十速科技股份有限公司
tenx technology inc.

TM87P18M

4-Bit Microcontroller with LCD Driver

User Manual

tenx technology inc

Rev 1.0, 2012/10/05

AMENDMENT HISTORY

Version	Date	Description
V1.0	Jul, 2012	New release

CONTENTS

Chapter 1 General Description

GENERAL DESCRIPTION

The TM87P18M is a One Time PROM embedded high-performance 4-bit microcontroller with LCD driver. It contains all the following functions on a single chip: 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, Resistance to Frequency Converter (RFC), LCD driver, look-up table, watchdog timer and key matrix scanning circuitry.

FEATURES

1. Powerful instruction set (173 instructions).
- Binary addition, subtraction, BCD adjusts, logical operation in direct and index addressing mode.
- Single-bit manipulation (set, reset, decision for branch).
- Various conditional branches.
- 16 working registers and manipulation.
- Table look-up.
- LCD driver data transfer.

2. Memory capacity.
- Program ROM capacity 4096 x 16 bits
- Data RAM capacity 512 x 4 bits.

3. Input/output ports.
- Port IOA 4 pins (with internal pull-low), muxed with SEG24~SEG27.
- Port IOB 4 pins (with internal pull-low), muxed with SEG28~SEG30, DC31.
- Port IOC 4 pins (with internal pull-low, low-level-hold, chattering prevention clock).
- Port IOD 4 pins (with internal pull-low, chattering prevention clock).

4. 8-level subroutine nesting.

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

6. Built-in Alarm, Frequency or Melody generator.

7. BZB, BZ (Muxed with IOB3/SE30, IOB4/DC31).

8. Built-in R to F Converter circuit.
- CX, RR, RT, RH (Muxed with IOA1~IOA4/ SEG24 ~ SEG27).

9. Built-in KEY_BOARD scanning function.
- K1~K16 (Share with SEG1~SEG16).
- KI1~KI4 (Muxed with IOC1~IOC4).

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

11. Watch dog timer.

12. LCD driver output.

- 8 common outputs and 32 segment outputs (drive up to 256 LCD segments).
- 1/2 ~ 1/8 Duty for LCD/LED.
- 1/2 Bias or 1/3 Bias for LCD/LED selected by option.
- Single instruction to turn off all segments.
- COM5~8, DC9/OD9, SEG17~ SEG23, DC31/OD31, SEG40, SEG41 can be defined as CMOS or P_open drain output type output in mask option.
- 32 LCD address.

13. Built-in Voltage doubler, halve charge pump circuit.

14. Dual clock operation

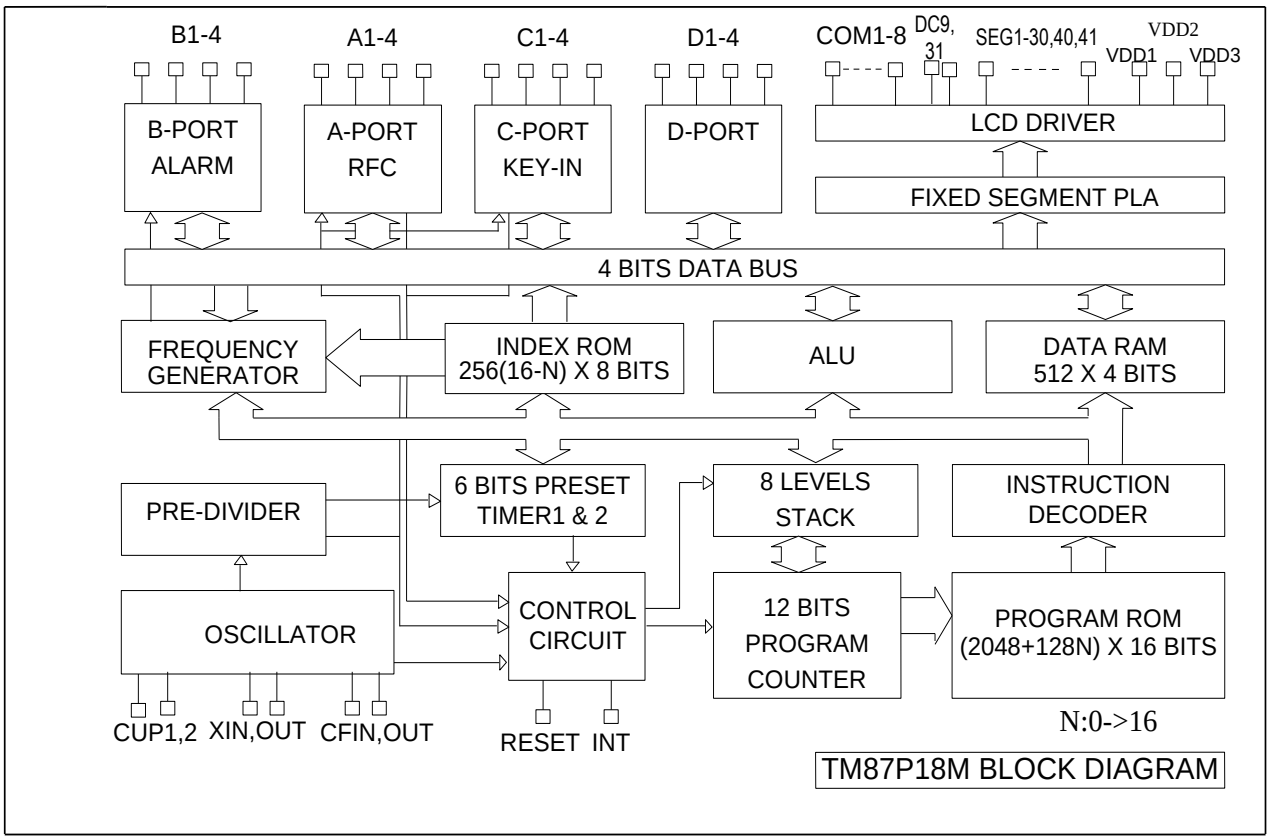
- Slow clock oscillation can be defined as X'tal or external RC type oscillator in mask option.
- Fast clock oscillation can be defined as 3.58 MHz ceramic resonator, internal R in external R type oscillator by mask option.

15. HALT function.

16. STOP function.

17. ROM code protect fuse.

BLOCK DIAGRAM



PAD COORDINATE

No	Name	No	Name
1	BAK	33	SEG13 (K13)
2	XIN	34	SEG14 (K14)
3	XOUT	35	SEG15 (K15)
4	CFIN	36	SEG16 (K16)
5	CFOUT	37	SEG17/DC17/OD17
6	GND	38	SEG18/DC18/OD18
7	VDD1	39	SEG19/DC19/OD19
8	VDD2	40	SEG20/DC20/OD20
9	VDD3	41	SEG21/DC21/OD21
10	CUP1	42	SEG22/DC22/OD22
11	CUP2	43	SEG23/DC23/OD23
12	COM1	44	SEG24/IOA1/CX
13	COM2	45	SEG25/IOA2/RR
14	COM3	46	SEG26/IOA3/RT
15	COM4	47	SEG27/IOA4/RH
16	COM5/DC5/OD5	48	SEG28/IOB1
17	COM6/DC6/OD6	49	SEG29/IOB2
18	COM7/DC7/OD7	50	SEG30/IOB3/BZB
19	COM8/DC8/OD8	51	DC31/OD31/IOB4/BZ
20	DC9/OD9	52	IOC1/KI1
21	SEG1 (K1)	53	IOC2/KI2
22	SEG2 (K2)	54	IOC3/KI3
23	SEG3 (K3)	55	IOC4/KI4
24	SEG4 (K4)	56	IOD1
25	SEG5 (K5)	57	IOD2
26	SEG6 (K6)	58	IOD3
27	SEG7 (K7)	59	IOD4
28	SEG8 (K8)	60	SEG40/DC40/OD40
29	SEG9 (K9)	61	SEG41/DC41/OD41
30	SEG10 (K10)	62	RESET
31	SEG11 (K11)	63	INT
32	SEG12 (K12)	64	VPP

PIN DESCRIPTION

Name	I/O	Description
BAK	P	Positive Back-up voltage, connect a 0.1u capacitor to GND. Positive voltage is needed to BAK pin for Serial Program/Read Mode.
VDD2	P	Positive supply voltage. Connect +3.0V battery positive pin to VDD2. Positive voltage is needed to VDD2 for Serial Program/Read Mode.
VDD1, 3	P	LCD supply voltage, and positive supply voltage. Positive voltage is needed to VDD3 for Serial Program/Read Mode.
RESET	I	Input pin from LSI reset request signal, with internal pull-down resistor. Instruction Reset Time can select “PH15/2” or “PH12/2” by option. Reset Type can select “Level” or “Pulse” by option. Signal for Serial Program/Read Mode.
INT	I I/O	Input pin for external INT request signal. Falling edge or rising edge triggered by option. Internal pull-down or pull-up resistor is selected by option. Signal for Serial Program/Read Mode.
CUP1, 2	O	Switching pins for supply the LCD driving voltage to the VDD1, VDD2, VDD3 pins. Connect the CUP1 and CUP2 pins with non-polarized electrolytic capacitor if 1/2 or 1/3 bias mode has been selected. In no BIAS mode, these pins should be open.
XIN XOUT	**I** O	Low speed oscillator, generates clock for time base functions (clock specified, LCD alternating frequency, Alarm signal frequency) or system clock oscillation. 32KHz Crystal oscillator for Slow Clock.
CFIN CFOUT	**I** O	High speed oscillator, system clock oscillation for FAST clock only or DUAL clock operation. The usage of 3.58 MHz ceramic/resonator oscillator or external R type oscillator is defined by mask option.
COM1~8	O	Output pins for driving the common pins of the LCD or LED panel. COM5~8 is muxed with DC/Open Drain, and set mask option.
DC9,31	O	DC/Open Drain
SEG1-30,40, 41	O	Output pins for driving the LCD or LED panel segment.
IOA1-4	I/O	Input / Output port A, can use software to define internal pull-low Resistor. This port is muxed with SEG24~27, and set by option.
IOB1-4	I/O	Input / Output port B, can use software to define internal pull-low Resistor. This port is muxed with SEG28~30, DC31 / BZB, BZ, and set by option.
IOC1-4	I/O	Input / Output port C, can use software to define internal pull-low / low-level-hold Resistor and Chattering clock to reduce input bounce. This port is muxed with KI1~4, and set by option.
IOD1-4	I/O	Input / Output port D, can use software to define internal pull-low Resistor, and Chattering clock to reduce input bounce.
(RFC)CX RR/RT/RH	I O	1 input pin and 3 output pins for RFC application. This port is muxed with SEG24~27 / IOA1~4, and set by option.
(ALM) BZB/BZ	 O	Output port for alarm, frequency or melody generator This port is muxed with SEG 30, DC 31 / IOB3, 4, and set by option.
KI1~4	I	Keyboard scanning input port. This port is muxed with IOC1~4, and set by option.
GND	P	Negative supply voltage. Connect for Serial Program/Read Mode.
VPP	P	High Voltage is need to VPP for Serial Program/Read Mode. Connected VDD2 to VPP or floating for Normal Mode.

Serial Program/Read Connect Pins:

VPP, VDD2, VDD3, GND, RESET, INT, BAK

CHARACTERIZATION

1. ABSOLUTE MAXIMUM RATINGS

(GND= 0V)

Name	Symbol	Range	Unit
Maximum Supply Voltage	VDD1	-0.3 to 5.5	V
	VDD2	-0.3 to 5.5	V
	VDD3	-0.3 to 8.5	V
	VPP	-0.3 to 13.5	V
Maximum Input Voltage	Vin	-0.3 to VDD1/VDD2+0.3	V
Maximum Output Voltage	Vout1	-0.3 to VDD1/VDD2+0.3	V
	Vout2	-0.3 to VDD3+0.3	V
Maximum Operating Temperature	Topg	-40 to +80	℃
Maximum Storage Temperature	Tstg	-25 to +125	℃

2. POWER CONSUMPTION

(At VDD2=3.0V, Ta=-40℃ to 80℃, GND= 0V)

Name	Sym.	Condition	Min.	Typ.	Max.	Unit
HALT mode	IHALT	Only 32.768 KHz Crystal oscillator operating, without loading. BCF = 0, 1/4 duty, ph0=BCLK		3	6	uA
STOP mode	ISTOP				1	uA
Normal Mode	I_{32K}	Only 32.768 KHz Crystal oscillator operating, without loading. BCF = 0, 1/4 duty, ph0=BCLK	8			uA
External R	$I_{Ext.\ R}$	R = 150 KΩ oscillator operating, without loading. BCF = 0, 1/4 duty, ph0=BCLK	36			uA
3.58 MHz ceramic resonator	$I_{3.58Mcr}$	Only 3.58 MHz ceramic resonator operating, without loading. BCF = 0, 1/4 duty, ph0=BCLK	480			uA

Note: When External R oscillator mode is operating, the current consumption will depend on the frequency of oscillation.

