is set to 0 by the default setting and reset back to 1 by executing the SF 2 instruction during the normal operation.

The back up function performs differently with different power mode options, as shown in the following table.

1.5V battery mode:

TM8722 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)

TM8722 status	BCF = 0	BCF = 1
32.768 KHz Crystal Oscillator	Small driver	Large driver
Voltage on BAK pin	VDD1	VDD1
Internal operating voltage	VDD1	VDD1

3V battery or higher mode:

TM8722 status	BCF flag status
Initial reset cycle	BCF = 1 (hardware controlled)
After initial reset cycle	BCF = 1 (hardware controlled)
Executing SF 2h instruction	BCF = 1
Executing RF 2h instruction	BCF = 0
HALT mode	Previous state
STOP mode	BCF = 1 (hardware controlled)

	BCF = 0	BCF = 1
32.768 KHz Crystal Oscillator	Small driver	Large driver
Voltage on BAK pin	VDD1	VDD2
Internal operating voltage	VDD1	VDD2

Ext-V power mode:

TM8722 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)

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

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

3. Control Function

3.1 NTERRUPT FUNCTION

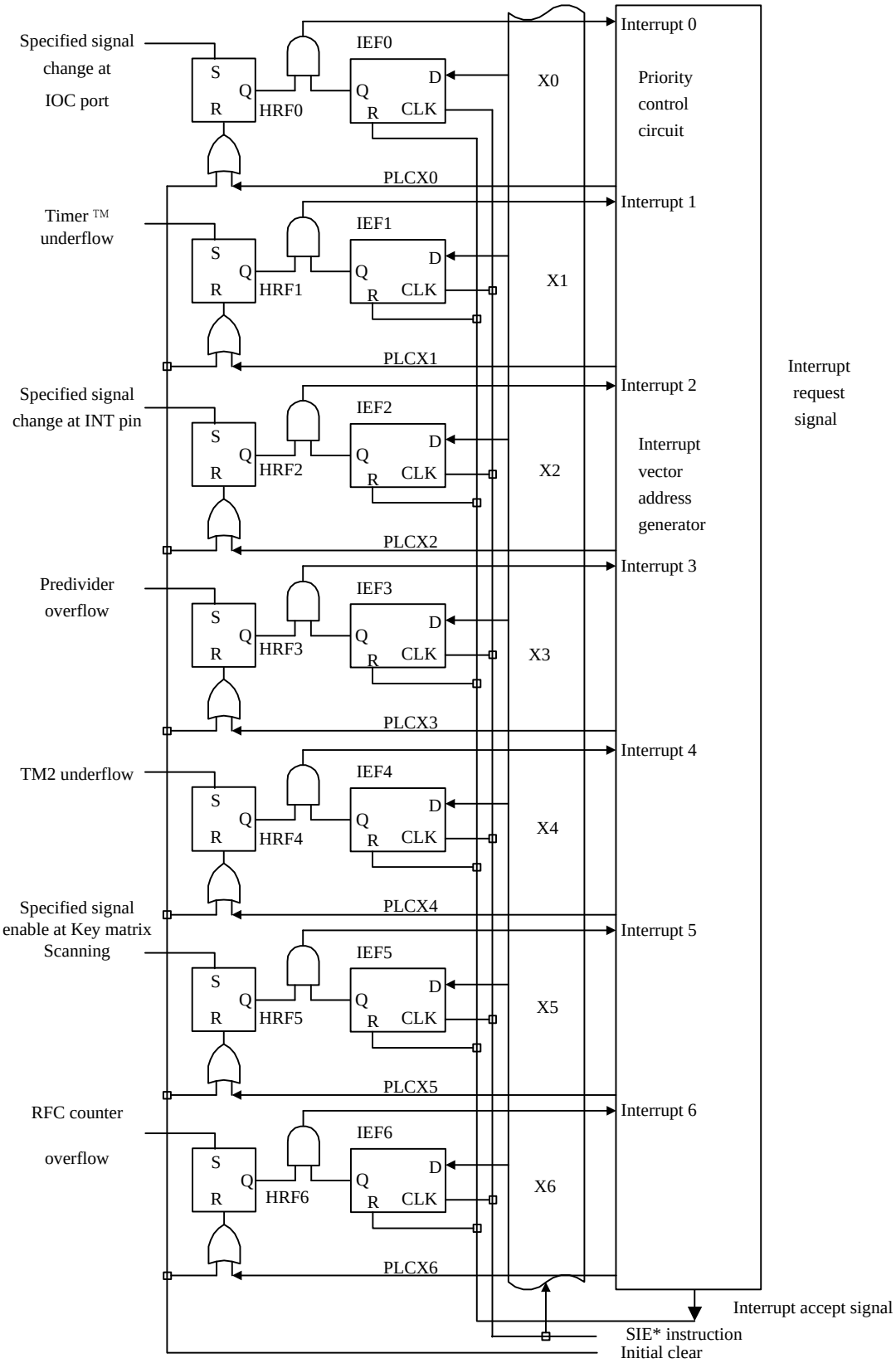
There are 7 different kinds of interrupt: 3 external and 4 internal interrupt. When an interrupt is accepted, the program in execution is suspended temporarily and the corresponding interrupt service routine specified by a pre-determined address in the program memory (ROM) will be called.

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

Table 3-1 Interrupt information

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

The following figure shows the Interrupt Control Circuit



3.1.1 ERRUPT REQUEST AND SERVICE ADDRESS

3.1.1.1 External interrupt factor

The external interrupts are generated by the INT pin, the IOC ports, and the key-matrix scanning function.

1. External INT pin interrupt request

In mask option, either a rising edge or falling edge of the signal on the INT pin can be selected for generating an interrupt. If the interrupt enable flag 2 (IEF2) is set beforehand and a signal change on the INT pin matches the mask option, it will generate a HRF2, the interrupt request 2. As soon as MCU accepts this interrupt request, the instruction at address10H will be executed automatically. It is necessary to hold the signal level for at least 1 machine cycle before and after the change edge of the signal.

2. I/O port IOC interrupt request.

An interrupt request signal (HRF0) will be generated when an input signal change on the I/O port (IOC) matches what is specified by the SCA instruction. In this case, if the interrupt enable flag 0 (IEF0) is set to 1, interrupt 0 is accepted and the instruction at address 14H will be executed automatically.

3. key-matrix scanning function interrupt request.

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

3.1.1.2 Internal interrupt factor

The internal interrupts are generated by Timer1, Timer2, RFC counter and the pre-divider.

1. Timer1/2 interrupt request

An interrupt request signal (HRF1/4) is generated when Timer1/2 underflows. In this case, if the interrupt enable flag 1/4 (IEF1/4) was set beforehand and interrupt request 1/4 is accepted, the instruction at address 18H / 20H will be executed automatically.

2. Pre-divider interrupt request

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

3. The 16-bit counter of RFC (CX pin control mode) interrupt request

An interrupt request signal (HRF6) is generated when the control signal applied on CX pin is inactive and the 16-bit counter stops to operate. In this case, if the interrupt enable flag6 (IEF6) is set beforehand and interrupt 6 is accepted, the instruction at address 28H will be executed automatically.

3.1.2 INTERRUPT PRIORITY

If all interrupt requests are issued simultaneously and all interrupts were enabled beforehand, the pre-divider interrupt is given the highest priority and other interrupts are held. When the interrupt service routine is initiated, all of the interrupt enable flags (IEF0 ~ IEF6) are cleared and they can be set by executing the SIE instruction again. Please refer to Table 3-1.

Example:

; Assuming all interrupts are requested simultaneously and all interrupts are enabled beforehand,

; all the pins of IOC port have been defined as input mode.

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

;……                    ; all interrupts are requested simultaneously.

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

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

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

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

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

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

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

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

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

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

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

SIE*    20h             ; Enable the interrupt request (except  the predivider, TMR1,
                        ; TMR2, RFC counter, IOC port, and INT)

;Interrupt caused by the Key matrix Scanning, and interrupt service is concluded.
;All interrupt requests have been processed.
```

3.1.3 INTERRUPT SERVICING

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

(1) The address of the suspended instruction will be stored into the stack register (STACK) as the return address of the interrupt service routine.

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

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

When an interrupt occurs, TM8722 will follow the procedure below:

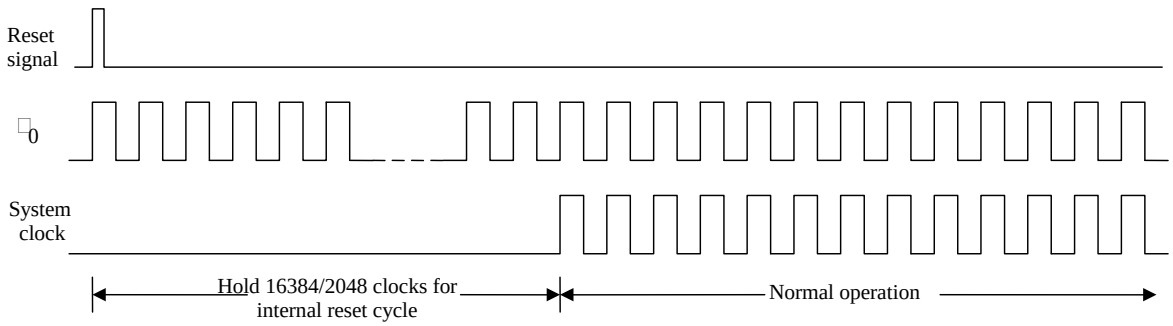
```
Instruction 1          ; An interrupt is accepted by the MCU.
NOP                    ; Store the address of Instruction 1 into the STACK.
                       ; The current program is suspended and a NOP instruction cycle is
                         inserted.

Instruction A          ;The program jumps to the interrupt service routine.
Instruction B
Instruction C
 .............
RTS                    ;Finish the interrupt service routine
Instruction 1*         ;Re-execute the instruction 1, which is interrupted.
Instruction 2
```

Note: If instruction 1 is the HALT instruction, MCU will return to halt mode 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) can be set again in the interrupt service routine if required.

3.2 RESET FUNCTION

TM8722 provides four kinds of reset functions: power-on reset, RESET pin reset, IOC port reset and watchdog timer reset.

When a reset signal is received, TM8722 will generate a time period for initial reset cycle, and there are two types of initial reset cycle time can be selected in mask option, the one is PH15/2 and the other is PH12/2.



Initial reset cycle time is PH15/2

MASK OPTION table:

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

In this option, the reset cycle time will last for 16384 clocks (PH0).

Initial reset cycle time is PH12/2

MASK OPTION table:

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

In this option, the reset cycle time will last for 2048 clocks (PH0).

3.2.1 POWER ON RESET

M8722 provides a power on reset function. If the power (VDD) is turned on or the power supply drops below 0.6V, it will generate a power-on reset signal.

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

MASK OPTION table:

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

Note: It is recommended to connect a capacitor between VDD and GND in order to get the better performance for the power-on reset function.

3.2.2 RESET PIN RESET

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

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

Once a “H” signal is applied on the RESET pin, TM8722 will not enter the initial reset cycle until the signal on the RESET pin returns to “0”. As soon as the signal applied on the reset pin returns to 0, TM8722 launches the initial reset cycle immediately.

The following table shows the initial condition of TM8722 in reset cycle.

Program counter	(PC)	Address 000H
Start condition flags 1 to 7	(SCF1-7)	0
Backup flag	(BCF)	1 (Ag, Li version) 0 (EXTV version)
Stop release enable flags 4,5,7	(SRF3,4,5,7)	0
Switch enable flags 4	(SEF3,4)	0
Halt release request flag	(HRF 0~6)	0
Halt release enable flags 1 to 3	(HEF1-6)	0
Interrupt enable flags 0 to 3	(IEF0-6)	0
Alarm output	(ALARM)	DC 0
Pull-down flags in I/OC 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*
EL panel driver pumping clock source and duty cycle	Celp	PH0, duty cycle is 3/4
EL panel driver clearing clock source and duty cycle	Celc	PH8, duty cycle is 1/4
Frequency generator clock source and duty cycle	Cfq	PH0, duty cycle is 1/4, output is inactive
Resistor frequency converter	(RFC)	Inactive, RR/RT/RH output 0
LCD driver output		All lighted (mask option)*
Timer 1/2		Inactive
Watchdog timer	(WDT)	Reset mode, WDF = 0
Clock source	(BCLK)	XT clock (slow speed clock in dual clock option)

Notes: 1. PH3 is the 3rd output of the predivider
2. PH10 is the 10th output of the predivider
3. All the LCD segment pins can output all ON or all OFF signals during reset cycle depending on the mask option.

3.2.3 KEY RESET from IOC Port or Key-Matrix scanning input port

The Key reset function can be selected in mask option. When the IOC port or the key-matrix scanning input (KI1~4) is activated and a '0' signal is applied to all the input pins, a reset signal will be delivered. (the key-matrix scanning function will not deliver the reset signal until the a scanning clock signal arrives)

MASK OPTION table:

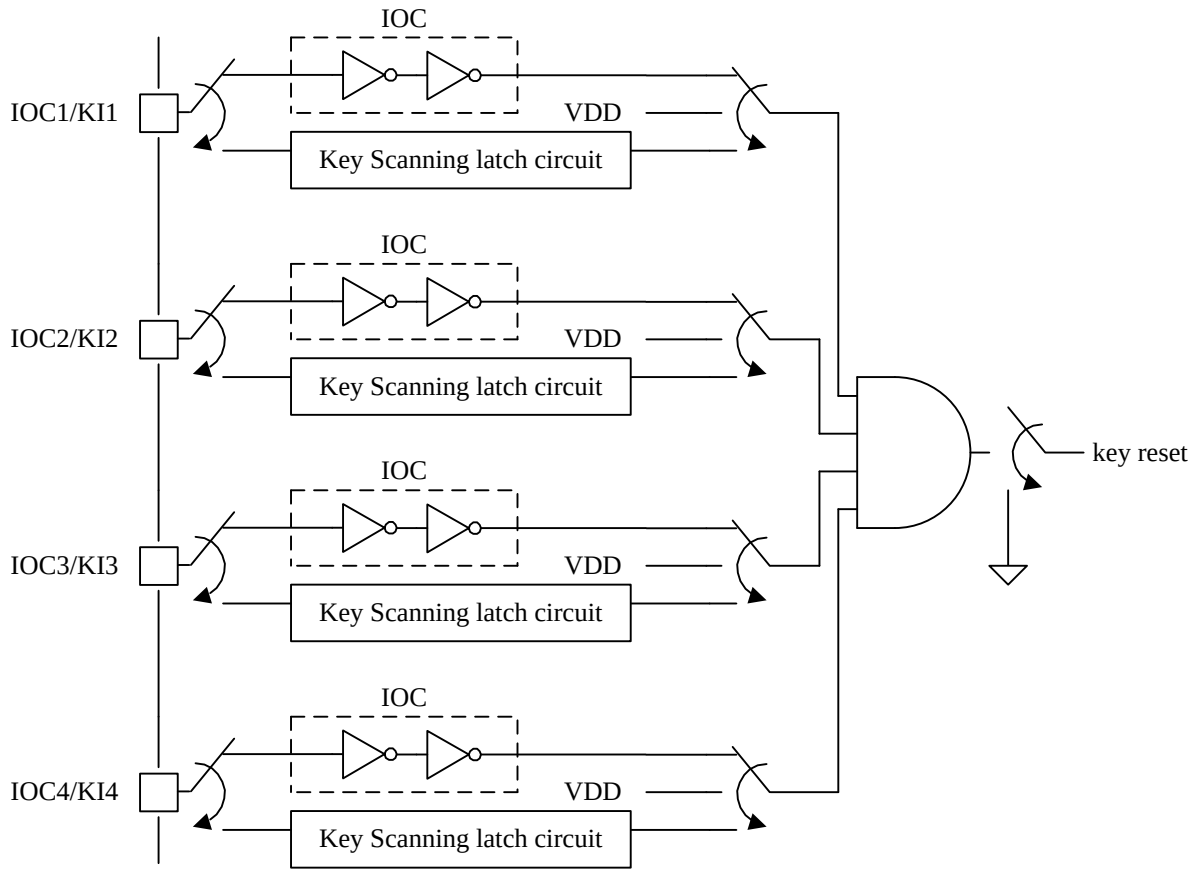
IOC or KI pins are used for key reset:

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

IOC or KI pins are not used for key reset:

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

The following figure shows the key reset organization.

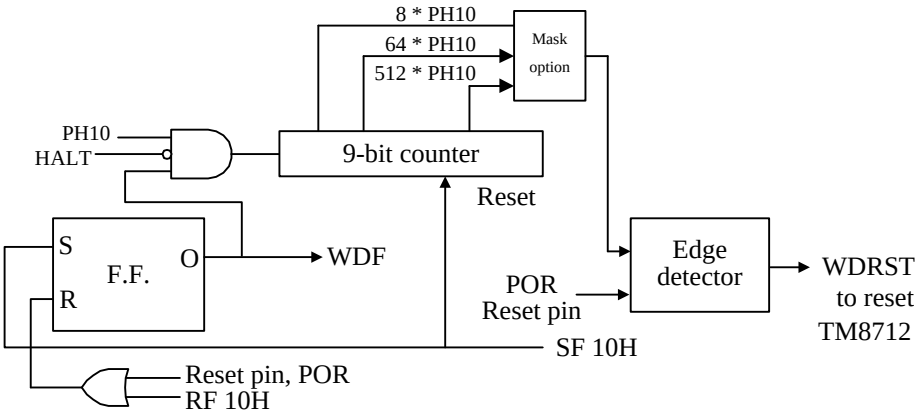


3.2.4 WATCHDOG TIMER RESET

The watchdog timer can prevent the unexpected execution sequences caused by a run-away software. The watchdog timer consists of a 9-bit binary counter. The clock source of watchdog timer comes from the 10th stage output of the pre-divider.

When the watchdog timer overflows, it will generate a reset signal to reset TM8722. Most of the functions in TM8722 will be re-initiated except for the watchdog timer itself(which is still active), the WDF flag will not be affected and PH0 ~ PH10 of the pre-divider will not be reset.

The following figure shows the watchdog timer organization.



During the initial reset cycle(power on reset, key reset or reset pin reset), the timer is inactive and the watchdog flag (WDF) is reset. The Instruction SF 10h will enable the watchdog timer and set the watchdog flag (WDF) to 1. At the same time, the content of the watchdog timer will be cleared. Once the watchdog timer is enabled, the timer will pause when the program enters the halt mode or the stop mode. When TM8722 wakes up from the halt or the stop mode, the timer operates continuously. It is recommended to execute a SF 10h instruction before the program enters the halt or the stop mode. This will prevent the MCU from the unexpected reset when it releases from halt or stop mode.

Once the watchdog timer is enabled, the program must execute the SF 10h instruction to clear the watchdog timer periodically. This will prevent the watchdog timer from being overflow.

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

MASK OPTION table:

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

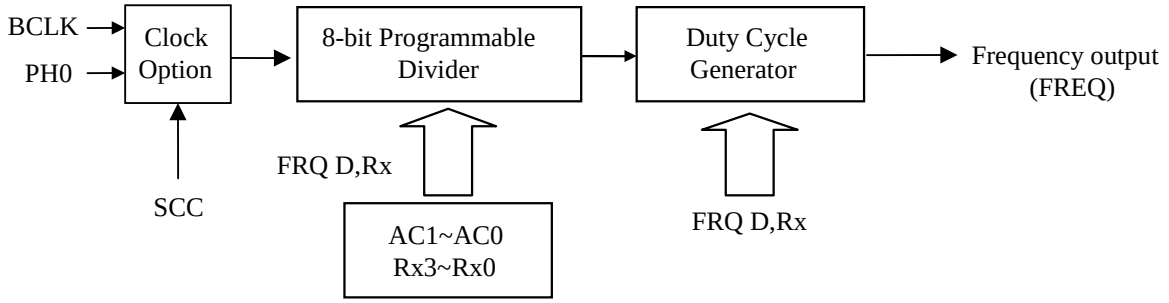
Note : the timer overflow time interval is about 16 seconds when PH0 = 32.768KHz

3.3 CLOCK GENERATOR

3.3.1 FREQUENCY GENERATOR

The Frequency Generator is a versatile programmable divider that is capable of outputting **a clock signal** with wide frequency range and different duty cycles. The output of the frequency generator can be the clock source for the alarm function, timer1, timer2 and 16-bit counter of RFC.

The following shows the organization of the frequency generator.



Executing a SCC instruction will select the clock source for the frequency generator. Executing a FRQ related instructions will set the output frequency and duty cycle of the frequency generator.

A FRQ related instruction presets a scaling data N for the programming divider and a data D for setting the duty cycle, and then causes the frequency generator to start outputting the clock signals with the following formula:

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

The scaling data N is preset by the content of data memory and the accumulator (AC), the table ROM data or the operand data specified in the FRQX instruction. The following table shows the bit pattern of the combination.

The following table shows the bit pattern of the preset scaling data N

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

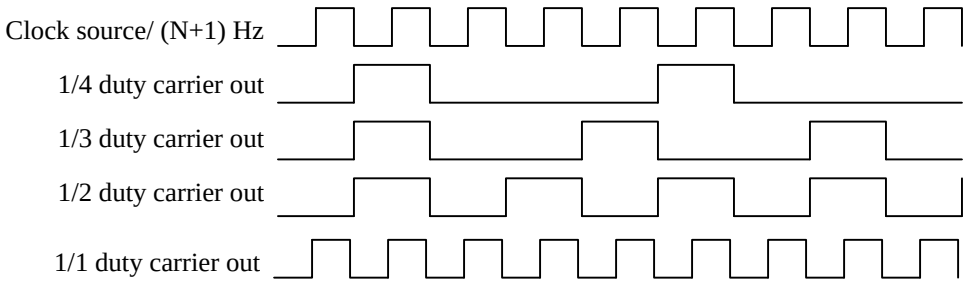
Notes: 1. T0 ~ T7 represents the data of table ROM.

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

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

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

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



3.3.2 FOR MELODY APPLICATION

The frequency generator can generate specified frequencies to compose a musical melody. The note table for these specified frequenies is shown below:

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

The following table shows the note table for melody application

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

Note:

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

For the melody application, the output signals of the frequency generator have to be conveyed to the buzzer output(BZB, BZ) in order to accomplish the complete function. For more detail information about the Buzzer output function, please refer to section 3-4.

3.3.3 Halver/Doubler/Tripler

The halver/doubler/tripler circuitry generates the necessary bias voltage for LCD driver, this circuitry consists of a combination of PH2, PH3, PH4, PH5. When using a Li battery power supply, the halver circuitry generates a 1/2 VDD voltage to drive the MCU's functoins which are not related to the input / output operation.

3.3.4 Alternating clock for LCD driver

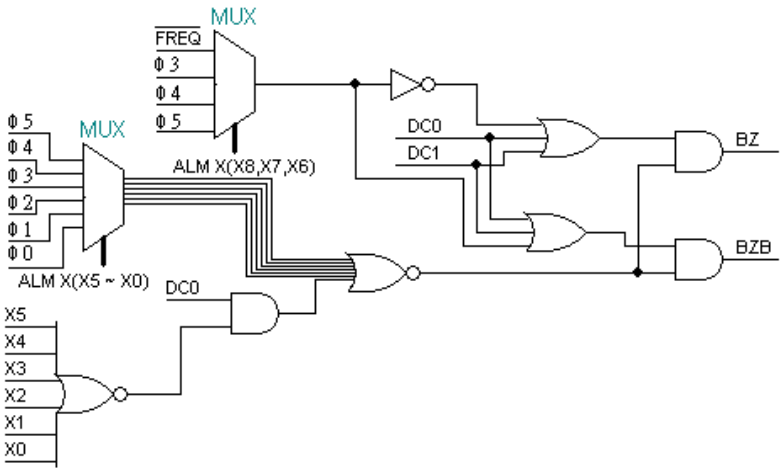
The alternating clock is the basic clock for LCD driver. Both COM and SEG pins shall change their output waveforms according to the alternating clock.

3.4 BUZZER OUTPUT PINS

TM8722 provides a pair of buzzer output pins known as BZB and BZ, which are pin-shared with I/O pins, IOB3 and IOB4, and are configured in mask option respectively. BZB and BZ pins are versatile output pins with complementary output polarity. When the buzzer output function combined with the clock source comes from the frequency generator, it can generate a melody, a sound effect or the carrier output for the remote controller.

MASK OPTION table:

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



This figure shows the organization of the buzzer output.

3.4.1 FOR SOUND EFFECT APPLICATION

It is recommended to drive the buzzer with a transistor on either one output pin (BZ or BZB) or drive the buzzer with both BZ and BZB pins directly. The buzzer output pins (BZ, BZB) are capable of outputting a modulation waveform combining the frequency generator's output signal, PH3(4096Hz), PH4(2048Hz) and PH5(1024Hz) as the carrier with an envelope waveform in one of the following frequencies: 32Hz(PH10), 16Hz(PH11), 8Hz(PH12), 4Hz(PH13), 2Hz(PH14), 1Hz(PH15).The frequency combination for the output waveform can be specified by executing an ALM instruction.

Note:

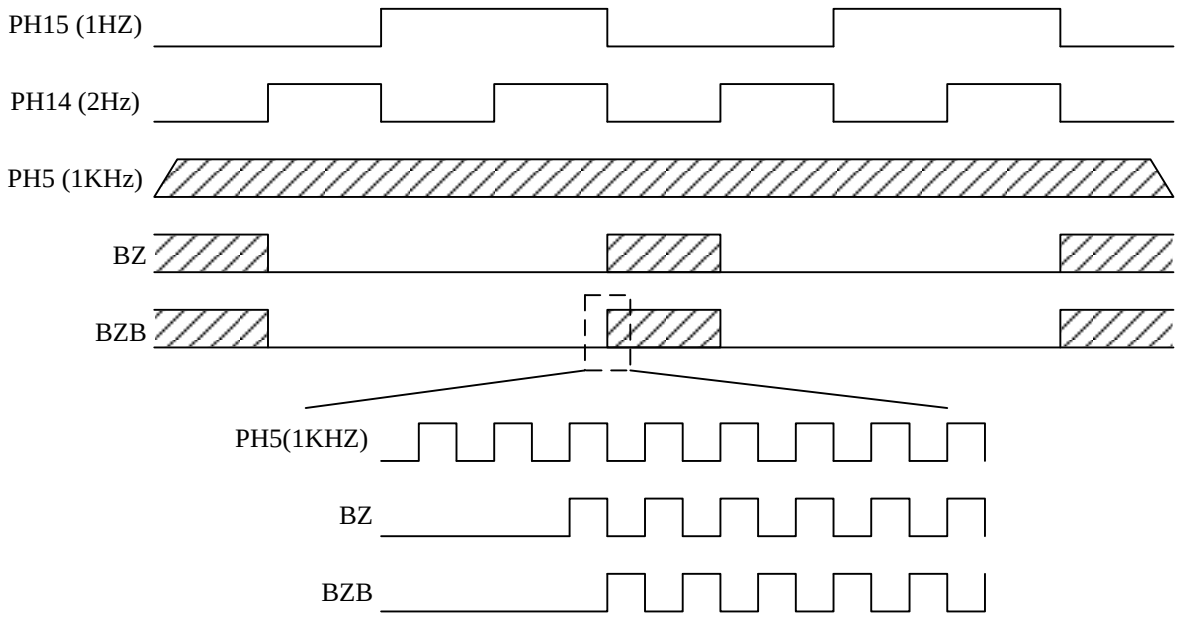
1.The higher frequency clock source should be only one of PH3, PH4, PH5 or FREQ, and the lower frequency may be the combinations from any/all of PH10 ~ PH15.
2. The frequencies in () corresponding to the input clock of the pre-divider (PH0) is 32768Hz.
3. The BZ and BZB pins output DC0 after the initial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

Example:

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

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

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



3.4.2 FOR REMOTE CONTROLLER APPLICATION

If the buzzer output combines with the timer and the frequency generator, the output signals on the BZ pin may produce the waveform for the IR remote controller. For the usage of remote controller, the preset scaling data N of the frequency generator must be greater than or equal to 3, and the ALM instruction must be executed immediately following the FRQ related instructions in order to deliver the FREQ signal to the BZ pin.

Example:

```
SHE     2       ; Enable timer 1 halt release enable flag.
TMSX    3Fh     ; Set the initial value of Timer1 to 3Fh and the clock source to PH9.
SCC     40h     ; Set the clock source of  the frequency generator to BCLK.
FRQX    2, 3    ; FREQ=BCLK/ (4*2), preset scaling data of the frequency generator
                ; to 3 and duty cycle to 1/2.
ALM     1C0h    ; FREQ signal is outputted. This instruction must be executed
                ; following the FRQ related instructions.
HALT            ; Waiting for the halt release(Timer 1 underflows).
……              ; Halt released.
ALM     0       ; Stop the buzzer output.
```

3.5 INPUT/OUTPUT PORTS

Three I/O ports are available in TM8722: IOA, IOB and IOC. Each I/O port has the same basic function and consists of 4 bits.

When the I/O pins are defined as non-IO function in mask option, the input / output function of the pins will be disabled.

3.5.1 IOA PORT

IOA1~IOA4 pins are MUX with CX/SEG24, RR/SEG25, RT/SEG26 and RH/SEG27 pins respectively as defined in mask option.

MASK OPTION table:

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

The default setting of the IOA port is “input mode” during the initial reset cycle. After the initial cycle, each bit of the port can be defined as either “input mode” or “output mode” respectively by executing a SPA instruction. Executing an OPA instruction can output the content of the specified data memory to those pins which are defined as output mode.

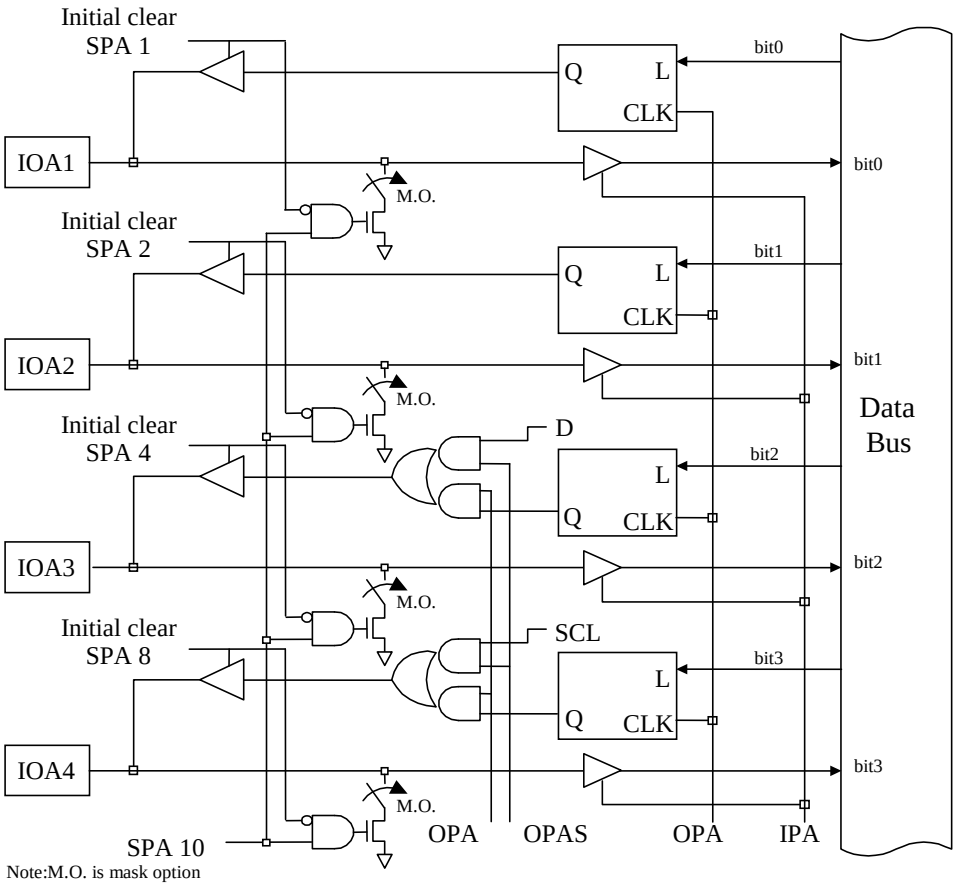
Executing an IPA instruction can store the IO pins’ signals into the specified data memory. When the IO pins are defined as output mode, executing an IPA instruction will store the content that stored in the output latch into the specified data memory.

Before changing the I/O pins to “output mode”,, the OPA instruction must be executed first to output the data to those output latches. It will prevent the chattering signal on the I/O pin when the I/O mode changes.

IOA port has a built-in pull-low resistor which can be selected in mask option and can be enabled / disabled by executing a SPA instruction.

Pull-low function option

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



The figure shows the organization of IOA port.

Note: The pins in "input mode" should never be left floating, otherwise a large current (straight-through current) will flow into the input buffer.

3.5.1.1 Pseudo Serial Output

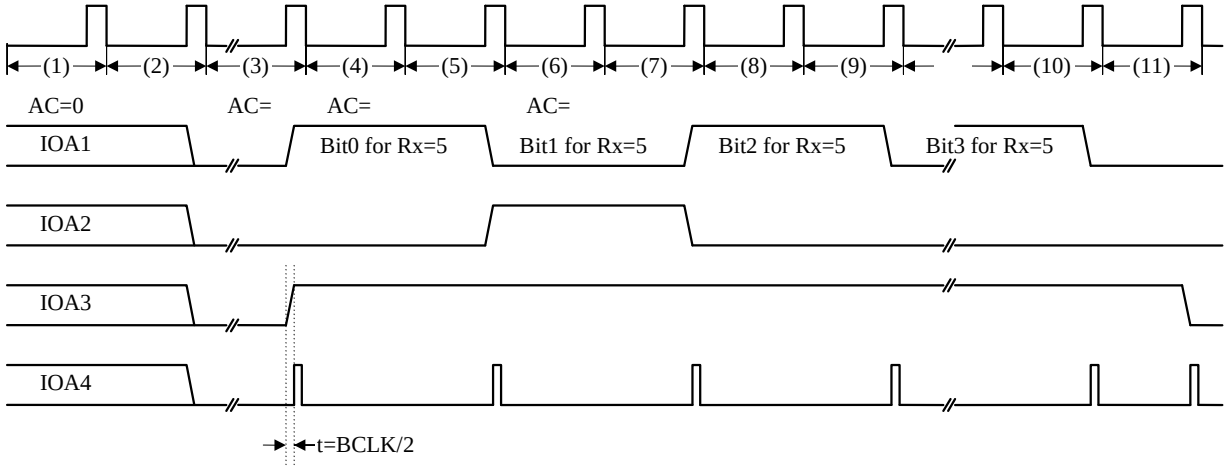
The IOA port may operate as a pseudo serial output port by executing an OPAS instruction. The IOA port must be defined as output mode before executing an OPAS instruction.

1. BIT0 and BIT1 of the port deliver RAM data.
2. BIT2 of the port delivers the constant data(D) in operand.
3. BIT3 of the port delivers a pulse.

Below is a sample program using the OPAS instruction to perform a serial output function.