3. ALLOWABLE OPERATING CONDITIONS

(At Ta=-40℃ to 80℃, GND= 0V)

Name	Symb.	Condition	Min.	Max.	Unit
Supply Voltage	VDD2		2.4	5.25	V
	VDD3		2.4	8.0	V
	VPP		2.4	12.5	V
Oscillator Start-Up Voltage	VDD_{stup}	32.768 KHz Crystal Mode	1.4		V
		3.58 ceramic resonator Mode	1.8		V
Oscillator Sustain Voltage	VDD_{sut}	32.768 KHz Crystal Mode	1.3		V
		3.58 ceramic resonator Mode	1.55		V
Supply Voltage	VDD2	Li Mode	2.4	3.6	V
Input “H” Voltage	Vih1	Li Battery Mode	VDD2-0.7	VDD2+0.7	V
Input “L” Voltage	Vil1		-0.7	0.7	V
Input “H” Voltage	Vih2	OSCIN at Li Battery Mode	0.8xVDD2	VDD2	V
Input “L” Voltage	Vil2		0	0.2xVDD2	V
Input “H” Voltage	Vih3	CFIN at Li Battery	0.8xVDD2	VDD2	V
Input “L” Voltage	Vil3		0	0.2xVDD2	V
Operating Freq	Fopg1	32.768 KHz Crystal Mode	32		KHz
	Fopg2	External R mode	10	1000	KHz

4. ALLOWABLE OPERATING FREQUENCY

(At Ta=-40℃ to 80℃, GND= 0V)

Condition	Max. Operating Frequency
BAK=3V	4 MHz

5. INTERNAL RC FREQUENCY RANGE

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

6. ELECTRICAL CHARACTERISTICS

At#1:VDD2=3.0V(Li); At#2:VDD2=5.0V;

- **Input Resistance**

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
"L" Level Hold Tr(IOC)	Rllh1	Vi=0.2VDD2,#1	10	40	100	KΩ
	Rllh2	Vi=0.2VDD2,#2	5	20	50	KΩ
IOA,B,C Pull-Down Tr	Rmad1	Vi=VDD2,#1	200	500	1000	KΩ
	Rmad2	Vi=VDD2,#2	100	250	500	KΩ
INT Pull-up Tr	Rintu1	Vi=VDD2,#1	200	500	1000	KΩ
	Rintu2	Vi=VDD2,#2	100	250	500	KΩ
INT Pull-Down Tr	Rintd1	Vi=GND,#1	200	500	1000	KΩ
	Rintd2	Vi=GND,#2	100	250	500	KΩ
RES Pull-Down R	Rres	Vi=GND or VDD2,#1,#2	10	45	100	KΩ

7. DC Output Characteristics

At#3:VDD2=2.4V(Li); At#4:VDD2=4.0V;

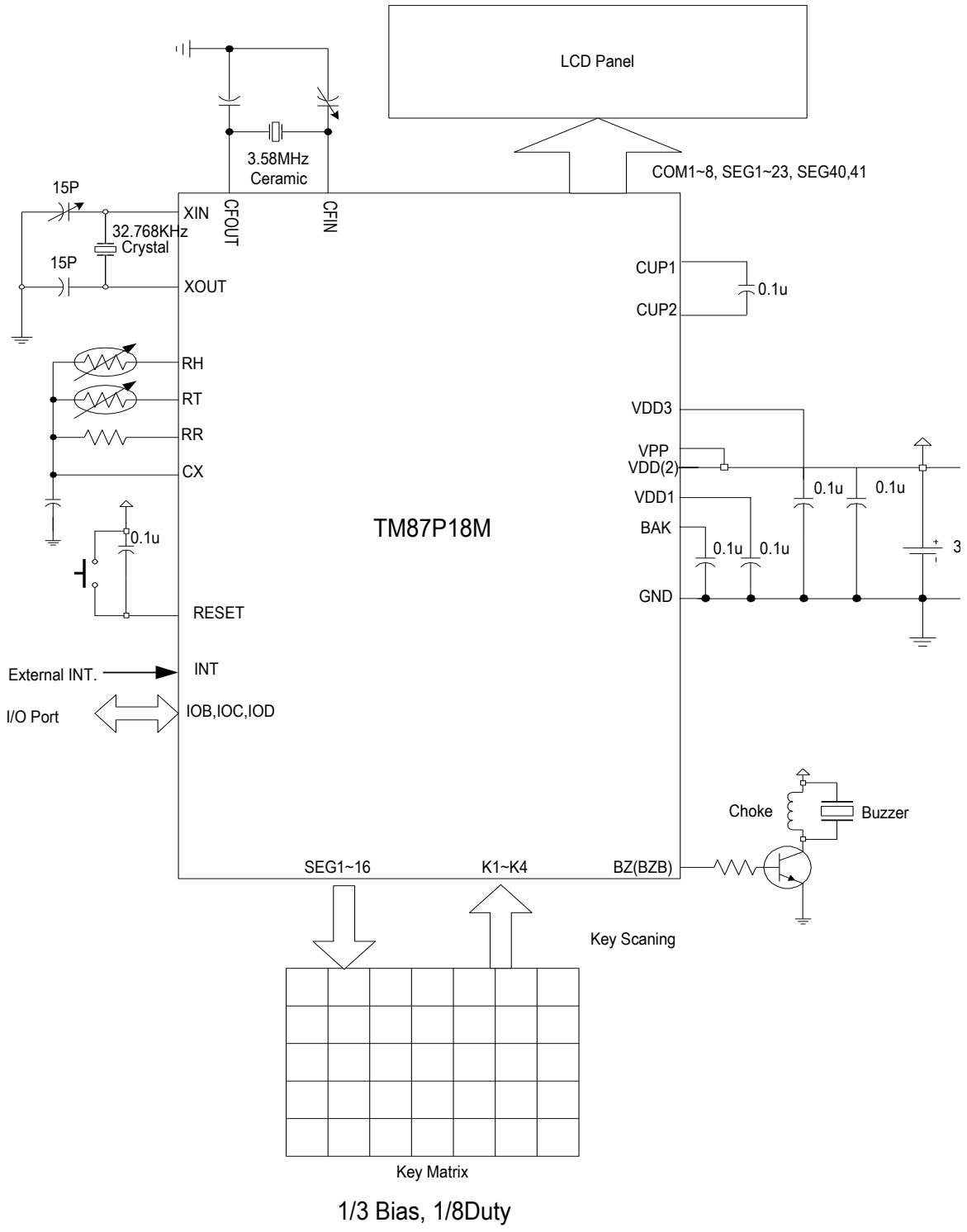
Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit
Output "H" Voltage	Voh3c	Ioh=-1 mA,#3	COM5~8,DC9 SEG28~30,DC31, SEG40,41, IOC, IOD	2.0			V
	Voh4c	Ioh=-3 mA,#4		3.2			V
Output "L" Voltage	Vol3c	Iol=2 mA,#3				0.4	V
	Vol4c	Iol=6 mA,#4				0.8	V
Output "H" Voltage	Voh3c	Ioh=-3 mA,#3	SEG24~27,INT	2.0			V
	Voh4c	Ioh=-5 mA,#4		3.2			V
Output "L" Voltage	Vol3c	Iol=5 mA,#3				0.4	V
	Vol4c	Iol=10 mA,#4				0.8	V

8. Segment Driver Output Characteristics

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
1/2 Bias Display Mode							
Output “H” Voltage	Voh3f	Ioh=-1 uA,#3	SEG-n	2.2			V
	Voh4f	Ioh=-1 uA,#4		3.8			V
Output “L” Voltage	Vol3f	Iol=1 uA,#3				0.2	V
	Vol4f	Iol=1 uA,#4				0.2	V
Output “H” Voltage	Voh3g	Ioh=-10 uA,#3	COM-n	2.2			V
	Voh4g	Ioh=-10 uA,#4		3.8			V
Output “M” Voltage	Vom3g	Iol/h=+/-10 uA,#3	COM-n	1.0		1.4	V
	Vom4g	Iol/h=+/-10 uA,#4		1.8		2.2	V
Output “L” Voltage	Vol3g	Iol=10 uA,#3				0.2	V
	Vol4g	Iol=10 uA,#4				0.2	V
1/3 Bias display Mode							
Output “H” Voltage	Voh3i	Ioh=-1 uA,#3	SEG-n	3.4			V
	Voh4i	Ioh=-1 uA,#4		5.8			V
Output “M1” Voltage	Vom13i	Iol/h=+/-10 uA,#3		1.0		1.4	V
	Vom14i	Iol/h=+/-10 uA,#4		1.8		2.2	V
Output “M2” Voltage	Vom23i	Iol/h=+/-10 uA,#3		2.2		2.6	V
	Vom24i	Iol/h=+/-10 uA,#4		3.8		4.2	V
Output “L” Voltage	Vol3i	Iol=1 uA,#3				0.2	V
	Vol4i	Iol=1 uA,#4				0.2	V
Output “H” Voltage	Voh3j	Ioh=-10 uA,#3	COM-n	3.4			V
	Voh4j	Ioh=-10 uA,#4		5.8			V
Output “M1” Voltage	Vom13j	Iol/h=+/-10 uA,#3		1.0		1.4	V
	Vom14j	Iol/h=+/-10 uA,#4		1.8		2.2	V
Output “M2” Voltage	Vom23j	Iol/h=+/-10 uA,#3		2.2		2.6	V
	Vom24j	Iol/h=+/-10 uA,#4		3.8		4.2	V
Output “L” Voltage	Vol3j	Iol=10 uA,#3				0.2	V
	Vol4j	Iol=10 uA,#4				0.2	V

TYPICAL APPLICATION CIRCUIT

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



1/3 Bias, 1/8Duty

Chapter 2 TM87P18M Internal System Architecture

1. Power Supply

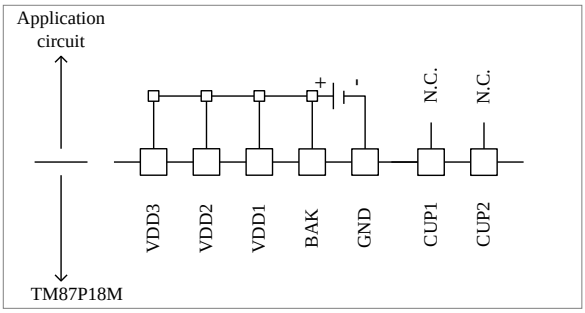
TM87P18M can operate at Li, all of these operating types are defined by mask option. The power supply circuitry is also generated by the necessary voltage level to drive the LCD panel with different bias. Shown below are the connection diagrams for 1/2 bias, 1/3 bias application.

LI BATTERY POWER SUPPLY

Operating voltage range: 2.4V ~ 5.25V.

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

1-1. NO BIAS USING A Li BATTERY POWER SUPPLY



MASK OPTION TABLE:

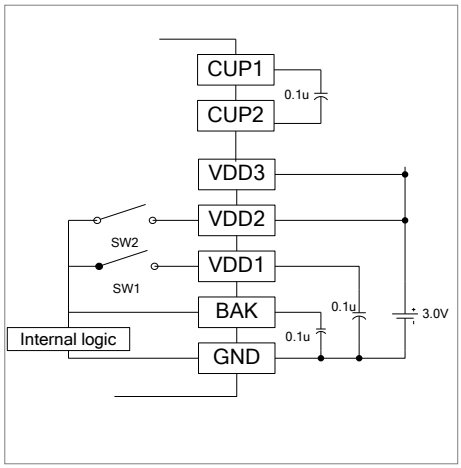
Mask Option name	Selected item
POWER SOURCE	(2) 3V BATTERY OR HIGHER
BIAS	(1) NO BIAS

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

1-2. 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
BIAS	(2) 1/2 BIAS

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

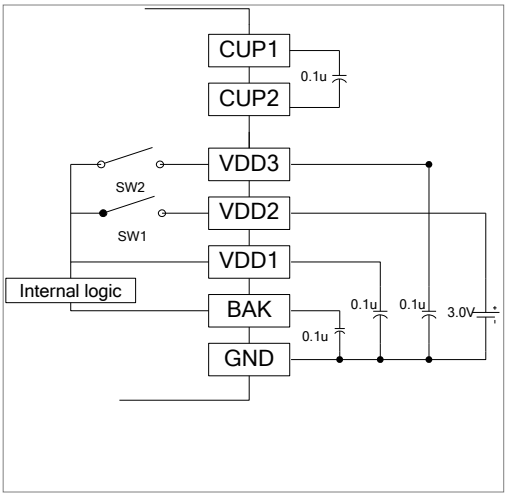
Note 2: The backup flag (BCF) is set to 1 in the initial cycle. When the backup flag is set to 1, the internal logic signal operated on VDD2 and the driving power of the oscillator circuit increases and the operating current also increases. Therefore, unless it is required, otherwise, the backup flag must be reset to 0 after the initial cycle. For the backup flag, please refer to 2-17.

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.

1-3. BIAS AT Li BATTERY POWER SUPPLY

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

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



MASK OPTION TABLE:

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

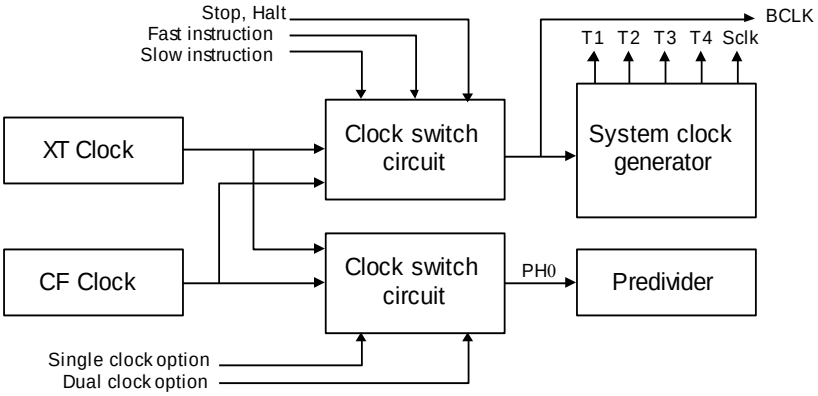
Note 1: The input/output ports operate between GND and VDD2.

Note 2: The backup flag (BCF) is set to 1 in the initial cycle. When the backup flag is set to 1, the internal logic signal is 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 cycle. For the backup flag, please refer to 2-17.

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. SYSTEM CLOCK

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



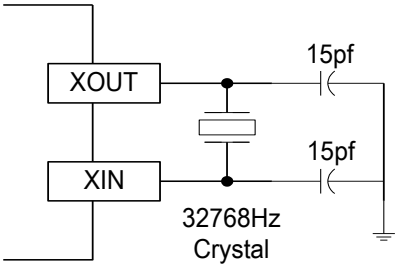
The system clock generator provides the necessary clock signals for the execution of instructions. The pre-divider generates various clock signals of different frequencies for the LCD driver, frequency generator, etc…

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

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

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

This 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, the oscillator will be stopped.



(1) X'tal

When the backup flag (BCF) is set to 1, the oscillator operates with a higher driving ability in order to reduce the start-up time of the oscillator. However, it increases the power consumption. Therefore, the backup flag should be reset unless otherwise required.

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

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

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 provide a faster clock source to the system. In single clock operation (fast only), this oscillator provides the clock signals to the system clock generator, pre-divider, timer, 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.

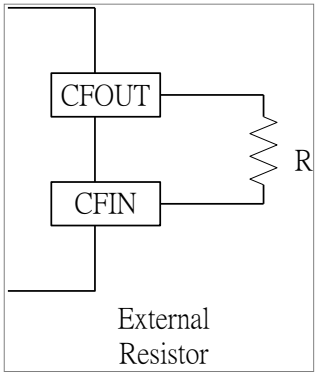
When the dual clock option is selected in mask option, this oscillator is inactive most of the time except when the FAST instruction is executed. After the FAST instruction is executed, the clock source (BCLK) of the system clock generator will be switched to CF clock, but the clock source for other 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:

2-2-1. RC oscillator with External Resistor, connection diagram is shown below:

MASK OPTION TABLE:

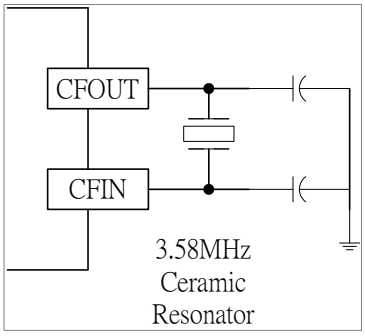
Mask Option name	Selected item
FAST CLOCK TYPE FOR FAST ONLY OR DUAL	(3) EXTERNAL RESISTOR



2-2-2. External 3.58 MHz Ceramic Resonator Oscillator

MASK OPTION TABLE:

Mask Option name	Selected item
FAST CLOCK TYPE FOR FAST ONLY OR DUAL	(4) 3.58 MHz Ceramic Resonator



Notes: 1. If it is required to reset the BCF to 0 in Li battery power mode, do not use a 3.58 MHz Ceramic Resonator as the oscillator.

2-2-3. Internal RC Oscillator

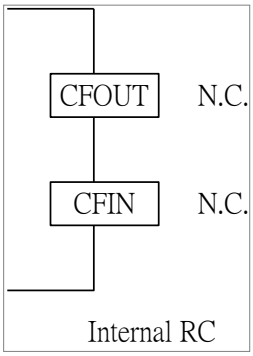
MASK OPTION TABLE:

For 250 KHz output frequency:

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

For 250 KHz output frequency:

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



FREQUENCY RANGE OF INTERNAL RC OSCILLATOR

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

2-3. THE COMBINATION OF THE CLOCK SOURCES

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

2-3-1. Dual Clock

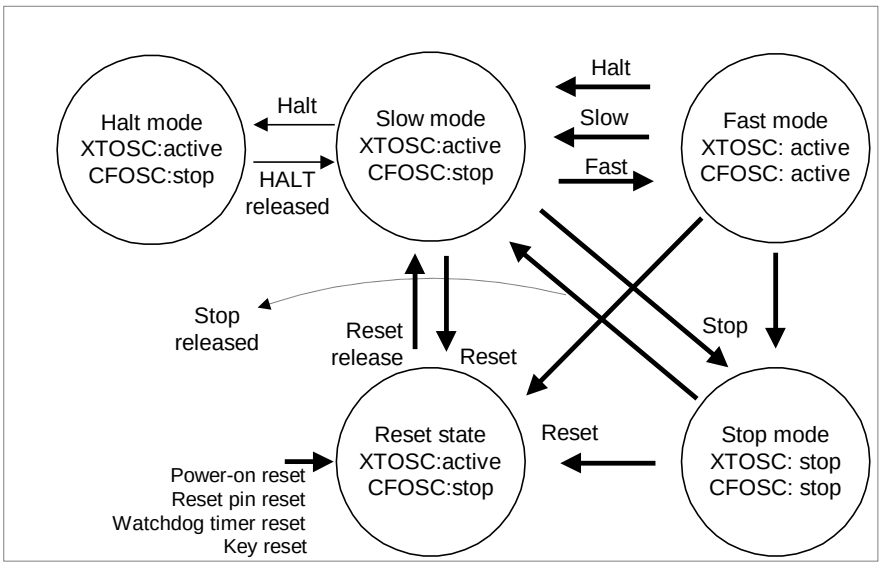
MASK OPTION TABLE:

Mask Option name	Selected Item
CLOCK SOURCE	(3) DUAL

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

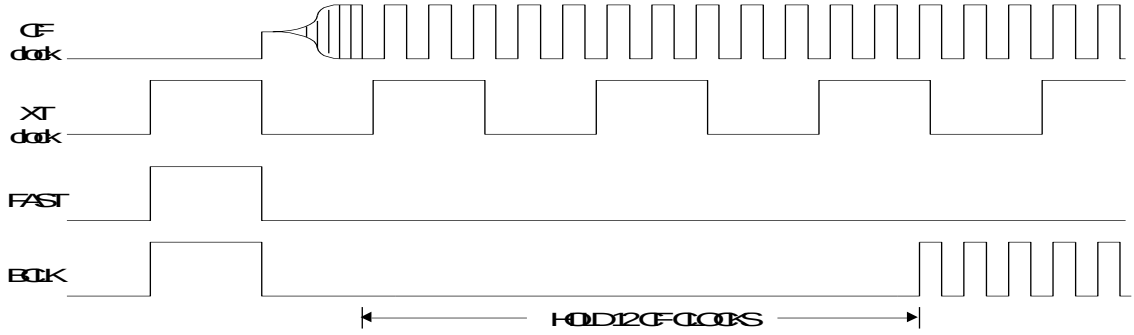
When this mode is selected in mask option, the clock source (BCLK) of the system clock generator will switch between the XT clock and the CF clock according to the user's program. When the HALT and STOP instructions are executed, the clock source (BCLK) will switch to the XT clock automatically.

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.



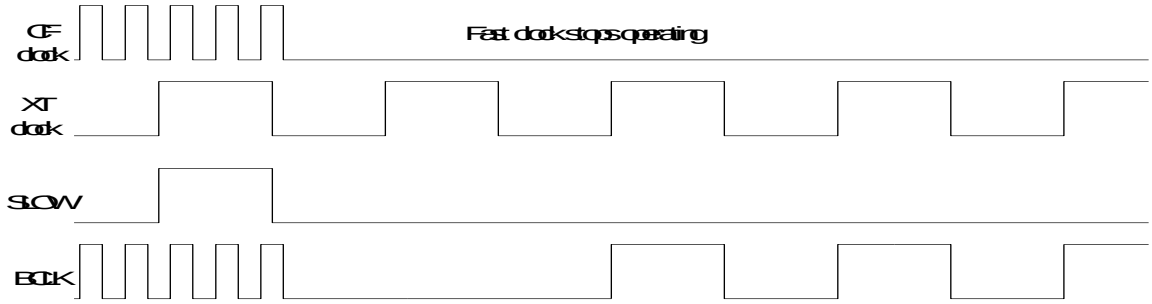
The state diagram of the dual clock mode is shown the above figure.

After the execution of the FAST instructions, 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 Switching 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 Switching from Fast to Slow

2-3-2. Single Clock

MASK OPTION TABLE:

For Fast clock oscillator only

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

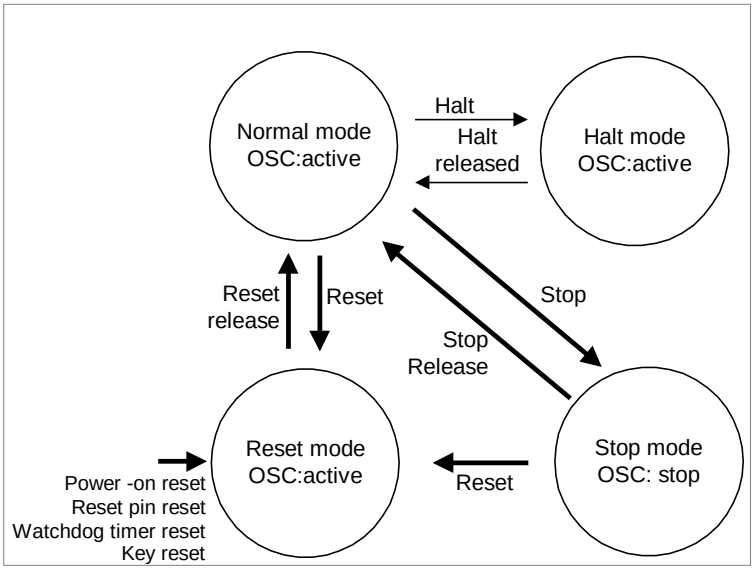
For slow clock oscillator only

Mask Option name	Selected item
CLOCK SOURCE	(2) SLOW ONLY

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

Either XT or CF clock may be selected by mask option in this mode. The FAST and SLOW instructions will perform as the NOP instruction in this option.

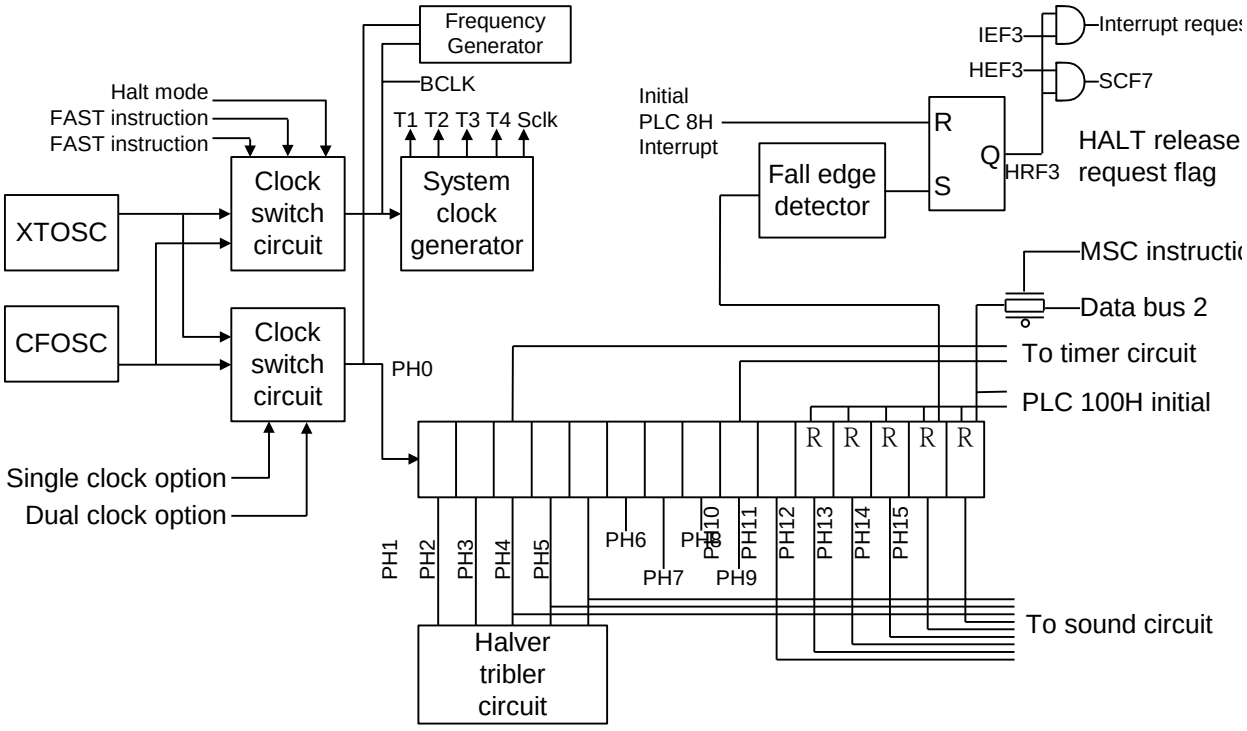
The backup flag (BCF) will be set to 1 automatically before the program enters the stop mode. This can ensure the Crystal oscillator will start up in a better condition.



This figure shows the State Diagram of Single Clock Option

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) as input. 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, LCD driver, 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 signal flag (HRF3), in this case, if the pre-divider interrupt enable mode (IEF3) is set in advance, the interrupt coming from predivider is accepted; and if the halt release enable mode (HEF3) is 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:

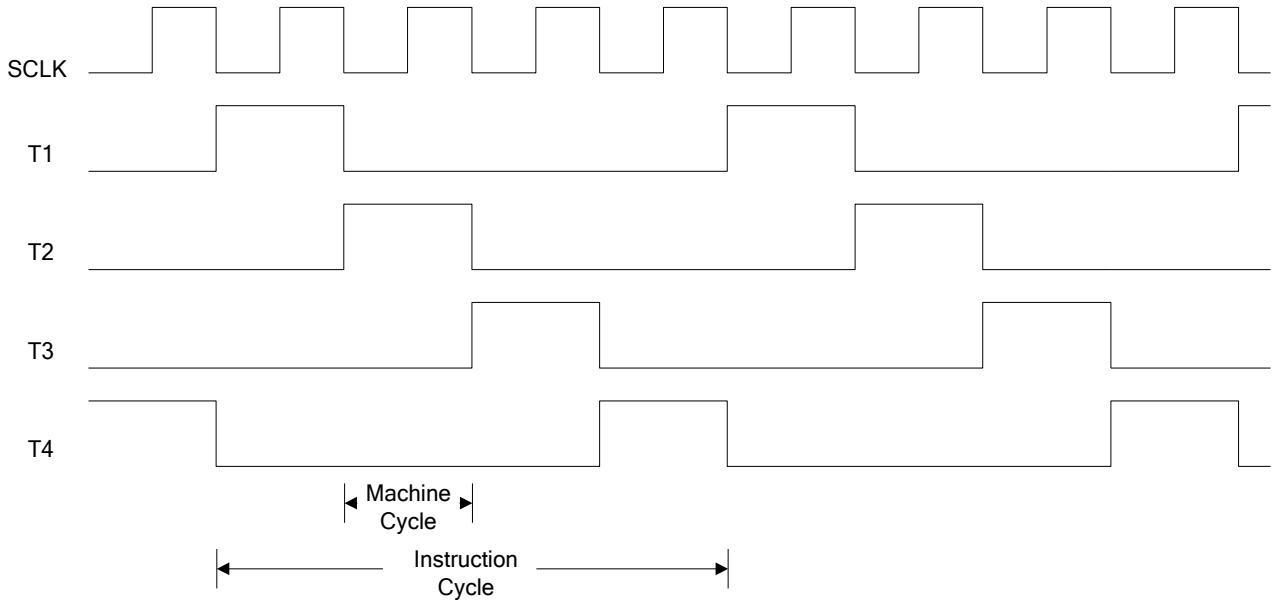
Mask Option name	Selected item
PH0 <-> BCLK FOR FAST ONLY	(1) PH0 = BCLK
PH0 <-> BCLK FOR FAST ONLY	(2) PH0 = BCLK/4
PH0 <-> BCLK FOR FAST ONLY	(3) PH0 = BCLK/8
PH0 <-> BCLK FOR FAST ONLY	(4) PH0 = BCLK/16

2-5. System Clock Generator

The system clock generator provides the necessary clocks 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 as shown below:



3. PROGRAM COUNTER (PC)

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

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

  PC PC + 1

- When executing JMP instructions, subroutine call instructions (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. All these relative jump instructions can only

be used to address the current page, that is when the current page is page 0 (PC11=0), only the range from 000h ~ 7FFh is accessible; when the current page is page 1 (PC11=1), only the range from 800h ~ FFFh is accessible.

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

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

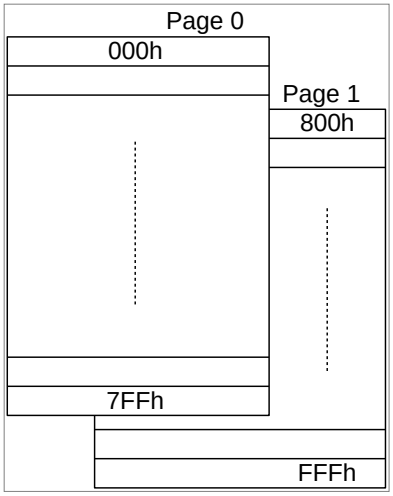
P10 to P0: the 11 Low-order bits of instruction operand

P11: page register

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

4. PROGRAM/TABLE MEMORY

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



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

Instruction ROM memory space = 2048 + (128 * N) words,

Table ROM memory space = 256(16 - N) bytes (N = 0 ~ 16).