```
(1)  LDS    0AH, 0
(2)  OPA    0AH     ; IOA1 is the serial output data pin,
     SPA    0FH     ; IOA3 enables the serial output function
     :              ; IOA4 is the serial output clock pin.
     :
     LDS    1,5
(3)  OPAS   1,1     ; Bit 0 output, enable the serial output function
(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     ; Output the last bit data
(11) OPAS   1,0     ; Disable the serial output function
```

The above program is illustrated by the timing chart below:



If IOA1 pin is used as the CX pin for RFC function and the other IOA pins (IOA2 ~ IOA3) are used for normal IO pins in mask option, the IOA1 function must be set as “output mode” at the begining of the program in order to prevent the signal change on CX pin from getting into the IOA1 function.

However, the IOA1 function in “output mode” can not change the output signal because the output signal of IOA1 function will affect the counting of RFC counter through the CX pin when the RFC counter function is enabled.

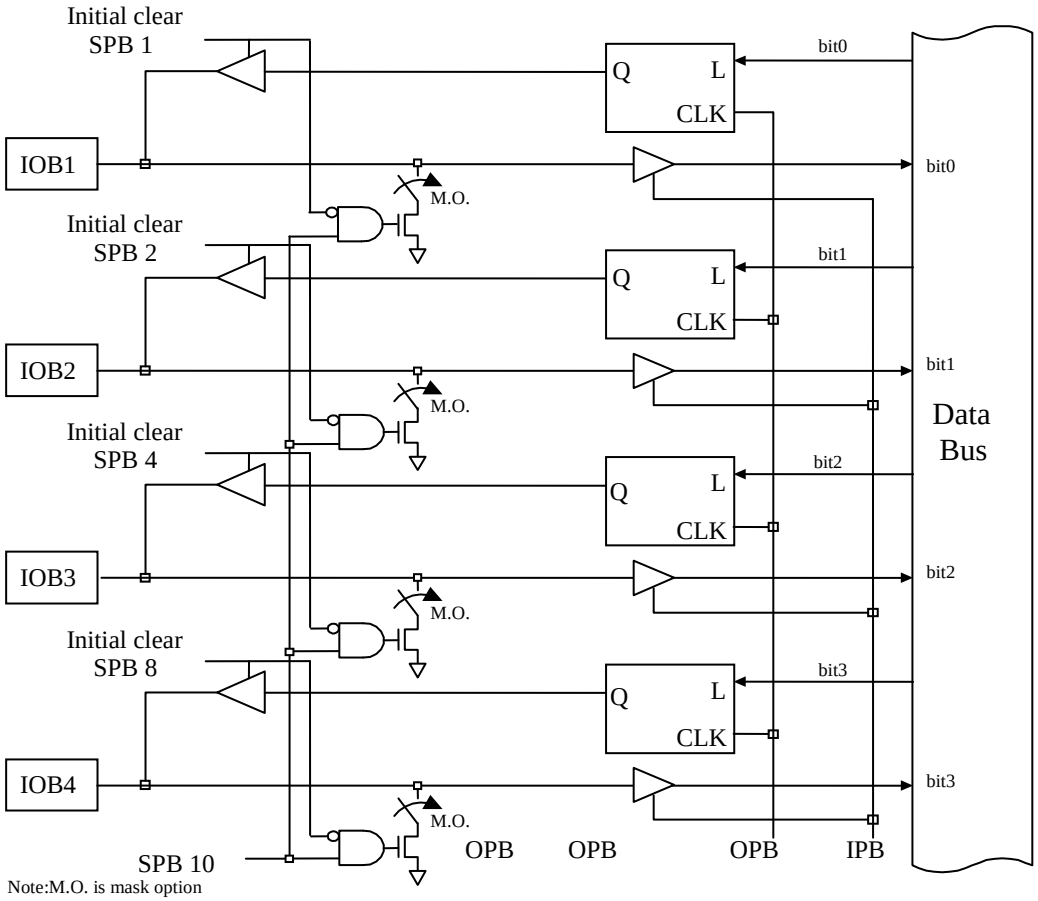
3.5.2 IOB PORT

IOB1~IOB4 pins are MUXed with ELC/SEG28, ELP/SEG29, BZB/SEG30 and BZ/SEG31 pins respectively as defined in mask option.

MASK OPTION table:

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

The following figure shows the organization of IOB port.



Note: The pins in "input mode" should never be left floating,,otherwise a large current (straight-through current) will flow into the input buffer.

The default setting of IOB port is "input mode" during the initial reset cycle. After the initial cycle, each bit of port can be defined as either "input mode" or "output mode" respectively by executing a SPB instruction. Executing an OPB instruction can output the content of data memory specified to those pins which had been defined as "output mode".

Executing an IPB instruction can store the IO pins' signals into the specified data memory. When the IO pins are defined as "output mode", executing an IPB instruction will store the content that is stored in the output latch into the specified data memory.

Before changing the I/O pins to “output mode”, it is required to output the data to those output latches by executing an OPB instruction first. It will prevent the chattering signal from occurring on the I/O pin when the I/O mode changes.

IOB port has a built-in pull-low resistor which can be selected in mask option and can be enabled / disabled by executing a SPB instruction.

Pull-low function option

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

3.5.3 IOC PORT

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

MASK OPTION table:

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

The default setting of IOC port is “input mode” during the initial reset cycle. After the initial cycle, each bit of port can be defined as either “input mode” or “output mode” respectively by executing a SPC instruction. Executing an OPC instruction can output the content of data memory specified to those pins which had been defined as “output mode”.

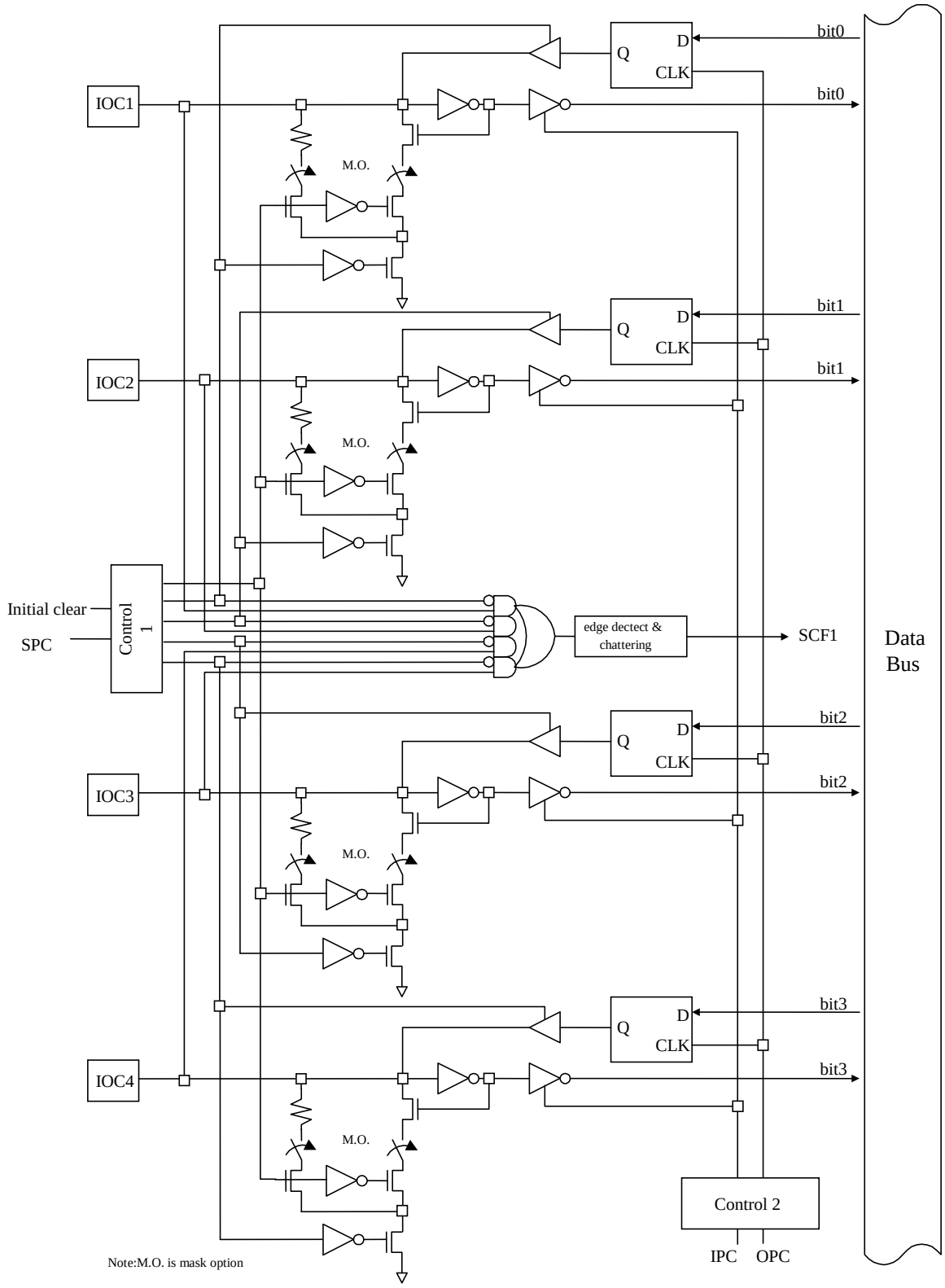
Executing an IPC instruction can store the IO pins’ signals into the specified data memory. When the IO pins are defined as “output mode”, executing an IPC instruction will store the content that is stored in the output latch into the specified data memory.

Before changing the I/O pins to “output mode”, it is required to output the data to those output latches by executing an OPC instruction first. It will prevent the chattering signal from occurring on the I/O pin when the I/O mode changes.

IOC port has a built-in pull-low resistor which can be selected in mask option and can be enabled / disabled by executing a SPC instruction.

There are two types of pull-low device in IOC port, the first type is a pull-low resistor and the other is a low-level hold device. Both of them can be selected in mask option and enabled/disabled by executing a SPC instruciton. When the pull-low resistor and low-level hold device are both selected in mask option, the default setting will enable the pull-low device and the low-level hold device will be disabled. Executing the SPC 10h instruction can enable the pull-low resistor and disable the low-level hold device and executing a SPC 0h can disable the pull-low resistor and enable the low-level hold device.

Once an IOC pin is defined as the output mode, both the pull-low resistor and the low-level hold devices will be disabled.



This figure shows the organization of the IOC port.

Note: in “input mode” should never be left floating,,otherwise a large current (straight-through current) will flow into the input buffer when both the pull-low device and the L-level hold device are disabled.

MASK OPTION table:

Pull-low function option

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

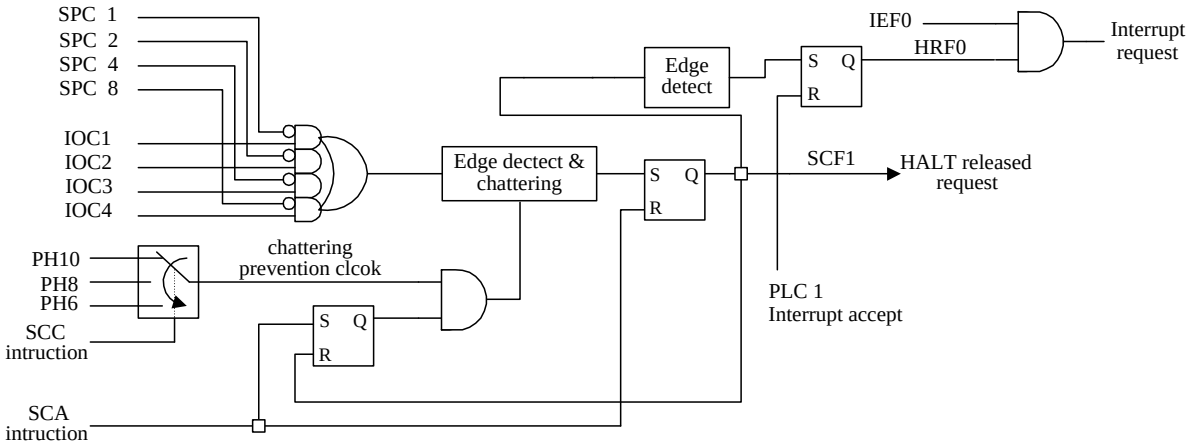
The low-level-hold device can not be selected individually in mask option without also selecting the pull-low resistor.

The Low-level-hold function option

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

3.5.3.1 Chattering Prevention Function and Halt Release

The port IOC is capable of preventing the chattering signals (bounce) applied on IOC1 to IOC4 pins. The de-bounce time can be selected as PH10 (32ms), PH8 (8ms) or PH6 (2ms) by executing a SCC instruction. The default selection is PH10 after the reset cycle. The following figure shows the organization of chattering prevention circuitry.



Note: The default prevention clock is PH10

The chattering prevention function will be invoked when the signal on the applicable pin (ex. IOC1) ~~is~~ changes 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 on the input pins of the IOC port specified by the SCA instruction and stays for at least two chattering clock (PH6, PH8, PH10) cycles, the control circuit on the input pins will generate the halt release request signal (SCF1). At that time, the chattering prevention clock will stop due to the delivery of SCF1. SCF1 can be reset to 0 by executing a SCA instruction and the chattering prevention clock will be enabled at the same time. If SCF1 has been set to 1, a halt release request flag 0 (HRF0)

will be generated. In this case, if the interrupt enable mode (IEF0) of the port IOC is set, the interrupt will be accepted.

Since no flip-flop is available to hold the information of the signal on the input pins IOC1 to IOC4, the input data on the port IOC should be stored into the RAM immediately after the halt mode is released.

3.6 EL PANEL DRIVER

TM8722 provides an EL panel driver for the backlight of the LCD panel. This circuitry can output a pumping voltage up to AC 150V or above to drive the EL panel and only requires a few external components. The pumping voltage level is determined by pumping frequency, duty cycle and ON / OFF frequency.

The ELC and ELP output are MUXed with IOB1 / SEG28 and IOB2 / SEG29, and can be selected in mask option.

MASK OPTION table:

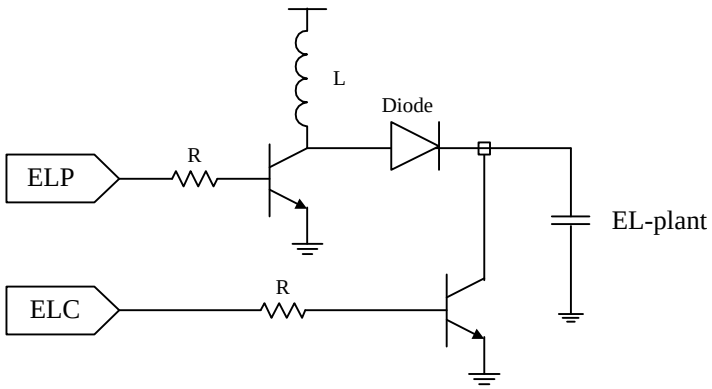
Mask Option name	Selected item
SEG28/IOB1/ELC	(3) ELC
SEG29/IOB2/ELP	(3) ELP

The ELP pin outputs the pumping clock signals and the ELC pin outputs the discharging pulses. The EL-**panel** driver will not operate until the SF 4h instruction is executed. While the driver is enabled, the ELC pin will output a pulse to discharge the EL-panel first and then ELP pin will output the pumping clock signal signals. This will ensure that there is no residual voltage to damage the EL-panel before the first pumping clock signal is applied.

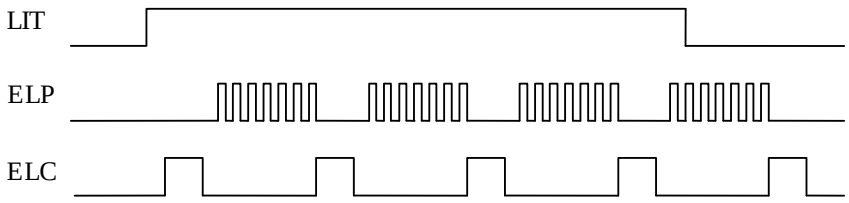
When the ELC pin outputs a discharge pulse, the pumping clock on the ELP pin will be inhibited.

Executing the RF 4h instruction will disable the EL-panel driver. At the same time, the ELC pin will output a pulse to discharge the EL- panel after the last pumping clock signal is delivered to the ELP pin.

This figure shows the application circuit of EL- **panel**.



This figure shows the output waveform of EL- **panel** driver



The ELP and ELC pin's pumping clock frequency, discharge clock frequency and duty cycle can be defined by executing an ELC instruction.

For ELP setting:

(X8, X7, X6)	Pumping clock frequency	(X5, X4)	Duty cycle
000	PH0	00	3/4 duty
100	BCLK	01	2/3 duty
101	BCLK/2	10	1/2 duty
110	BCLK/4	11	1/1 duty
111	BCLK/8		

For ELC setting:

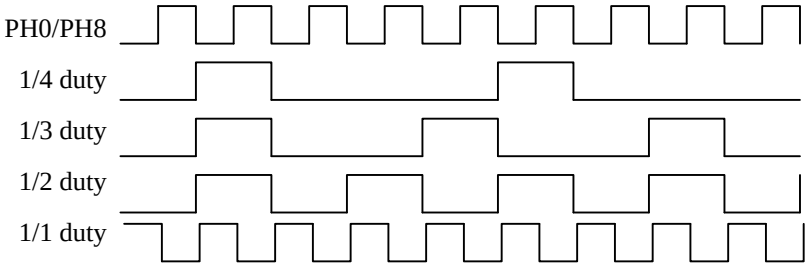
(X3, X2)	Discharge pulse frequency	(X1, X0)	Duty cycle
00	PH8	00	1/4 duty
01	PH7	01	1/3 duty
10	PH6	10	1/2 duty
11	PH5	11	1/1 duty

The default setting after the initial reset is:

ELP: PH0 clock of the pre-divider and 3/4 duty cycle

ELC: PH8 clock of the pre-divider and 1/4 duty cycle

The timing of the duty cycle is shown below:



Example:

```
ELC     110h    ; ELP outputs a BCLK pumping clock with 1/3 duty cycle
                ; and ELC outputs a PH8 discharging pulse with 1/4 duty cycle.
SF      4h      ; Enable the EL-light driver.
……
RF      4h      ; Disable the EL-light driver.
```

3.7 EXTERNAL INT PIN

There are 3 input types can be selected in mask option for the INT pin, pull-up, pull-down and high impendance. A signal change (either rising edge or falling edge specified in mask option) will set the halt release request flag 2 (HRF2). If the halt release enable flag (HEF2) is set, the start condition flag 2 will be set and a corresponding sigal is delivered. If the INT pin interrupt enable mode (IEF2) is set, the interrupt will be accepted.

MASK OPTION table:

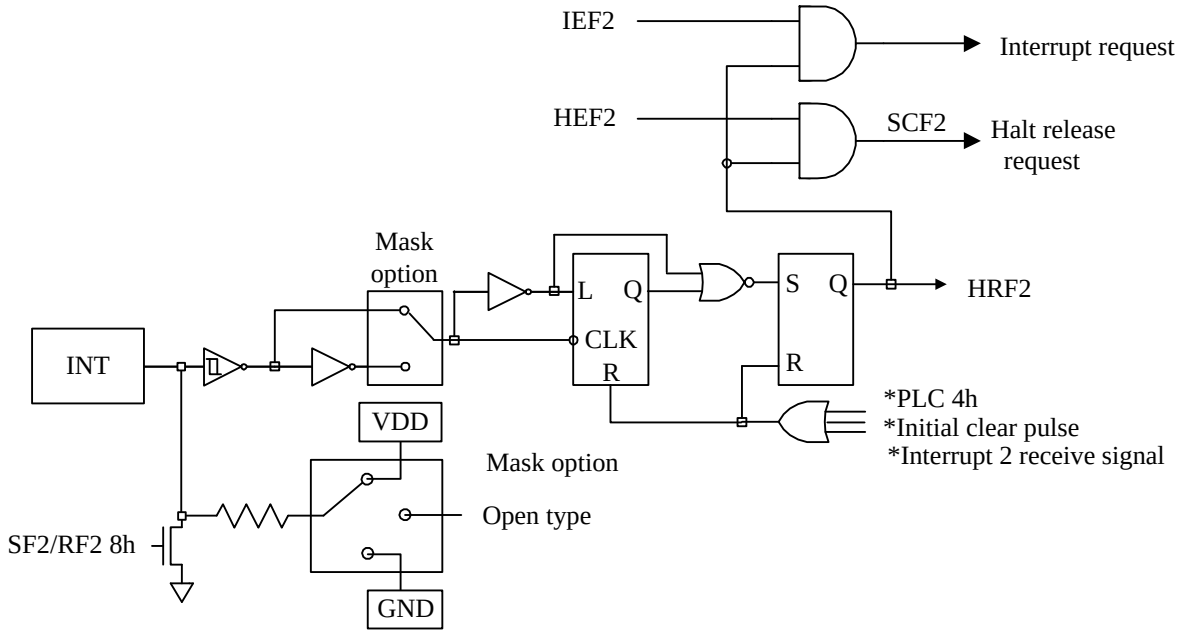
For internal resistor type:

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

For input triggered type:

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

This figure shows the INT Pin Configuration

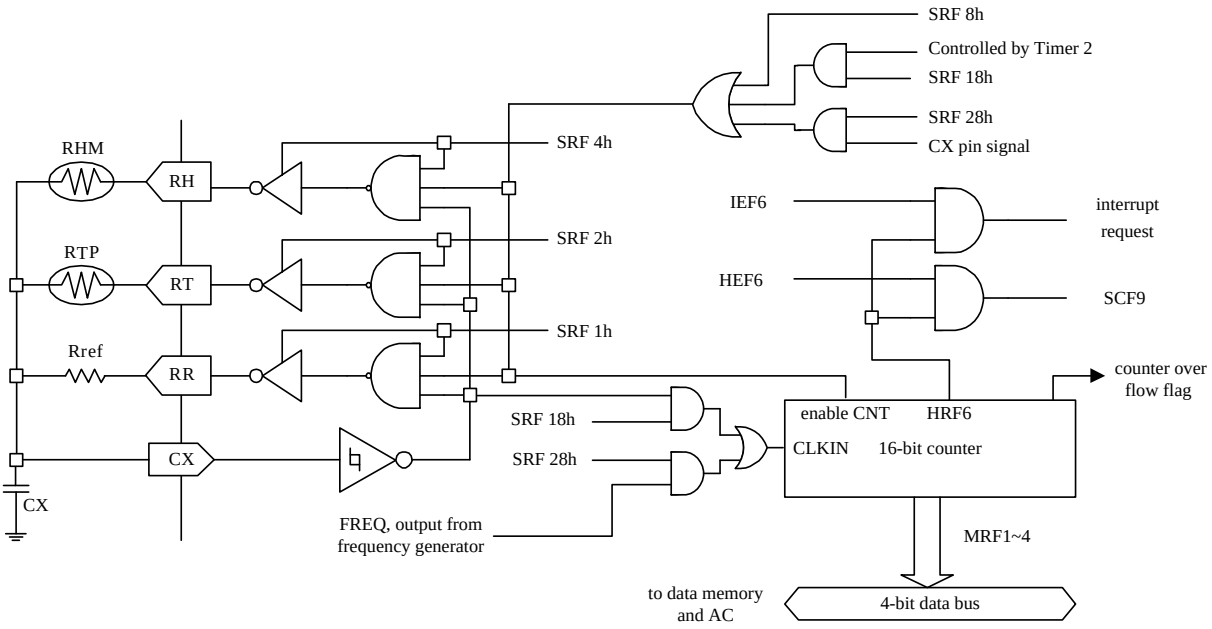


Note: For Ag battery power supply, positive power is connected to VDD1; the other kind of battery power supply, it is connected to VDD2.

3.8 Resistor to Frequency Converter (RFC)

The resistor to frequency converter (RFC) converts a specified resistance to a corresponding frequency. With a reference resistor and a sensor (resistance type), RFC will generate 2 corresponding frequencies. Based on these 2 frequencis, the MCU can calculate the resistance of the sensor.

The figure below shows the block diagram of RFC.



RFC consists of four external pins:

CX: the oscillation Schemmit trigger input pin

RR: the reference resister output pin

RT: the temperature sensor output pin

RH: the humidity sensor output pin (this pin can also be used with another temperature sensor or left floating)

These CX, RR, RT and RH pins are MUXed with IOA1/SEG37 to IOA4/SEG40 respectively and selected in mask option.

Mask Option name	Selected item
SEG24/IOA1/CX	(3) CX
SEG25/IOA2/RR	(3) RR
SEG26/IOA3/RT	(3) RT
SEG27/IOA4/RH	(3) RH

3.8.1 RC Oscillation Network

The RFC circuitry can be used to build up to 3 RC oscillation networks by connecting sensors or resistors between CX and any one of RR, RT or RH pins. Only one RC oscillation network can be active at a time. When one of the oscillation networks is activated by executing SRF 1h, SRF 2h or SRF 4h instructions to enable RR, RT, RH networks respectively and clock signals with specified frequency corresponding to the resistance will be generated and transferred to the 16-bit counter as the clock source. The 16-bit counter will count these clock signals and then calculate their frequency.

How to build up the RC oscillation network:

1. Connect the resistor and capacitor to the RR, RT, RH and CX pins. The Fig. 2-24 illustrates the connection of these networks.

2. Execute SRF 1h, SRF 2h, or SRF 4h instructions to activate the output pins(RR, RT, RH) for the RC networks respectively. The inactive output pins will become tri-state output pins.

3. Execute SRF 8, SRF 18h or SRF 28h instructions to enable the RC oscillation network and the 16-bit counter. The RC oscillation network will not operate until these instructions are executed. The output pin of RC oscillation network (one of the RR, RT, RH pins) will be set to output 0 state before its corresponding network is activated.

To get a better oscillation clock from the CX pin, activate the output pin for each RC network before the counter is enabled.

There is an extended bit (the 17^{th} bit) for the 16-bit counter. This bit also called the overflow flag (RFOVF). It can be checked by executing a MSD instruction.

The following mask option can determine whether the 16-bit counter will stop counting or not when overflow occurs.

Mask Option name	Selected item
RFC OVERFLOW DISABLE COUNTER	(1) USE
RFC OVERFLOW DISABLE COUNTER	(2) NO USE

If “NO USE” is selected, the RFOVF will be set to “1” when the 16-bit counter overflows. The counter will keep on counting up from “0000h” and the RFOVF flag will retrun to 0 when next overflow occurs,. In this option, the RFOVF flag is just used as the 17th bit of the counter only.

If “USE” is selected, the 16-bit counter will stop counting when overflow occurs and the RFOVF flag will be set to 1.

There are 3 operation modes for the 16-bit counter. Each mode is described in the following sections:

3.8.2 Enable/Disable the Counter by Software

In this mode, the clock input of the 16-bit counter receives signals from the CX pin and the counter is enabled/disabled by the S/W. When the SRF 8h instruction is executed, the counter will be enabled and start to count the clock signals from the CX pin. The counter will be disabled when the SRF 0 instruction is executed. Executing MRF1~4 instructions will load the content of the 16-bit counter into the specified data memory and AC.

Each time the 16-bit counter is enabled, the content of the counter will be cleared automatically.

Example:

If you want to count the amount of clock signals for a time period from the CX pin, you can enable the 16-bit counter by executing a SRF 8 instruction and set timer1 to control the time period. The overflow flag (RFOVF) of the 16-bit counter will be checkd during the time period. If the overflow flag is not set to 1, read the content of the counter; if the overflow flag is set to 1, the program has to reduce the time period and repeat the previous procedure again. In the following example, the RR network generates the clock source on CX pin.

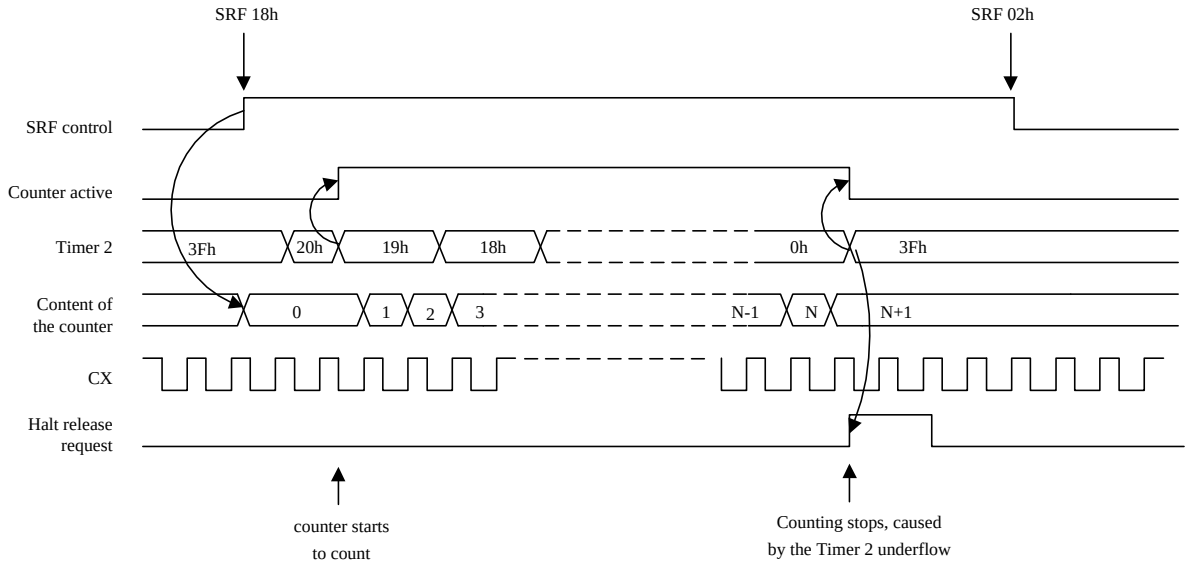
```
;Timer 1 is used to enable/disable the counter
        LDS     0, 0      ; Set the TMR1 clock source (PH9)
        LDS     1, 3      ; initiate TMR1 setting value to 3F
        LDS     2, 0Fh
        SHE     2         ; enable halt release by TMR1
RE_CNT:
        LDA     0
        OR*     1         ;combine the TMR1 setting value
        TMS     2         ;enable the TMR1
        SRF     9         ;build up the RR network and enable the counter
        HALT
        SRF     1         ;stop the counter when TMR1 underflows
        MRF1    10h       ;read the content of the counter
        MRF2    11h
        MRF3    12h
        MRF4    13h
        MSD     20h
        JB2     CNT1_OF       ; check the overflow flag of counter
        JMP     DATA_ACCEPT
CNT1_OF:
        DEC*    2             ; decrement the TM1 value
        LDS     20h, 0
        SBC*    1
        JZ      CHG_CLK_RANGE     ; change the clock source of TMR1
        PLC     1                 ; clear the halt release request flag of TMR1
        JMP     RE_CNT
```

3.8.3 Enable/Disable the Counter by Timer 2

In this mode, the 16-bit counter receives signals from the CX pin as clock input of and is activated by the operation of TMR2. When the counter is enabled by a SRF 18 instruction, the 16-bit counter will not start counting until TMR2 is enabled and the first clock falling edge has been applied on the clock input of TMR2. When the TMR2 underflow occurs, the 16-bit counter will stop counting immediately.

TMR2 can produce an accurate time period to control the counting of 16-bit counter. For a detail description of the operation of TMR2, please refer to 2-12.

Each time the 16-bit counter is enabled, the content of the counter will be cleared automatically.



The figure shows the timing of the RFC counter controlled by timer 2

Example:

```
                    ; In this example, the RT network is used to generate the clock source.
SRF     1Ah         ; Build up the RT network and enable the counter
                    ; controlled by TM2
SHE     10h         ; enable the halt release caused by TM2
TM2X    20h         ; set the PH9 as the clock signals for TM2 and the count down
                    ; value is 20h.
HALT
PLC     10h         ; Clear the halt release request flag of TM2
MRF1    10h         ; read the content of the counter.
MRF2    11h
MRF3    12h
MRF4    13h
```

3.8.4 Enable/Disable the Counter by CX Signal

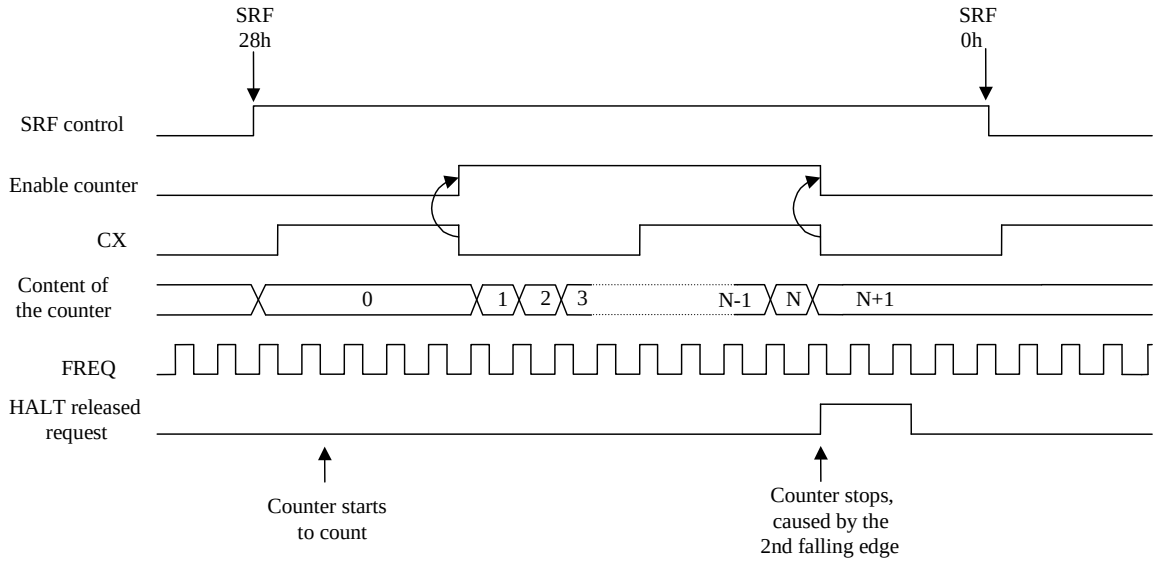
This is another way to use the 16-bit counter but it has nothing to do with the RFC function. In applications described in the previous section, CX is used as the clock source for the 16-bit counter and S/W or TMR2 to produce a time period in order to control the 16-bit counter.

In this mode, however, the 16-bit counter operates differently, the clock signal on CX pin turns into the controlled signal to enable/disable the 16-bit counter and the clock source of the 16-bit counter comes from the output of the frequency generator (FREQ).

While the 16-bit counter is enabled, it counts the clock (FREQ) after the first rising edge signal applies to the CX pin. Once the second rising edge applies to the CX pin, a halt release request (HRF6) will be

delivered and the 16-bit counter stops counting. In this case, if the interrupt enable flag 6(IEF6) is set, the interrupt will be accepted; and if the halt release enable flag 6(HEF6) is set, the halt release request signal will be delivered to set the start condition flag 9 (SCF9) in status register 4 (STS4).

Each time the 16-bit counter is enabled, the content of the counter will be cleared automatically.



The figure above shows the timing of the counter controlled by the CX pin

Example:

```
SCC     0h      ; Select the base clock of the frequency generator that comes from
                ; PH0 (XT clock)
FRQX    1, 5    ; set the frequency generator to FREQ = (PH0/6) / 3
                ; the frequency generator is set to 5 and FREQ
                        ; is set to 1/3 duty waveform.
SHE     40h     ; enable the halt release caused by 16-bit counter
SRF     28h     ; enable the counter controlled by CX signal
HALT
PLC     40h     ; a halt release request is caused by the 2nd rising edge on CX pin and
                ; clear the halt release request flag
MRF1    10h     ; read the content of the counter
MRF2    11h
MRF3    12h
MRF4    13h
```

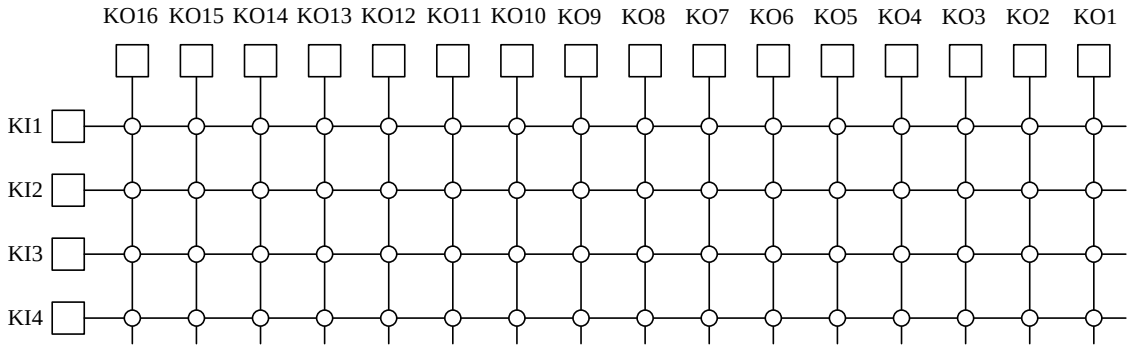
3.9 Key-Matrix Scanning Function

The key matrix scanning function is made up of four input pins KI1 ~ KI4, 16 output pins (shared with the LCD output pins SEG1 ~ SEG16. For ease of explanation, these will be referred to as KO1~KO16 in the rest of the document), and the external matrix keyboard. The input port of the key matrix circuitry is composed of KI1~KI4 pins (these pins are muxed with SEG32~SEG35 pins and selected in mask option).

MASK OPTION table:

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

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



Executing the SPK X instruction can set the different scanning types. The bit pattern of this instruction is shown below:

Instruction	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPK X	X6	X5	X4	X3	X2	X1	X0

Bit Patten	Setting : Halt Release by
X6=0	Normal Key Scanning
X6=1	Scanning Cycle

The bit pattern of X (for Key Matrix scanning output to KO1~KO16)

X5	X4	X3	X2	X1	X0	KO1	KO2	KO3	KO4	KO5	KO6	KO7	KO8	KO9	KO10	KO11	KO12	KO13	KO14	KO15	KO16
0	0	0	0	0	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	0	1	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	1	0	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	1	1	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	0	0	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	1	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z
0	0	1	1	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z
0	0	1	1	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z
0	0	1	1	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z

Notes: 1. The data"1" in KO1~KO16 output table represents,
For LED driver, the output voltage level is logic High.
For LCD driver, the output voltage level is logic Low.
2. KO1~16 = SEG1~16 output in scanning interval

IF the KI1~4 are selected as the Key matrix scanning input in mask option, it is necessary to execute a SPC instruction to set the unused IOC port as output mode before the key matrix scanning function is activated.

The organization of the Key matrix scanning input port is shown in the next page.

Once one of the KI1~4 pins detects the signal changes from “Hi-z” to “1”, TM8722 will set HRF5 to 1. If HEF5 has been set to 1 already, SCF7 will be set and release the HALT mode. After the key scanning cycle finishes, the states of KI1 ~ 4 pins are stored into the output latch of IOC port. Executing an IPC instruction can store these states into data RAM.

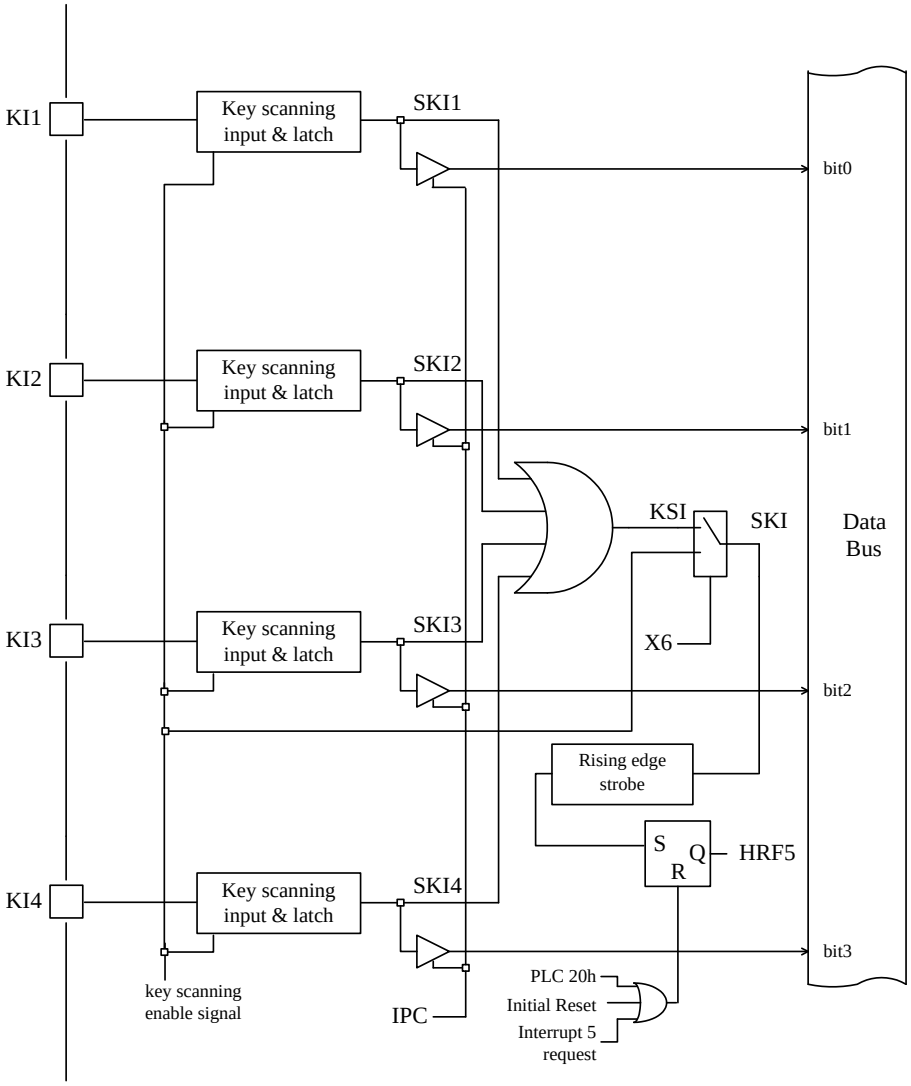
Executing a PLC 20h instruction can clear HRF5 flag.

Since the key matrix scanning function steals a part of the LCD/LED driver’s waveform as the scanning output signal, so the scanning frequency is the same as the LCD driver’s alternating clock frequency. The formula for key matrix scanning frequency is shown below:

key matrix scanning frequency (Hz) = (LCD frame frequency) x (LCD duty cycle) x 2

Note: “2” is a factor

For example, if the LCD frame frequency is 32Hz, and duty cycle is 1/5 duty, the scanning frequency for key matrix will be: 320Hz (32 x 5 x 2).



This figure shows the organization of Key matrix scanning input

Example:

```
        SPC     0fh             ; Disable all the pull-down devices on internal IOC port.
                                ; Set all the IOC pins as output mode.
        SPK     10h             ; Generate a HALT released request when key depressed
                                ; Scanning all columns simultaneously during each cycle.
        PLC     20h             ; Clear Flag HRF5
        SHE     20h             ; Set HEF5.
        HALT                    ; wait for the halt release caused by the key matrix.
        MCX     10h             ; Check SCF8 (SKI).
        JB0     ski_release
        ……
        ……
ski_release:
        IPC     10h             ; read KI1~4 input latch state.
        JB0     ki1_release
        JB1     ki2_release
        JB2     ki3_release
        JB3     ki4_release
        .
        .
ki1_release:
        SPK     40h                     ; Check if the key depressed on K1 column.
        PLC     20h                     ; Clear Flag HRF5 to avoid the false HALT released
        CALL    wait_scan_again; Waiting for the next key matrix scanning cycle.
                                        ; The waiting period must be longer than the key matrix
                                        ; scanning cycle.
        IPC     10h                     ; Read the KI1 input latch state.
        JB0     ki1_seg1
        ……
        ……
        SPK     4fh                     ; Only enable SEG16 scanning output.
        PLC     20h                     ; Clear HRF5 to avoid the false HALT released
        CALL    wait_scan_again         ; Wait for time longer than the halt LCD clock cycle
                                        ; to ensure scan again.
        IPC     10h                     ; Read KI1 input latch state.
        JB0     kil_seg16
        ……
        ……
wait_scan_again:
        HALT
        PLC     20h
        RTS
```

4. LCD/LED DRIVER OUTPUT

4.1 LCD DRIVER OUTPUT

TM8722 provides 35 segment output pins and 5 common output pins to drive LCD or LED. SEG1~23 output pins can also be used as DC output ports (mask option). If there are more than one LCD driver output pins need to be defined as the DC output pin, the following mask option must be selected.

MASK OPTION table:

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

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

When all of SEG1~23 are used for DC output port & SEG24~35 are not used for SEG:

Mask Option name	Selected item
LCD ACTIVE TYPE	(4) O/P

During the initial reset cycle, all the LCD patterns can be selected either on or off in mask option. All LCD patterns will keep the initial setting state till the the LCD related instructions are executed to change the LCD patterns.

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 TM8722

There are several settings in the LCD lighting system that can be selected in mask option in TM8722:

- 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,

All these lighting system options are combined into 2 kinds of mask options, the “LCD DUTY CYCLE” and the “LCD BIAS”.

MASK OPTION table:

LCD duty cycle option

Mask Option Name	Selected Item
LCD/LED DUTY CYCLE	(1) O/P
LCD/LED DUTY CYCLE	(2) DUPLEX (1/2 duty)
LCD/LED DUTY CYCLE	(3) 1/3 DUTY
LCD/LED DUTY CYCLE	(4) 1/4 DUTY
LCD/LED 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 can be selected in mask option.

(All the LCD frame frequencies in the following tables are based on the clock source frequency of the pre-divider (PH0), which is 32768 Hz).

The LCD alternating frequency at duplex (1/2 duty)

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

The LCD alternating frequency at 1/3 duty

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

The LCD alternating frequency at 1/4 duty

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

The LCD alternating frequency at 1/5 duty

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

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

LCD Lighting System	The Maximum Number of Driving LCD Segments	Remarks
Static	35	Connect VDD3 to VDD2
Duplex	70	Connect VDD3 to VDD2
1/2bias 1/3duty	105	Connect VDD3 to VDD2
1/2bias 1/4duty	140	Connect VDD3 to VDD2
1/2bias 1/5duty	175	Connect VDD3 to VDD2
1/3 bias 1/3 duty	105	
1/3 bias 1/4 duty	140	
1/3 bias 1/5 duty	175	

It is recommended to choose the frame frequency higher than 24Hz. If the frame frequency is lower than 24Hz, the pattern on the LCD panel will start to flicker.

4.1.2 DC OUTPUT

TM8722 allows all LCD/LED driver output pins to be defined as CMOS type DC output or P open-drain DC output ports in mask option. It is also possible to utilize some LCD/LED driver output pins for DC output and the rest of the LCD/LED driver output pins for LCD/LED driver. Please refer to 4-1-3-4 for more details.

The configurations of CMOS output type and P open-drain output type are shown below.

When the LCD/LED driver output pins (SEG) are defined as DC output, the output data on the port will not be affected even the program enters the stop mode or LCD turn-off mode.

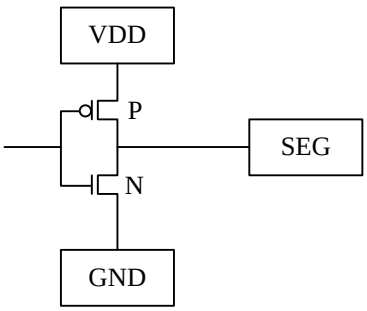


Figure 5-1 CMOS Output Type

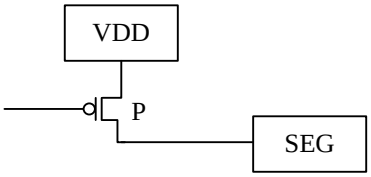


Figure 5-2 P Open-Drain Output Type

4.1.3 SEGMENT PLA CIRCUIT FOR LCD DISPLAY

4.1.3.1 PRINCIPLE OF OPERATION OF LCD DRIVER SECTION

Fig. 5-3 below explains how the LCD driver module operates when the LCD-related instructions are executed.

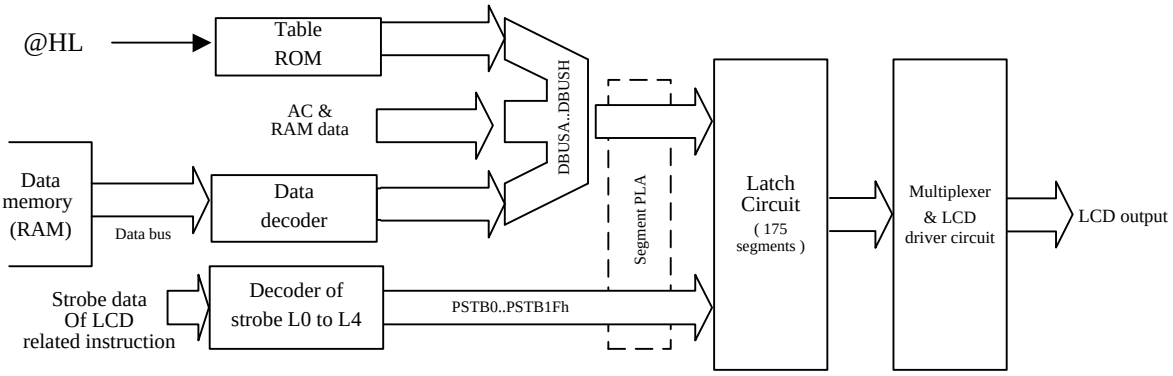


Figure 5-3 Primary Diagram of LCD Driver module

The LCD driver module contains the following units:

- Data decoder : decode the data received from RAM or table ROM
- L0 to L4 decoder : decode the Lz data specified in the LCD-related instructions
- Latch circuit : store LCD lighting information
- Segment PLA circuit : connect to data decoder, the L0 to L4 decoder and the latch circuit.
- Multiplexer : select 1/2duty, 1/3duty, 1/4duty, 1/5duty
- LCD driver circuitry

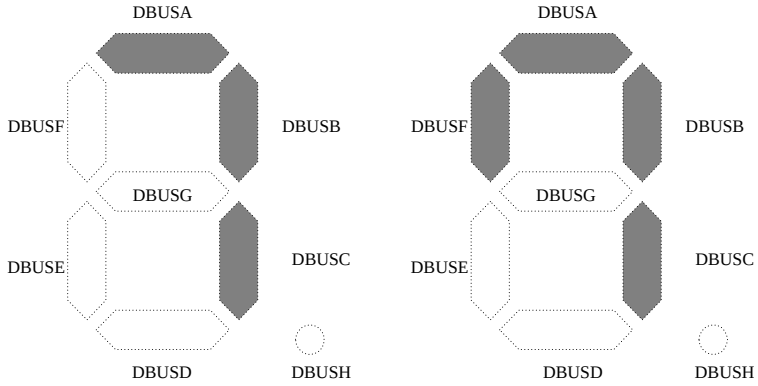
The data decoder converts the content of the working register specified in LCD-related instructions into the data format of 7-segment pattern on LCD panel.

The data decoder table is shown below:

Content of the data memory	Output of the 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 data decoder output, DBUSF, can be selected as 0 or 1 in mask option and

it will control the display the digit “7” in two shapes accordingly as below:



DBUSF=0 **DBUSF=1**

MASK OPTION table:

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

Both the LCT and LCB instructions decode the content of the data memory according to the data-decoder table. When the content of the data memory that specified by a LCB instruction is “0”, the output data of DBUSA ~ DBUSH will be all “0”. (this is used for blanking the leading digit ”0” on the LCD panel).

The LCP instruction transfers the content of the RAM(Rx) and accumulator(AC) to ” DBUSA~DBUSH” bus directly, and bypass the data decoder.

The LCD instruction transfers the table ROM data (T@HL) to ” DBUSA~DBUSH” bus directly, and bypass the data decoder.

Table 2- 2 The bit mapping table for 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

A 8-bit data bus (DBUSA~DBUSH, DBUS) conveys the pattern data which will be displayed on the LCD panel. The DBUS data will be stored into the latch circuit.

The L0 to L4 decoder can decode the Lz data up to 32 strobe signals (PSTB 0h~PSTB 1Fh, PSTB) which can store the DBUS data into a specified latch in the latch circuitry.

Table 2- 3 The mapping table of strobe signals and Lz data in the LCD-related instructions

strobe signal for LCD latch	Lz data in LCT, LCB, LCP, LCD instructions The values of Lz in"LCT Lz, Q": *
PSTB 0	0H
PSTB 1	1H
PSTB 2	2H
PSTB 3	3H
PSTB 4	4H
PSTB 5	5H
……	…….
PSTB 1Ah	1AH
PSTB 1Bh	1BH
PSTB 1Ch	1CH
PSTB 1Dh	1DH
PSTB 1Eh	1EH
PSTB 1Fh	1FH

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

If we define one "pixel" is a pattern on LCD panel which corresponds to a specified segment and common, TM8722 can drive a LCD panel which contains up to 175(75 SEGs and 5 COMs) pixels. Each pixel needs a "pixel latch" to store its display information (ON or OFF), so there are total 175 pixel latches in the latch circuitry. The input data of the pixel latch comes from DBUS data and the storbe signal comes from PSTB signal.

The segment PLA determines the connection between DBUS data and the data input of a latch circuit, and so does the connection between PSTB signals and the strobe signal. The connection is controlled in mask option. Each latch circuit can select one of 8 DBUS data and select one of 32 PSTB signals. In this way, the configuration for the LCD panel's pixel is very flexible.

The following figure shows the relationship between segment PLA (mask option) and latch circuit.

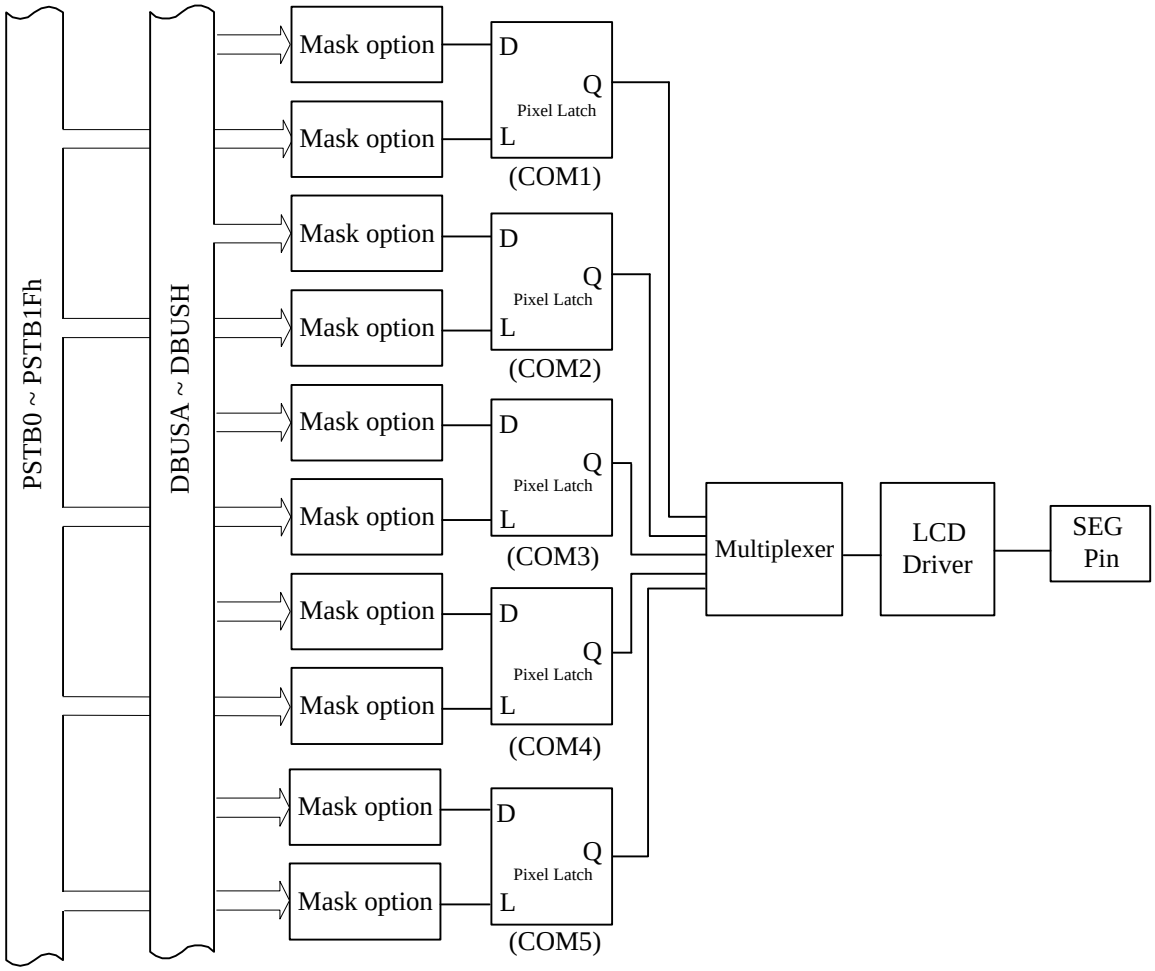


Figure: The diagram of Segment PLA Option

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

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

4.1.3.2 LCD-Related Instructions

1. LCT Lz, Ry

Decodes the content specified in Ry and stores the DBUS data into the latch circuit specified by Lz.

2. **LCB Lz, Ry**
   Decodes the content specified in Ry and stores the DBUS data into the latch circuit specified by Lz. All the DBUS data will be 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 DBUS and stores the DBUS data into the latch circuit specified by Lz. The mapping table is shown in Table2-2.

4. **LCP Lz, Ry**
   The data of the RAM and accumulator (AC) are transferred directly to DBUS and stores the DBUS data into the latch circuit specified by Lz. The mapping table is shown in Table2-4.

5. **LCT Lz, @HL**
   Decodes the content specified in index RAM (@HL) and stroes the DBUS data into the latch circuit specified by Lz.

6. **LCB Lz, @HL**
   Decodes the content specified in index RAM (@HL) and stores the DBUS data into the latch circuit specified by Lz. All the DBUS data will be 0 when the input data of the data decoder is 0.

7. **LCP Lz, @HL**
   The content of the index RAM(@HL) and accumulator (AC) are transferred directly to DBUS and stores the DBUS data in to the latch circuit specified by Lz. The mapping table is shown in Table2-4.

8. **SF2 4h**
   Turns off the LCD display.

9. **RF2 4h**
   Turns on the LCD display.

Table 2- 4 The mapping table for the 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

4.1.3.3 THE CONFIGURATION FILE FOR MASK OPTION

The *.cfg file(LCD configuration file) contains all the necessary information for segment PLA in mask option. This file records the SEG and COM pin Number that are connected to a pixel latch. It also records the DBUS data and PSTB strobe signal that a pixel latch needs to connect in segment PLA.

The syntax in *.cfg file is as follows:

SEG COM PSTB DBUS

SEG: Specifies the SEG pin No. that are connected to a pixel latch.

COM: Specifies the COM pin Number that are connected to a pixel latch. In this column, only 0, 1, 2, 3, 4, 5 and 9 can be entered into this field.

Data “1”~”5” represents the COM No., data ”0” represents this SEG pin is defined as the CMOS type DC output and data ”9” represents this SEG pin is defined as the P open-drain DC output.

PSTB: Specifies the strobe signal for the pixel latch.

DBUS: Specifies the DBUS data for the lpixel atch.

4.2 LED DRIVER OUTPUT

If the LED mode option is selected in the mask option, TM8722 will switch the LCD driver to the LED driver. TM8722 provides 35 segment pins(SEG) and 5 common pins(COM) to drive a LED module with 175 pixels.

For LED application, the COM pin can be selected as active low LED display or active high LED display in mask option. There are options for static, 1/2 duty, 1/3 duty, 1/4duty or 1/5 duty lighting systems. There are only 2 bias options can be selected in mask option, the one is 1/2 bias and the other is “No bias” option for the bias system.

In the LED mode, the segment output pins’ (SEG) waveforms are low active type.

MASK OPTION table:

When COM pins drives the high active LED panel

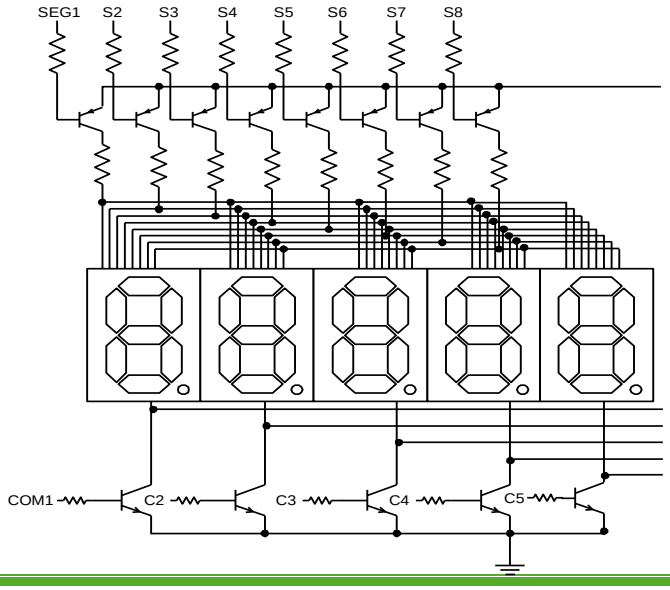
Mask Option name	Selected item
LCD/LED ACTIVE TYPE	(2) LED HIGH ACTIVE

When COM pins drives the low active LED panel

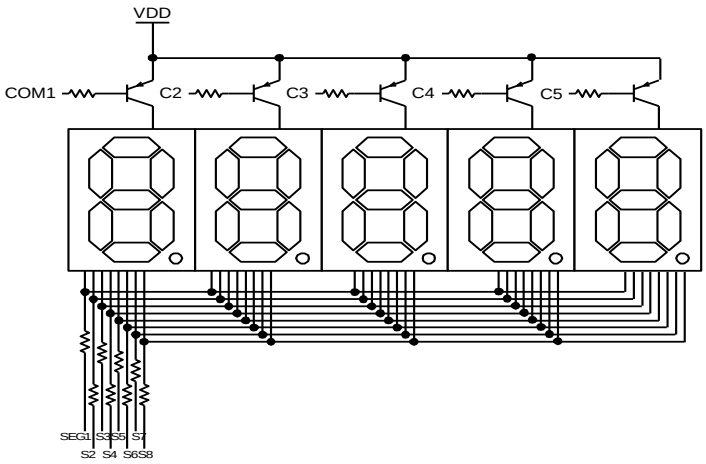
Mask Option name	Selected item
LCD/LED ACTIVE TYPE	(3) LED LOW ACTIVE

The following schematics will illustrate the difference between high active mode and low active mode:

(1) High Active Mode



(2) Low Active Mode



Note: Please limit the total sink current under 40mA for each COM pin at Low Active Mode..

The LED alternating frequency can be selected in mask option. (All the LED alternating frequencies are based on the predivider's clock source frequency, which is 32768Hz.)

The LED alternating frequency in 1/2 duty mode

LED duty cycle	1/2 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	32Hz	64Hz	128Hz

The LED alternating frequency in 1/3 duty mode

LED duty cycle	1/3 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	42Hz	85Hz	171Hz

The LED alternating frequency in 1/4 duty mode

LED duty cycle	1/4 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	32Hz	64Hz	128Hz

The LED alternating frequency in 1/4 duty mode

LED duty cycle	1/5 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	51Hz	102Hz	205Hz

LED Lighting System and Maximum Number of Driving LED Segments

LED Lighting System	Maximum Number of Driving LED Segments
Static	35
Duplex	70
1/3duty	105
1/4duty	140
1/5duty	175

The SEG pins can not only serve as the LED driver output pins but also as the CMOS type DC output port or the P open-drain DC output port in mask option. TM8722 allows some SEG pins to be the DC output ports and the remaining of the SEG pins to be the LED driver outputs.

In the LCD configuration file (*.cfg), if the data in the ”COM” column is “0”, the segment pin will be defined as the CMOS type output port. If the data in the “COM” column is “9”, the segment pin will be defined as the P open-drain type output port.

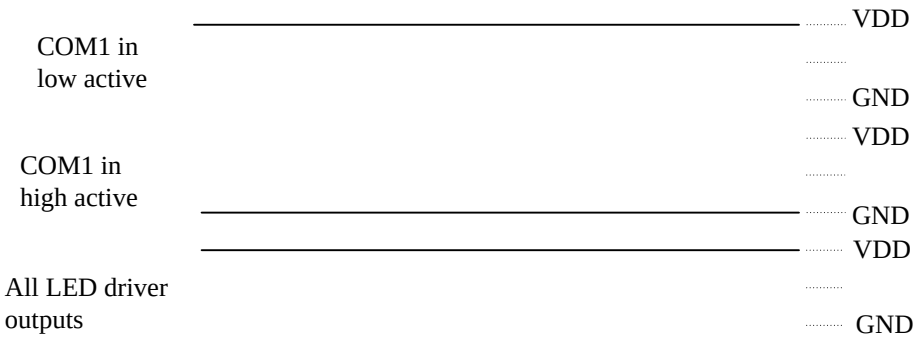
All the LED driver outputs can be selected as the CMOS type or the P open-drain type output in mask option. When a SEG pin is defined as the DC output port, the output data will remain intact even if the MCU enters the STOP mode or the LED turn-off mode is active.

During the initial reset cycle, all the LED pixels will be turned off as defined in the default setting because turning on all the LED pixels will cause large current consumption. All the LED output data will keep their initial settings until LED related instructions are executed to change their settings in the program.

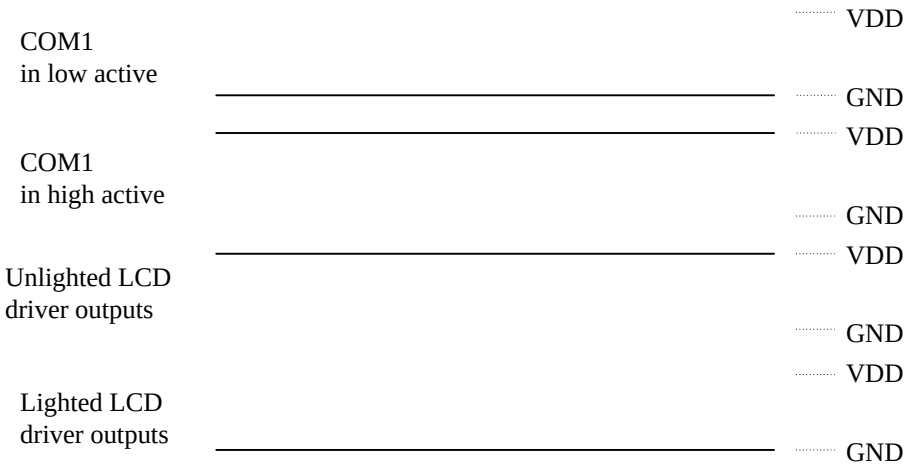
The waveform on the COM output and LED driver output for each LED lighting system are shown below.

4.2.1 STATIC LIGHTING SYSTEM FOR LED DRIVER

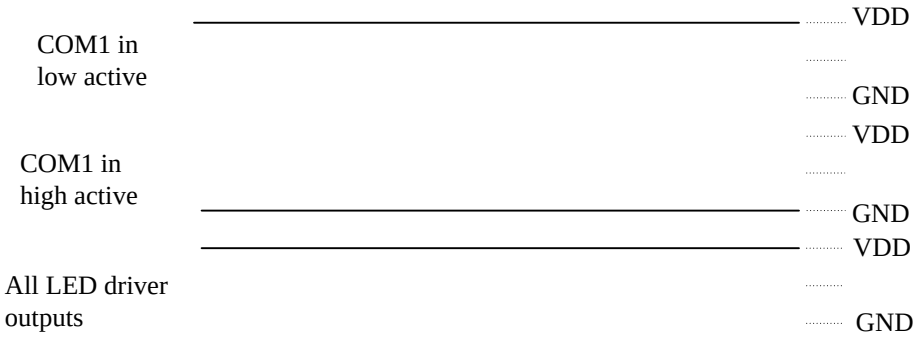
(i) Initial reset cycle



(ii) Normal operation mode



(iii) Display Turned Off



(iv) STOP Mode

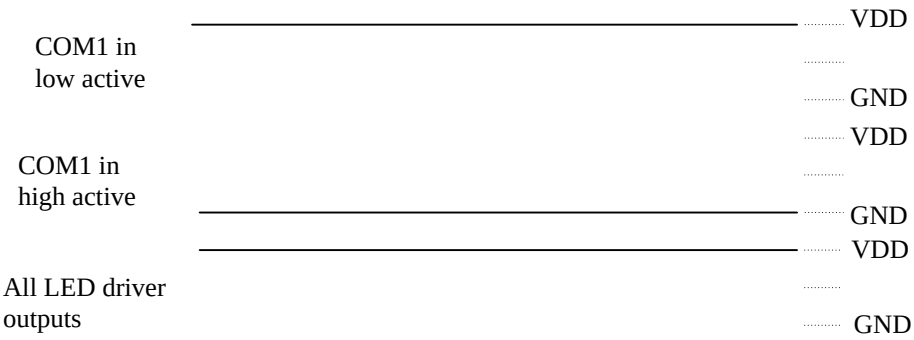
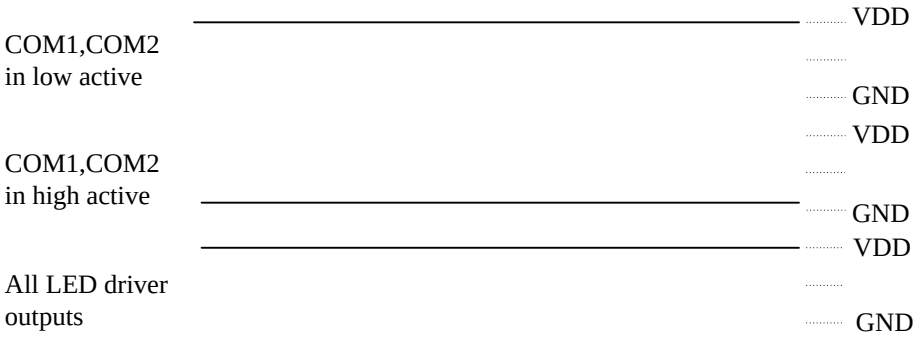


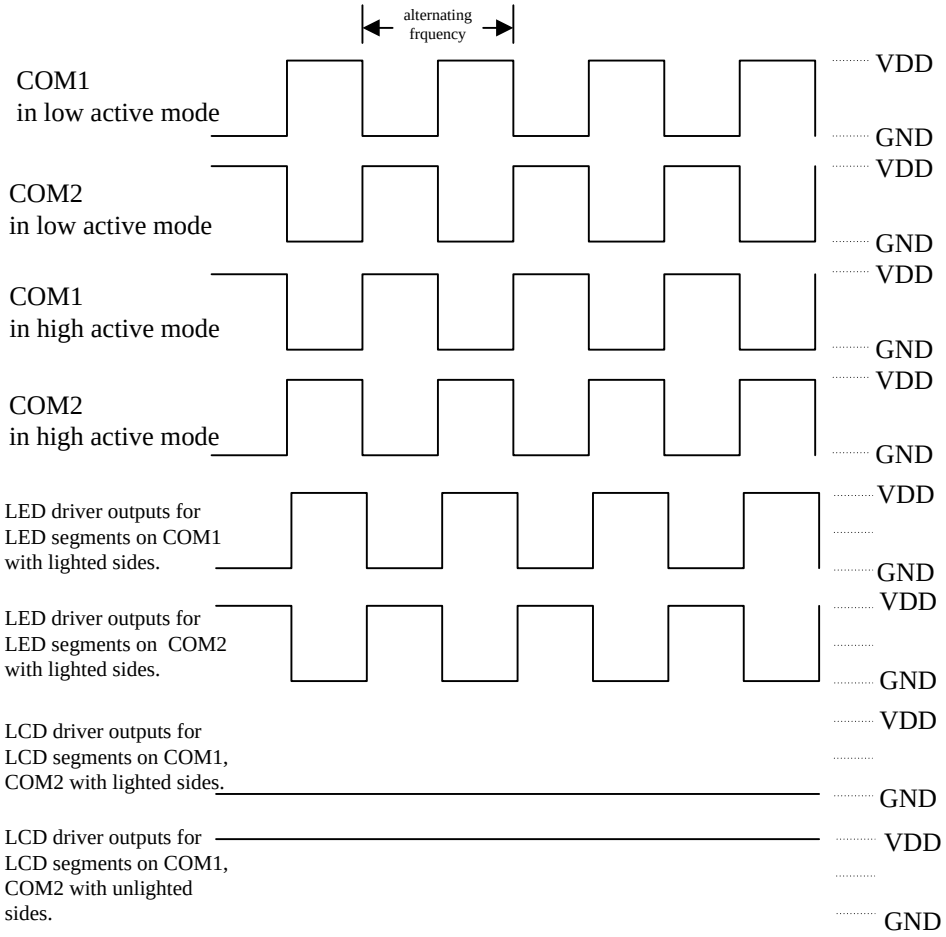
Figure 2- 39 Static LED Waveform

4.2.2 1/2 DUTY LIGHTING SYSTEM FOR LED DRIVER

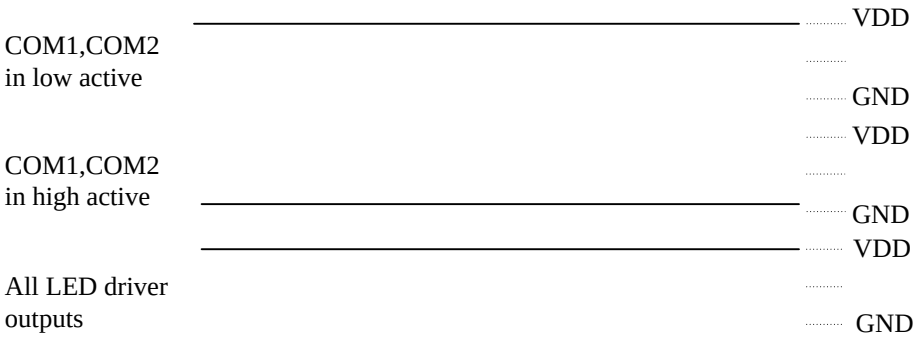
(i) Initial reset cycle



(ii) Normal operation mode



(iii) Display Turn Off



(iv) STOP Mode

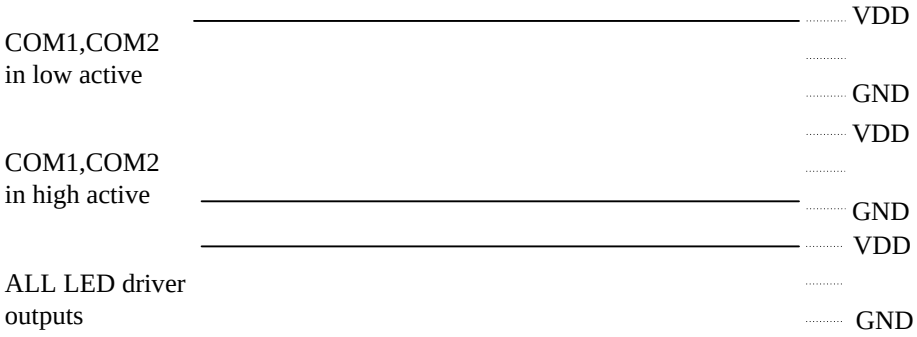
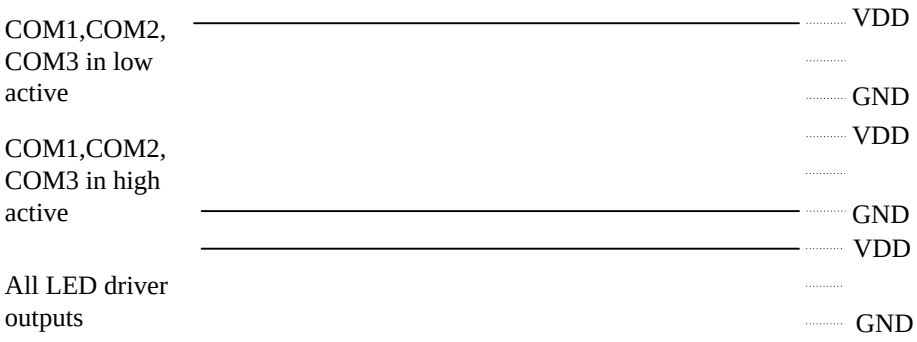


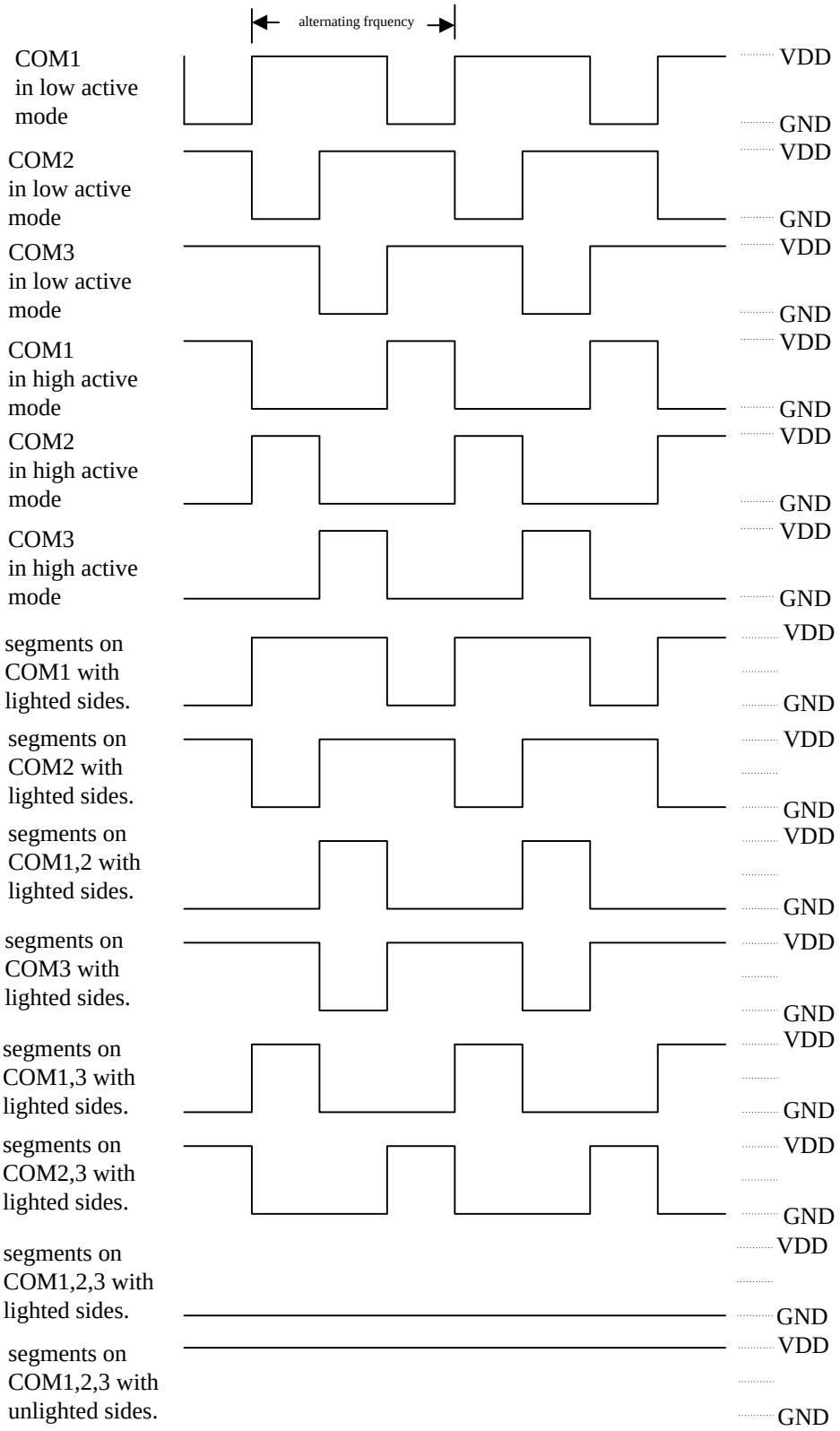
Figure 2- 40 Duplex(1/2 duty) LED Waveform

4.2.3 1/3 DUTY LIGHTING SYSTEM FOR LED DRIVER

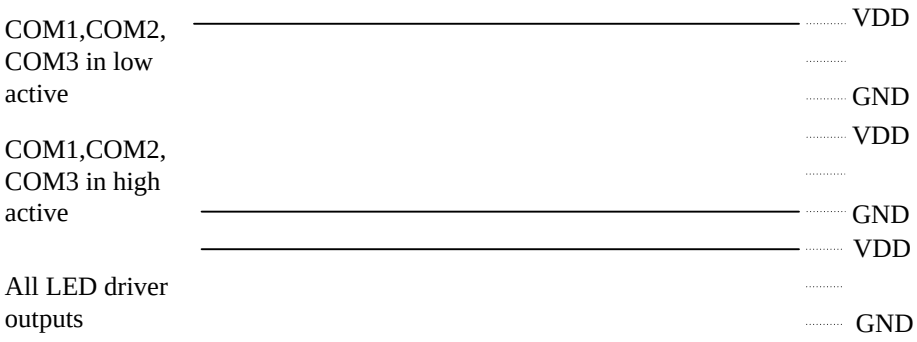
(i) Initial reset cycle



(ii) Normal operation mode



(iii) Display Turned Off



(iv) STOP mode

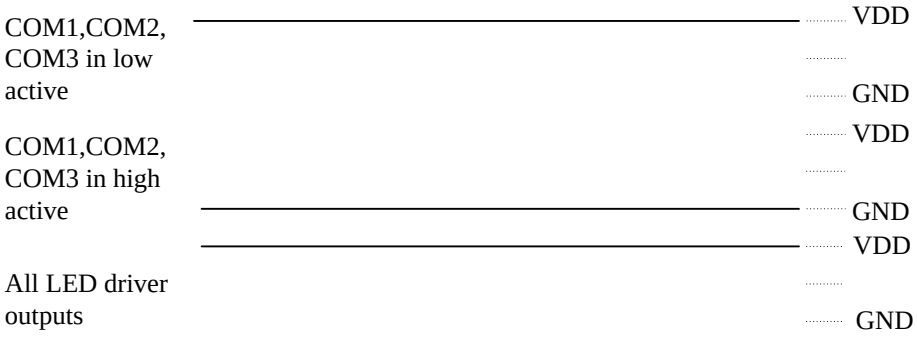
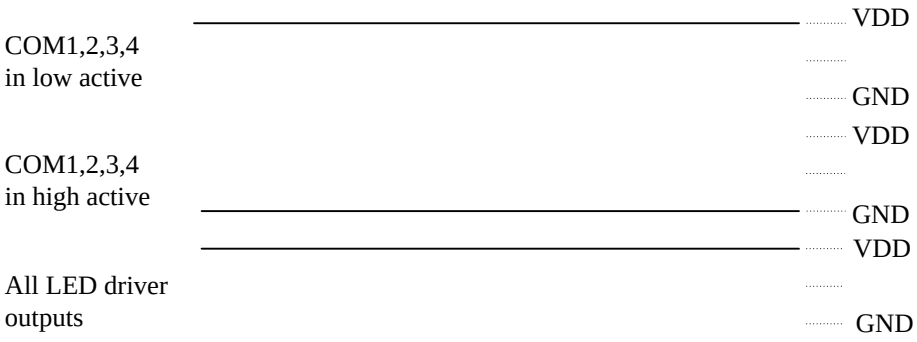


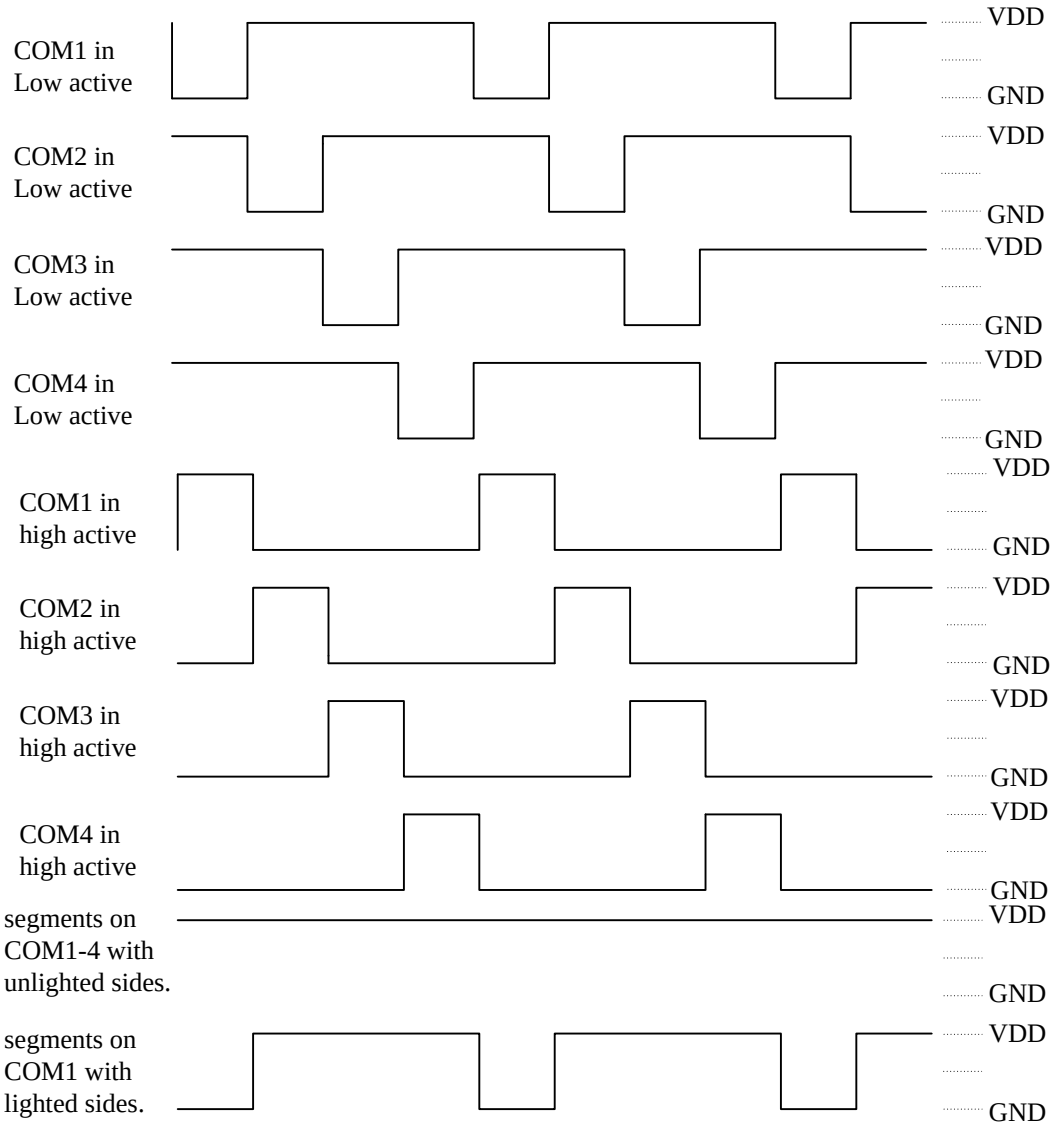
Figure 2- 41 1/3duty LED Waveform

4.2.4 1/4 DUTY LIGHTING SYSTEM FOR LED DRIVER

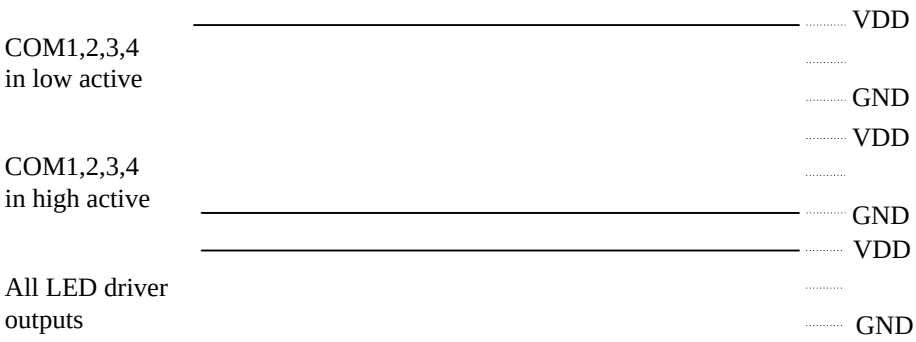
(i) Initial reset cycle (lighting)



(ii) Normal operation mode



(iii) Display Turned Off



(iv) Stop mode

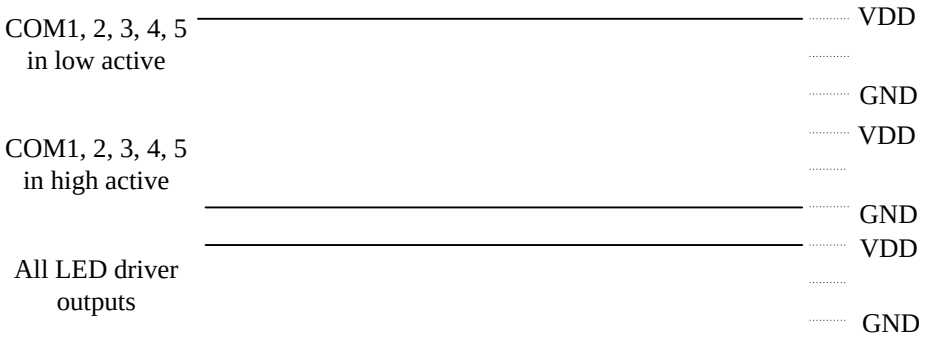
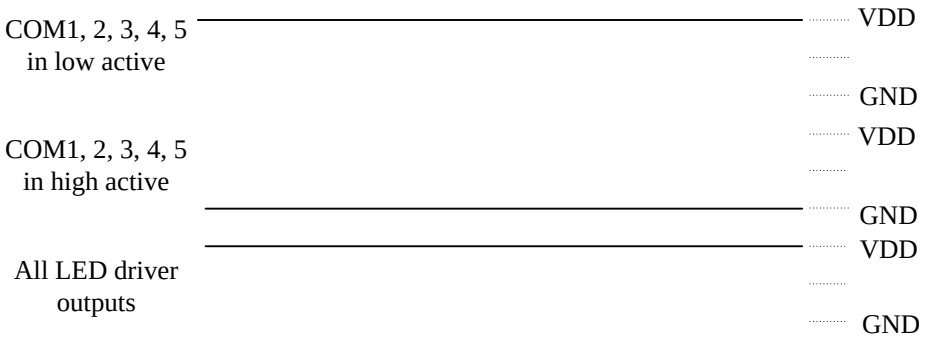


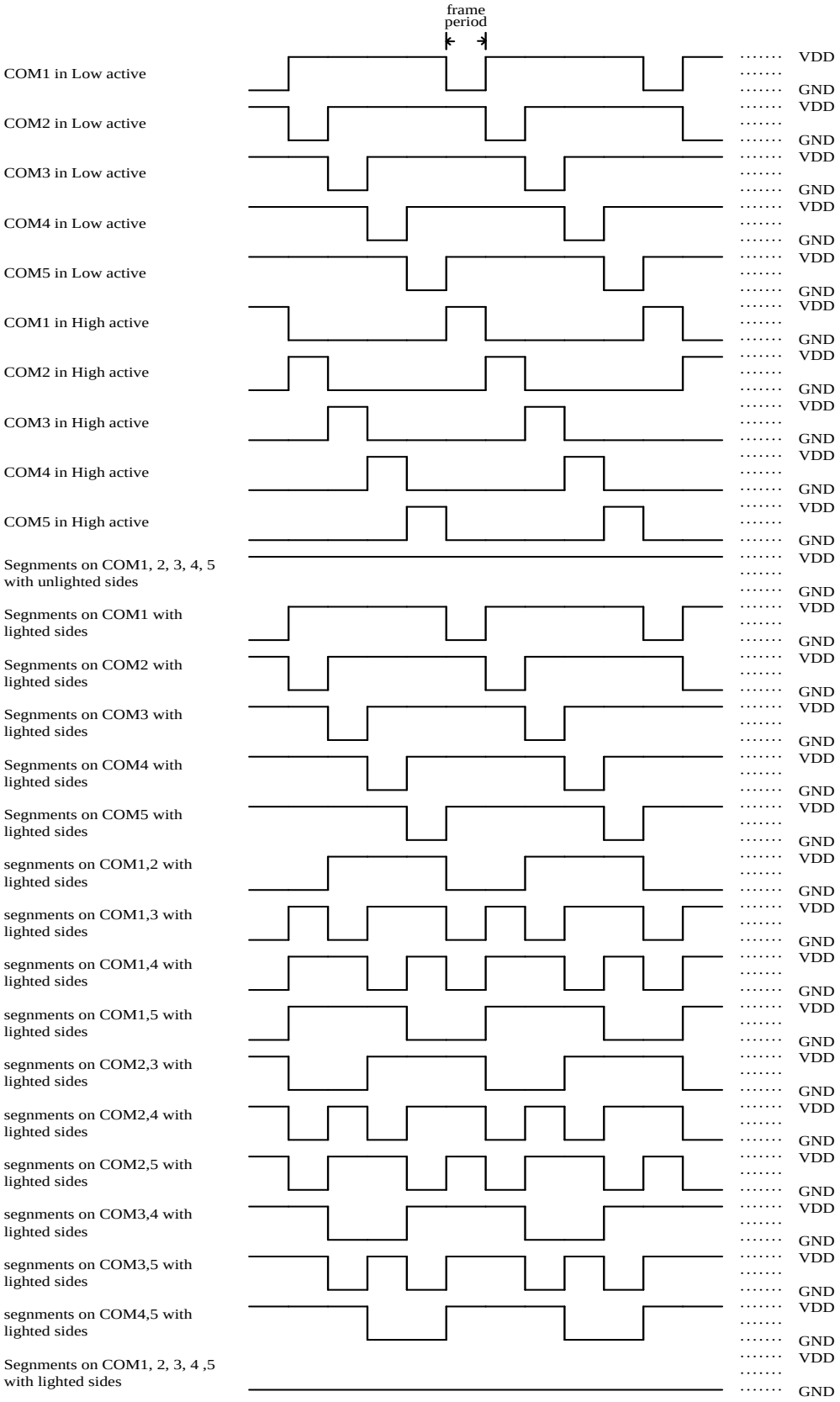
Figure 2- 42 1/4 duty LED Waveform

4.2.5 1/5 DUTY LIGHTING SYSTEM FOR LED DRIVER

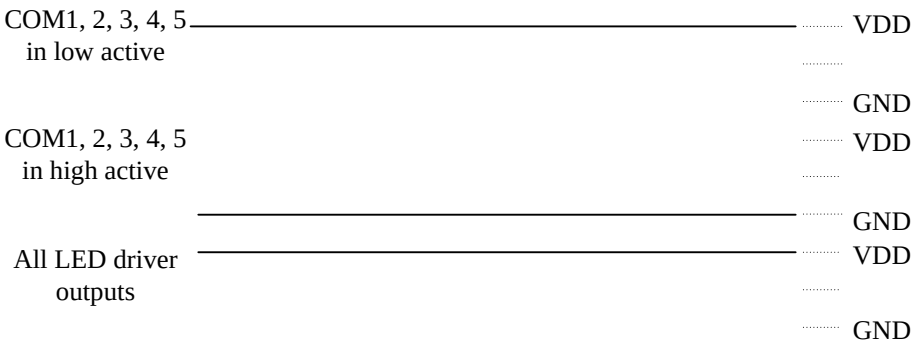
(i) Initial reset cycle (lighting)



(ii) Normal operation mode



(iii) Display Turned Off



(iv) Stop mode

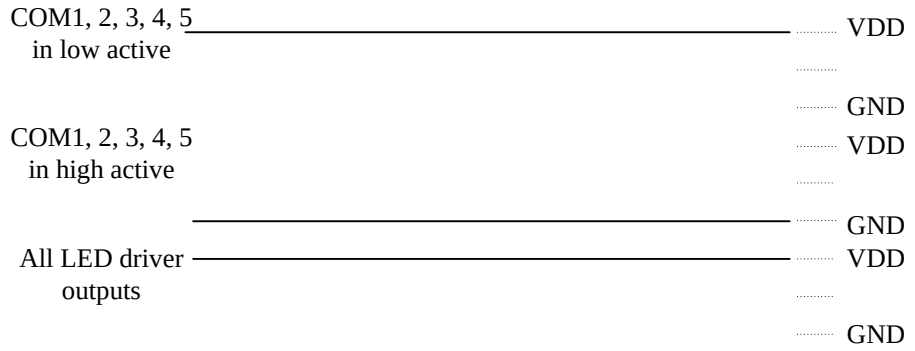


Figure 2- 43 1/5 duty LED Waveform

5. Detail Explanation of TM8722 Instructions

- It is recommended to initialize the content of data memory because the initial value of them is unknown.

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

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

Note: Ry: the Address of working register, the range of addresses specified by Rx is from 00H to 7FH.
Rx: the 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 register. This register can address all data memory and table ROM. The contents of the index address register can be changed 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 register (@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 LCD pixel latch which was configured in the segment PLA; the address range specified by Lz is from 00H to 1FH.

5.1 INPUT / OUTPUT INSTRUCTIONS

LCT Lz, Ry
function: LCD latch Lz ← data decoder ← (Ry)
description: The content of working register 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 content of working register 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 content of working register 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 are loaded to the LCD latch specified by Lz directly.

LCT Lz, @HL
function: LCD latch Lz ← data decoder ← (R@HL)
description: The content of index RAM specified by @HL are 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 content of index RAM specified by @HL are 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 content 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 the 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 has been set as input mode. The description of each bit of X(X3 X2 X1 X0) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable IOA pull low R	X4=0	Disable IOA pull low R
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 the I/OA port.

OPAS Rx, D

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

description: The Content of Rx is outputted to the IOA port. D is outputted to IOA3, and pulse is outputted to IOA4.
D=0 or 1

IPA Rx

function: Rx, AC ← (IOA)

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

SPB X

function: Defines the input/output mode of each pin for the 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 has been set as input mode. The description 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 content of Rx are outputted to the I/OB port.

IPB Rx

function: Rx, AC ← (IOB)

description: The data of the 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 has been set as input mode.

The description of each bit of X(X4 X3 X2 X1 X0) is shown below:

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enables all the pull-low and disables the low-level hold devices	X4=0	Disables all 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 the I/OC port.

IPC Rx
function: Rx, AC ← (IOC)
description: The data of the I/OC port is loaded to AC and data memory Rx.

SPK X
function: Sets the Key Matrix scanning output state.
description: When using the key matrix scanning function, at less one of SEG1~16 pin has to be defined as the LCD/LED driver pin in mask option, and the scanning output state is set by X(X6~0)

Bit Patten	Setting : Halt Release by
X6=0	Normal Key Scanning
X6=1	Scanning Cycle

The bit pattern of X (for Key Matrix scanning output to KO1~16)

X5	X4	X3	X2	X1	X0	KO1	KO2	KO3	KO4	KO5	KO6	KO7	KO8	KO9	KO10	KO11	KO12	KO13	KO14	KO15	KO16
0	0	0	0	0	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	0	1	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	1	0	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	0	1	1	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	0	0	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	0	1	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	0	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	1	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z	Hi-z
0	0	1	1	0	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z	Hi-z
0	0	1	1	1	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1	Hi-z
0	0	1	1	1	1	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	1
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z	Hi-z

Notes: 1. 1=H/L (LED/LCD)
2. KO1~16 = SEG1~16 output in scanning interval

ALM X

function: Sets the buzzer output frequency.

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

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

X8	X7	X6	clock source (higher frequency)
1	1	1	FREQ*
1	0	0	DC1
0	1	1	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 () is based on the ϕ0 32768Hz.

ELC X

function: The bit control of EL-light driver.

description: The description of each bit specified by X(X8 ~ X0) is shown below:

For ELP pin setting:

(X8, X7, X6)	Pumping clock frequency	(X5,X4)	Duty cycle
000	ϕ0	00	3/4 duty
100	BCLK	01	2/3 duty
101	BCLK/2	10	1/2 duty
110	BCLK/4	11	1/1 duty(original)
111	BCLK/8		

For ELC pin setting:

(X3, X2)	Discharge pulse frequency	(X1,X0)	Duty cycle
00	ϕ8	00	1/4 duty
01	ϕ7	01	1/3 duty
10	ϕ6	10	1/2 duty
11	ϕ5	11	1/1 duty(original)

SRF X

function: The operation control for RFC.

description: The meaning of each control bit(X5 ~ X0) is shown below:

X0=1	enables the RC oscillation network of RR	X0=0	disables the RC oscillation network of RR
X1=1	enables the RC oscillation network of RT	X1=0	disables the RC oscillation network of RT
X2=1	enables the RC oscillation network of RH	X2=0	disables the RC oscillation network of RH
X3=1	enables the 16-bit counter	X3=0	disables the 16-bit counter
X4=1	Timer 2 controls the 16-bit counter. X3 must be set to 1 when this bit is set to 1.	X4=0	Disables timer 2 to control the 16-bit counter.
X5=1	The 16-bit counter is controlled by the signal on CX pin. X3 must be set to 1 when this bit is set to 1.	X5=0	Disables the CX pin to control the 16-bit counter.

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

5.2 ACCUMULATOR MANIPULATION INSTRUCTIONS AND MEMORY MANIPULATION INSTRUCTIONS

MRW Ry, Rx

function: AC, Rx ← (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 the meaning of all the bits that after executing this instruction are as below:

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.
* The 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.
* The Carry flag (CF) will be affected.

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

DEC* @HL
function: R@HL, AC ← (R@HL)-1
description: Substract 1 from the content of data memory specified by @HL; the result is loaded to data memory specified by @HL and AC.
* The 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.
* The 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.
* The 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.
* The 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.
* The 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.
. The 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.
* The 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.
. The 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.
* The 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.
. The 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.
* The 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.
. The 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.
* The 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.
. The 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.
* The 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.
* The 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.
* The 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,Rx ← (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 ← (Rx)+ (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,Rx ← (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, Rx ← (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,Rx ← (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, Rx ← (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, Rx ← (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, Rx ← (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, Rx ← (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.

MRF1 Rx
function: Rx , AC ← RFC[3 ~ 0]
description: Loads the lowest nibble data of the16-bit counter of RFC to AC and data memory specified by Rx.
Bit 3 ← RFC[3]
Bit 2 ← RFC[2]
Bit 1 ← RFC[1]
Bit 0 ← RFC[0]

MRF2 Rx

function: Rx , AC ← RFC[7 ~ 4]

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

MRF3 Rx

function: Rx , AC ← RFC[11 ~ 8]

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

MRF4 Rx

function: Rx , AC ← RFC[15 ~ 12]

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

5.5 CPU CONTROL INSTRUCTIONS

NOP

function: no operation

description: no operation

HALT

function: Enters the halt mode

description: The following 3 conditions will 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 the completion of the interrupt service.

STOP

function: Enters the stop mode and stops all oscillators

description: Before executing this instruction, all signals on IOC port must be set to low.
The following 3 conditions cause the stop mode to be released.
1) One of the signals on KI1~4 is “H”/”L”(LED/LCD) during the scanning interval of the key-matrix scanning function.
2) A signal change in the INT pin.
3) One of the signals on the IOC port is "H".

SCA X

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

description: Specified the pins of IOC port which can release the Halt mode. 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 the interrupt enable flag

description:

X0=1	The IEF0 is set so that interrupt 0(Signal change at port IOC specified by SCA) can be accepted.
X1=1	The IEF1 is set so that interrupt 1 (underflow from timer 1) can be accepted.
X2=1	The IEF2 is set so that interrupt 2(the signal change at the INT pin) can be accepted.
X3=1	The IEF3 is set so that interrupt 3(overflow from the predivider) can be accepted.
X4=1	The IEF4 is set so that interrupt 4(underflow from timer 2) can be accepted.
X5=1	The IEF5 is set so that interrupt 5(key scanning) can be accepted.
X6=1	The IEF6 is set so that interrupt 6(overflow from the RFC counter) can be accepted.

X7 is reserved

SHE X

function: Set/Reset halt release enable flag

description:

X1=1	The HEF1 is set so that the halt mode can be released by TMR1 underflow.
X2=1	The HEF2 is set so that the halt mode can be released by signal changed on INT pin.
X3=1	The HEF3 is set so that the halt mode can be released by predivider overflow.
X4=1	The HEF4 is set so that the halt mode can be released by TMR2 underflow.
X5=1	The HEF5 is set so that the halt mode can be released by th signal is “H”/”L”(LED/LCD) on KI1~4 during scanning interval.
X6=1	The HEF6 is set so that the halt mode can be released by RFC counter overflow.

X7 is reserved

SRE X

function: Set/Reset stop release enable flag

description:

X4=1	The SRF4 is set so that the stop mode can be released by the signal changed on IOC port.
X5=1	The SRF5 is set so that the stop mode can be released by the signal changed on INT pin.
X7=1	The SRF6 is set so that the stop mode can be released by the signal is “H”/”L”(LED/LCD) on KI1~4 during scanning interval.

X6,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,BCF2,BCF
description: The contents of the SCF1, SCF2 and BCF flags are loaded to AC and the data memory specified by Rx.
The content of AC and the meaning of all the bits after the 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 SCF4,5,6,7,8,9	Halt release caused by the IOC port	The backup mode status

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 the meaning of all the bits 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

MCX Rx
function: AC, Rx ← SCF8,SCF6,SCF9
description: The SCF8,SCF6,SCF9 contents are loaded to AC and the data memory specified by Rx.
The content of AC and the meaning of all the bits after execution of this instruction are as follows:

Bit 3	Bit 2	Bit 1	Bit 0
Start condition flag 9 (SCF9)	NA	Start condition flag 6 (SCF6)	Start condition flag 8 (SCF8)
Halt release caused by RFC counter overflow	NA	Halt release caused by TM2 underflow	Halt release caused by the signal change to “H”/”L”(LED/LCD) on KI1~4 in scanning interval

MSD Rx

function: Rx, AC ← WDF,CSF,RFOVF

description: The watchdog flag, system clock status and overflow flag of 16-bit counter are loaded to data memory specified by Rx and AC.
The content of AC and the meaning of all the bits after execution of this instruction are as follows:

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

5.6 INDEX ADDRESS INSTRUCTIONS

MVH Rx

function: (@H) ← (Rx),(AC)

description: Loads the 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 the 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 binary format, and then restores to AC.
When this instruction is executed, the AC must be the result of an add instruction.
* The carry flag (CF) will be affected.

DAA* Rx

function: AC, Rx ← BCD(AC)

description: Converts the content of AC to binary format, and then restores to AC and data memory specified by Rx.
When this instruction is executed, the AC must be the result of an add 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 an add 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 of a substract 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 a substract 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 a substract 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 bit0 of AC is 0, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JB1 X

function: Program counter jumps to X if AC1=1.

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

JB2 X
function: Program counter jumps to X if AC2=1.
description: If bit2 of AC is 1 , jump occurs.
If bit2 of AC is 0 , the PC will increment by 1.
The range of X is from 000H to 7FFH.

JB3 X
function: Program counter jumps to X if AC3=1.
description: If bit3 of AC is 1 , jump occurs.
If bit3 of AC is 0, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JNZ X
function: Program counter jumps to X if (AC) != 0.
description: If the content of AC is not 0 , jump occurs.
If the content of AC is 0, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JNC X
function: Program counter jumps to X if CF=0.
description: If the content of CF is 0 , jump occurs.
If the content of CF is 1, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JZ X
function: Program counter jumps to X if (AC)=0.
description: If the content of AC is 0 , jump occurs.
If the content of AC is 1, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JC X
function: Program counter jumps to X if CF=1.
description: If the content of CF is 1 , jump occurs.
If the content of CF is 0, the PC will increment by 1.
The range of X is from 000H to 7FFH.

JMP X
function: Program counter jumps to X.
description: Unconditional jump.
The range of X is from 000H to 7FFH.

CALL X
function: STACK ← (PC)+1
Program counter jumps to X.
description: A subroutine is called.
The range of X is from 000H to 7FFH.

RTS
function: PC ← (STACK)
description: A return from a subroutine occurs.

5.9 MISCELLANEOUS INSTRUCTIONS

SCC X

function: Setting the clock source for IOA,IOC chattering prevention, PWM output and frequency generator.

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

Bit pattern	Clock source setting	Bit pattern	Clock source setting
X6=1	The clock source of frequency generator comes from the system clock (BCLK).	X6=0	The clock source of frequency generator comes from the PH0.
(X2,X1,X0)=001	Chattering prevention clock is PH10	(X2,X1,X0)=010	Chattering prevention clock is PH8
(X2,X1,X0)=100	Chattering prevention clock is PH6		

X7,5,4,3 is reserved

FRQ D, Rx

function: Frequency generator ← D, (Rx), (AC)

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

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

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

FRQ D, @HL

function: Frequency generator ← D, (T@HL)

description: Loads the content of Table ROM specified by @HL and D to frequency generator to set the duty cycle and the initial value. The following table shows the preset data and the duty cycle settings:

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

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

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

FRQX D, X

function: Frequency generator ← D, X

description: Loads the data X(X7 ~ X0) and D to frequency generator to set the duty cycle and the initial value. The following table shows the preset data and the duty cycle settings:

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

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

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

1. FRQ D, Rx
   Use the contents of Rx and AC as preset data N.
2. FRQ D, @HL
   Use the contents of tables TOM specified by index address buffer as preset data N.
3. FRQX D, X
   Use the data of operand in the instruction assigned as preset data N.

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
1	1	FREQ

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
1	1	FREQ

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
1	1	FREQ

TM2 Rx

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

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

The following table shows the bit pattern for this instruction:

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

The clock source setting for timer 2

AC3	AC2	clock source
0	0	ϕ9
0	1	ϕ3
1	0	ϕ15
1	1	FREQ

TM2 @HL

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

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

The following table shows the bit pattern for this instruction:

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

The clock source setting for timer 2

Bit7	Bit6	clock source
0	0	ϕ9
0	1	ϕ3
1	0	ϕ15
1	1	FREQ

TM2X X

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

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

The following table shows the bit pattern for this instruction:

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

The clock source setting for timer 2

X8	X7	X6	clock source
0	0	0	ϕ9
0	0	1	ϕ3
0	1	0	ϕ15
0	1	1	FREQ
1	0	0	ϕ5
1	0	1	ϕ7
1	1	0	ϕ11
1	1	1	ϕ13

SF X

function: Sets flags

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.

X2: "1" The EL-light driver output pin is active.

X3: "1" For X2=1 , when the SF instruction is executed at X3=1 , the EL-light driver is active and the halt request signal is outputted, then the chip enters the halt mode.

X4: "1" The watchdog timer is initiated and active.

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

X6,5 is reserved

RF X

machine code: 1111 0100 X700X4 0X2X1X0

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.

X2: "1" The EL-light driver is inactive.

X4: "1" The watchdog timer is inactive.

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

X6,5,3 is reserved

SF2 X

function: Sets flag

description: Description of each flag

X3: “1” Enable the low-resistance pull-low device on INT pin

X2: "1" Disables the LCD/LED segment output.

X1: "1" Sets the DED flag. Refer to 2-12-3 for detail.

X0: "1" Enables the re-load function of timer 2.

X7~6 is reserved

RF2 X

function: Resets flags

description: Description of each flag

X3: “1” Disable the low-resistance pull-low device on INT pin

X2: "1" Enables the LCD/LED segment output.

X1: "1" Resets the DED flag. Please refer to 2-12-3 for details.

X0: "1" Disables the re-load function of timer 2.

X7~6 is reserved

PLC

function: Pulse control

description: The pulse corresponding to the data specified by X is generated.

X0: "1" the Halt release request flag HRF0 caused by the signal at I/O port C is reset.

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

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

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

X4: "1" the Halt release request flag HRF4 caused by underflow from the timer 2 is reset and stops the operating of timer 2(TM2).

X5: "1" the Halt release request flag HRF5 caused by the signal change to “H”/”L”(LED/LCD) on KI1~4 during scanning interval is reset.

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

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

ORDERING INFORMATION

The ordering information:

Ordering number	Package
TM8722-COD	Wafer/Dice with code

Appendix A TM8722 Instruction Table

Instruction		Machine Code	Function		Flag/Remark
NOP		0000 0000 0000 0000	No Operation		
LCT	Lz,Ry	0000 001Z ZZZZ YYYY	(Lz)	←7SEG ← (Ry)	
LCB	Lz,Ry	0000 010Z ZZZZ YYYY	(Lz)	←7SEG ←(Ry)	Blank Zero
LCP	Lz,Ry	0000 011Z ZZZZ YYYY	(Lz)	← (Ry) , (AC)	
LCD	Lz,@HL	0000 100Z ZZZZ 0000	(Lz)	← (R@HL)	
LCT	Lz,@HL	0000 100Z ZZZZ 0001	(Lz)	←7SEG ←(R@HL)	
LCB	Lz,@HL	0000 100Z ZZZZ 0010	(Lz)	←7SEG ←(R@HL)	Blank Zero
LCP	Lz,@HL	0000 100Z 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)	
FRQ	D,Rx	0001 00DD 0XXX XXXX	FREQ D=00 D=01 D=10 D=11	← (Rx) , (AC) : 1/4 Duty : 1/3 Duty : 1/2 Duty : 1/1 Duty	
FRQ	D,@HL	0001 01DD 0000 0000	FREQ	←(T@HL)	
FRQX	D,X	0001 10DD XXXX XXXX	FREQ	← X	
MVL	Rx	0001 1100 0XXX XXXX	(@L)	← (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)	
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)	

Instruction		Machine Code	Function		Flag/Remark
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) B2 : SCF2(HRx) B1 : SCF1(CPT) B0 : BCF
MSC	Rx	0100 1100 0XXX XXXX	(AC),(Rx)	← STS3	B3 : SCF7(PDV) B2 : PH15 B1 : SCF5(TM1) B0 : SCF4(INT)
MCX	Rx	0100 1101 0XXX XXXX	(AC),(Rx)	← STS3X	B3 : SCF9(RFC) B2 : (No use) B1 : SCF6(TM2) B0 : SCF8(SKI)
MSD	Rx	0100 1110 0XXX XXXX	(AC),(Rx)	← STS4	B3 : (No use) B2 : RFOVF B1 : WDF B0 : CSF
SR0	Rx	0101 0000 0XXX XXXX	(AC)n, (Rx)n (AC)3, (Rx)3	← (Rx)(n+1) ← 0	

Instruction		Machine Code	Function		Flag/Remark
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	
MRF1	Rx	0110 0100 0XXX XXXX	(AC),(Rx)	← (RFC)3-0	
MRF2	Rx	0110 0101 0XXX XXXX	(AC),(Rx)	← (RFC7)-4	
MRF3	Rx	0110 0110 0XXX XXXX	(AC),(Rx)	← (RFC)11-8	
MRF4	Rx	0110 0111 0XXX XXXX	(AC),(Rx)	← (RFC)15-12	
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	0110 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
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 0X00 0XXX	X6 = 1 X6 = 0 X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cfq = BCLK : Cfq = PH0 : Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1101 1010 000X 0000	X4	: Enable SEF4	C1-4
SPA	X	1101 1100 000X XXXX	X4	: Set IOA4-1 Pull-Low	

Instruction		Machine Code	Function		Flag/Remark
			X3~0	: Set IOA4-1 I/O	
SPB	X	1101 1101 000X XXXX	X4 X3~0	: Set IOB4-1 Pull-Low : Set IOB4-1 I/O	
SPC	X	1101 1110 000X XXXX	X4 X3-0	: Set IOC4-1 Pull-Low / Low-Level-Hold : Set IOC4-1 I/O	
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 = 11 X7,6 = 10 X7,6 = 01 X7,6 = 00 X5~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
SPK	X	1110 0011 0XXX XXXX	X6=1 X6=0 X5 =1 X4 =1 X3~0	: KEY_S release by scanning cycle : KEY_S release by normal key scanning : Set all Hi-z : Set all = 1 : Set n of 16	 IOC=normal IOC=KEY SCAN IOC=KEY SCAN
TM2	Rx	1110 0100 0XXX XXXX	Timer2	← (Rx) & (AC)	
TM2	@HL	1110 0101 0000 0000	Timer2	← (T@HL)	
TM2X	X	1110 011X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer2 Value	
SHE	X	1110 1000 0XXX XXX0	X6 X5 X4 X3 X2 X1	: Enable HEF6 : Enable HEF5 : Enable HEF4 : Enable HEF3 : Enable HEF2 : Enable HEF1	RFC KEY_S TMR2 PDV INT TMR1
SIE*	X	1110 1001 0XXX XXXX	X6 X5 X4 X3 X2 X1 X0	: Enable IEF6 : Enable IEF5 : Enable IEF4 : Enable IEF3 : Enable IEF2 : Enable IEF1 : Enable IEF0	RFC KEY_S TMR2 PDV INT TMR1 CPT
PLC	X	1110 101X 0XXX XXXX	X8 X6-0	: Reset PH15~11 : Reset HRF6-0	
SRF	X	1110 1100 00XX XXXX	X5 X4 X3 X2 X1 X0	: Enable Cx Control : Enable TM2 Control : Enable Counter : Enable RH Output : Enable RT Output : Enable RR Output	 ENX EHM ETP ERR
SRE	X	1110 1101 X0XX 0000	X7 X5 X4	: Enable SRF7 : Enable SRF5 : Enable SRF4	SRF7(KEY_S) 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	: Reload 1 Set	RL1

Instruction		Machine Code	Function		Flag/Remark
			X4 X3 X2 X1 X0	: WDT Enable : HALT after EL : EL LIGHT On : BCF Set : CF Set	WDF BCF CF
RF	X	1111 0100 X00X 0XXX	X7 X4 X2 X1 X0	:Reload 1 Reset : WDT Reset : EL LIGHT Off : BCF Reset : CF Reset	RL1 WDF BCF CF
SF2	X	1111 1000 0000 XXXX	X3 X2 X1 X0	: Enable INT powerful Pull-low : Close all Segments : Dis-ENX Set : Reload 2 Set	INTPL RSOFF DED RL2
RF2	X	1111 1001 0000 XXXX	X3 X2 X1 X0	: Disable INT powerful Pull-low : Release Segments : Dis-ENX Reset : Reload 2 Reset	INTPL RSOFF DED RL2
ALM	X	1111 101X XXXX XXXX	X8,7,6=111 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: FREQ : DC1 : PH3 : PH4 : PH5 : DC0 ← PH15~10	
ELC	X	1111 110X XXXX XXXX	X8=1 X8=0 X7,6=11 X7,6=10 X7,6=01 X7,6=00 X5,4=11 X5,4=10 X5,4=01 X5,4=00 X3,2=11 X3,2=10 X3,2=01 X3,2=00 X1,0=11 X1,0=10 X1,0=01 X1,0=00	BCLKX PH0 BCLK/8 BCLK/4 BCLK/2 BCLK 1/1 1/2 2/3 3/4 PH5 PH6 PH7 PH8 1/1 1/2 1/3 1/4	ELP - CLK BCLKX ELP - DUTY ELC - CLK ELC - DUTY
HALT		1111 1110 0000 0000	Halt Operation		
STOP		1111 1111 0000 0000	Stop Operation		

Symbol Description

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

Appendix B TM87 series Application Note

1. AP-TM87XX_11EV10

TITLE

How to keep the content of data RAM for the duration of the Reset

APPLICATION NOTE

In TM8722 MCU, the content of data RAM (specified in 00h, 10h, 20h, 30h … addresses) will be disturbed when MCU entering reset cycle **(Note-1)**.

If it is required to keep the content of data RAM during the reset cycle, please do not store such data to the specified addresses that mentioned above.

However, TM87 series MCUs do not guarantee such application can keep the content of data well while the MCU released from reset cycle. Please avoid using this application as possible in case of other uncertainty factors.

Note-1:

The Reset cycle can be triggered by RESET pin when activated, Watch dog timer overflows and Key Reset activates.

2. AP-TM87XX_13EV10

TITLE

How to minimize the power noise in the system design which will cause MCU malfunction

APPLICATION NOTE

There are two reference tips to minimize the power noise in system design:

1. Active external RESET pin function with “level reset” option only, the Mask Option items are as below:

   POWER ON RESET: NO USE

   RESET PIN TYPE: LEVEL

   IOC1/KI1 FOR KEY RESET: NO USE

   IOC2/KI2 FOR KEY RESET: NO USE

   IOC3/KI3 FOR KEY RESET: NO USE

   IOC4/KI4 FOR KEY RESET: NO USE

2. Take care of the placement and route on PCB Layout, the details are as below:

- Minimize the routes between MCU and power capacitors.
- Minimize the routes between MCU and the capacitor which connect to RESET pin.
- Minimize the routes between MCU and Crystal and trim the external capacitors for the oscillator driver as well.
- Insert the ground shielding on PCB as you can.

Minimize the route for each signal on PCB.

3. AP-SZ061_02EV10

TITLE

Use I/O port to implement the Key Control function Control

APPLICATION NOTE

I. Basic features

1. CPU working voltage 3V
2. CPU working frequency：32.768 KHz Crystal
3. 3 control key operating functions

II. The definition of the keys

Key A：Connect to IOC1，key pressed，EL lights up and goes out after 2 seconds.
Key B：Connect to IOC2，key pressed for 2 seconds, will generate key tone.
Key C：Connect to IOC3，key pressed and will generate key tone after release.

III. Talbe ROM addressing

For example: to find the content of the address 386H in the Talbe ROM （use TM8726 as an example）, there are two steps to accomplish the above function：

1. Specify address.
2. Read the data from the address.

Specific instructions are as follows:（R0, R1 are data registers）