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

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

MASK OPTION table:

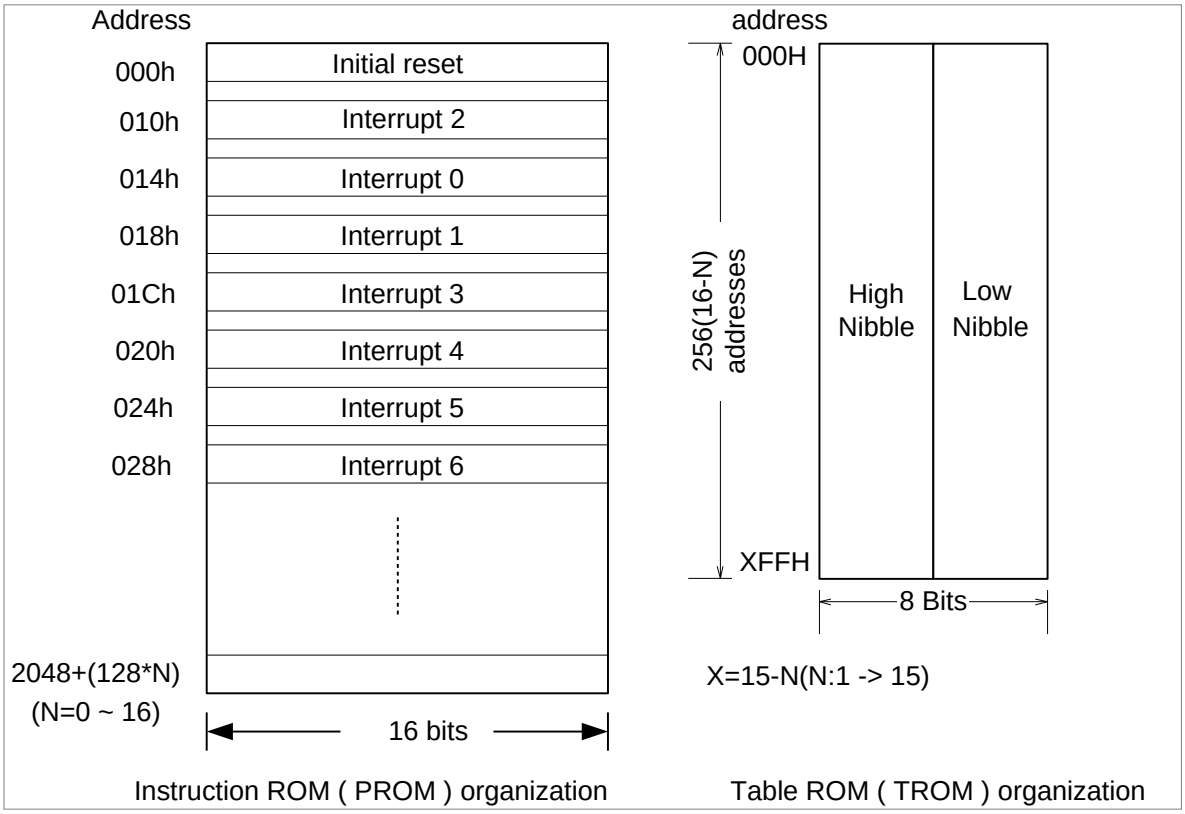
Mask Option name	Selected item	Instruction ROM memory space (Words)	Table ROM memory space (Bytes)
INSTRUCTION ROM <-> TABLE ROM	1 (N=0)	2048	4096
INSTRUCTION ROM <-> TABLE ROM	2 (N=1)	2176	3840
INSTRUCTION ROM <-> TABLE ROM	3 (N=2)	2304	3584
INSTRUCTION ROM <-> TABLE ROM	4 (N=3)	2432	3328
INSTRUCTION ROM <-> TABLE ROM	5 (N=4)	2560	3072
INSTRUCTION ROM <-> TABLE ROM	6 (N=5)	2688	2816
INSTRUCTION ROM <-> TABLE ROM	7 (N=6)	2816	2560
INSTRUCTION ROM <-> TABLE ROM	8 (N=7)	2944	2304
INSTRUCTION ROM <-> TABLE ROM	9 (N=8)	3072	2048
INSTRUCTION ROM <-> TABLE ROM	A (N=9)	3200	1792
INSTRUCTION ROM <-> TABLE ROM	B (N=10)	3328	1536
INSTRUCTION ROM <-> TABLE ROM	C (N=11)	3456	1280
INSTRUCTION ROM <-> TABLE ROM	D (N=12)	3584	1024
INSTRUCTION ROM <-> TABLE ROM	E (N=13)	3712	768
INSTRUCTION ROM <-> TABLE ROM	F (N=14)	3840	512
INSTRUCTION ROM <-> TABLE ROM	G (N=15)	3968	256
INSTRUCTION ROM <-> TABLE ROM	H (N=16)	4096	0

4-1. INSTRUCTION ROM (PROM)

There are some special locations that serve as interrupt service routines, such as reset address (000H), interrupt 0 address (014H), interrupt 1 address (018H), interrupt 2 address (010H), interrupt 3 address (01CH), interrupt 4 address (020H), interrupt 5 address (024H), and interrupt 6 address (028H), in the program memory.

When the valid address range of PROM exceeds 2048 addresses (800h), the memory space of PROM will automatically be defined as 2 pages. Refer to section 2-3.

This figure shows the Organization of ROM



4-2. TABLE ROM (TROM)

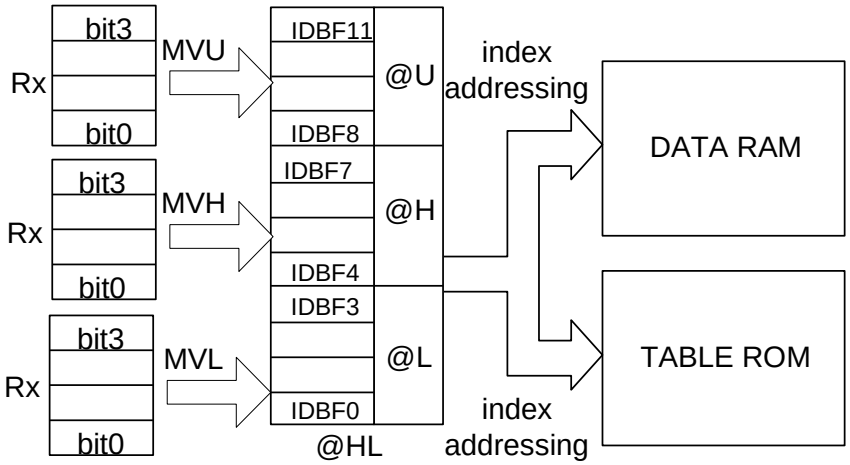
The table ROM is organized into 256(16-N) x 8 bits that shares the memory space with the instruction ROM (as shown in the figure above). This memory space stores the constant data or look up tables for the usage of the main program. All the table ROM addresses can be specified by the index address register (@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.

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 content of the register can be modified by executing MVH, MVL and MVU instructions. The execution of the MVL instructions 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 and MVU instructions will load the content of the data RAM (Rx) to the higher nibble of the register @H and @U, respectively.

@U register				@H register				@L register			
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit3	Bit2	Bit1	Bit0
IDBF11	IDBF10	IDBF9	IDBF8	IDBF7	IDBF6	IDBF5	IDBF4	IDBF3	IDBF2	IDBF1	IDBF0

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



This figure shows the diagram of the index address register

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

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

The comparison bit pattern is shown below:

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

Example:

```
……                ; @HL = 30h
CPHL   30h
SIE*   0h         ; disable IEF
```

```
JMP     lable1          ; this instruction will not be executed (NOP)
JMP     lable2          ; this instruction will be executed and than jump to lable2
……..
lable1:
……..
lable2:
```

6. STACK REGISTER (STACK)

Stack is a special design register following the first-in-last-out rule. It is used to save the contents of the program counter sequentially during subroutine call or execution of the interrupt service routine.

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

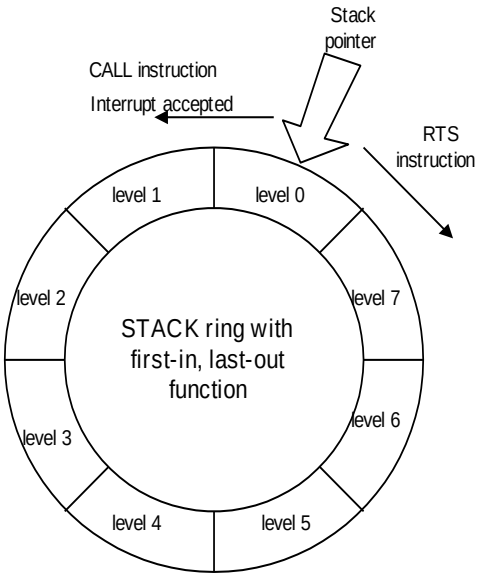
The stack register is organized using 11 bits by 8 levels but with no overflow flag; hence only 8 levels of subroutine call or interrupt are allowed (If the stacks are full, and either interrupt occurs or subroutine call executes, the first level will be overwritten).

Once the subroutine call or interrupt causes the stack register (STACK) overflow, the stack pointer will return to 0 and the content of the level 0 stack will be overwritten by the PC value.

The contents of the stack register (STACK) are returned sequentially to the program counter (PC) during execution of the RTS instruction.

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

The following figure shows the diagram of the stack.



7. DATA MEMORY (RAM)

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

The data memory may be accessed using two methods:

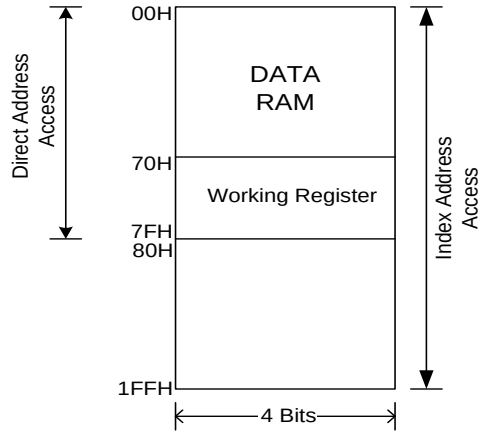
(1) Direct addressing mode

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

(2) Index addressing mode

The index address register (@HL) specifies the address of the data memory and all address space from 00H to 1FFH can be accessed.

The 16 specified addresses (70H to 7FH) in the direct addressing memory are also used as 16 working registers. The function of working register will be described in detail in section 2-8.



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

8. WORKING REGISTER (WR)

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

(1) To 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 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 of a working register and then output to the LCD PLA circuit. Such as:

(4) LCT, LCB, LCP

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.

10. ALU (Arithmetic and Logic Unit)

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

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

11. HEXADECIMAL CONVERT TO DECIMAL (HCD)

Decimal format is another number format for TM87P18M. When the content of the data memory has been assigned as decimal format, it is necessary to convert the results to decimal format after the execution of ALU instructions. When the decimal converting operation is processing, all of the operand data (including the contents of the data memory (RAM), accumulator (AC), immediate data, and look-up table) should be in the decimal format, or the results of conversion will be incorrect.

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

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

Example 1:

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

```
DAA*  10h        ; Convert the content of AC to decimal format.
                 ; The result in the data memory address 10H is"0"and in
                 ; the CF is "1". This represents the decimal number"10".
```

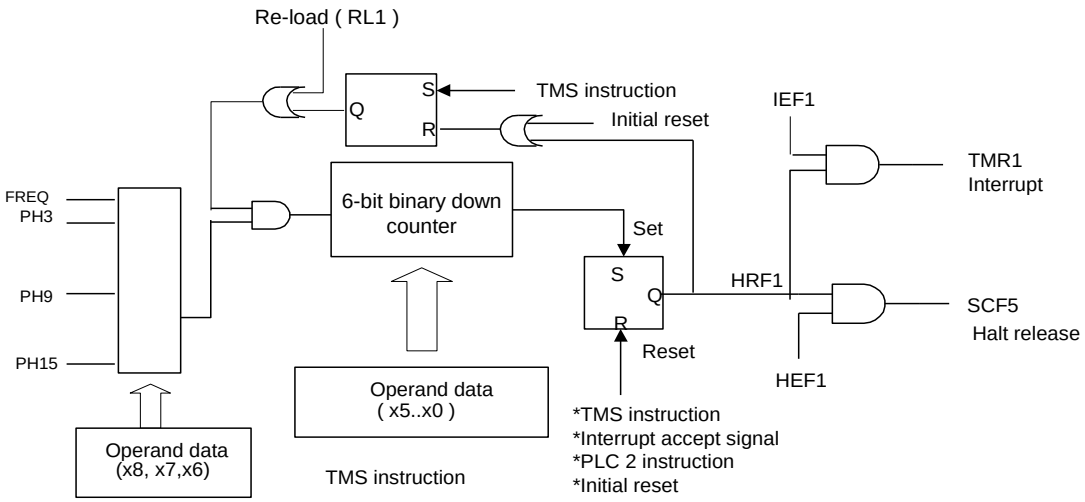
Instructions DAS, DAS*, DAS @HL can convert the data from hexadecimal format to decimal format after any subtraction operation. The conversion rules are shown in the following table and illustrated in Example 2.

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

Example 2:

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

12. TIMER 1 (TMR1)



This figure shows the TMR1 organization.

12-1. NORMAL OPERATION

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

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

When HRF1 = 1, and the TMR1 interrupt enable flag (IEF1) = 1, the interrupt is generated.

When HRF1 = 1, if the IEF1 = 0 and the TMR1 halt release enable (HEF1) = 1, program will escape from halt mode (if CPU is in halt mode) and then set the start condition flag 5 (SCF5) to 1 in the status register 3 (STS3).

After power on reset, the default clock source of TMR1 is PH3.

If watchdog reset occurs, the clock source of TMR1 will still keep the previous selection.

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

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

The following table shows the clock source setting for TMR1.

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

Notes:

1. n the TMR1 clock is PH3

   TMR1 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the TMR1 clock is PH9

   TMR1 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the TMR1 clock is PH15

   TMR1 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
4. When the TMR1 clock is PH5

   TMR1 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)
5. When the TMR1 clock is PH7

TMR1 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)

6. When the TMR1 clock is PH11

   TMR1 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)

7. When the TMR1 clock is PH13

   TMR1 set time = (Set value + error) * 8192 * 1/fosc (KHz) (ms)

   Set value: Decimal number of timer set value

   error: the tolerance of set value, 0 < error <1.

   fosc: Input of the predivider

   PH3: The 3rd stage output of the predivider

   PH5: The 5th stage output of the predivider

   PH7: The 7th stage output of the predivider

   PH9: The 9th stage output of the predivider

   PH11: The 11th stage output of the predivider

   PH13: The 13th stage output of the predivider

   PH15: The 15th stage output of the predivider

8. When the TMR1 clock is FREQ

   TMR1 set time = (Set value + error) * 1/FREQ (KHz) (ms).

   FREQ: refer to section 3-3-4.

12-2. RE-LOAD OPERATION

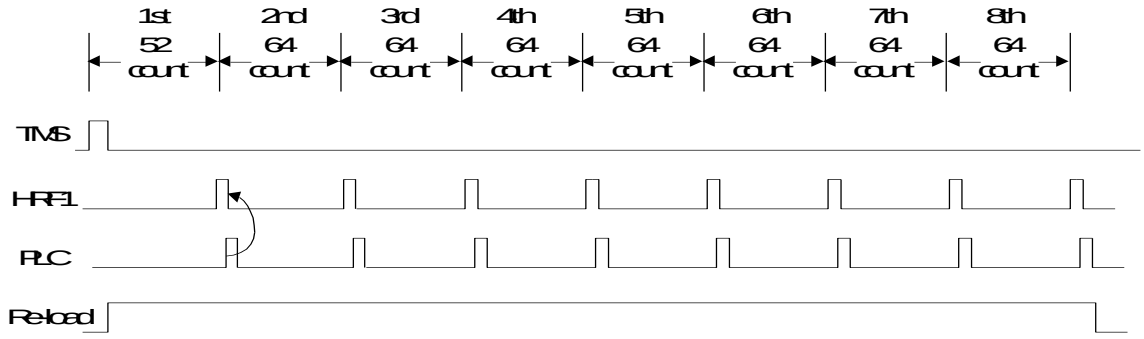
TMR1 provides a re-load function, which can last for a time interval longer than 3Fh. The SF 80h instruction enables the re-load function and RF 80h instruction disables it.

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

- It is necessary to execute either the TMS or the TMSX instructions to initiate the count down value before the re-load function is enabled, otherwise, TMR1 will automatically 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 not executed after each 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 PLC instruction. After a halt release or an interrupt occurs 8 times, disable the re-load function and then the counting is completed.

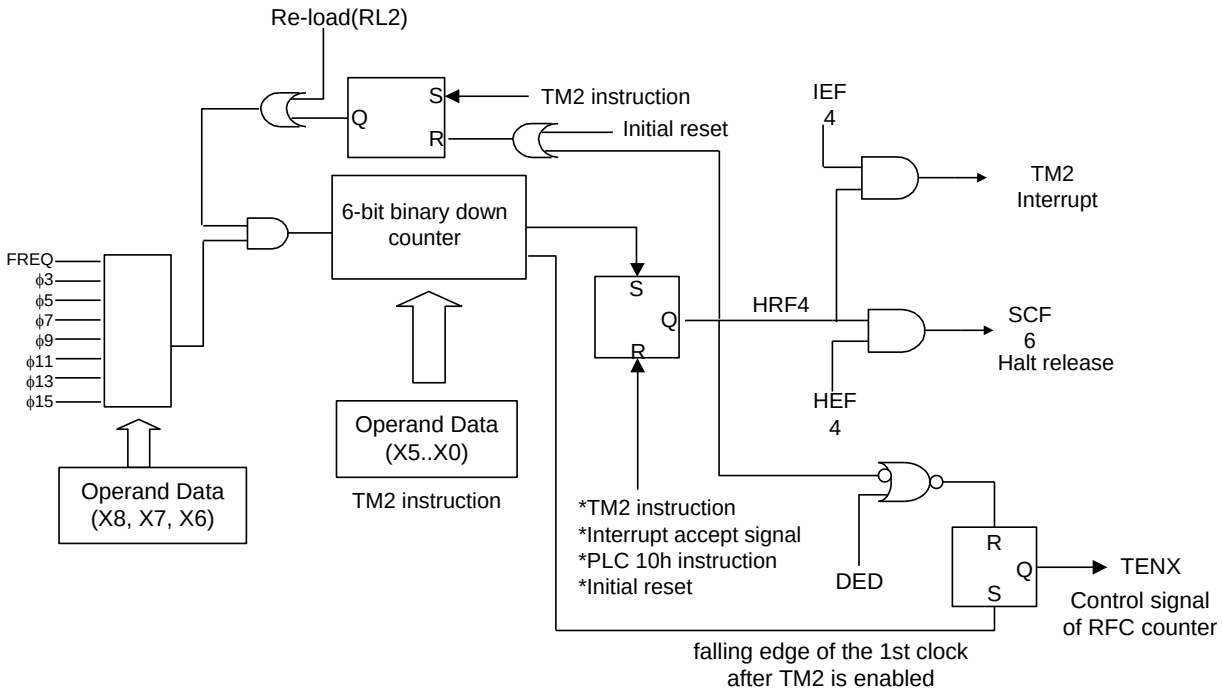


In the following example, S/W enters the halt mode to wait for the underflow of TMR1.

```
        LDS     0, 0            ; Initiate the underflow counting register
        PLC     2
        SHE     2               ; Enable the HALT release caused by TMR1
        TMSX    34h             ; Initiate the TMR1 value (52) and clock source is 9
        SF      80h             ; Enable the re-load function
RE_LOAD:
        HALT
        INC*    0               ; Increase the underflow counter
        PLC     2               ; Clear HRF1
        JB3     END_TM1         ; If the TMR1 underflow counter is equal to 8, exit subroutine
        JMP     RE_LOAD
END_TM1:
        RF      80h             ; Disable the re-load function
```

13. TIMER 2 (TMR2)

The following figure shows the TMR2 organization.



13-1. NORMAL OPERATION

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

Once TMR2 counts down to 3Fh, it stops counting, then generates an underflow signal and sets the halt release request flag 4 (HRF4) to 1.

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

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

If a watchdog reset occurs, the clock source of TMR2 will remain the same.

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

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

The following table shows the clock source setting for TMR2

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

Notes:

1. n the TMR2 clock is PH3

   TMR2 set time = (Set value + error) * 8 * 1/fosc (KHz) (ms)
2. When the TMR2 clock is PH9

   TMR2 set time = (Set value + error) * 512 * 1/fosc (KHz) (ms)
3. When the TMR2 clock is PH15

   TMR2 set time = (Set value + error) * 32768 * 1/fosc (KHz) (ms)
4. When the TMR2 clock is PH5

   TMR2 set time = (Set value + error) * 32 * 1/fosc (KHz) (ms)
5. When the timer clock is PH7

   TMR2 set time = (Set value + error) * 128 * 1/fosc (KHz) (ms)
6. When the TMR2 clock is PH11

   TMR2 set time = (Set value + error) * 2048 * 1/fosc (KHz) (ms)
7. When the TMR2 clock is PH13

   TMR2 set time = (Set value + error) * 8192 * 1/fosc (KHz) (ms)

   Set value: Decimal number of timer set value

   error: the tolerance of set value, 0 < error <1.

   fosc: Input of the predivider

   PH3: The 3rd stage output of the predivider

   PHn: The nth stage output of the predivider (n=5,7,9,11,13)
8. When the TMR2 clock is FREQ

   TMR2 set time = (Set value + error) * 1/FREQ (KHz) (ms).

   FREQ: refer to section 3-3-4.

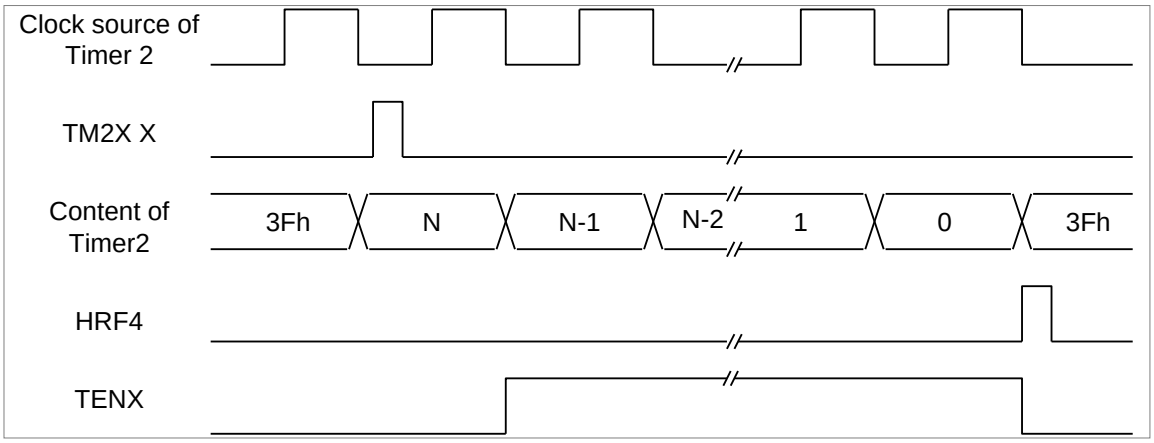
13-2. RE-LOAD OPERATION

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

13-3. TIMER 2 (TMR2) IN RESISTOR TO FREQUENCY CONVERTER (RFC)

TMR2 also controls the operation of the RFC function.

TMR2 will set 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. Refer to section 3-8 for more detail information on controlling the RFC counter. The following figure shows the operating timing of TMR 2 in RFC mode.



TMR2 also provides the re-load function when controlls the RFC function.

The SF2 1h instruction enables the re-load function, and the DED flag should be set to 1 by SF2 2h instruction. Once DED flag has been set to 1, TENX flag will not be cleared to 0 while 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 delivery. After the last underflow (HRF4) of TMR2 occurs, disable the re-load function by executing RF2 1h instruction.

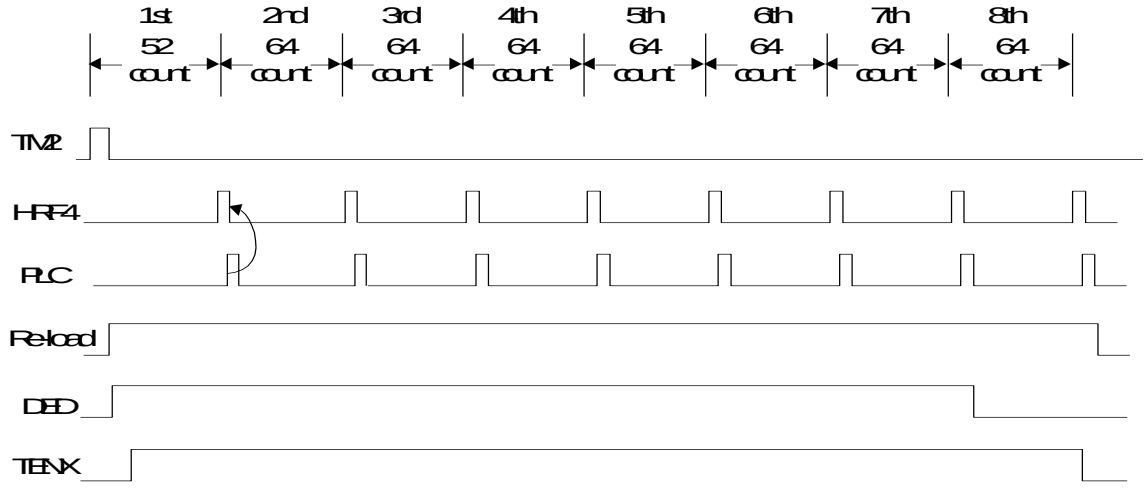
For example, if the target set value is 500, it will be divided as 52 + 7 * 64.

1. Set the initiate value of TMR2 to 52 and start counting.
2. Enable the TMR2 halt release or 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 halt release or interrupt occurs, clear the HRF4 flag by PLC instruction and increase the counting value to count the underflow times.
5. When halt release or interrupt occurs for the 7^{th} time, reset the DED flag.
6. When halt release or interrupt occurs for the 8^{th} time, disable the re-load function and the counting is completed.

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

```
LDS     0,0             ; Initiate the underflow counting register
PLC     10h
```