```
LDS     R0 ,   03H    ；Set R0=03H
MVU     R0            ；Specify the high portion of the address is 3
LDS     R0 ,   08H
MVH     R0            ；Specify the next portion of the address is 8
LDS     R0 ,   06H
MVL     R0            ；Specify the low portion of the address is 6, now the address is
                        completely specified.
```

```
LDL     R0 ,   @HL   ；Read the lower nibble of the data to R0.
LDH     R1 ,   @HL   ；Read the lower nibble of the data to R1.
```

If the address 386H of the Talbe ROM contains the data of 36H, R0=6H，R1=3H.

Some IC such as TM8722, TM8723 etc. do not have 'MVU' instruction, therefore, their addressing method will be slightly different from that of the TM8726.

The instructions are as follows：

```
LDS     R0,    08H
LDS     R1,     03H    ；Set Acc=03H
MVH     R0             ；Specify the high address portion is 38 H
LDS     R0,    00H
MVL     R0
LDL     R0,    @H L
LDH     R1,    @H L
```

IV. RAM addressing

There are two addressing modes for RAM addressing: Direct addressing and indirect addressing. RAM locations between 00H~7FH can be addressed using both methods. RAM location after 80 H can only be addressed by the indirect addressing method.

When using indirect addressing, the RAM addressing mode is very similar to the ROM addressing mod. The following is a simple example using TM8726. If the memory location 235H of RAM will be accessed, the address needs to be specified first then read from the location.