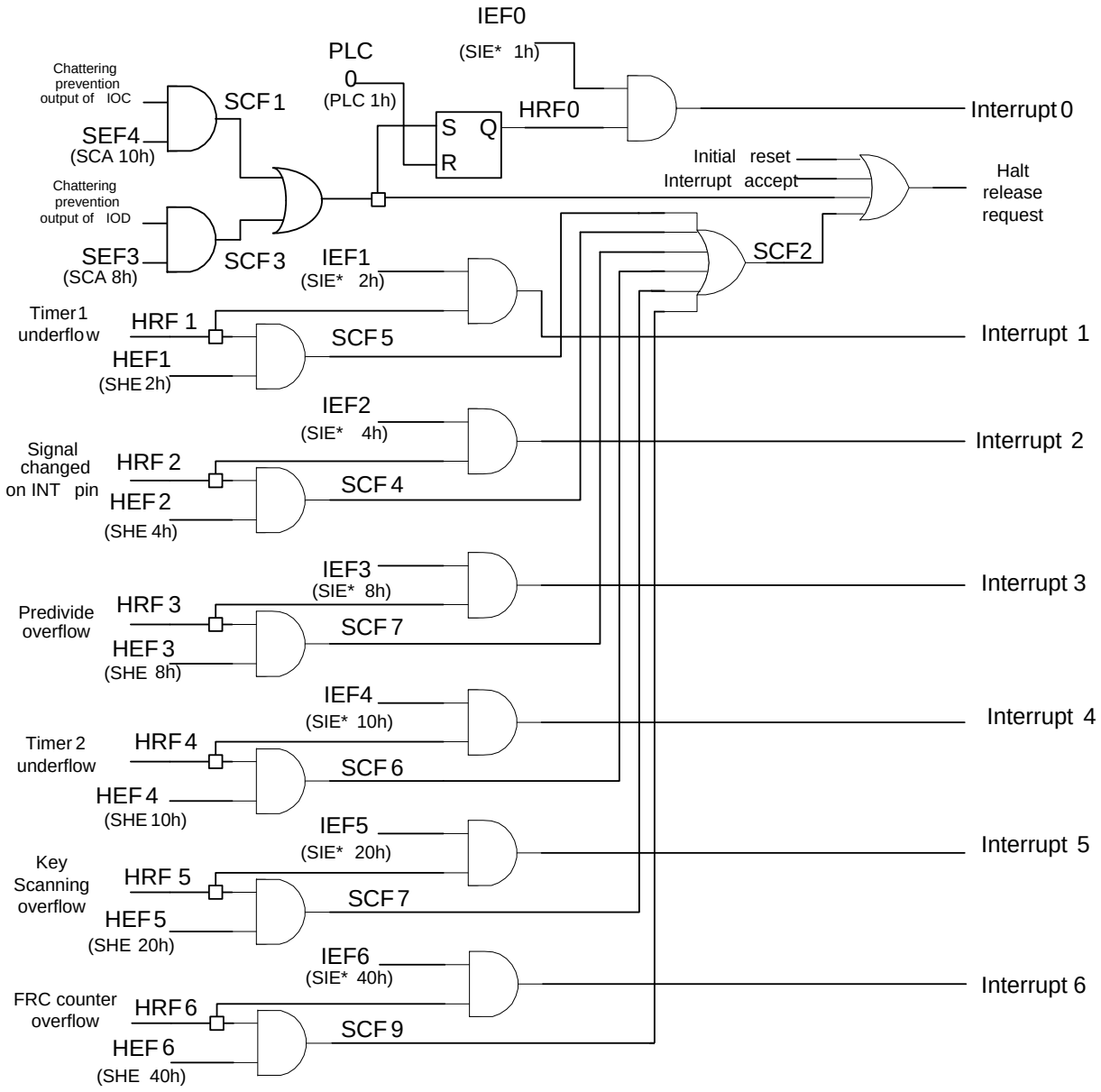
```
        SHE     10h         ; Enable the halt release caused by TM2
        SRF     19h         ; Enable RFC, and controlled by TM2
        TM2X    34h         ; Initiate the TM value(52) and clock source is 9
        SF2     3h          ; Enable the re-load function and set DED flag to 1
RE_LOAD:
        HALT
        INC*    0           ; Increase the underflow counter
        PLC     10h         ; Clear HRF4
        LDS     20h, 7
        SUB     0           ; When halt is released for the 7th time, reset DED flag
        JNZ     NOT_RESET_DED
        RF2     2           ; Reset DED flag
NOT_RESET_DED:
        LDA     0           ; Store 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

14. STATUS REGISTER (STS)

The status register (STS) is organized with 4 bits and comes in 4 types: status register 1 (STS1) to status register 4 (STS4). The following figure shows the configuration of the start condition flags for TM87P18M.



14-1. STATUS REGISTER 1 (STS1)

Status register 1 (STS1) consists of 2 flags:

(1) Carry flag (CF)

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

(2) Zero flag (Z)

Indicate 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	Bit0
Carry flag (AC)	Zero flag (Z)	NA	NA
Read / write	Read only	Read only	Read only

14-2. STATUS REGISTER 2 (STS2)

Status register 2 (STS2) consists of start condition flag 1, 2, 3 (SCF1, SCF2, SCF3) 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), and the status register 2 (STS2) is read-only.

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

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

Start Condition Flag 3 (SCF3)

When a signal change occurs on port IOC due to the execution of SCA instruction to and the halt mode is released as a result, SCF3 will be set. Executing the SCA instruction will cause SCF3 to be reset to 0.

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 IOA and IOC cause the halt mode to be released, SCF2 will be set to 1. 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 of 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.

14-3. STATUS REGISTER 3 (STS3)

When the halt mode is released by the start condition flag 2 (SCF2), the status register 3 (STS3) will update 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 output.

   There are two methods to reset the Start Condition Flag 4 (SCF4), one is to execute the PLC instruction to reset the halt release request flag 2 (HRF2) and the other 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 output.

   There are two methods to reset the Start Condition Flag 5 (SCF5), one is to execute the PLC instruction to reset the halt release request flag 1 (HRF1) and the other 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) is 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 output.

   There are two methods to reset the Start Condition Flag 7 (SCF7), one is to execute the PLC instruction to reset the halt release request flag 3 (HRF3) and the other is to execute the SHE instruction to reset the halt release enable flag 3 (HEF3).

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

The MSC instruction is used to transfer the contents of 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

14-4. STATUS REGISTER 3X (STS3X)

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

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

1. Start Condition Flag 8 (SCF8)

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

2. Start Condition Flag 6 (SCF6)

   SCF6 is set to 1 when an underflow signal from timer 2 (TMR2) causes the halt release request flag 4 (HRF4) to be output and the halt release enable flag 4 (HEF4) is set beforehand. To reset the Start Condition Flag 6 (SCF6), the PLC instruction must be used to reset the halt release request flag 4 (HRF4) or the SHE instruction must be used to reset the halt release enable flag 4 (HEF4).

3. Start Condition Flag 9 (SCF9)

   SCF9 is set when a finish signal from mode 3 of RFC function causes the halt release request flag 6 (HRF6) to be output and the halt release enable flag 9 (HEF9) is set beforehand. In this case, the 16-counter of RFC function must be controlled by CX pin; please refer to 2-16-9. To reset the Start Condition Flag 9 (SCF9), the PLC instruction must be used to reset the halt release request flag 6 (HRF6) or the SHE instruction must be used to reset the halt release enable flag 6 (HEF6).

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

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

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

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 (SCG) is in use. Executing the 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 the 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 section 2-2-3.

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 the SRF instruction.

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

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

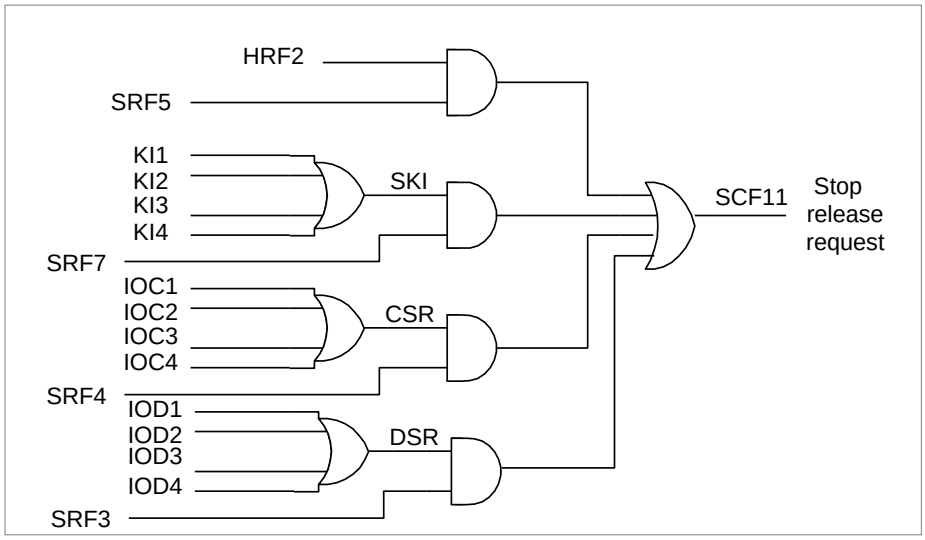
Bit 3	Bit 2	Bit 1	Bit 0
NA	The overflow flag of 16-bit counter of RFC (RFVOF)	Watchdog timer Enable flag (WDF)	System clock selection flag (CSF)
Read only	Read only	Read only	Read only

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 received from the OR-ed output via the pins defined as input mode in the IOC port, it causes the stop release flag of the IOC port (CSR) to output, the stop release enable flag 4 (SRF4) has to be set beforehand.
- A high level signal received from the OR-ed output via the pins defined as input mode in the IOD port, it causes the stop release flag of the IOD port (DSR) to output, the stop release enable flag 3 (SRF3) has to be set beforehand.
- A high level signal received from the OR-ed output of the signals latch buffer on KI1~4 pins, it causes the stop release flag of the Key Scanning (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, DSR, HRF2) are specified by the stop release enable flags (SRFx). These flags should be cleared before the chip enters stop mode. All of the pins in the IOA and IOC ports have to be set in input mode and keep in 0 state before the chip enters the STOP mode, otherwise the program can not enter STOP mode.

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

The following table shows the stop release request flags.

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

15. CONTROL REGISTER (CTL)

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

15-1. CONTROL REGISTER 1 (CTL1)

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

1. itch Enable Flag 4 (SEF4)

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

2. Switch Enable Flag 3 (SEF3)

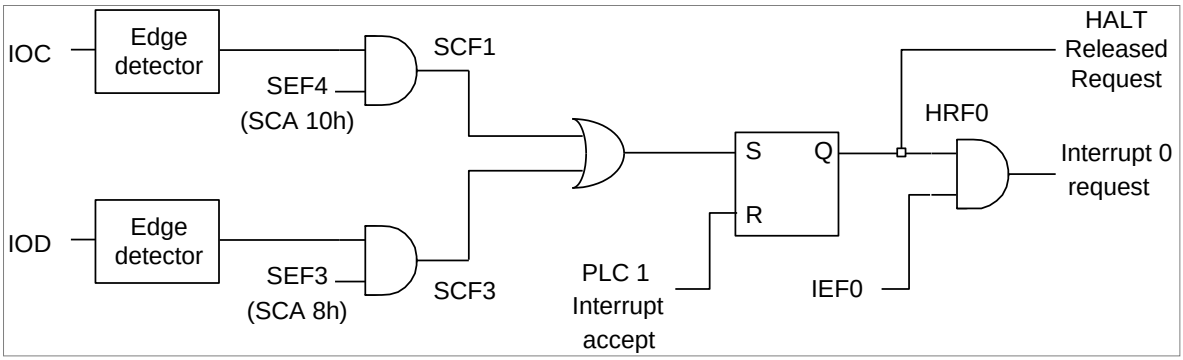
   It stores the status of the input signal change on IOD pins which have been defined as input mode that causes the halt mode or the stop mode to be released.

Executing the SCA instruction can set or reset these flags.

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

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

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



15-1-1. The Setting for Halt Mode

If SEF4 (SEF3) is set to 1, a signal change on the IOC (IOD) port will cause the halt mode to be released and SCF1 (SCF3) will be set to 1. Because the signal change on the IOC (IOD) port is an ORed output of IOC1~4, it is necessary to keep the rest of input pins in "0" state when one of the input signal on the IOC (IOD) port pin is changing.

15-1-2. The Setting for Stop Mode

If SRF4 (SRF3) and SEF4 (SEF3) are set, the stop mode will be released and set the SCF1 (SCF3) when a high level signal is applied to one of the input mode pins of IOC (IOD) port and other pins stay in "0" state.

It is applied to one of the IOC (IOD) port pins in input mode.

After the stop mode is released, TM87P18M enters the halt mode.

The high level signal must hold for a period long enough to allow the chattering prevention circuitry of the IOC (IOD) port to detect this signal and then set SCF1 (SCF3) to release the halt mode, otherwise the chip will return to stop mode again.

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 changes on the input pins of IOC (IOD) port will cause MCU to deliver the SCF1 (SCF3) when SEF4 (SEF3) has been set to 1 by executing the SCA instruction. After delivering the status of SCF1 (SCF3) flag, the halt release request flag (HRF0) will be set to 1.

In this case, if the interrupt enable flag 0 (IEF0) is set to 1 by executing the SIE instruction beforehand, it will also deliver the interrupt request flag 0 (interrupt 0) to interrupt the program.

Once the interrupt 0 is accepted by MCU, the later interrupt requests come from interrupt 0 will be inhibited until executing the SCA instruction to release this inhibition. Refer to section 2-16-1-1.

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

When the halt release enable flag 6 (HEF6) is set, the stop signal from the 16-bit counter of RFC causes the halt mode to be released. In the same manner, when HEF1 to HEF4 are set to 1, the following conditions will cause the halt mode to be released, respectively : an underflow signal from TMR1, the signal change at the INT pin, an overflow signal from the pre-divider and an underflow signal from TMR2, 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.

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 as 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 or IOD port signal to be changed (HRF0)		
Interrupt flag	Interrupt 0		

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

15-4. CONTROL REGISTER 4 (CTL4)

The Control register 4 (CTL4) is 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 (IOD)

When the stop release enable flag 7 (SRF7) is set to 1, an input signal change on the 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 will change on the IOC (IOD) port pin in input mode and a signal change on the INT pin will cause the stop mode to be released as well.

Example:

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