```
LDS     R0,    02H
MVU     R0             ；Specify the high portion of RAM address.
LDS     R0,    03H
MVH     R0             ；Specify the next portion of RAM address.
LDS     R0,    05H
MVL     R0             ；Specify the low portion of RAM address.
```

Instructions to Read are：MWR RX, @HL；LDA @HL etc.

Instructions to Write are：MRW @ HL, RX ；INC* HL；STA @HL etc.

V. A few applications tips

1. Use the SCC instruction to choose the debounce time such as 32 ms（Ph10）, 8 ms（Ph8）,2 ms（Ph6）. The default value is 32 ms（Ph10）
2. When the IOC pins is set to be the key input pins, the IOC pins will generate HALT release signal or interrupt signal at the rising or falling edge. Therefore, if inquiry about the duration of a key is pressed or released is desired, the timer interrupt (or HALT release) needs to be enabled for polling the pin states and processed properly to prevent the fault signals.
3. When release a key, it will generate the falling edge and HALT signal (or interrupt) at the same time, so the key processing procedure should be ended when the value of the input pin is 0 after halt release (or interrupt).
4. When a key is pressed and another key is pressed at the same time, the HALT signal or interrupt will not be generated. Therefore, when handling multiple pressed keys, the timer interrupt (or HALT release) needs to be enabled to do the key scanning procedure.

VI. EL light control explanation

A：Description of the usages:

1. Turning on EL light consumes large current. Therefore, the BCF should be set to 1, that is execute instruction SF 02H, before turning on EL light.
2. ELC instruction should be executed to choose the relevant parameters.
3. SF 04 H should be executed to activate the EL panel driver.
4. Choose the ELC and ELP pin functions in Mask option.
5. Use RF 4H instruction to turn off EL light.

B：Setting the parameters of ELC/ELP pins by executing ELC instruction:

(X represents the operand in ELC instruction)

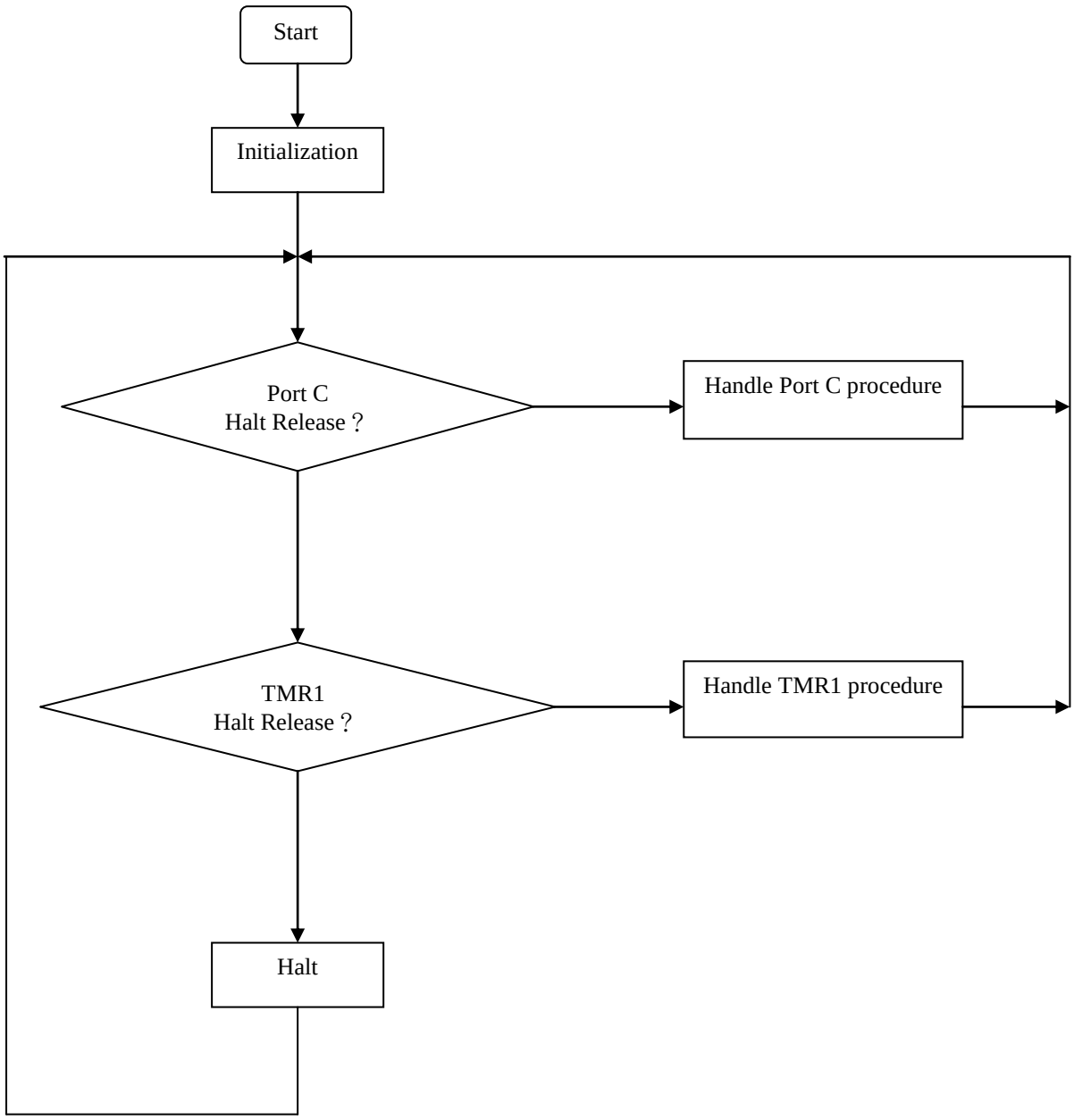
1. ELP pin setup ○○○

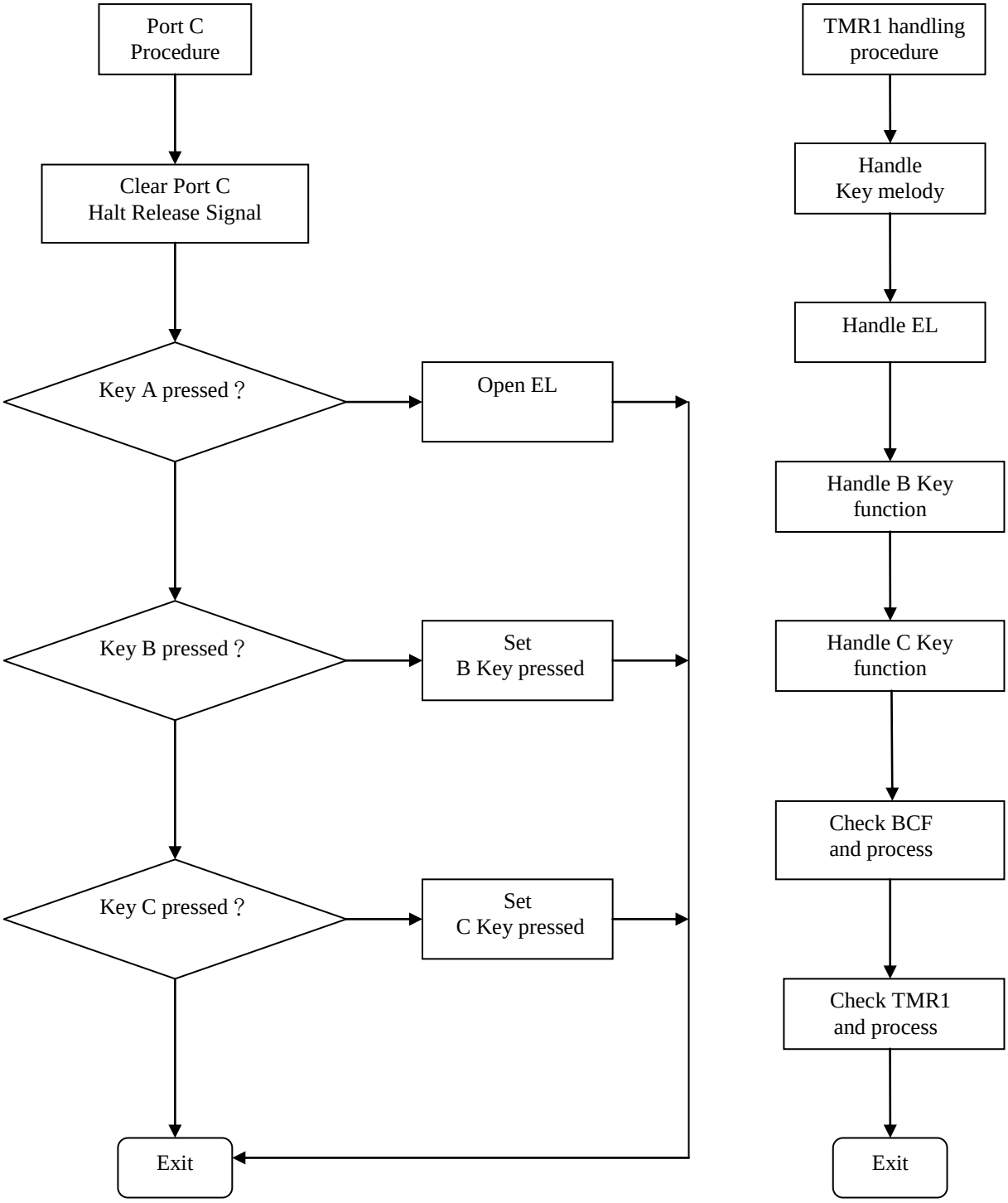
X8 X7 X6	clock frequency for rising voltage	X5 X5 X4	duty cycle
0 0 0	φ_O	1 0 0	3/4 duty
1 0 0	BCLR	1 1 1	2/3 duty
1 0 1	BCLR/2	X 1 0	1/2 duty
1 1 0	BCLR/4	X 1 1	1/1 duty
1 1 1	BCLR/8	0 0 1	1/3 duty
		0 0 0	1/4 duty

2. ELC pin setup

X3 X2	Discharge pulse frequency	X1 r0	duty cycle
0 0	φ8	0 0	1/4 duty
0 1	φ7	0 1	1/3 duty
1 0	φ6	1 0	1/2 duty
1 1	φ5	1 1	1/1 duty

VII. Flow Chart




Port C
Procedure
Clear Port C
Halt Release Signal
Key A pressed ?
Open EL
Key B pressed ?
Set
B Key pressed
Key C pressed ?
Set
C Key pressed
Exit
TMR1 handling
procedure
Handle
Key melody
Handle EL
Handle B Key
function
Handle C Key
function
Check BCF
and process
Check TMR1
and process
Exit

4. AP-SZ062_01EV10

TITLE

Introduction to all the clock modes in the TM87 series

APPLICATION NOTE

We can use mask option to choose the clock sources. There are three clock sources to choose from for the TM87 series MCU. Fast only, Slow only and Dual clock.

I. Fast only mode

- There are three clock sources to choose from in the Fast only mode: the first one is to use an external ceramic resonator, the second is to use external resistor to generate oscillation, and the third is to use internal RC oscillation (choose 250KHz or 500KHz).
- In the Fast only mode, the clock signals generated by the fast clock oscillator can be used as the basic frequency for the circuitries such as the system clock generator, pre-Divider, timer, I/O chattering prevention and LCD driver module etc.
- When the MCU executes the STOP instruction and enters the STOP mode, the fast clock oscillator will be disabled and stop oscillating.

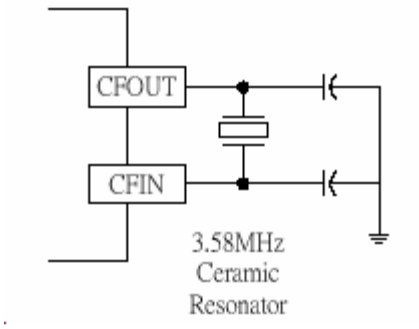
1. Using external ceramic resonator

- Do not use the 3.58 MHZ ceramic resonator in the 1.5V Ag battery mode.
- Set the BCF flag to 1 before enable 3.58MHZ ceramic resonator in the 3V Li battery mode。

For TM8726 example & Mask Option Program:

2.CLOCK SOURCE :
☑ A.FAST ONLY ☐ .SLOW ONLY ☐ .DUAL

33.FAST CLOCK TYPE FOR ONLY OR DUAL :
☐ A.INTERNAL RESISTOR FOR 250KHz ☐ .INTERNAL RESISTOR FOR 500KHz
☐ .EXTERNAL RESISTOR ☑ .3.58MHz CERAMIC RESONATOR

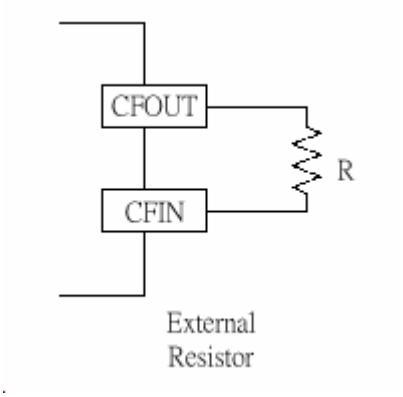


2. Using the external resistor to generate oscillation

For TM8726 example & Mask Option Program:

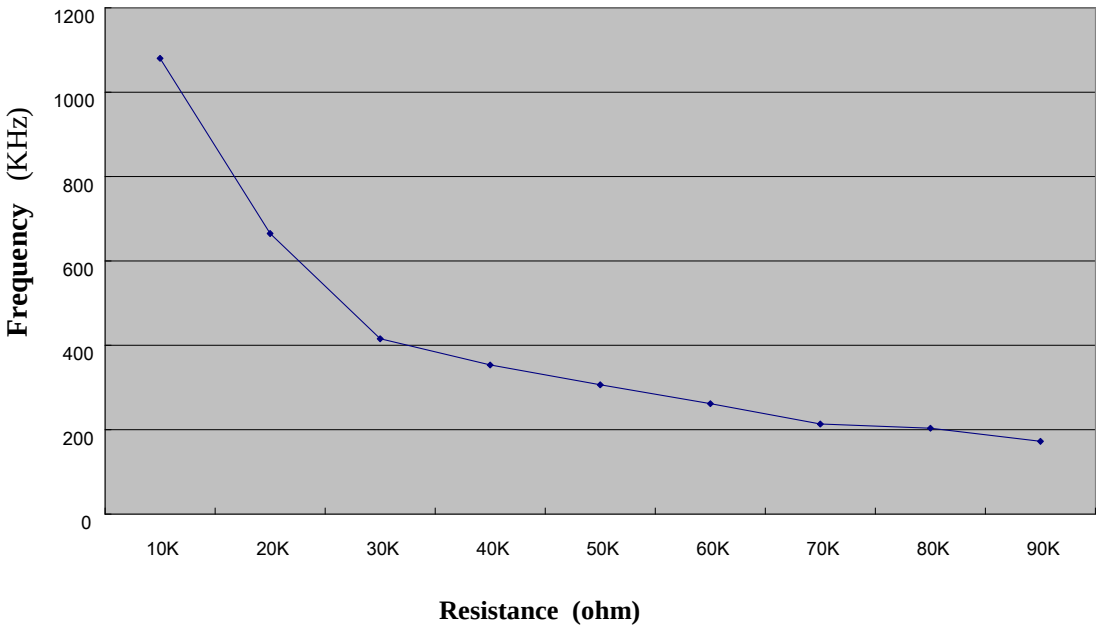
2.CLOCK SOURCE :
☑ A.FAST ONLY ☐ .SLOW ONLY ☐ .DUAL

33.FAST CLOCK TYPE FOR ONLY OR DUAL :
☐ A.INTERNAL RESISTOR FOR 250KHz ☐ .INTERNAL RESISTOR FOR 500KHz
☑ .EXTERNAL RESISTOR ☐ .3.58MHz CERAMIC RESONATOR

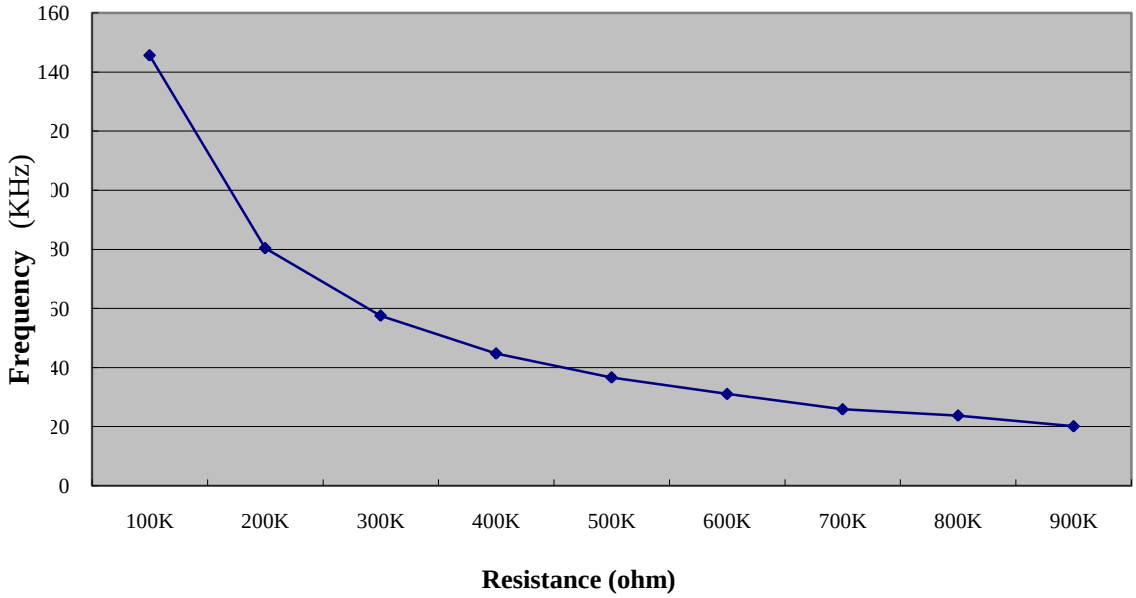


- When using the external resistor to generate oscillation in 3V operating application, the resistance and frequency relation chart will be:

TM8722 3.0V fast Ext-R R v.s. F

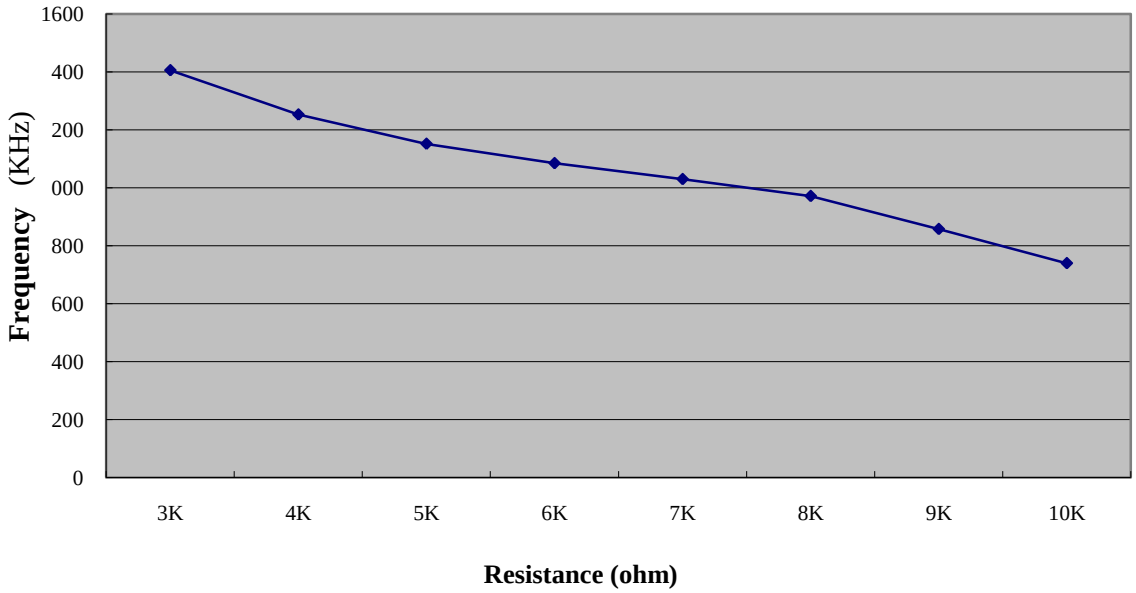


TM8722 3.0V fast Ext-R R v.s. F

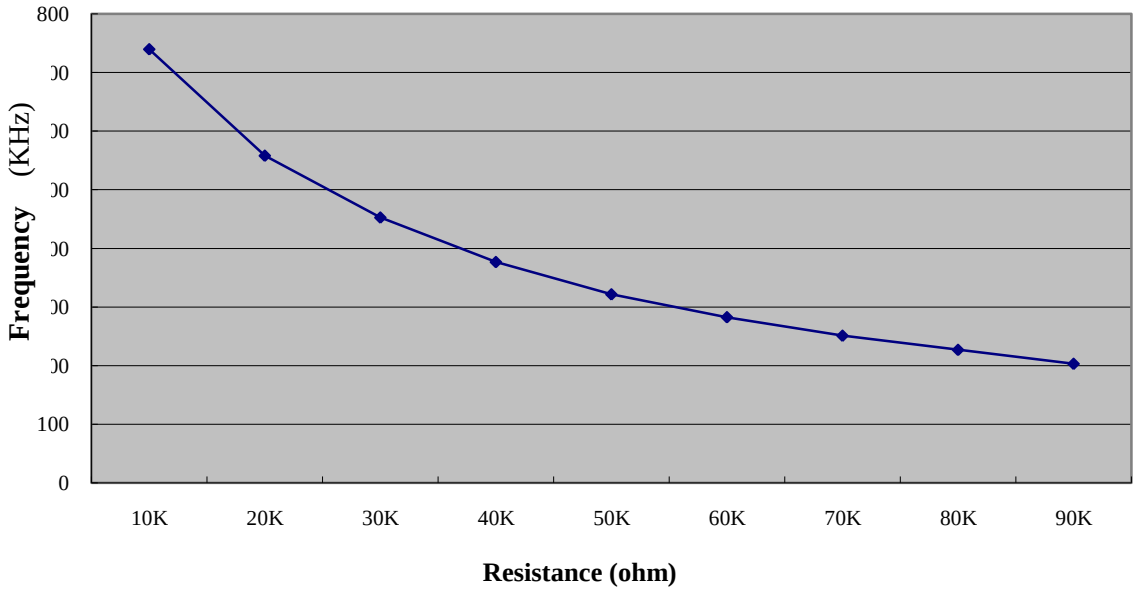


- When using the external resistor to generate oscillation in 1.5V operating application, the resistance and frequency relation chart will be:

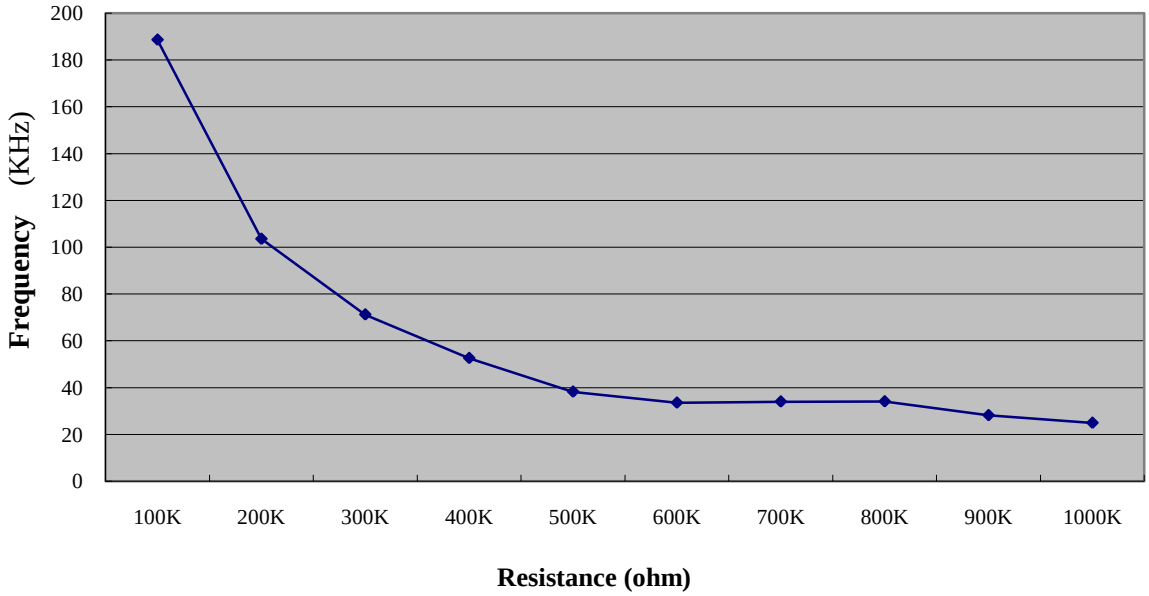
TM8722 1.5V fast EXT-R R v.s. F



TM8722 1.5V fast EXT-R R v.s. F



TM8722 1.5V fast Ext-R R v.s. F

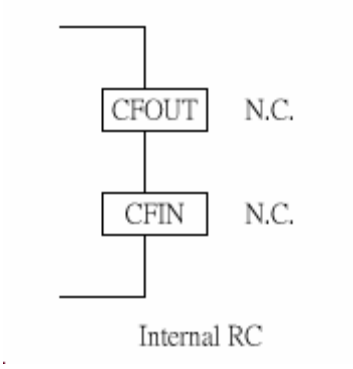


3. Using internal RC oscillation, choose 250KHz or 500KHz

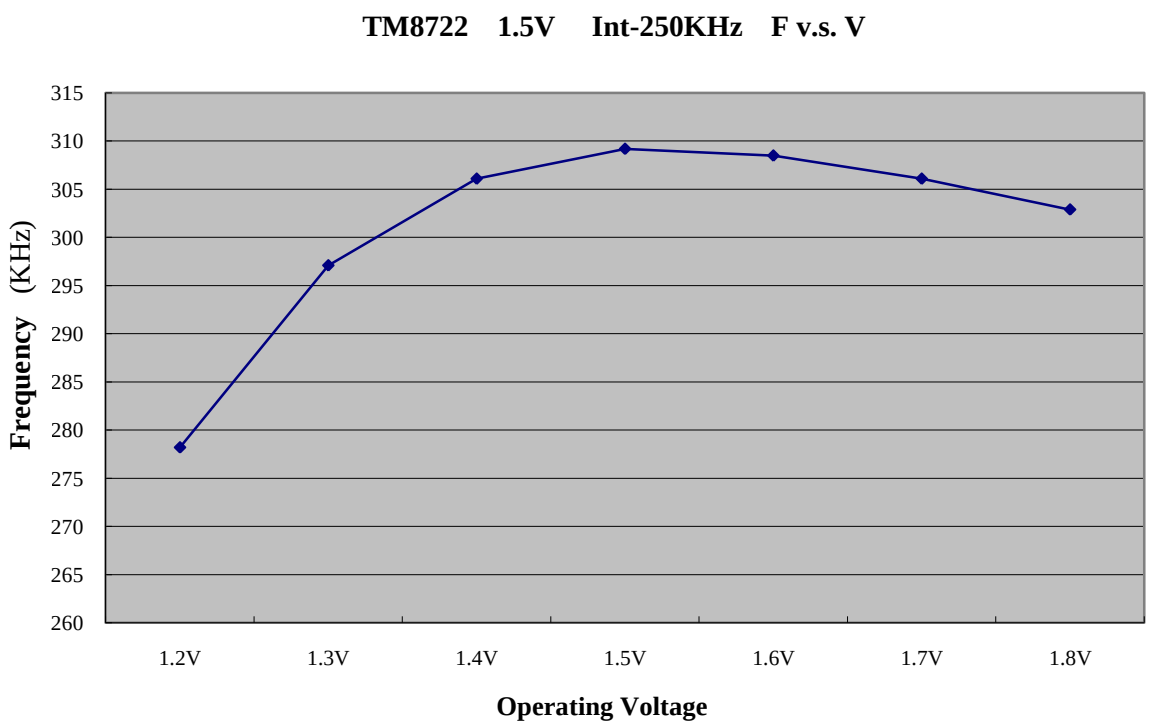
For TM8726 example & Mask Option Program:

2.CLOCK SOURCE :
☑ A.FAST ONLY ☐ .SLOW ONLY ☐ .DUAL

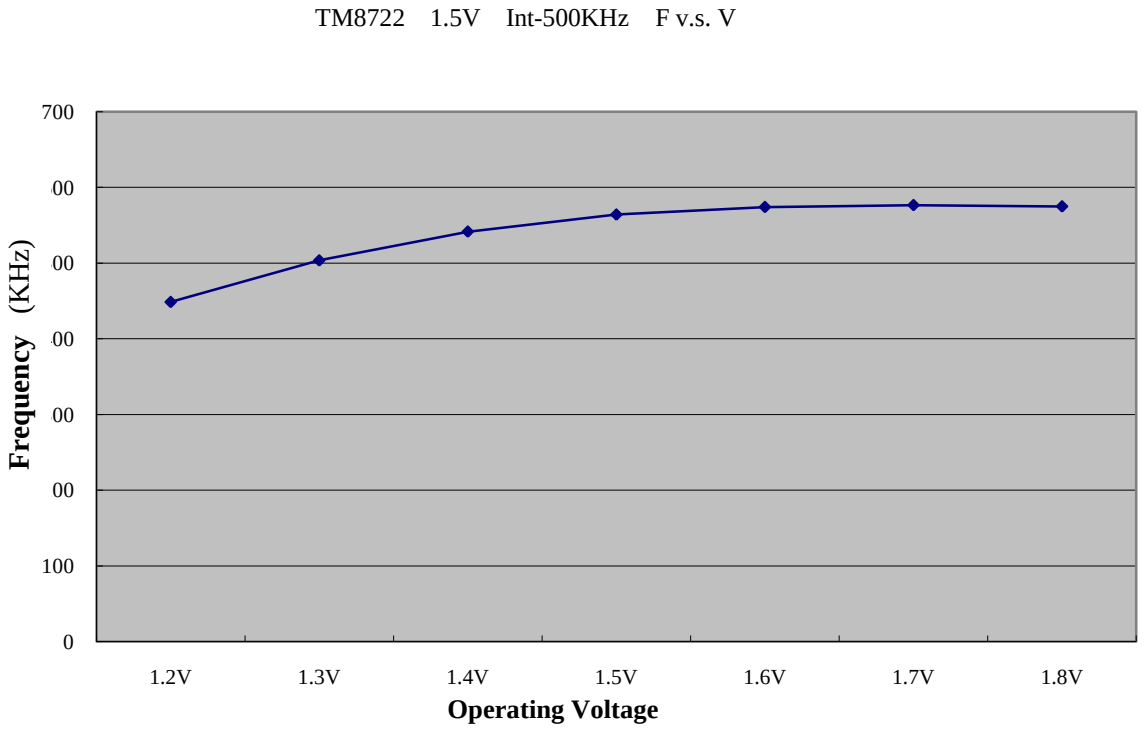
33.FAST CLOCK TYPE FOR ONLY OR DUAL :
☑ A.INTERNAL RESISTOR FOR 250KHz ☐ .INTERNAL RESISTOR FOR 500KHz
☐ .EXTERNAL RESISTOR ☐ .3.58MHz CERAMIC RESONATOR



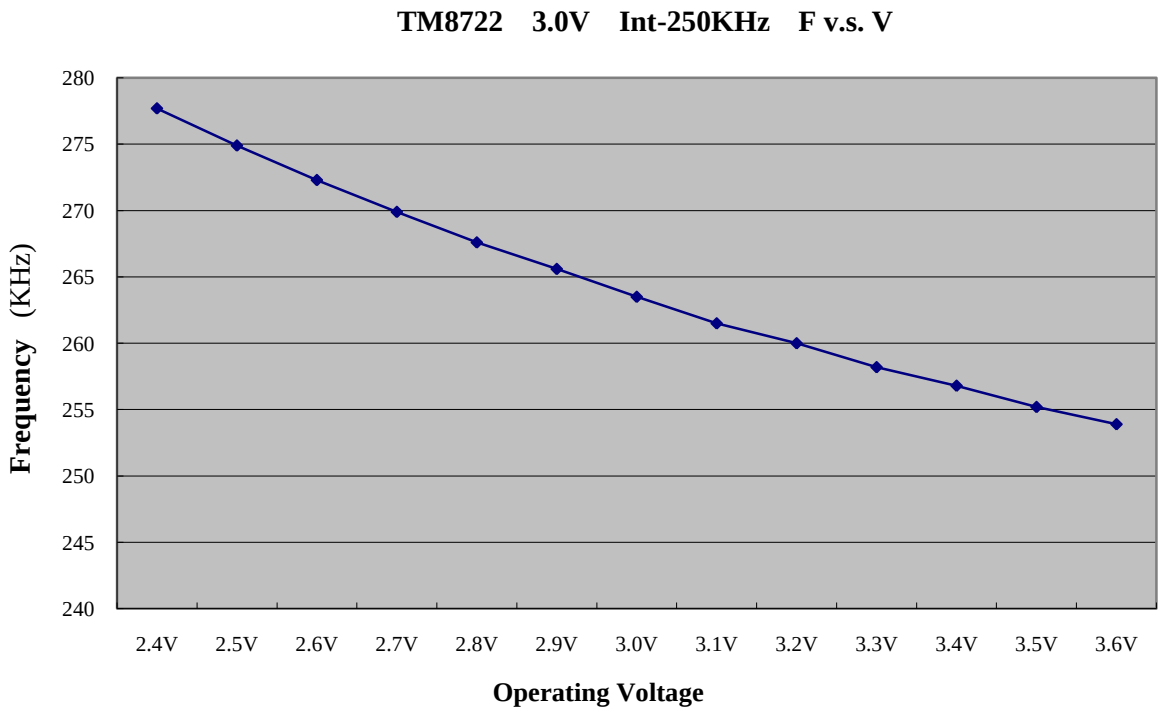
■ When using the internal RC oscillation in 1.5V operating application, the operating voltage and frequency relation chart (**choose 250KHz**) will be:



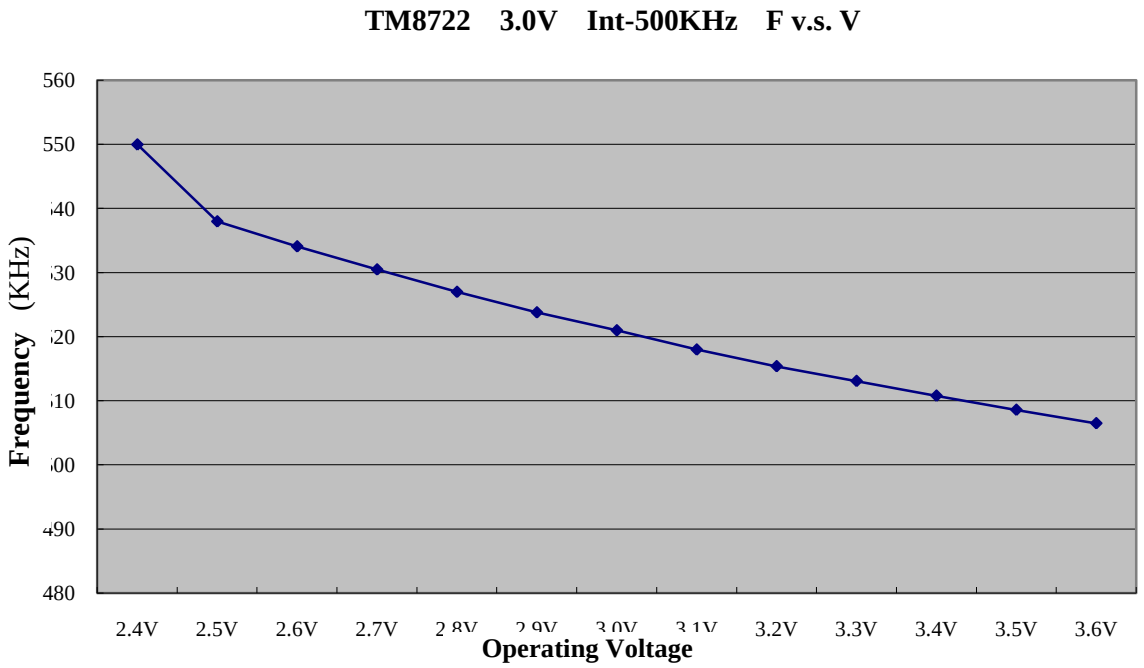
■ When using the internal RC oscillation in 1.5V operating application, the operating voltage and frequency relation chart (**choose 500KHz**) will be:



■ When using the internal RC oscillation in 3V operating application, the operating voltage and frequency relation chart (**choose 250KHz**) will be:



■ When using the internal RC oscillation in 3V operating application, the operating voltage and frequency relation chart (**choose 500KHz**) will be:



II. Slow only mode

- There are two clock sources to choose from in the Slow only mode: one is to use the external 32.768KHZ Quartz crystal oscillator and the other is to use external RC to couple the XIN and XOUT to generate oscillation.
- In the Slow only mode, the clock signals generated by the slow clock oscillator can be used as the basic frequency for the circuitries such as the system clock generator, pre-Divider, timer, I/O chattering prevention and LCD driver module etc.
- When the mask option is set to Fast only mode, the slow clock oscillation module will be disabled permanently.
- When the MCU execute the STOP instruction and enters the STOP mode, the slow clock oscillation module will be disabled and stop oscillating.

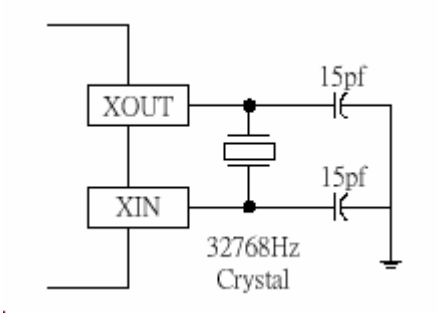
1. Using the external 32.768KHZ Quartz crystal oscillator

■ When the BCF flag is set to 1, the crystal oscillator will produce higher driving capability to sustain the oscillation, but it will consume higher current.

For TM8726 example & Mask Option Program:

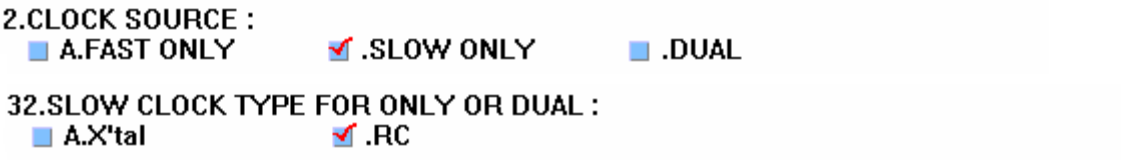
2.CLOCK SOURCE :
☐ A.FAST ONLY ☑ .SLOW ONLY ☐ .DUAL

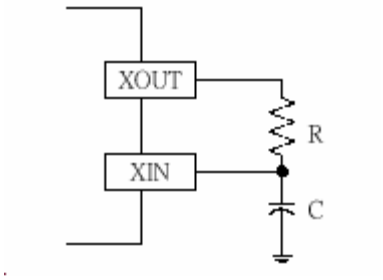
32.SLOW CLOCK TYPE FOR ONLY OR DUAL :
☑ A.X'tal ☐ .RC



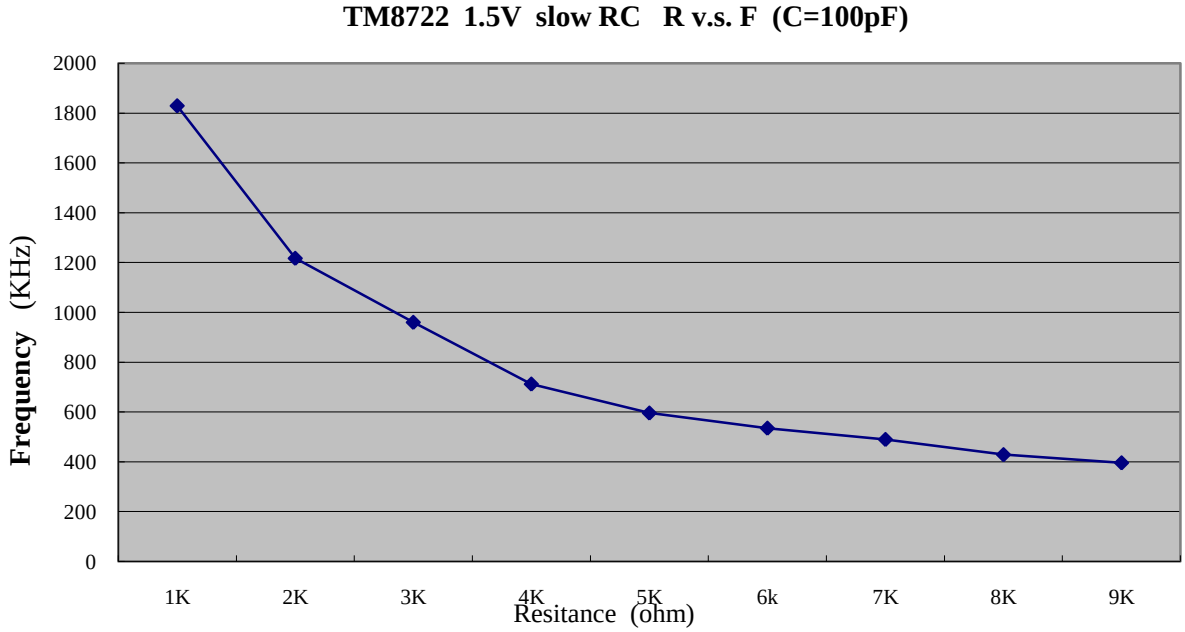
2. Using external RC to couple the XIN and the XOUT to generate oscillation

For TM8726 example & Mask Option Program:

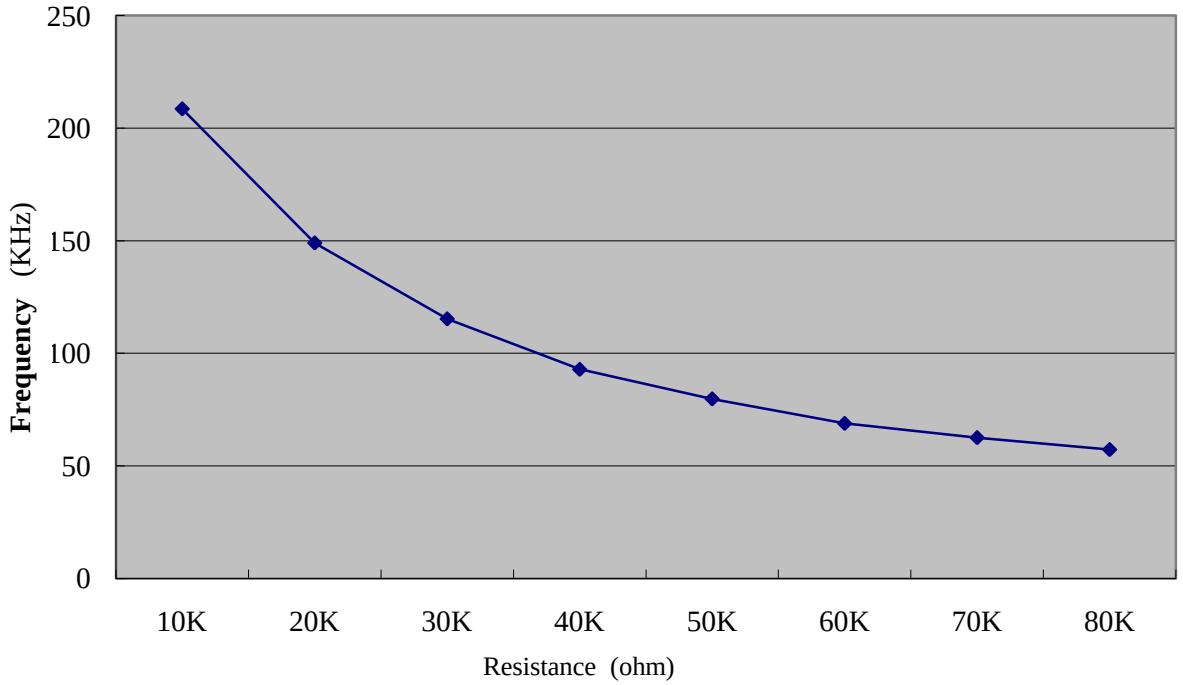




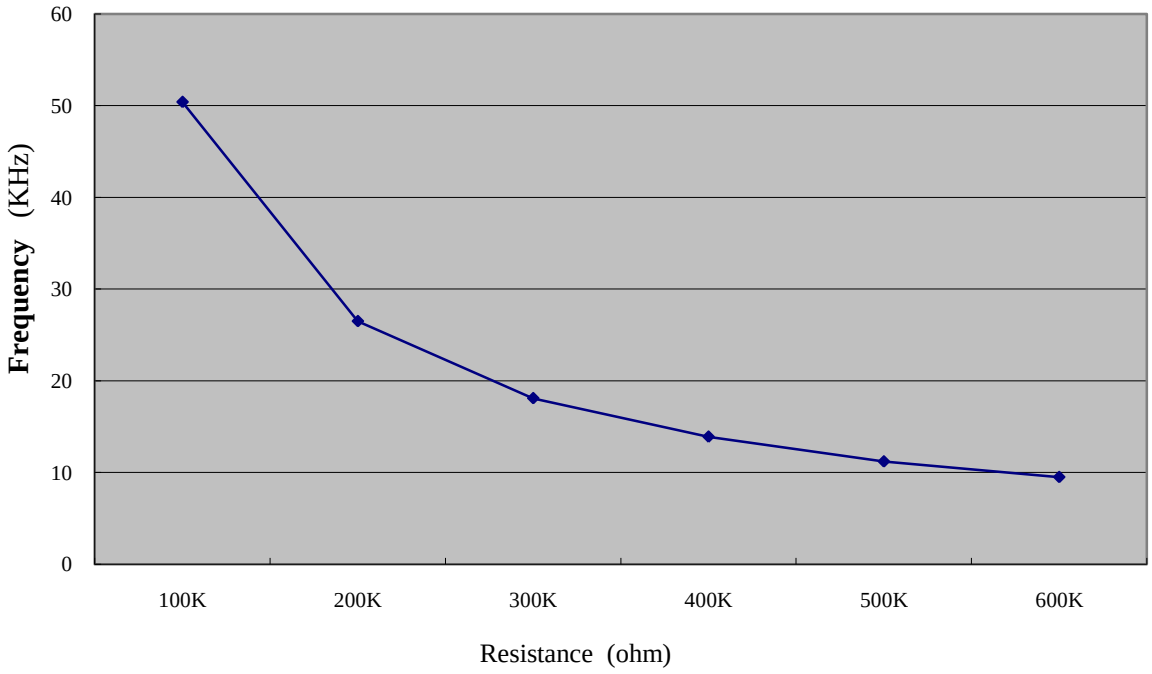
- When using the external RC oscillation in 1.5V operating application, the resistance and frequency relation chart (capacitance 100pF) will be:



TM8722 1.5V slow RC R v.s. F (C=100pF)

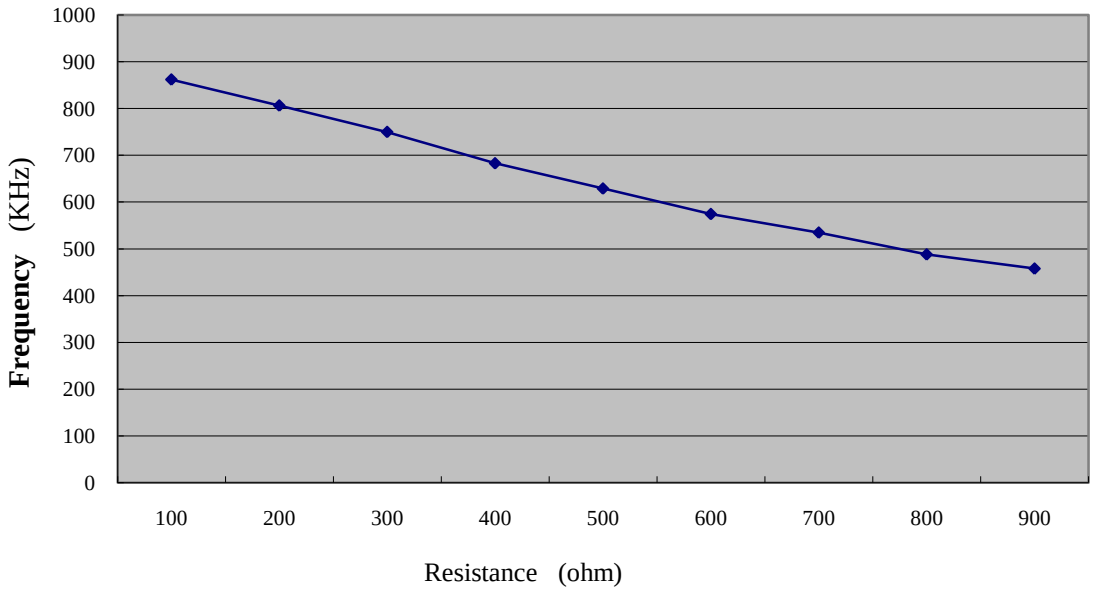


TM8722 1.5V slow RC R v.s. F (C=100pF)

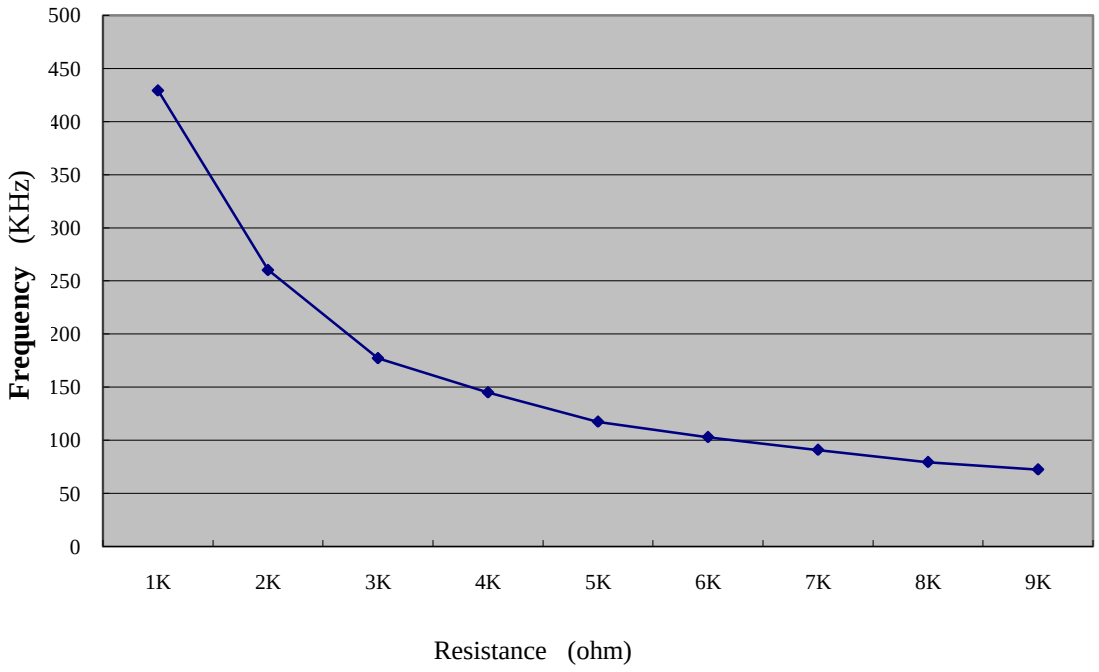


■ When using the external RC oscillation in 1.5V operating application, the resistance and frequency relation chart (capacitance 1000pF) will be:

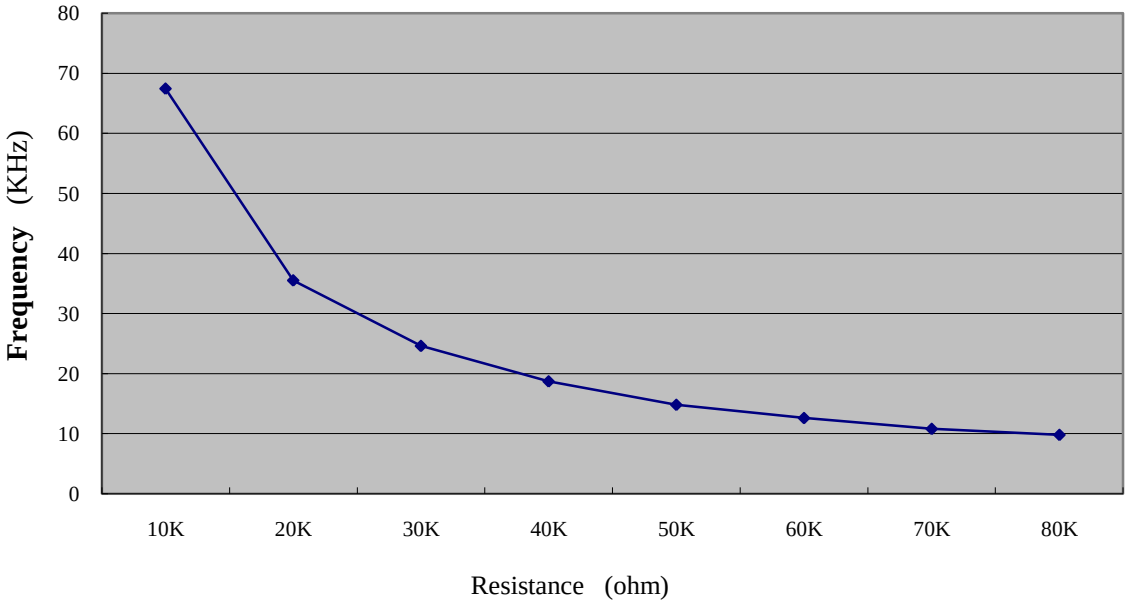
TM8722 1.5V slow RC R v.s. F (C=1000pF)



TM8722 1.5V slow RC R v.s. F (C=1000pF)

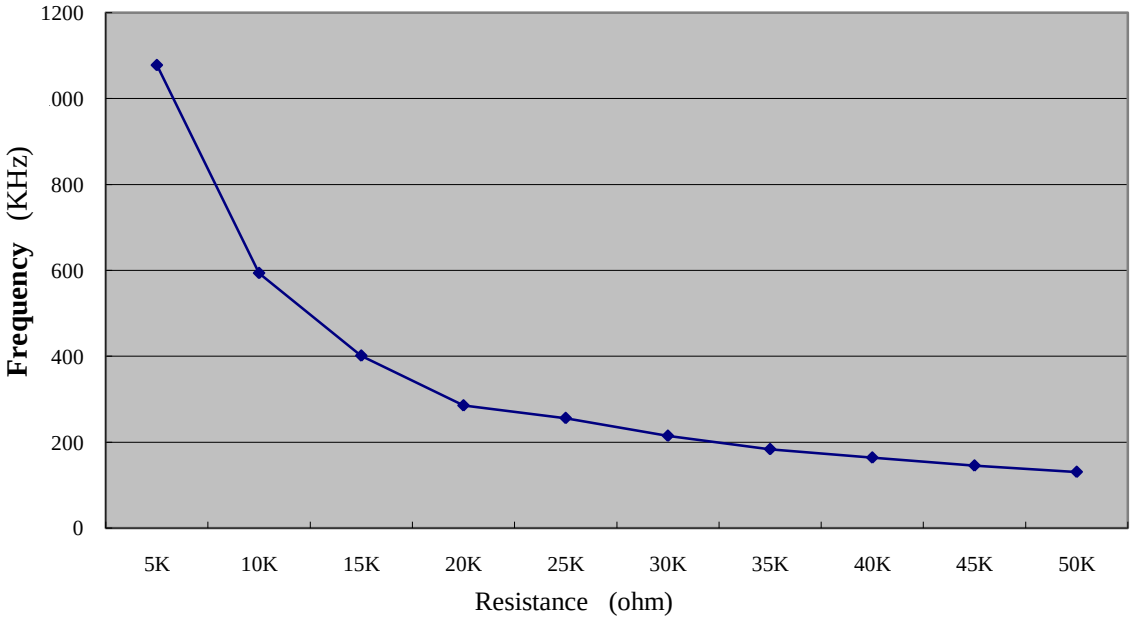


TM8722 1.5V slow RC R v.s. F (C=1000pF)

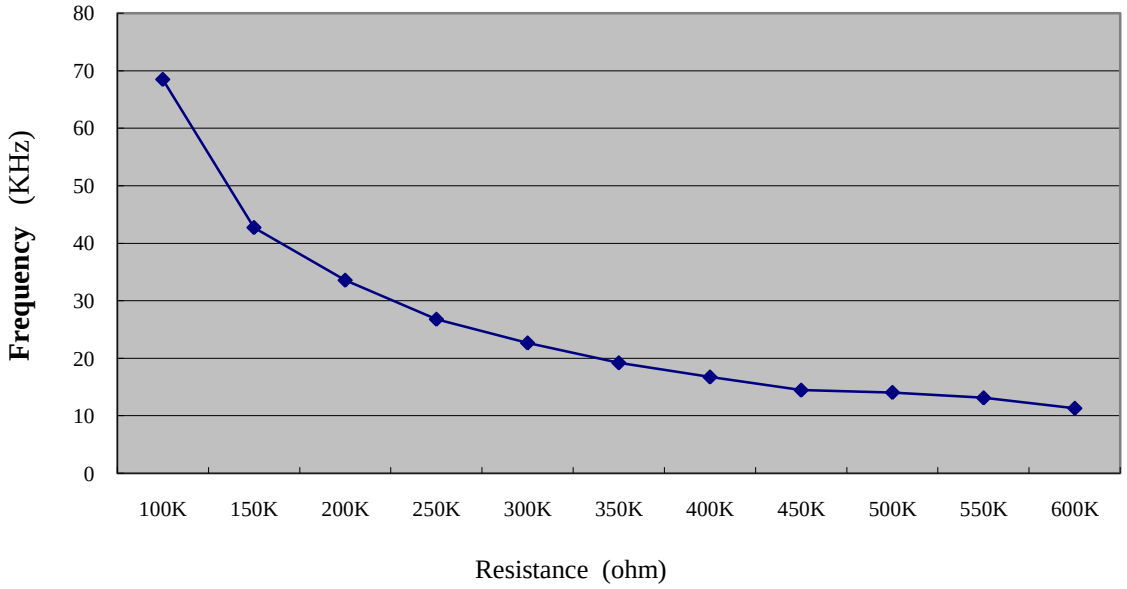


■ When using the external RC oscillation in 3V operating application, the resistance and frequency relation chart (capacitance100pF) will be:

TM8722 3.0V slow RC R v.s. F (C=100pF)

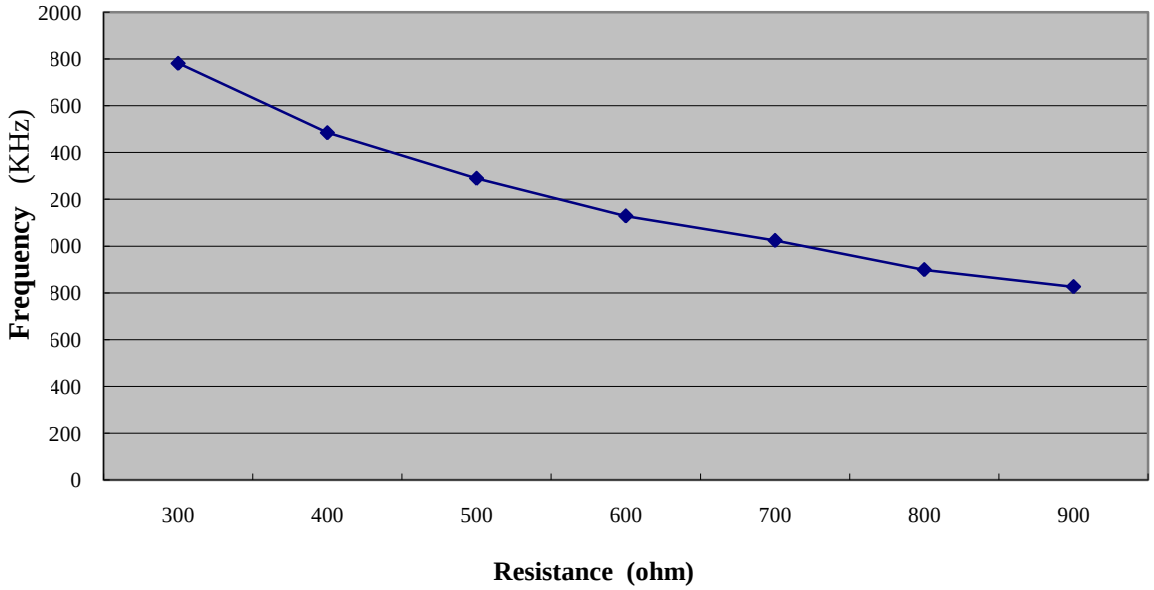


TM8722 3.0V slow RC R v.s. F (C=100pF)

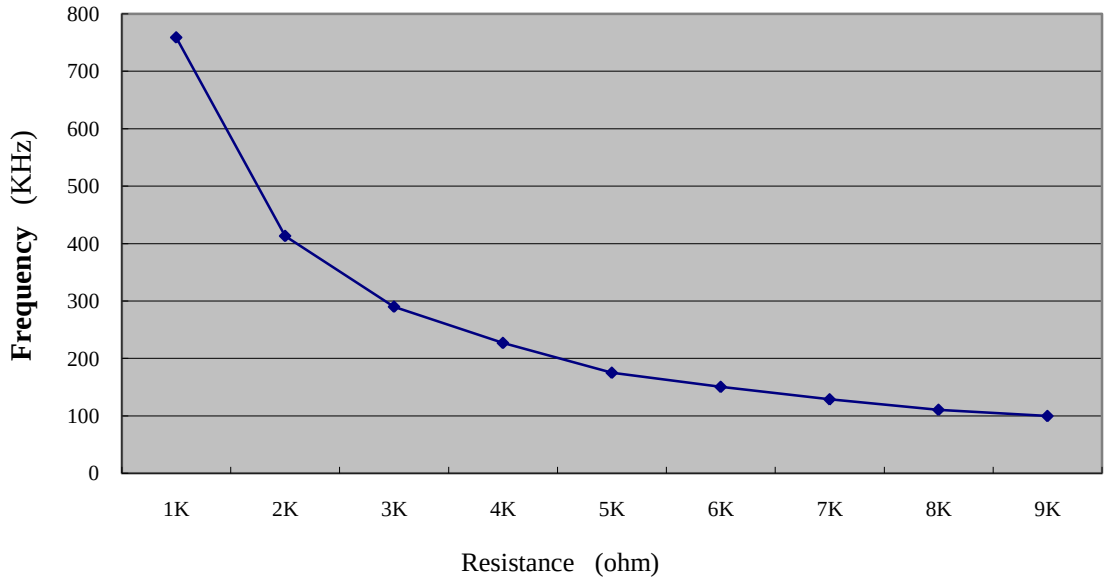


■ When using the external RC oscillation in 3V operating application, the resistance and frequency relation chart (capacitance 1000pF) will be:

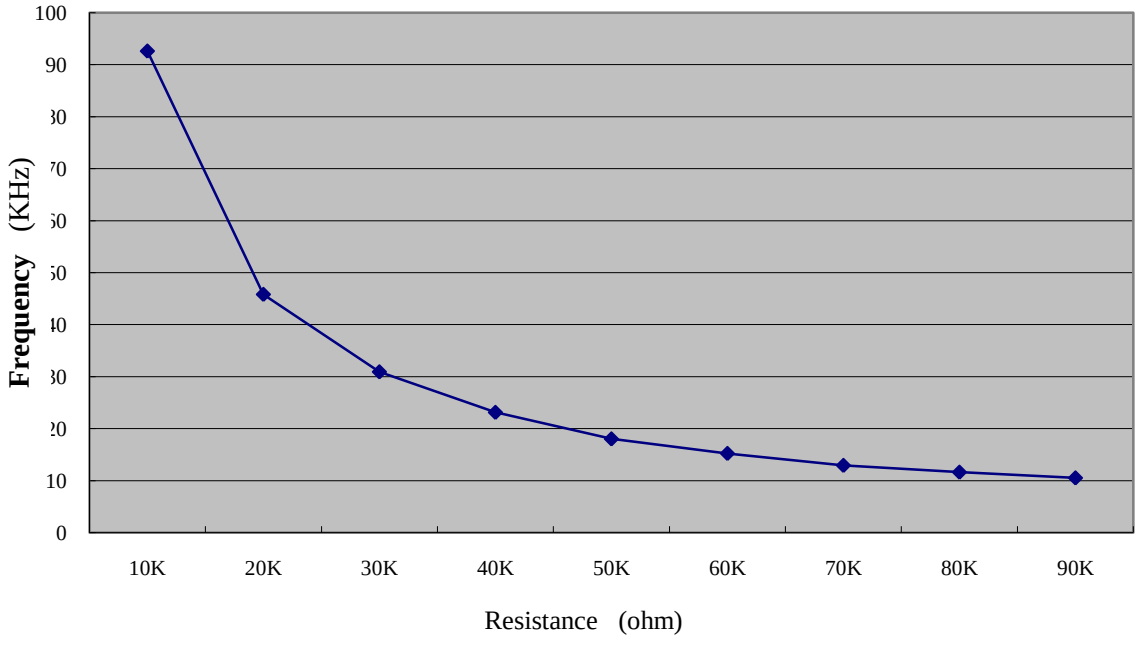
TM8722 3.0V slow RC R v.s. F (C=1000pF)



TM8722 3.0V slow RC R v.s. F (C=1000pF)



TM8722 3.0V slow RC R v.s. F (C=1000pF)



III. Dual Clock mode

The TM87 series MCU can have the fast clock and the slow clock functions simultaneously through the use of mask option. The user can use the (Fast, slow) instructions to switch between the clock sources of the system clock generator so that the system can function under both the fast clock mode and the slow clock mode.

When the HALT instruction is executed, the system clock (BCLK) will switch to the slow clock automatically and use the clock signals generated by the slow clock oscillator as the basic frequency for the circuitries such as LCD driver module etc.

The graphs and tables provided in this AP-note are a statistical summary based on a limited number of samples and are provided for informational purposes only.

The performance characteristics listed herein are not tested or guaranteed.In some graphs or tables, the data presented may be outside the specified operating range (e.g., outside specified operating frequency range) and therefore outside the warranted range.

5. AP-SZ077_01EV10

TITLE

How to make bonding option for TM87/89 series MCU?

APPLICATION NOTE

I. Introduction:

When making bonding option, the simplest method is to connect the input pin to High (1) or Low (0) directly. In doing so, there are two states for each input pin and if there are enough input pins, all the bonding options can be completed in this method. In most cases, there are not enough input pins. However, there might be leftover output pins (for example, the COMMON pin) due to the fact that the TM87/89 series share the input pins with SEGMENT or other special pins. Therefore, the output pins need to match up with the input pins. There are three states for each input pin: High, Low, and Connection. The actual number of Connection will be determined by the number of output pins.

II. Use I/O for bonding option

The TM87/89 series MCUs excel in low power consumption and the majority of the developed products are also low power consumption. Therefore, the biggest challenge for making the bonding option is not to increase the power consumption of the product. The specifics of how to avoid such mistakes will be discussed as follows:

The input pins of the TM87/89 series MCUs all come with one pull down resistor. This resistor can be disabled in program. When using the input pins without the pull down resistor for bonding option, the program will be simpler. There is no need to read and restore its state after power on reset; it is fine to read when needed. Please pay attention to the following point when using this method: the input pin must

be connected to the High or Low during manufacturing. If it is matched up with an output pin, the output pin must also have definitive High or Low state (High impedance state can not be allowed.) The main purpose of doing so is to be able to read the correct states and, at the same time, to avoid the large current generated when the input pin is in Floating state.

The disadvantage for making the bonding option as mentioned above is that customers will need to specify all the bonding out for the input pins. Here is another way to allow customers to avoid such kind of trouble: that is, to use input pins with pull down resistor. The advantage of this method is that those input pins in Low state will not need to be bonded out. The disadvantage is that there are some processes needed to avoid unnecessary current in the bonding point.

The specifics of the program is as follows(use PortA as an example)：