```
PLC     25h         ; Reset the HRF0, HRF2 and HRF5.
SHE     24h         ; Set HEF2 and HEF5, the signal change on INT or KI1~4 pin
                    ; will cause the start condition flag 4 or 8 to be set.
SCA     10h         ; Set SEF4, the signal change 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         ; Checks the signal change on KI1~4 pins that causes the stop
                    ; mode to be released.
```

16. HALT FUNCTION

The halt function is provided to minimize the current dissipation of the TM87P18M when the LCD is still operating. During halt mode, the program memory (ROM) is not in operation; only the oscillator circuit, pre-divider circuit, sound circuit, I/O port chattering prevention circuit, and LCD driver output circuit are in operation (If the timer has started operating, the timer counter still operates in the halt mode).

After executing the HALT instruction, and no halt release signals (SCF1, SCF3, HRF1 ~ 6) are delivered, the CPU enters halt mode.

The following 3 conditions are available to release halt mode

(1) An interrupt is accepted.

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

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

(2) A signal change on IOC or IOD port is specified by the SCA instruction (SCF1) or (SCF3).

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

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

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

17. BACK UP FUNCTION

TM87P18M provides a back up mode to avoid system malfunctioning under heavy loading, such as active buzzer, LED lighting, etc..., since heavy loading will cause a large voltage drop in the supply voltage, the system will malfunction under this condition.

In back up mode, the 32.768 KHz Crystal oscillator will increase the driving ability and switch the internal power (BAK pin) from VDD1 to VDD2 (Li power option only). Under this condition, all the functions in TM87P18M will work under VDD2 voltage level. It will improve the power noise immunity of TM87P18M but it also increases the power consumption.

If it is not in back up mode, the 32.768 KHz Crystal oscillator operates with a normal driving ability and the internal power (BAK pin) switches from VDD2 to VDD1 when BCF flag is cleared. In 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 0.1 uf capacitor between BAK and GND pins to regulate the internal power voltage.

Exit the back up mode anytime if it is not needed and reset the BCF flag to 0 in order to reduce the current consumption for low power applications.

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

In order to shorten the start-up time of the 32.768 KHz Crystal oscillator, TM87P18M sets the BCF to 1 during the initial reset cycle and reset BCF to 0 by executing the RF 2 instruction in Li power mode options.

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

3V battery or higher mode:

TM87P18M 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

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

18. STOP FUNCTION (STOP)

The stop function is another way to minimize the current dissipation for TM87P18M. In stop mode, all the functions in TM87P18M are put into hold state, including oscillators. All of the LCD corresponding signals (COM and Segment) will output "L" level. In this mode, TM87P18M will not dissipate any power. 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 IOD and IOC port pins which are defined as input mode must be in the "L" state, and no stop release signals (SRFn) will be delivered. The CPU will then enter stop mode by executing STOP instruction.

The following conditions will cause stop mode to be released.

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

When TM87P18M is released from stop mode, the TM87P18M will enter the halt mode immediately and will process the halt release procedure. If the "H" signal on the IOC (IOD) port does not hold long enough to set the SCF1 (SCF3), once the signal on the IOC port returns to "L", the TM87P18M will enter stop mode. The backup flag (BCF) will be set to 1 automatically after the MCU enters stop mode.

The following diagram shows the stop release procedure:

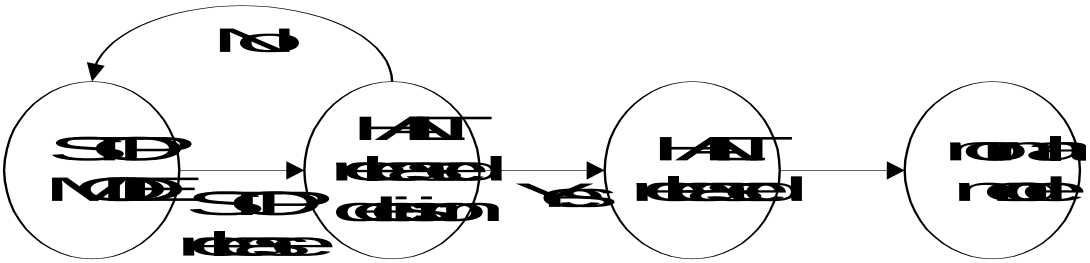


Figure: The stop release state machine

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 stop mode is released by an interrupt request, TM87P18M will enter the halt mode immediately. Once the interrupt is accepted, the halt mode will be released and then enters the interrupt service routine. The MCU will return to the stop mode again by executing the RTS instruction after the interrupt service is completed.

Once the MCU is released from the stop release, the execution of the MSB, MSC or the MCX instruction can test the halt release signals and the execution of the PLC instruction can reset the halt release signals. If the stop instruction is executed in the state that the stop release signal (SRF) is delivered, the CPU will not enter stop mode, but enter the halt mode. When stop mode is released and an interrupt is accepted, the halt release signal (HRF) is reset automatically.

Chapter 3 Control Function

1. ERRUPT FUNCTION

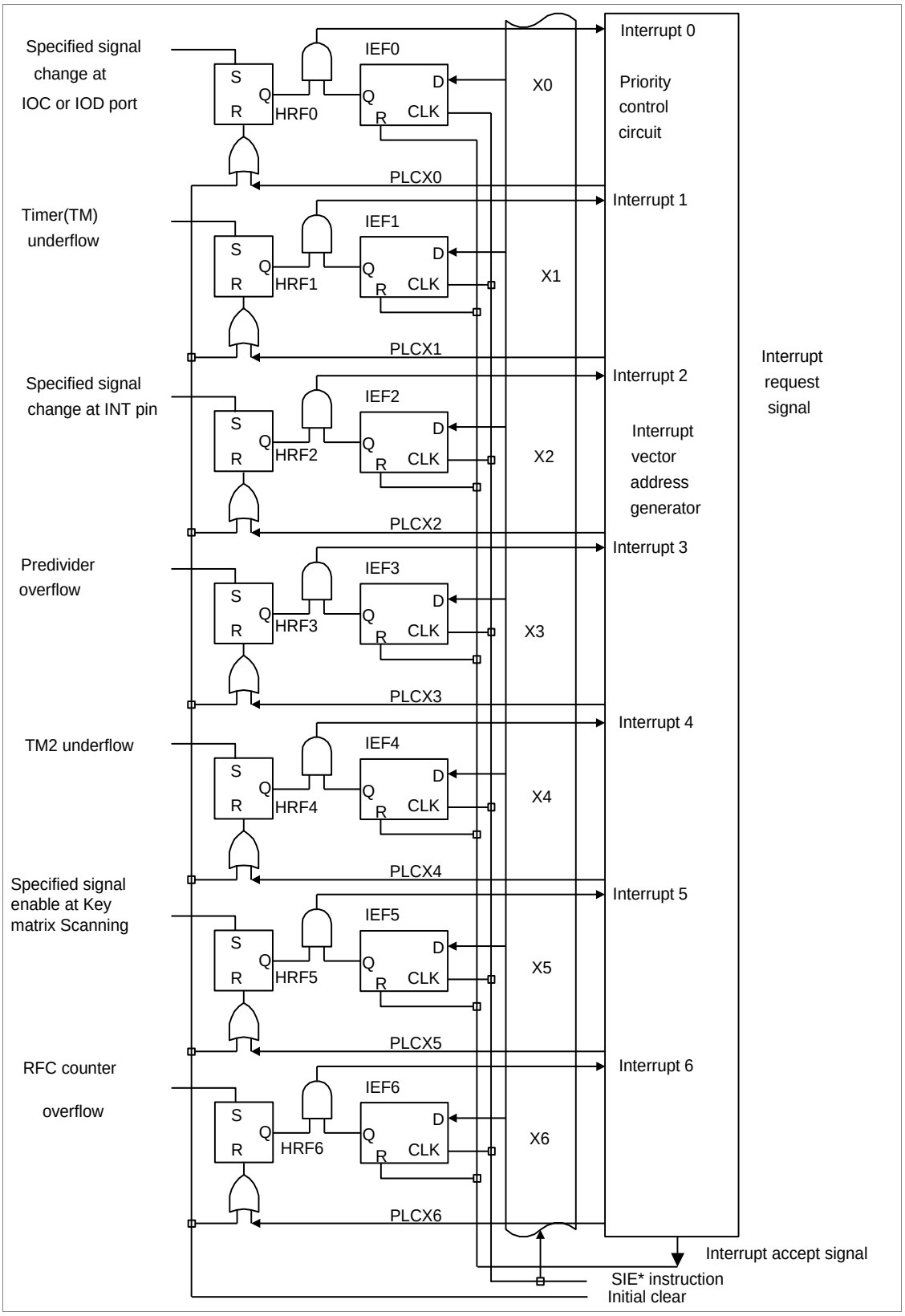
There are 7 different kinds of interrupt: 3 external interrupts and 4 internal interrupts. When an interrupt is accepted, the program in execution is suspended temporarily and the corresponding interrupt service routine specified by a pre-determined address in the program memory (ROM) will be called.

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

Table 3-1-1 Interrupt information

Interrupt source	INT pin	IOC or IOD 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	6^{th}	5^{th}	2^{nd}	1^{st}	3^{rd}	7^{th}	4^{th}
Interrupt request flag	Interrupt 2	Interrupt 0	Interrupt 1	Interrupt 3	Interrupt 4	Interrupt 5	Interrupt 6

The following figure shows the Interrupt Control Circuit



1-1. RUPT REQUEST AND SERVICE ADDRESS

1-1-1. External Interrupt Factor

The external interrupts are generated by the INT pin, the IOC or IOD ports, or Key-matrix scanning function.

(1) External INT pin interrupt request

In the mask option, 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 2. Once the interrupt request is accepted and the instruction at address 10H will be executed automatically. It is necessary to hold the signal level for at least 1 machine cycle after the signal edge changes.

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

An interrupt request signal (HRF0) will be generated when an input signal changes on the I/O port IOC (IOD) 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 interrupt request.

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

1-1-2. Internal Interrupt Factor

The internal interrupts are generated by Timer 1 (TMR1), Timer 2 (TMR2), RFC counter and the pre-divider.

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

An interrupt request signal (HRF1/4) is generated when Timer1/2 (TMR1/2) underflows. In this case, if the interrupt enable flag 1/4 (IEF1/4) is set beforehand and interrupt 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) is 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 the 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.

1-2. INTERRUPT PRIORITY

If all interrupts are requested simultaneously during a state when all interrupts are enabled, the pre-divider interrupt is given the first priority and other interrupts are held. When the interrupt service routine is initiated, all of the Interrupt Enable Flags (IEF0 ~ IEF6) are cleared and should be set with the next execution of the SIE instruction. Refer to Table 3-1.

Example:

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

; beforehand, all the IOC port pins are been defined as input mode.

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

;……..               ; All interrupts are requested simultaneously.

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

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

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

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

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

SIE*   65h          ; Enable the interrupt request (except the predivider, TMR1
                    ; and TMR2).
                    ; An interrupt caused by RFC counter overflow occurs, and
                    ; interrupt service is concluded.

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

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

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

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

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

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

1-3. INTERRUPT SERVICING

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

(1) The value of the program counter (PC) right before the interrupt service begins is saved on the stack register (STACK).

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

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

When an interrupt occurs, the TM87P18M 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 insert a NOP instruction cycle.
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
```

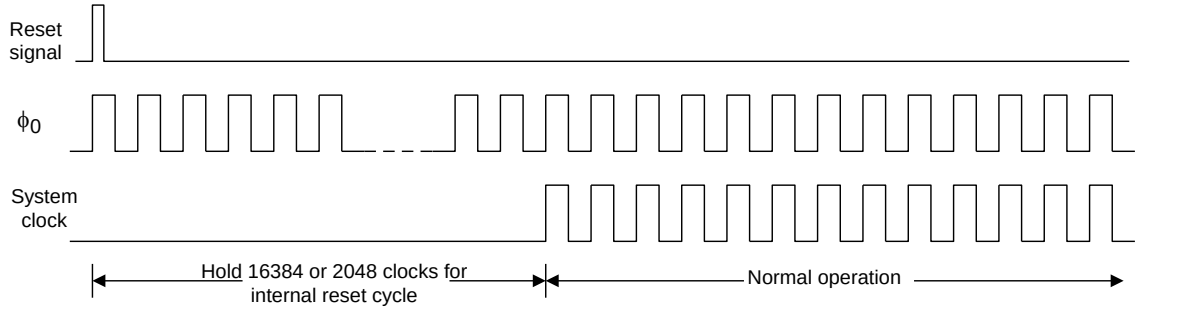
Note: If instruction 1 is the "halt" instruction, the MCU will return to "halt" mode after interrupt.

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.

2. RESET FUNCTION

TM87P18M contains four reset sources: power-on reset, RESET pin reset, IOC port reset and watchdog timer reset.

When reset signal is accepted, TM87P18M will generate a time period for internal reset cycle and there are two types of internal reset cycle time could be selected by mask option, the one is PH15/2 and the other is PH12/2.



Internal reset cycle time is PH15/2

MASK OPTION table:

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

In this option, the reset cycle time will be extended 16384 clocks

Internal reset cycle time is PH12/2

MASK OPTION table:

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

In this option, the reset cycle time will be extended 2048 clocks

2-1. POWER ON RESET

TM87P18M 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.

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

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.

There are two types of reset mode can be set for the RESET pin in mask option. One is level reset and the other is pulse reset.

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

2-2-1. Level Reset

Once an "H" signal is applied on the RESET pin, TM87P18M will not enter the initial reset cycle until the signal on the RESET pin is return to "0". Once the signal applied on the reset pin returns to 0, TM87P18M launches the initial reset cycle immediately.

MASK OPTION table:

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

2-2-2. Pulse Reset

Once a "1" signal is applied on the RESET pin, TM87P18M will escape from reset state and begin the normal operation after internal reset cycle automatically no matter whether the signal on RESET pin returns to "0" or not.

MASK OPTION table:

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

The following table shows the initial condition of TM87P18M in reset cycle.

Program counter	(PC)	Address 000H
Start condition flags 1 to 7	(SCF1-7)	0
Backup flag	(BCF)	1 (Li-B option)
Stop release enable flags 4,5,7	(SRF3,4,5,7)	0
Switch enable flags 4	(SEF3,4)	0
Halt release request flag	(HRF 0~6)	0
Halt release enable flags 1 to 3	(HEF1-6)	0
Interrupt enable flags 0 to 3	(IEF0-6)	0
Alarm output	(ALARM)	DC 0
Pull-down flags in I/OC, I/OD port		1(with pull-down resistor)
Input/output ports I/OA, I/OB, I/OC, I/OD	(PORT I/OA, I/OB, I/OC, I/OD)	Input mode
I/OC, I/OD port chattering clock	Cch	PH10*
Frequency generator clock source and duty cycle	Cfq	PH0, duty cycle is 1/4, output is inactive
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: the 3rd output of the predivider**

2. PH10: the 10th output of predivider

3. All the LCD segment pins can be set to output all-ON or all-OFF signals during reset cycle in mask option

2-2-3. IOC Port/Key Matrix RESET

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 the the '0' signal is applied to all the input pins, a reset signal is delivered (The key-matrix scanning function will not deliver the reset signal until the scanning clock signal arrives).

MASK OPTION table:

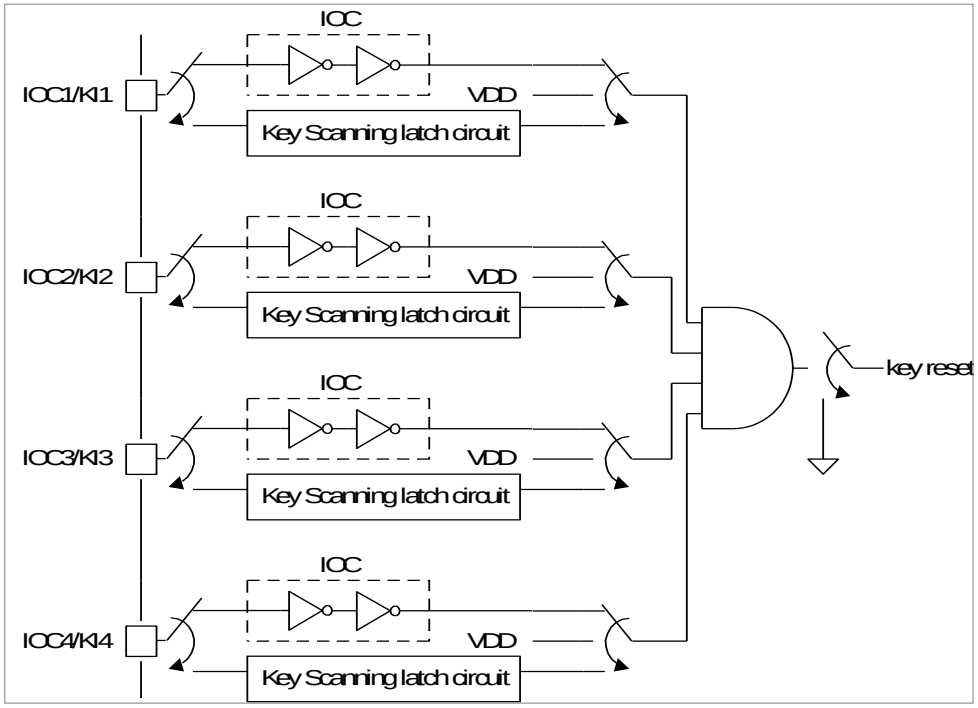
IOC or KI pins are used as key reset :

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

IOC or KI pins are not used as key reset:

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

The following figure shows the key reset diagram.