```
。。。
SPA     10000B    ；Set PortA to be the input pin with pull down resistor
IPA     OPTION    ；Read the bonding option and store in variable OPTION
OPA     OPTION    ；Send the content of variable OPTION to output buffer
SPA     1111B     ；Set PortA to be output pin
。。。
```

The function of Instructions 1 and 2 is to read and store the bonding option. When the program needs to use the bonding option, it only needs to access the variable OPTION. Therefore, the above program needs to be executed only once after power on reset. The function of Instruction 3 and 4 is to equalize the output state of all the pins in PortA to the bond option state; that is, all the bond options connected to VCC will output high state and all the bond options connected to GND or floating will output low state to make sure that there is no current consumption in each bond option. Note: The sequence of Instructions 3 and 4 can not be changed.

III. Use the INT pin for bonding option

In many cases, all the input pins are used for other functions. You can check the external interrupt pin INT in these cases. If this pin is not used for other function, it can be used as an input pin for the bonding option.

The specific of the program is as follows：

Assume the INT pin is set to internal pull-down, triggered by rising edge in the Mask Option and assume the definition of SEG1 output pin is “1, 0, 1f, a” in the CFG document(the second item “0” sets SEG1 to be COMS output pin).

```
1.      。。。
2.  plc     1100B       ；Clear the Halt release Request flag of 0.5 second and the external
                          interrupt pin INT
3.  she     1100B       ；Enable the Halt release from 0.5 second and the external interrupt pin
                          INT
4.  lds     Temp,00h    ；
5.  lcp     01fh,Temp   ；SEG1 output low state
6.  Nop                 ；Delay
7.  lds     Temp,0fh
```

```
8.    lcp    01fh,Temp    ；SEG1 Output high state
9.    Nop                 ；Delay
10.   lds    Temp,00h
11.   lcp    01fh,Temp    ；SEG1 Output low state
12.   halt
13    msc    Option       ；Store SCF4 (INT pin outputs HaltRelease signal) to variable Option
14.   lds    Temp,0001B   ；
15.   And    Optio        ；Set bit0 of variable Option to be the signal of the bonding option
；Option.1=1 means there is a connection between SEG1 and INT pin
                          ；Option.1=0 means there is no connection between SEG1 and INT  pin
16.   plc    1100B        ；Clear the Halt release Request flag of 0.5 second and the external
                            interrupt pin INT
```

The set up of Instructions 2 and 3 allows 0.5 second and INT pin Halt release. When there are no input signals on the INT pin, the 0.5 second signal will free the MCU from the Halt mode. The function of instruction 4 and 5 is to output low state on SEG1. The reason to do it this way is due to the fact that if customer sets LCD display in reset cycle to be ON in Mask Option, the outputting SEG1 will be set to high state after RESET. Therefore, it needs to be set to low state first (to produce rising edge). If customer sets it to OFF, these two instructions can be omitted. Instruction 7 and 8 sets the SEG1 to output high state to generate a rising edge on the INT pin(INT pin is set as triggered by rising edge in Mask Option). Instruction 10 and 11 sets SEG1 to output low state so that there is no voltage difference between SEG1 and INT (INT is set to with pull-down resistor in Mask Option). Instructions 12, 13, and 15 read and store bonding option.

The above mentioned method requires an output pin and the INT pin to form a bonding option.

If all the output pins are used, you can consider reuse. Buzzer is frequently used. The specifics are basically the same as mentioned above.

The specific of the program is as follows：

```
plc     1100B        ；Clear the Halt release Request flag of 0.5 second and the external interrupt pin
she     1100B        ；Allow 0.5 second and the Halt release in the external interrupt pin INT
alm     011000000B   ；BZ/BZB output square wave to INT pint
call    delay        ；Delay
alm     00h          ；BZ/BZB output low state
halt
msc     Option       ；Store SCF4(Halt release signal from INT pin) to variable Option
lds     Temp,0001B   ；

And     *Option      ；Set bit0 of variable Option to be the signal of the bonding option
                     ；Option.1=1 means there is a connection between BZ/BZB and INT pin
                     ；Option.1=0 means there is no connection between BZ/BZB and INT pin
Plc     1100B        ；Clear the Halt release Request flag of 0.5 second and the external interrupt pin
```

The delay time in the above program should be adjusted based on the requirement of application. The basic principle is to ensure the generation of rising edges. The main considering factor is the frequency of the system clock.

IV. Miscellaneous notes

Also, if the SEGMENT line is time shared and re-used to do keyscan, it can be re-used to make bonding option. The details will be divulged in the AP Note.

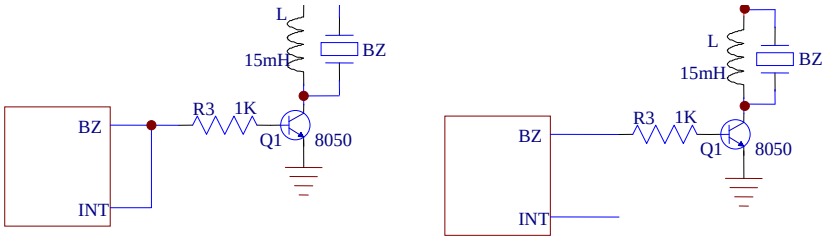
V. Application Circuit

Fig.1 Use the input pins without the pull down resistor for bonding option

Fig. 2 Use the input pins with the pull down resistor for bonding option

Fig. 3 Use external pin INT and output pin with SEG wire for bonding option

Fig. 4 Use external pin INT and pin BZ for bonding option



tenx	Title: SZ077		
	size B	number:	Revision:
	Data: Aug. 30, 2002		sheet of
	File: sz077.ddb		Drawnby:

6. AP-SZ081_01EV10

TITLE

TM87xx series KEYSCAN with shared segment

APPLICATION NOTE

I. Summary

This program illustrates the scanning method of key matrix shared with segment in TM87 series chips. 4*4 matrix is used as an example.

II. Description

In general, in TM87 series chips, there are 16 SEGEMENT that can be used to do keyscan output （S1~S16）and 4 input pins（KI1-KI4）. By selecting KEYSCAN options in MASKOPTION, they can be used to generate periodic scanning signals automatically through the hardware. The key movements can be judged by using these scanning signals. The scanning frequency formula is as follows:

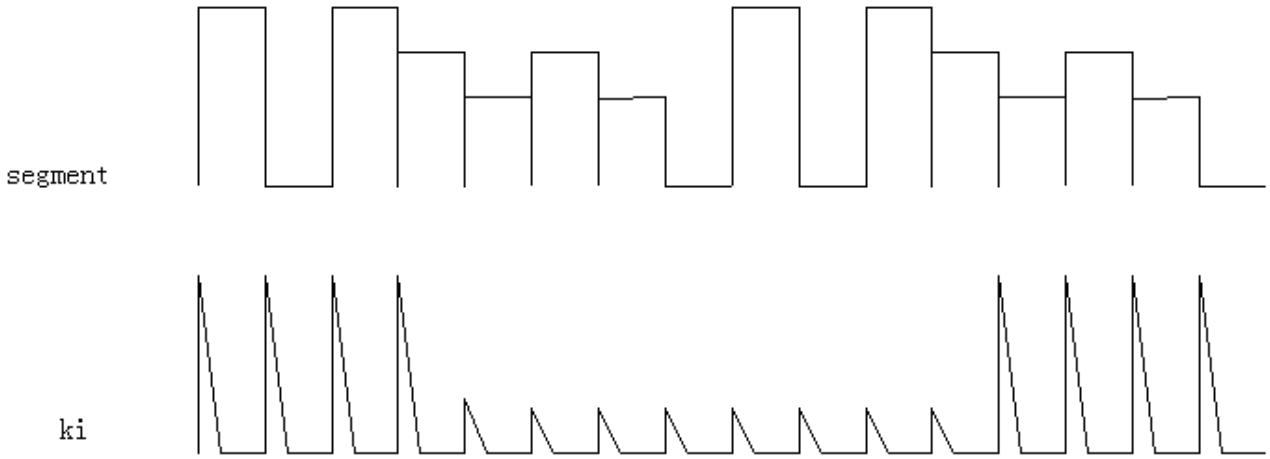
Key matrix scanning frequency (hz) = (LCD frame frequency)*(LCD duty cycle)*2

TM87 series chips offer two methods to generate keyscan Halt release signal. One method is to generate the HALT release signal after the key is pressed; the other is to generate HALT release signals on every scanning cycle. The Key matrix scanning is usually completed through a combination of these two methods. These two HALT release signals are controlled through the use of the SPK instruction. The details are as follows:

Spk function:（8705）
X6 = “ 0 “, When HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after the key pressed on the key matrix and then set SCF7 to 1.
“ 1 “, When HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after each scanning cycle no matter the key is pressed or not and then set SCF7 to 1.
X7X5X4 = 000, in this setting, each scanning cycle only check one specified column (K1 ~ K16) on key matrix. The specified column is defined by the setting of X3 ~ X0.
X3 ~ X0 = 0000, active K1 column
X3 ~ X0 = 0001, active K2 column
X3 ~ X0 = 1110, active K15 column
X3 ~ X0 = 1111, active K16 column
X7X5X4 = 001, in this setting, all of the matrix columns (K1 ~ K16) will be checked simultaneously in each scanning cycle. X3 ~ X0 don’t care.
X7X5X4 = 010, in this setting, the key matrix scanning function will be disable.
X3 ~ X0 don’t care.
X7X5X4 = 10X, in this setting, each scanning cycle check 8 specified columns on key matrix. The specified column is defined by the setting of X3.
X3 = 0, active K1 ~ K8 columns simultaneously
X3 = 1, active K9 ~ K16 columns simultaneously
X2 ~ X0 don’t care.
X7X5X4 = 110, in this setting, each scanning cycle check four specified columns on key matrix. The specified columns are defined by the setting of X3 and X2.
X3X2 = 00, active K1 ~ K4 columns simultaneously
X3X2 = 01, active K5 ~ K8 columns simultaneously
X3X2 = 10, active K9 ~ K12 columns simultaneously
X3X2 = 11, active K13 ~ K16 columns simultaneously

X1, X0 don’t care.
X7X5X4 = 111, in this setting, each scanning cycle check two specified columns on key matrix. The specified columns are defined by the setting of X3, X2 and X1.
X3X2X1 = 000, active K1 ~ K2 columns simultaneously
X3X2X1 = 001, active K3 ~ K4 columns simultaneously
X3X2X1 = 110, active K13 ~ K14 columns simultaneously
X3X2X1 = 111, active K15 ~ K16 columns simultaneously
X0 don’t care.

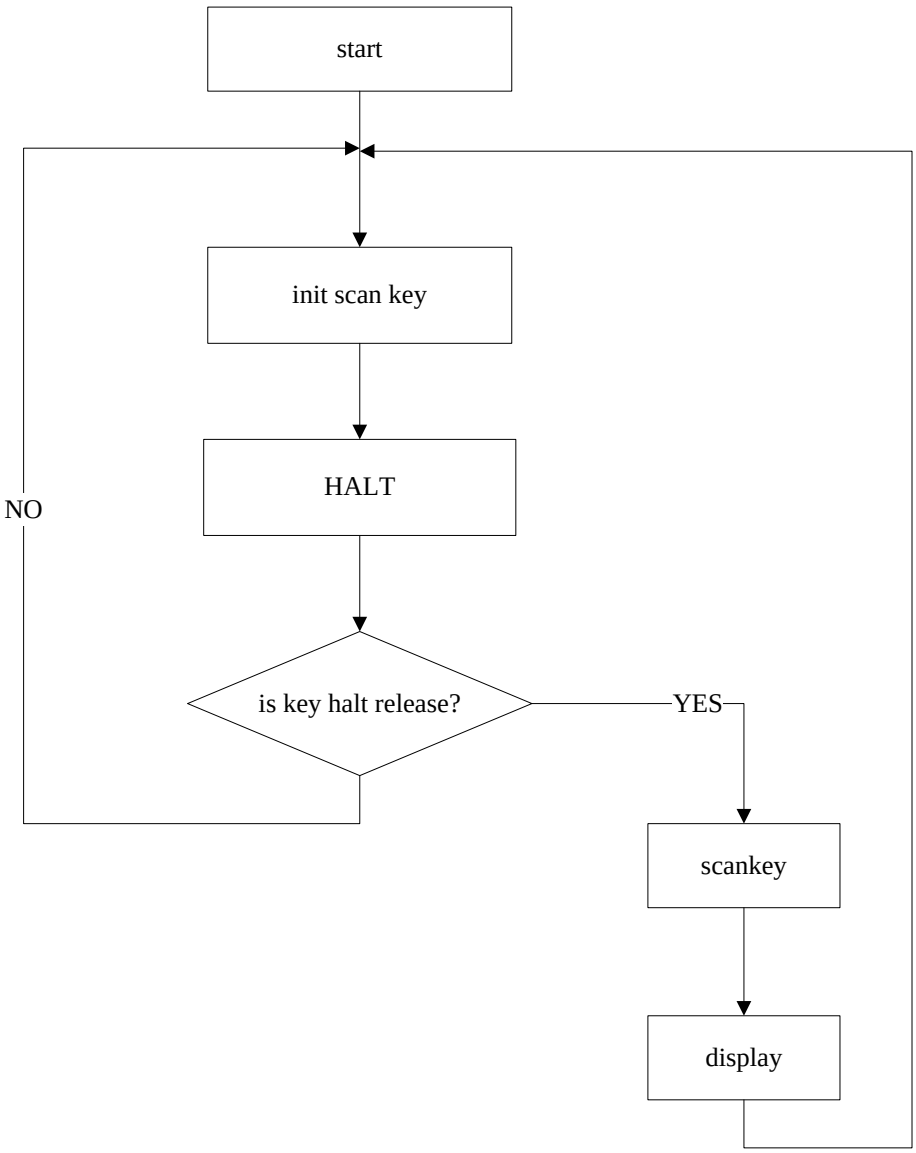
III. TM87 series chips’ hardware scanning signals

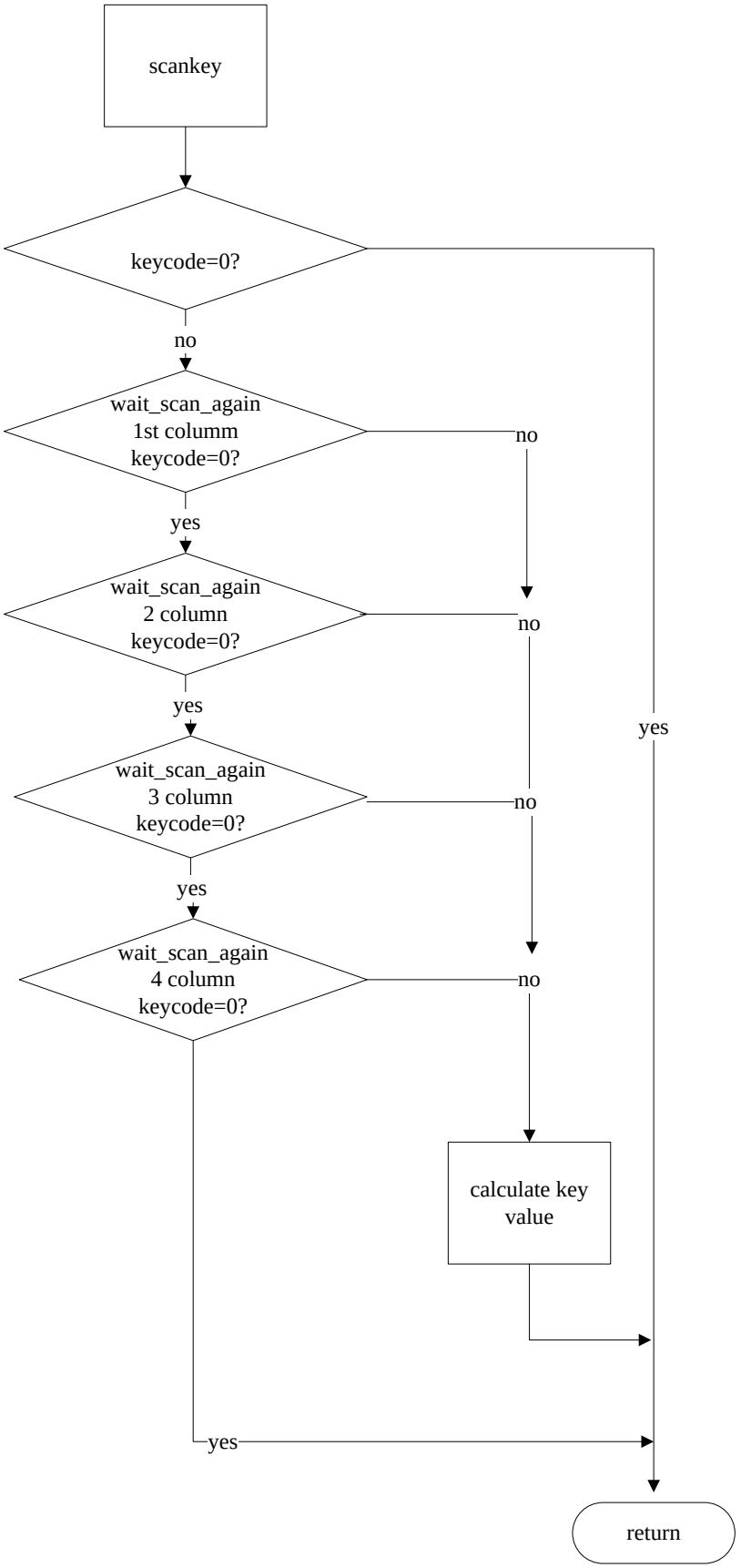


KI: Complete pulse signal when key is not pressed and the pulse signal will be pulled down when key is pressed.

S1-S16：The end of each LCD signal will output a short low level.

IV. The Entire KEYSCAN Process




scankey
keycode=0?
no
wait_scan_again
1st columm
keycode=0?
no
yes
wait_scan_again
2 column
keycode=0?
no
yes
wait_scan_again
3 column
keycode=0?
no
yes
wait_scan_again
4 column
keycode=0?
no
yes
calculate key
value
yes
return

V. The KEYSCAN program listing

```
.chip TM8712
.head
pstbnum0        equ 0
pstbnum1        equ 1
.endh
.data
org             00h
haltflag        dn 1
column          dn 1
across          dn 1
keycode         dn 2
org             70h
worktmp         dn 8
.endd

.code
org             00h

mainhalt:
spc             0fh             ;setting portc output and disable pull-down
spk             10h             ;halt release when key press
                                ;scanning all column simulltaneously for each cycle
she             20h             ;set HEF5
plc             20h             ;clear HRF5
halt
mcx haltflag                    ;Check SCF8(ski)
jb0 key_release
jmp mainhalt
key_release:
ipc worktmp                     ;check KI input
jz mainhalt                     ;key release
call scankey
call display
jmp mainhalt

scankey:                        ;check 1st column
spk             40h             ;bit6=1 When HEF5 is set to 1
                                ;the HALT released request(HRF5) will be set  to 1 after each scanning
                                  cycle
                                ;no matter the key depressed or  not and  then set SCF7 to 1.
lds across,0                    ;save column number
plc             20h             ;clear HRF5 to avoid the false halt released
call wait_scan_again            ;waiting for the nested key matrix scanning cycle
ipc column                      ;1st column key vlaue
jnz scanend

spk             41h             ;check 2 column
lds across,1
```

```
plc             20h
call wait_scan_again
ipc column
jnz scanend

spk             42h             ;check 3 column
lds across,2
plc             20h
call wait_scan_again
ipc column
jnz scanend

spk             43h             ;check 4 column
lds across,3
plc             20h
call wait_scan_again
ipc column
jnz scanend
rts

scanend:
lds worktmp+1, 0
lds worktmp, 0
calkeyl:
inc* worktmp
lda column
jb0 calkeyh
sr0 column
jmp calkeyl

calkeyh:
lda across
jz keyseccess; _rts
dec* across
addi* worktmp, 4
daa* worktmp
jnc calkeyh
inc* worktmp+1
jmp calkeyh

display:
mrw worktmp, keycode
mrw worktmp+1, keycode+1
lct pstbnum0, worktmp+1
lct pstbnum1, worktmp
rts
keyseccess:
mwr keycode, worktmp
mwr keycode+1, worktmp+1
rts
```

```
wait_scan_again:
halt
plc                 20h
rts

_rts: rts
.endc

.table
.endt
```

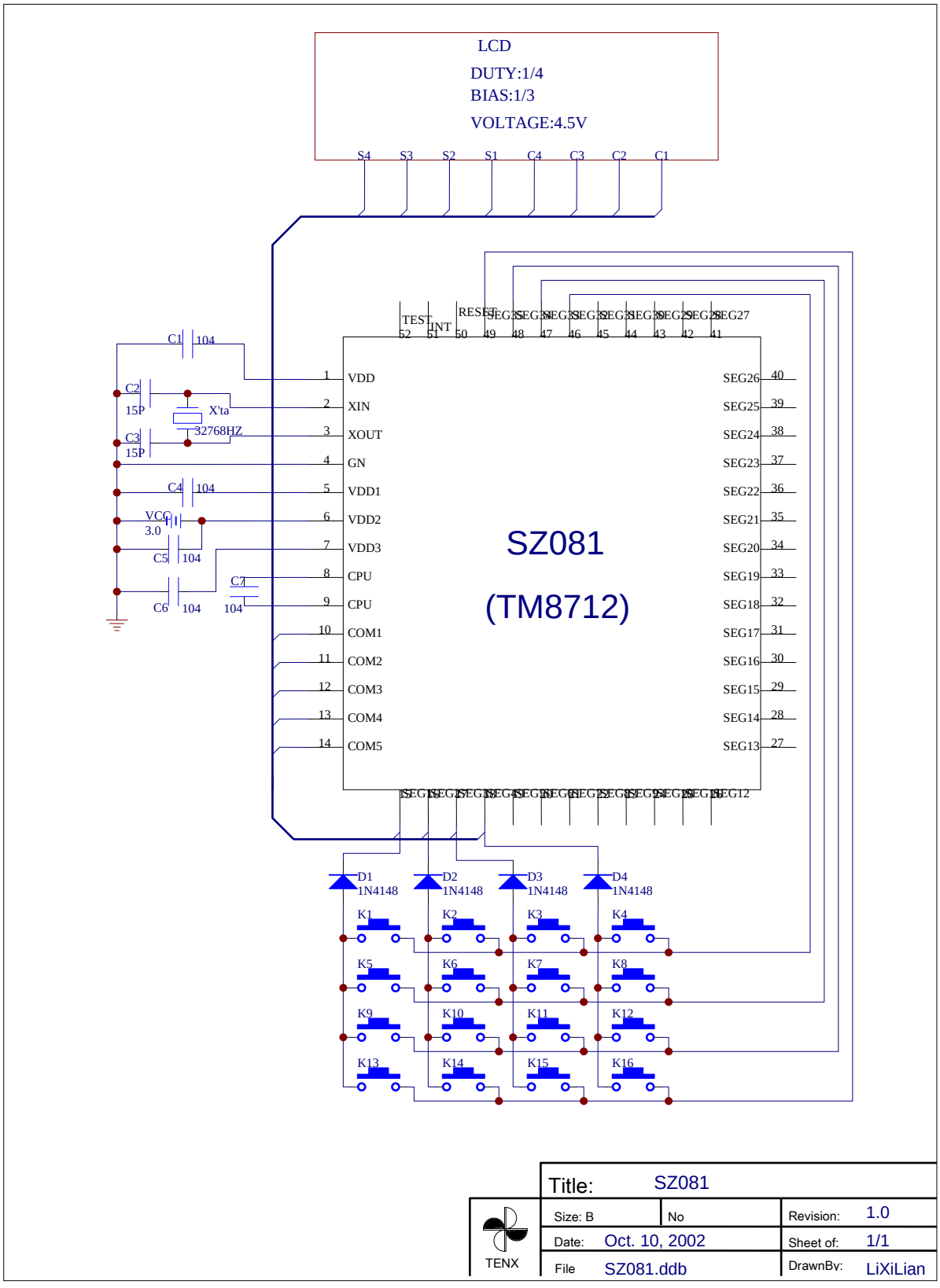
VI. Miscellaneous notes

1. The KEYSCAN function mentioned above is accomplished by sharing the SEGMENT lines. When multiple keys are pressed simultaneously, two or more of the SEGMENT lines will usually be short-circuited and degrade the effect of the LCD display. A diode should be added to every SEGMENT wire to prevent such scenario to happen.

2. In some situations, the KEYSCAN function is used to do jumper selection. But please be careful for the following: when TM87 series chip enters the STOP mode, there is a pull up resistor, about 500K, attached to the KI that is used for KEYSCAN and S1-16 will always output low level. Therefore, if the jumper selection connects them together, there will be a VDD/500K current passing through the wires.

3. LCD is a capacitive load. Its capacitance increases as the area gets larger. When the area increases up to certain degree, its capacitance will destabilize the KEYSCAN. The TM8797 DEMO BOARD will reflect the situation faithfully. Therefore, please use it to test.

VII. Application Circuit Diagram



7. AP-SZ082_01EV11

TITLE

TM87xx series Halt Release and Stop Release

APPLICATION NOTE

The methods to wake-up TM87 series MCU from the halt mode are as below:

1. IOC signal changes
2. INT signal changes
3. Timer1 Underflows
4. Tmer2 Underflows
5. Pre-divider overflows (0.5s)
6. Key matrix scanning wake-up
7. RFC Counter overflows

In addition, the MCU can also be released from halt mode when interrupt occurs during the halt mode. But it will back to the halt mode after returned from ISR, which means it does not execute the instruction next to the HALT instruction.

The following are the examples:

```
;*************************************
;*    timer1 halt release    *
;*************************************
initiation:                         ;initialization
plc             100101111B          ;clear all the halt release flags
tmsx            01111111B           ;64*8/32.768=15.625ms
sf              80h                 ;enable timer1 re_load
she             02h                 ;enable timer1 halt release
lds             halt_flag,00h       ;clear halt_flag
;*************************************
main:                               ;main program
    lda         halt_flag           ;if it is timer1 halt release
    jb1         timer1_Process      ;go to timer1_Process

    halt

    msc         halt_flag           ;Check timer1 overflow that causes the halt
                                    ;mode to be released.
    jmp         main
;*************************************
timer1_Process:                     ;timer1 halt release program processing
    plc         02h                 ;Clear timer1 Halt release request flag HRF1
    andi*       halt_flag,1101b     ;clear timer1 Halt release flag
      . . . . . .
```

```
    jmp         main
;*************************************************************************************
*******************

;************************************
;*    timer2 Halt Release    *
;************************************
initiation:                         ;initialization
plc     00101111B                   ;clear all the halt release flags
she     10h                         ;enable timer2 halt release
tm2x    001111111b                  ;64*8/32.768=15.625ms
sf2     01h                         ;enable timer2 re_load
lds     halt_flag,00h
;************************************
main:                               ;main program
    lda         halt_flag           ;if timer2 halt release
    jb1         timer2_Process      ;then go to timer2_Process

    halt

    mcx         halt_flag           ;Check timer2 overflow  that causes the halt
                                     ;mode to be released
    jmp         main
;************************************
timer2_Process:                     ;timer2 halt release program processing
    plc         10h                 ;Clear timer2 Halt release request flag HRF4
    andi*       halt_flag,1101b     ;clear timer2 Halt release flag
      ......
    jmp         main
;*************************************************************************************
*******************

;************************************
;*     ioc Halt Release      *
;************************************
initiation:                         ;initialization
plc             100101111B          ;clear all the halt release flags
sp              10000               ;enable ioc port output and pull-low
sca             10h                 ;enable ioc halt release
lds             halt_flag,00h
;************************************
main:                               ;main program
    lda         halt_flag           ;if ioc halt release
    jb1         portc               ;then go portc

    halt

    msb         halt_flag           ;Check ioc that causes the halt mode to be released
    jmp         main
```

```
;*************************************
portc:                          ;portc  program processing
      plc         01h           ;Clear ioc Halt release request flag HRF0
      andi*       halt_flag,1101b ;clear ioc Halt release flag
      sca         10h           ;enable ioc halt release
            ......
      jmp         main
;*****************************************************************************************
*******************


;*************************************
;       int halt release        *
;*************************************
initiation:                     ;initialization
plc               100101111B    ;clear all the halt release flags
she               04h                   ;enable int halt release
lds               halt_flag,00h
;*************************************
main:                           ;main program
      lda         halt_flag     ;if int halt release
      jb0         int_Process   ;then go int_Process

      halt

      msc         halt_flag     ;Check int that causes the halt mode to be released

      jmp         main
;*************************************
int_Process:                    ;int halt release program processing
      plc         04h           ;Clear int Halt release request flag HRF2
      andi*       halt_flag,1110b ;clear int Halt release flag
         ......
      jmp         main
;*****************************************************************************************
*******************


;*************************************************
;*predivider overflow（0.5s） halt release*
;*************************************************
initiation:                     ;initialization
plc       100101111B            ;clear all the halt release flags
she               08h           ;enable predivider overflow halt release
lds       halt_flag,00h
;*************************************
main:                           ;main program
      lda         halt_flag     ;if predivider overflow halt release
      jb3         halt0.5s      ;then go halt0.5s

      halt
```

```
    ms halt_flag                ;Check 0.5s that causes the halt mode to be released
    jmp         main
;************************************
halt0.5s:                       ;0.5s program processing
    plc         08h             ;Clear predivider overflow Halt release
                                ;request flag HRF1
    andi*   halt_flag,0111b     ;clear predivider overflow Halt release flag
      ......
    jmp         main
;*****************************************************************************************
*******************
```

```
;*************************************
;* key matrix scanning halt release *
;*************************************
```

TM87 series MCU shared the timing of LCD/LED waveform to scan the key matrix circuitry and these scanning output pins are SEG1~16(for easy to understand, named these pins as K1 ~ K16). The sharing time of LCD/LED waveform will not affect the display of LCD/LED panel. The input port of key matrix circuitry is composed by KI1 ~ KI4 pins (these pins are muxed with SEG32 ~ SEG35 pins and selected by mask option).

```
initiation                          ;initialization
        plc     100101111B          ;clear all the halt release flags
        spc     0f                  ;Disable all the pull-down device on internal
                                    ;IOC port
        lds     halt_flag,00h
;************************************
main:                   ;main program
        spk     10h             ;Generate HALT released request when key
                        ;depressed
                she     20h         ;Set HEF5
                plc     20h         ;Clear HRF5

                halt

                mcx     halt_flag   ;Check SCF8 (SKI)
                jb0     ski_release ;if key scan matrix halt release then go
                                    ;ski_release
                jmp     main
;************************************
ski_release:                        ;key matrix scanning program processing
    plc         20h                 ;Clear  HRF5
    andi*       halt_flag,1110b     ;clear key scan matrix  Halt release flag
      . . . . . .
    jmp         main
;*****************************************************************************************
*******************
```

The above are several typical methods of Halt Release. Using these methods combining with interrupt, the chip can be released from Halt Mode after return from ISR (interrupt service routine)

every time. (Tm87 series MCU have a special feature, that is the MCU enters Halt mode again after return from ISR when the interrupt have been accepted in halt Mode) The following example will show you how to release chip from halt mode by combining timer1 interrupt with INT halt release.

BZ is selected among the mask option seg31, IOB4 and BZ. BZ is connected with INT.

```
initiation:                             ;initialization
        plc     100101111B              ;clear all the halt release flags
        she     04h                     ;enable int halt release
        alm     01100000b               ;enable alm 4kHZ output, so the BZ
                                        ;will generate a falling or rising edge on
                                        ;the INT in order to set  HRF2(INT Halt release
                                        ;request ), and it will NOT be cleared unless
                                        ;reset the IC
        lds     temp+0,09h

delay:
        dec*    temp+0
        lds     temp+1,09h

delay1:
        dec*    temp+1
        jnz     delay1
        lda     temp+0
        jnz     delay

        alm     00                      ;disable alm
        tmSx    001111111b              ;64*8/32.768=15.625ms
        sf      80h                     ;enable timer1 re-load
        sie*    02h                     ;enable timer1 interrupt
;*************************************
main:                   ;main program
      . . . . . .
                                        ;
        halt
        she     00                      ;disable int halt release
      . . . . . .                       ;these instruction will be executed
                                        ;after Timer1 interrupt
        jmp     main
;*************************************
timer1_interrupt:                       ;interrupt subroutine
        sta     acc
        ma      cf
     . . . . . .
        she     04h                     ;enable int halt release
        sie*    02h                     ;enable timer1 interrupt
        lda     acc
        mra     cf
        rts
```

```
;*****************************************************************************************
*******************
```

The methods to be woken up from the stop mode with tm87 series MCU are as below:

1. IOC signal change
2. INT signal change
3. Key is pressed during key matrix scanning.

Be careful that the BCF flag should not be set to 0 immediately after the MCU released from the stop mode. (There should be at least a 100ms delay before setting BCF flag to 0 by executing the instruction "RF 02h" to wait for oscillation stable.) The following are the examples of waking up the MCU from Stop mode by the input source of IOC, INT, and keyscan.

```
;*************************************
;*    ioc stop release     *
;*************************************
initiation:                             ;initialization
        plc     100101111B              ;clear all the halt release flags
        spc     10000b                  ;enable ioc port output and pull-low
        sca     10h                     ;enable ioc halt release
        sre     10h                     ;enable ioc stop release
;*************************************
main:                                   ;main program
        . . . . . .
        plc     01h                     ;clear HRF0
        sca     10h                     ;enable ioc halt release

        stop

        call    delay                   ;adjust program delay
        rf      02h                     ;reset BCF flag
        msb     temp                    ;check SCF1
        jb1     ioc_stop
        jmp     main
;*************************************
ioc_stop:                               ;ioc stop release program processing
        . . . . . .
        jmp     main
;*************************************
delay:                                  ;delay 100ms
        lds     temp+0,0fh

delay0:
        dec*    temp+0
        lds     temp+1,0fh

delay1:
        dec*    temp+1
        nop
        nop
        jnz     delay1
        lda     temp+0
        jnz     delay0
```

```
        rts
;*************************************************************************************
*******************

;*************************************
;*         int stop release     *
;*************************************
initiation:                             ;initialization
        plc     100101111B              ;clear all the halt release flags
        she     04h                     ;enable int halt release
        sre     20h                     ;enable int stop release
;*************************************
main:                                   ;main program
. . . . . .
        plc     04h                     ;clear HRF2
        she     04h                     ;enable int halt release

        stop

        call    delay                   ;adjust program delay
        rf      02h                     ;reset BCF flag
        msc     temp                    ;check SCF4
        jb0     int_stop
        jmp     main
;*************************************
int_stop:                               ;int stop release program processing
      . . . . . .

        jmp     main
;*************************************
delay:                                  ;delay 100ms
        lds     temp+0,0fh

delay0:
        dec*    temp+0
        lds     temp+1,0fh

delay1:
        dec*    temp+1
        nop
        nop
        jnz     delay1
        lda     temp+0
        jnz     delay0
        rts
;*************************************************************************************
*******************

;*****************************************
```

```
;* key matrix scanning stop release *
;*****************************************
```

The segment line is used as the scanning output port for the key matrix scanning function, and IOC is used as the state input port. Select “KI1” function form the multi-function pad(SEG32 / IOC1 / KI1) in mask option table. (Keep the other IOC port as the IOC function.)

```
initiation:                             ;initialization
        plc     100101111B              ;clear all the halt release flags
        spc     0fh                     ;Disable all the pull-down device on internal
                                        ;IOC port
        lds     halt_flag,00h
;*************************************
main:                                   ;main program
        . . . . . .
        spk     10h                     ;Sets Key Matrix scanning output state
        she     20h                     ;Set HEF5 enable key matrix scanning halt
                                        ;release
        sre     80h                     ;set SRF6 enable key matrix scanning stop
                                        ;release
        plc     20h                     ;Clear HRF5


        stop

        call    delay                   ;adjust program delay
        rf      02h                     ;reset BCF flag
        mcx     temp                    ;check SCF8
        jb0     keyscan_stop
        jmp     main
;*************************************
keyscan_stop:                           ;key matrix scanning stop release program process
        . . . . . .
        jmp     main
;*************************************
delay:                                  ;delay 100ms
        lds     temp+0,0fh

delay0:
        dec*    temp+0
        lds     temp+1,0fh

delay1:
        dec*    temp+1
        nop
        nop
        jnz     delay1
        lda     temp+0
        jnz     delay0
        rts
;*****************************************************************************************
*******************
```

8. AP-TM87XX_02EV12

TITLE

The solution to "How to avoid the Crystal oscillator being stopped by external interference when BCF flag is cleared to 0?" as TM87 series products applying to 3V power supply and LCD 1/3 bias.

APPLICATION NOTE

Description:

When BCF flag is cleared to 0, IC internal logic circuit will operate under power-saving mode. At this point of time, the Crystal oscillator driver circuit is working under power-saving mode.

In such situations, for example, if there is an instant loss of electricity or interference from external force or touch, the incorrect setting of the BCF Flag, just remove the BAK capacitance on the original application and replace it by a 2M ohm resistor between VDD2 and BAK pin. (Shown in Figure 1) Although this approach will increase about 750nA of current, but it will ensure that the Crystal oscillator will be able to start-up automatically after interruption caused by the interference or abnormal environmental conditions. In general use, the standard connection is to connect the BAK in series with a capacitor to GND. (Shown in Fig. 2).

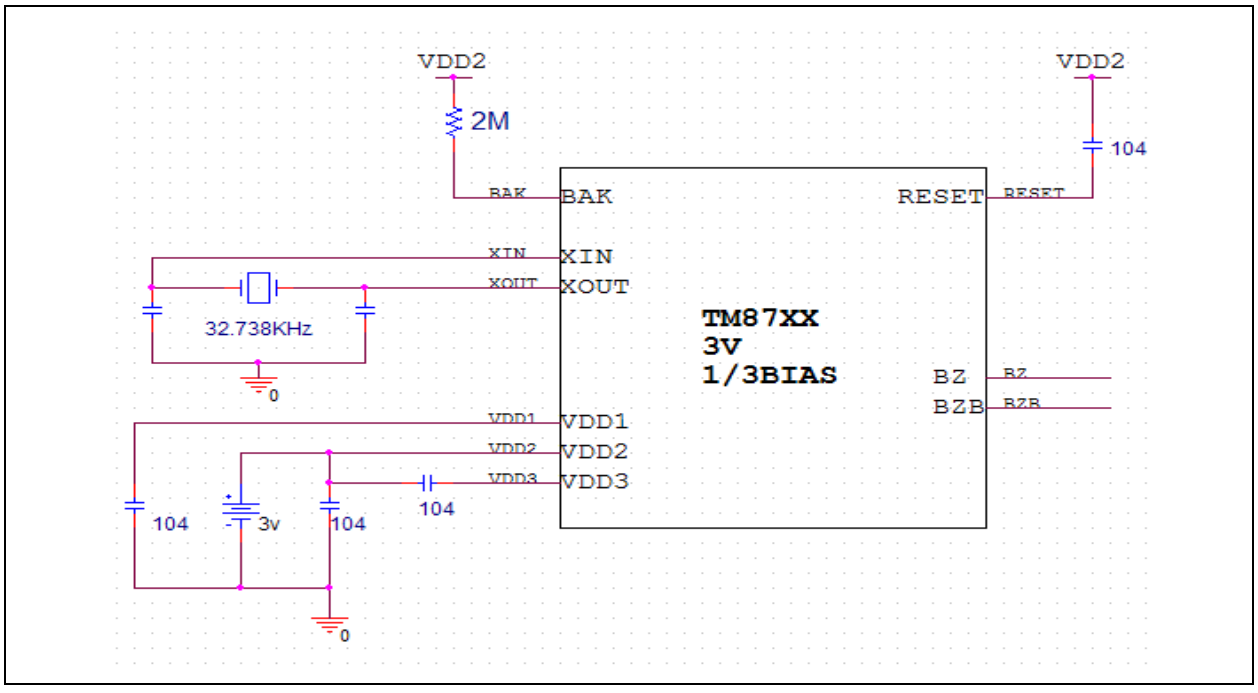


Fig.1

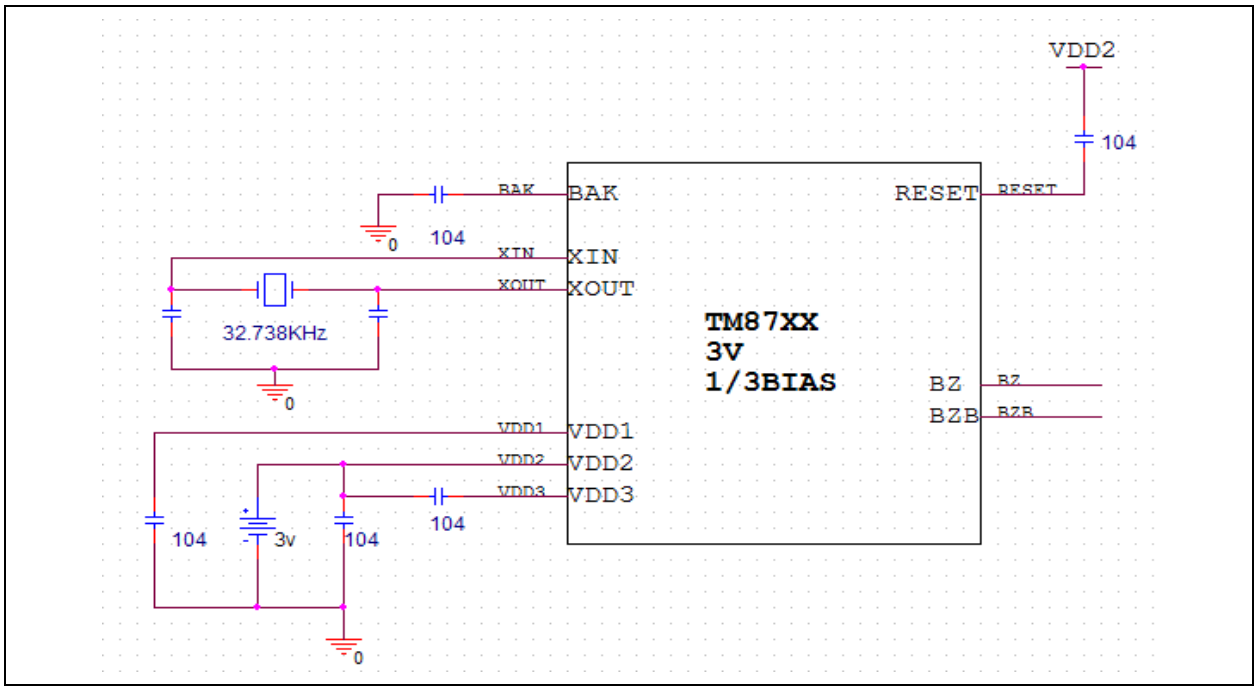

VDD2
104
BAK
BAK
RESET
RESET
0
104
XIN
XIN
XOUT
XOUT
TM87XX
3V
1/3BIAS
32.738KHz
0
BZ
BZ
BZB
BZB
VDD1
VDD1
VDD2
VDD2
VDD3
VDD3
104
104
3v
104
0

Fig.2

9. AP-TM87XX_10EV12

TITLE

The application of large sized LCD panel

APPLICATION NOTE

When users want to use the applications of large sized LCD panels, they should follow these recommendations:

Change the capacitance of Vdd1, Vdd2, Vdd3, CUP1, and CUP2 in the application circuit to 1uF.

In mask option table, the “LCD frame frequency” option please select “slow” item.

Since the voltage (VDD1-4) for LCD Waveform of TM8722 is generated by the charge-pump circuit, it relies on the external capacitor on VDD1~4 pins to stabilize the voltage. When the size of LCD panel is larger, the loading increases as well. Therefore, when TM8722 drives the large-size LCD panels, the voltages of VDD1~4 will be dragged down due to the charge sharing and a further cause of the flickering or unevenness on LCD panel.

Therefore, when TM8722 is driving a large-size LCD panels, the capacitance of Vdd1, Vdd2, Vdd3, CUP1, and CUP2 in the application circuit has to be changed to 1uF to minimize the influence of charge sharing.

How to define the specification of the large-size LCD panel for your application:

a. Prepare a working sample based on your application with tenx’s Demo Board and then provide this sample to the LCD manufacturer. Asking the LCD manufacturer to make some LCD panel samples which can meet the LCD waveform’s characteristics of your design.

b. In normal case, the liquid crystal material filled in the LCD panel is mostly based on a standard recipe. However, this recipe will not be suitable for the application of large-size LCD panels. Therefore, it is necessary to provide a complete module to the LCD manufacturer for testing. It is recommended to use the TN material and ask the LCD manufacturer to adjust the recipe of liquid crystal and view angle in order to get the better display quality.

c. Pay attention to the layout approach of the internal alignment within the LCD panel. Avoid routing wires too dense, too long, and the wire width too thin. These three problems will cause the unevenness of brightness and the panel to get interference easily by the flicker effect.

d. It is recommended to use the coating materials with lower resistance as the routine wire inside the LCD panel to reduce the voltage drop.

10. AP-TM87XX_12EV10

TITLE

The differences between TM8797 Demo Board/TM8795 and Mask type MCU in CX/SEG24 pin function

APPLICATION NOTE

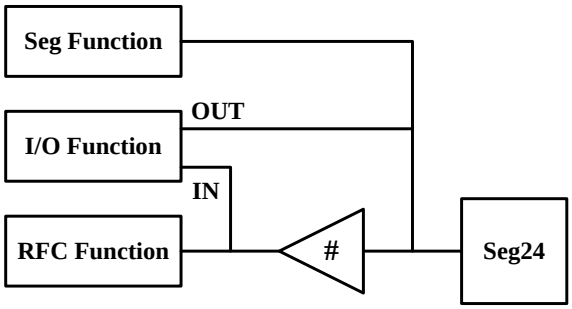


Figure 1. TM8797 demo board and TM8795

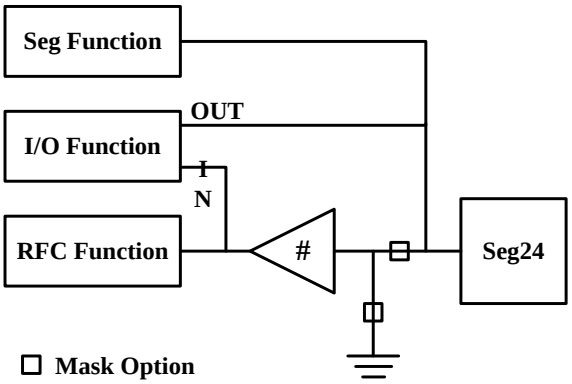


Figure 2. Mask type TM87 series MCU

Figure 1 shows the internal circuitry of CX/SEG24 pin of TM8797 demo board and TM8795.

When the MCU operates in 3V mode and the SEG24(multi-function pad) had been assigned as the LCD segment function, one of SEG24 pin's output voltage level(VDD1=1.5V) will cause extra current flows through the Schmitt-trigger inverter. However, the extra current consumption will not affect the internal functions of the MCUs.

When the MCU operates In 1.5V mode, the Schmitt-trigger inverter operates in 1.5V and all of the output voltage levels on SEG24 pin are equal to or higher than 1.5V. In this application, there is no extra current consumption flow through this inverter.

Figure 2 shows the internal circuitry of CX/SEG24 pin in mask type TM87 series MCUs.

When the mask type MCU operates in 3V mode and the SEG24(multi-function pad) had been assigned as the LCD segment function, the input of Schmitt-trigger inverter will be grounded and disconnected to SEG24 pin. The output voltage level (VDD1=1.5V) on SEG24 pin will not cause any extra current flow through Schmitt-trigger inverter.

11. AP-TM8722_01EV10

TITLE

TM8722 Electrical Characteristics

APPLICATION NOTE

A. The characteristics in this document are for reference only. The operating current is measured at room temperature(25℃) and in the absence of external loads. In mass-production, the characteristics will be influenced by process deviation, temperature, Option, loading and operating voltage etc.

B. Power Consumption
LCD：1/2Bias，1/4Duty * 9Seg，Size：1cm * 2.5cm

(1). At 3V，25°C

TM8722 (Crystal and Internal Fast 500 kHz 3V)											
Condition	1	2	3	4	5	6	7	8	9	10	11
3V	V	V	V	V	V	V	V	V	V	V	
LCD	on	on	on	on	on	on	on	OFF	OFF		
Operating	V	V	V	V							
Bcf Flag	1	0	1	0	1	1	0	1	0	1	0
Halt						V	V	V	V	V	
Stop					V						
500KHz			V	V							
32768Hz	V	V			V	V	V	V	V	V	V
Operating current (μ A)	9.497143	1.628571	107.1471	22.58714	0.078571	7.701429	1.29	7.068571	0.665714		
Freq. Tolerance (sec./day)										0.007143	-0.29143

TM8722 (Internal Fast Only 250 kHz 3V)							
Condition	1	2	3	4	5	6	7
3V	V	V	V	V	V	V	V
LCD	on	on	on	on	on	OFF	OFF
Operating	V	V					
Bcf Flag	1	0	1	1	0	1	0
Halt				V	V	V	V
Stop			V				
Operating current(μ A)	54.34857	14.57143	0.08	40.12	11.02857	39.21714	9.994286

(2). At 1.5V，25°C

TM8722 (Crystal and Internal Fast 500kHz 1.5V)											
Condition	1	2	3	4	5	6	7	8	9	10	11
1.5V	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	
LCD	on	on	on	on	on	on	on	OFF	OFF		
Operating	ˇ	ˇ	ˇ	ˇ							
Bcf Flag	1	0	1	0	1	1	0	1	0	1	0
Halt						ˇ	ˇ	ˇ	ˇ	ˇ	
Stop					ˇ						
500KHz			ˇ	ˇ							
32768Hz	ˇ	ˇ			ˇ	ˇ	ˇ	ˇ	ˇ	ˇ	ˇ
Operating current(μA)	2.094285	2.06	45.55428	45.53	0.08	1.405714	1.368571	1.277142	1.234285		
Freq. Tolerance (sec./day)										0.082857	-0.14

TM8722-980 (Internal Fast Only 250kHz 1.5V)							
1.5V	V	V	V	V	V	V	V
EXT-V							
LCD	on	on	on	on	on	OFF	OFF
Operating	V	V					
Bcf Flag	1	0	1	1	0	1	0
Halt				V	V	V	V
Stop			V				
Operating current(μ A)	27.10571	27.21143	0.08	20.16	20.24714	19.35857	19.35857

NOTE：

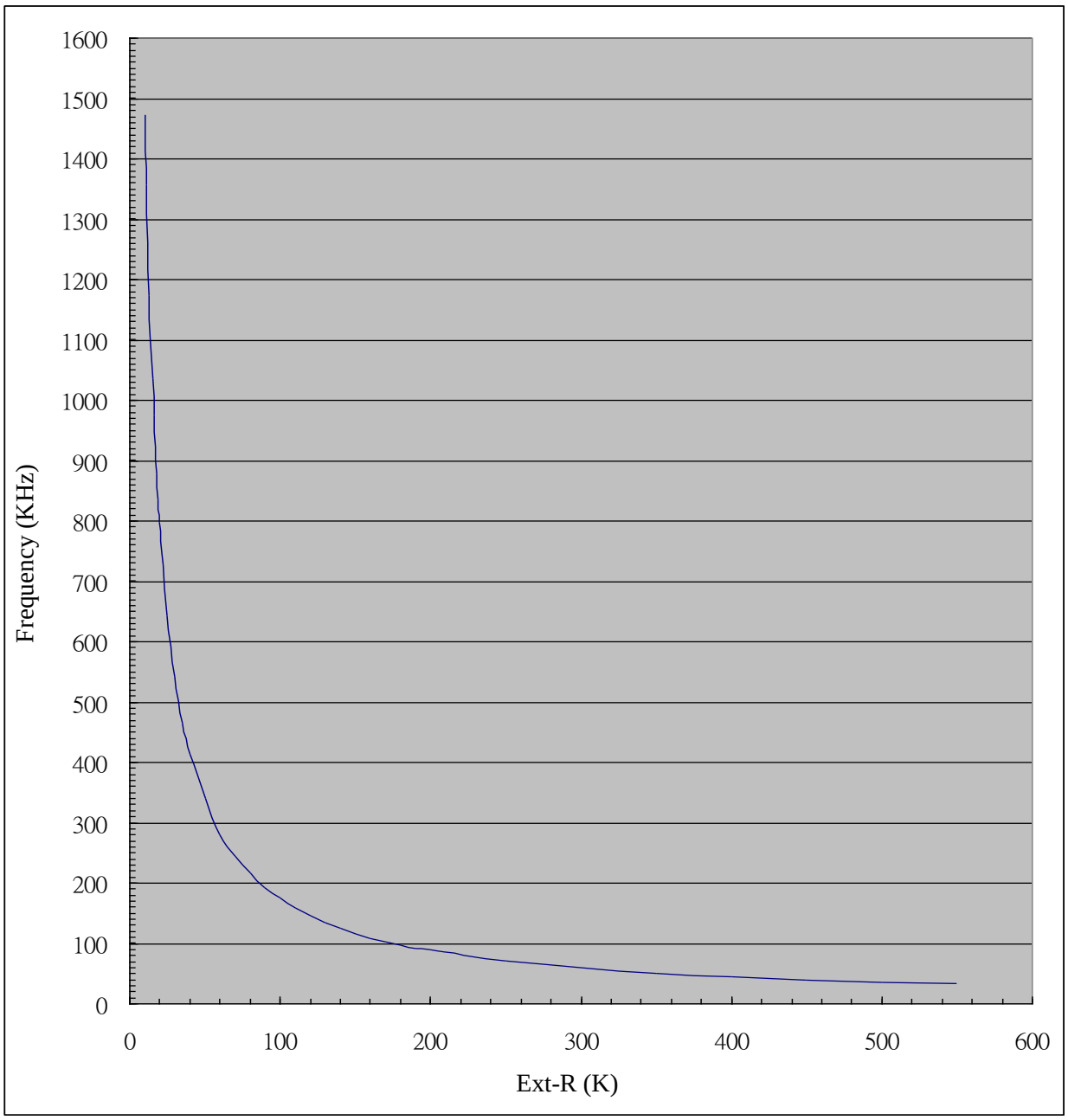
Frequency Tolerance means after trimming the capacitance of external capacitor of 32768 Hz Crystal oscillator, how many seconds the real time clock function will be fast or slow everyday.

Many factors will influence the frequency tolerance, such as the setting of BCF flag in MCU, foundry/lot No. / type of Crystal oscillator, PCB layout and quality of external capacitor.

C. Ext-R v.s. Frequency vs. Operating Current

(1). At 3V, 25℃

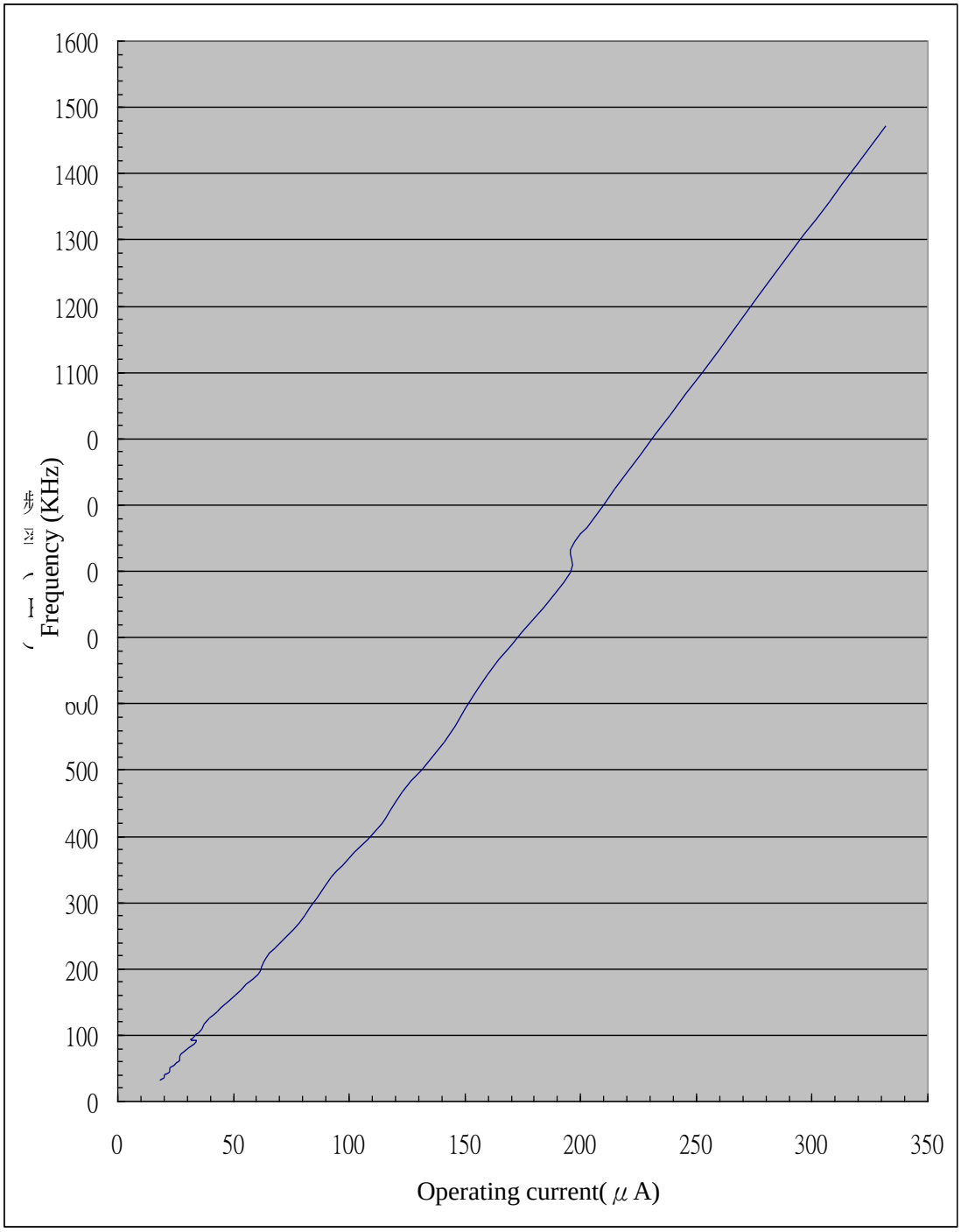
Ext-R vs. Frequency



(Figure 1)

(2). At 3V, 25℃

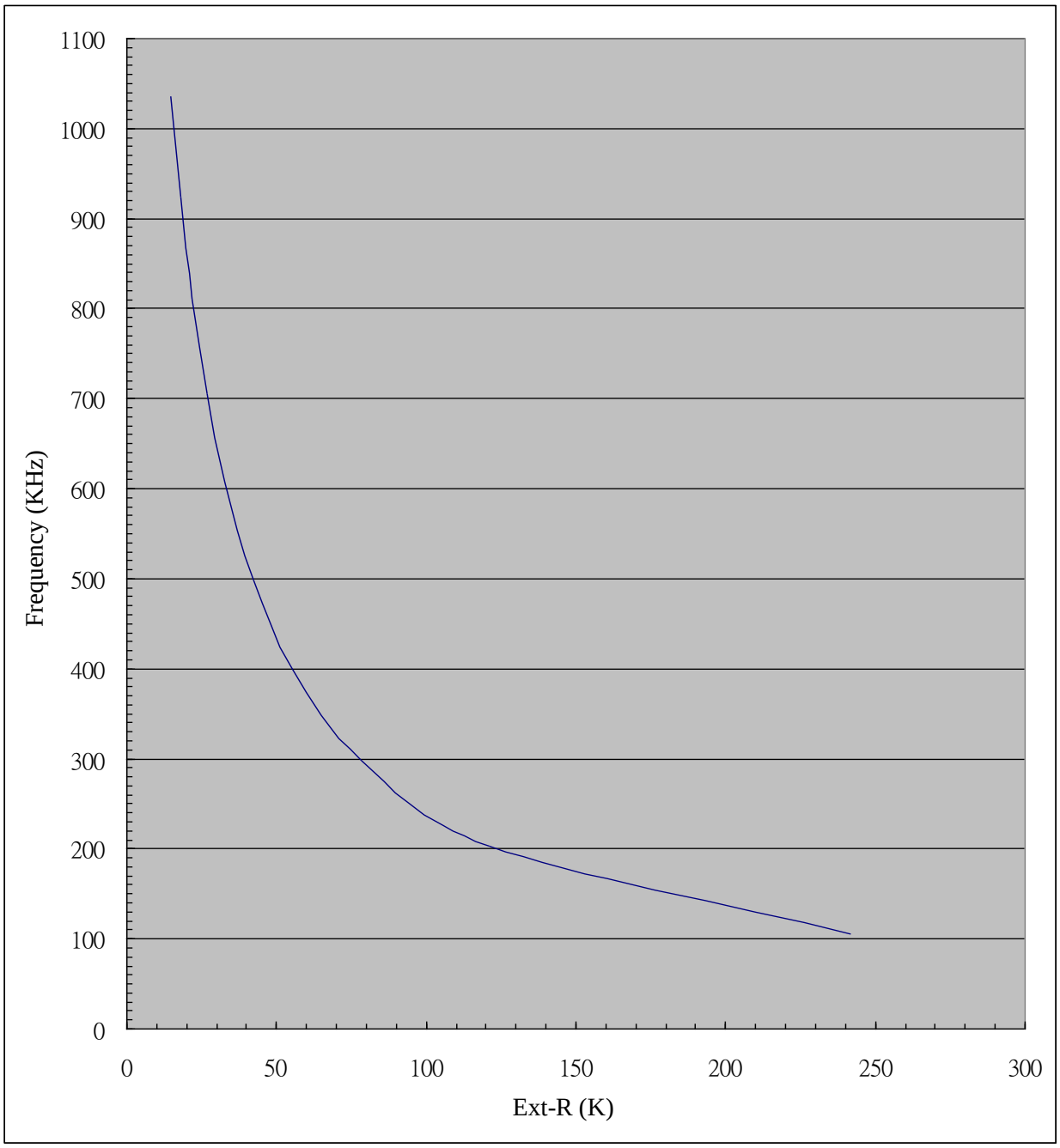
Frequency vs. Operating Current



(Figure 2)

(3). At 1.5V, 25℃

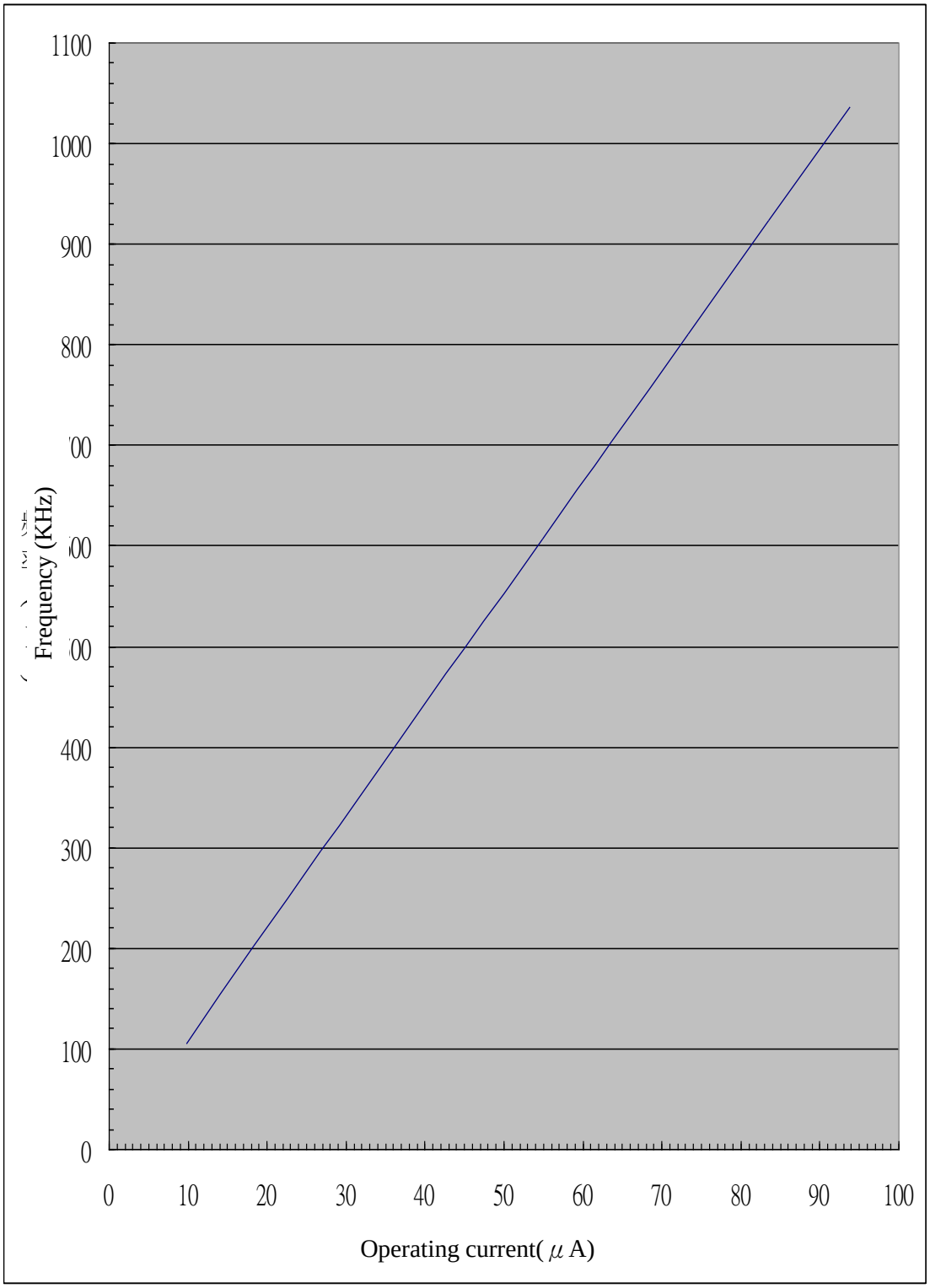
Ext-R vs. Frequency



(Figure 3)

(4). At 1.5V, 25℃

Frequency vs. Operating Current



(Figure 4)

D. To generate 32.768 kHz frequency by using Slow RC oscillator,
the RC value can be trimmed as follows:

(1). At 3V, 25℃: C=200pF and R=109 KΩ

(2). At 1.5V, 25℃: C=200pF and R=88 KΩ