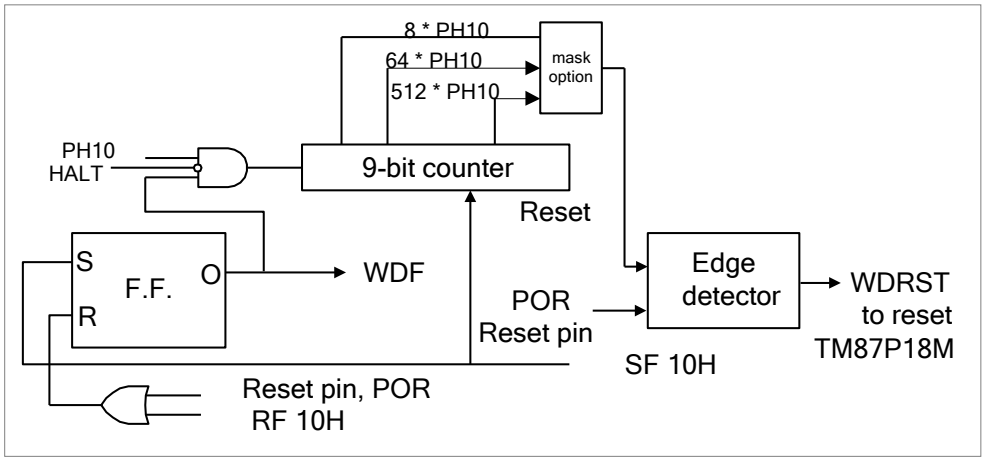


2-2-4. WATCHDOG RESET

The watchdog timer is used to detect unexpected execution sequences caused by 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 TM87P18M. Most of the functions in TM87P18M 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 diagram.



During initial reset (power on reset [POR] 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 watchdog timer will pause when the program enters the halt or the stop mode. When the TM87P18M 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 keep the MCU away from the unexpected reset when it is released 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; it will prevent the watchdog timer from overflow.

The overflow time interval of the 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

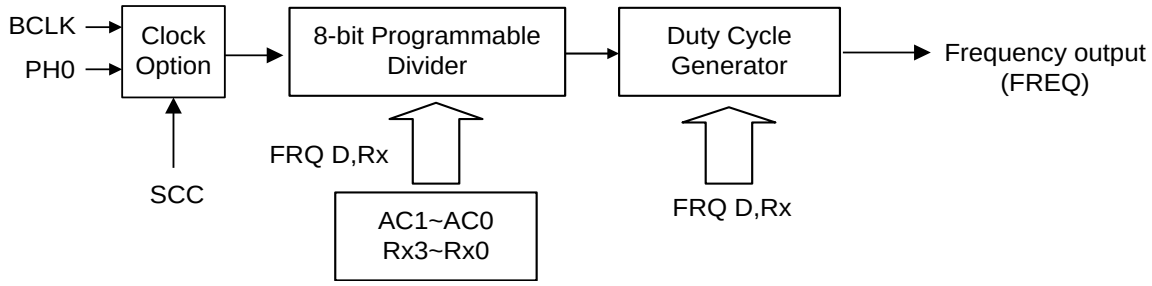
Note: timer overflow time interval is about 16 seconds when PH0 = 32.768 KHz

3. OCK GENERATOR

3-1. REQUENCY GENERATOR

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

The following shows the organization of the frequency generator.



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

The FRQ related instructions preset a scaling data N for the programming divider and a data D for setting the duty cycle, and then the frequency generator starts to output 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 letter N							
Programming divider	bit7	Bit6	bit 5	bit 4	bit 3	Bit 2	bit 1	bit 0
FRQ D,Rx	AC3	C2	AC1	AC0	Rx3	Rx2	Rx1	Rx0
FRQ D,@HL	T7	T6	T5	T4	T3	T2	T1	T0
FRQX D,X	X7	X6	X5	X4	X3	X2	X1	X0

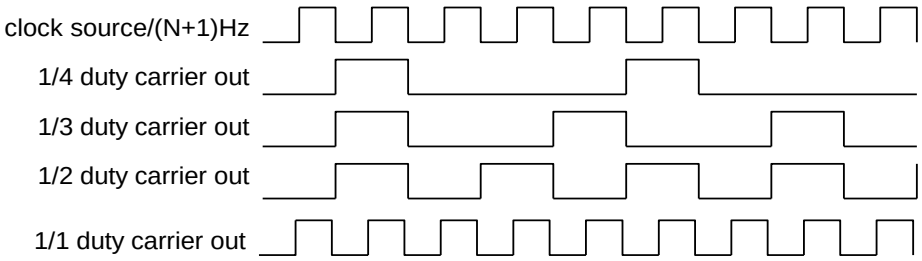
Notes: 1. T0 ~ T7 represents the data of table ROM.

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

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

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

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



3-2. Melody APPLICATION

The frequency generator may generate specified frequencies to compose melody music and the note table for those specified frequencies 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 = 65536 Hz, C3 - B5 may have more accurate frequency.

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

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, and PH5. When using Li battery power supply, halver circuitry generates a 1/2 VDD voltage for suppling the MCU's functoins which is not related to the input/output operation.

3-4. Alternating Frequency for LCD

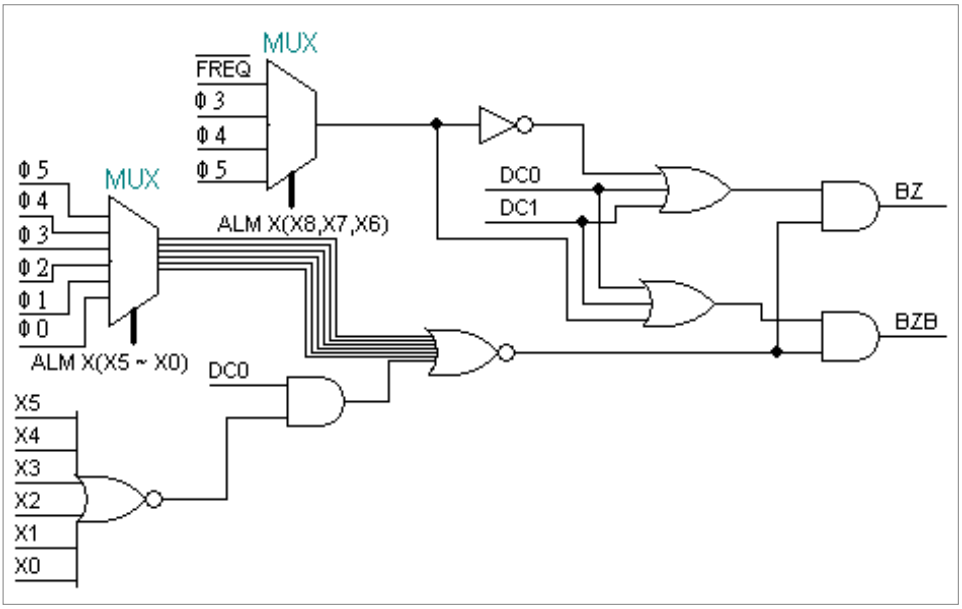
The alternating clock is the basic clock for LCD driver. Both COM and SEG pins shall change their output waveforms according to the alternating clock.

4. BUZZER OUTPUT PINS

TM87P18M provides a pair of buzzer output pins known as BZB and BZ, which are pin-shared with I/O pins, IOB3 and IOB4, and can be 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
DC31/IOB4/BZ	(3) BZ



This figure shows the organization of the buzzer output.

4-1. SOUND EFFECT APPLICATION

The buzzer output pins (BZ, BZB) are suitable for driving the buzzer through a transistor with one output pin or driving the buzzer with both BZ and BZB pins directly. It is capable of outputting a modulation waveform of any combination of the frequency generator's output signal, PH3 (1024 Hz), PH4 (2048 Hz), PH5 (1024 Hz) as the carrier, and with the envelope waveform of any combination of the following frequencies: 32 Hz (PH10), 16 Hz (PH11), 8 Hz (PH12), 4 Hz (PH13), 2 Hz (PH14), 1 Hz (PH15). Execute the ALM instruction to specify the frequency combination for the output waveform.

Note:

1. The higher frequency clock source should be only one of PH3, PH4, PH5 or FREQ, and the lower frequency may be any/all of the combinations from any/all of PH10 ~ PH15.

2. The frequency in parentheses corresponding to the input clock of the pre-divider (PH0) is 32768 Hz.

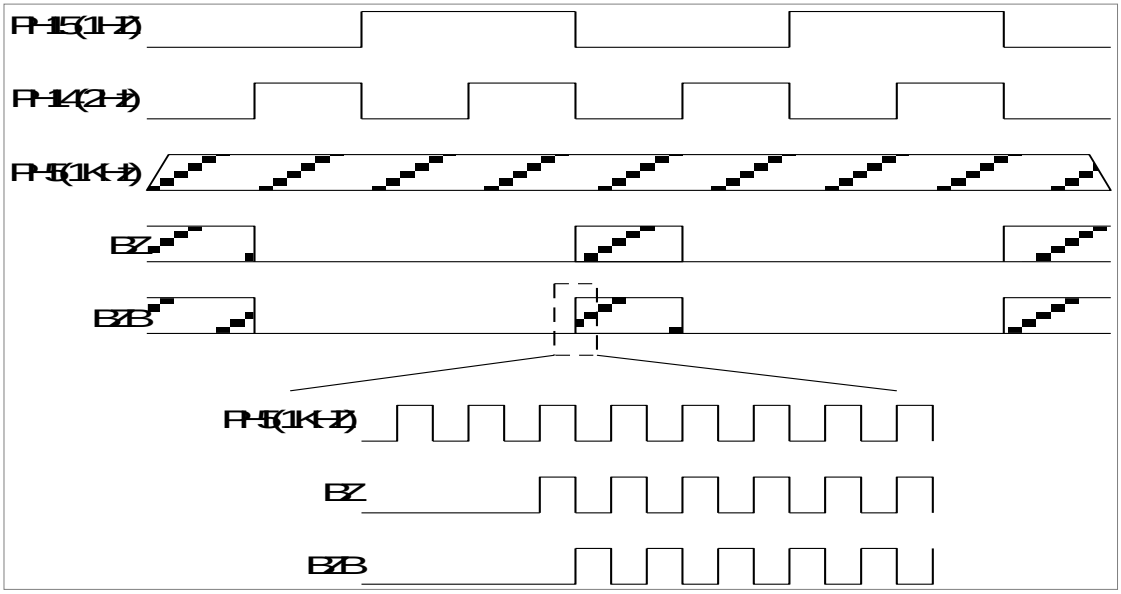
3. The BZ and BZB pins will output DC0 after the initial reset cycle.

Example:

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

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

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



4-2. REMOTE CONTROLLER APPLICATION

If the buzzer output is combined 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 after the FRQ related instructions in order to deliver the FREQ signal to the BZ pin.

Example:

```
SHE     1       ; Enable timer 1 halt release enable flag.
TMSX    3Fh     ; Set initial value of Timer 1 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 output. This instruction must be executed
                ; after the FRQ related instructions.
HALT            ; Waiting for the halt release (Timer 1 underflows).
                ; Halt released.
ALM     0       ; Stop the buzzer output.
```

5. INPUT / OUTPUT PORTS

Four I/O ports are available in TM87P18M: IOA, IOB, IOC and IOD. Each I/O port has the same basic function and consists of 4 bits.

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

5-1. OA PORT

IOA1 ~ IOA4 pins are MUX with CX / SEG24, RR / SEG25, RT / SEG26 and RH / SEG27 pins respectively by 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 IOA port is input mode in initial reset cycle, each bit of the port can be defined as 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 the pins which have been defined as output mode.

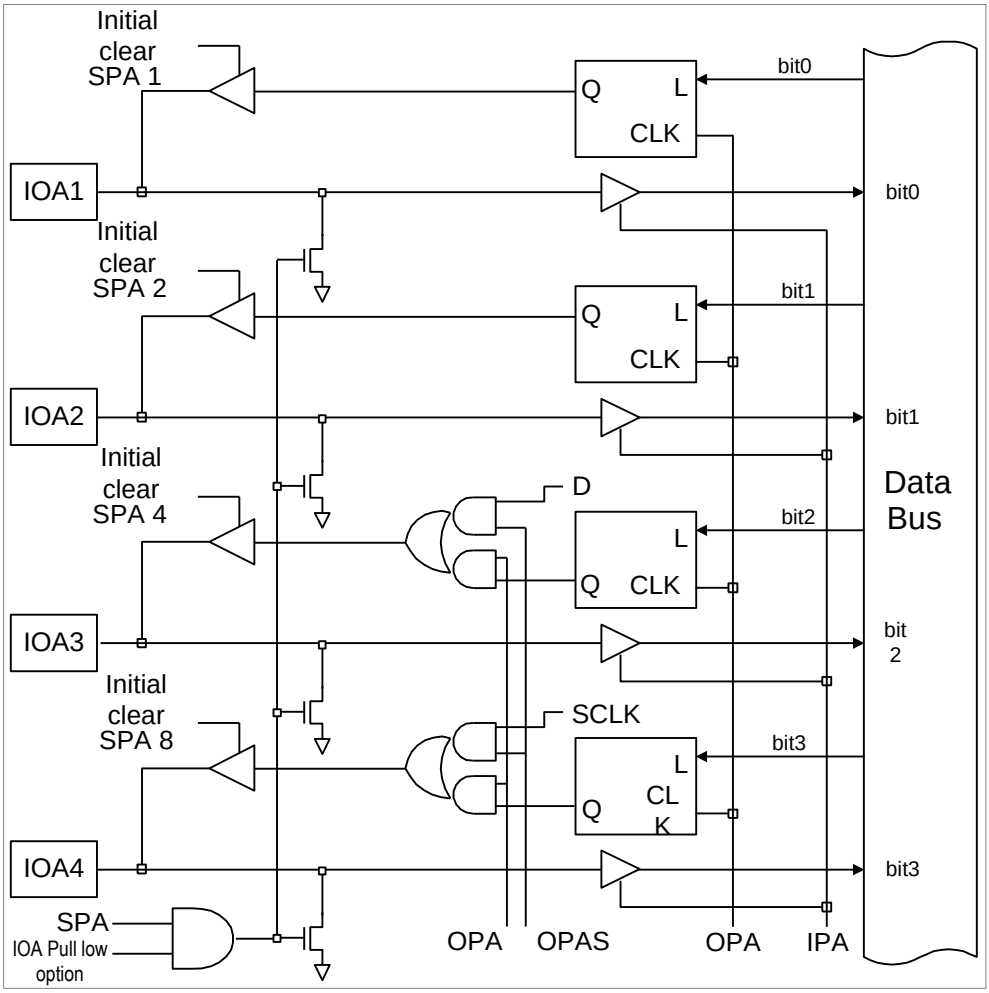
Executing an IPA instruction can store the I/O pins' signal into the specified data memory locations. When the IO pins are defined as output mode, executing an IPA instruction will store the content of the latch of the output pin into the specified data memory location.

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

The IOA port has a built-in pull-low resistor which can be selected in mask option and 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



This figure shows the organization of IOA port.

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

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.

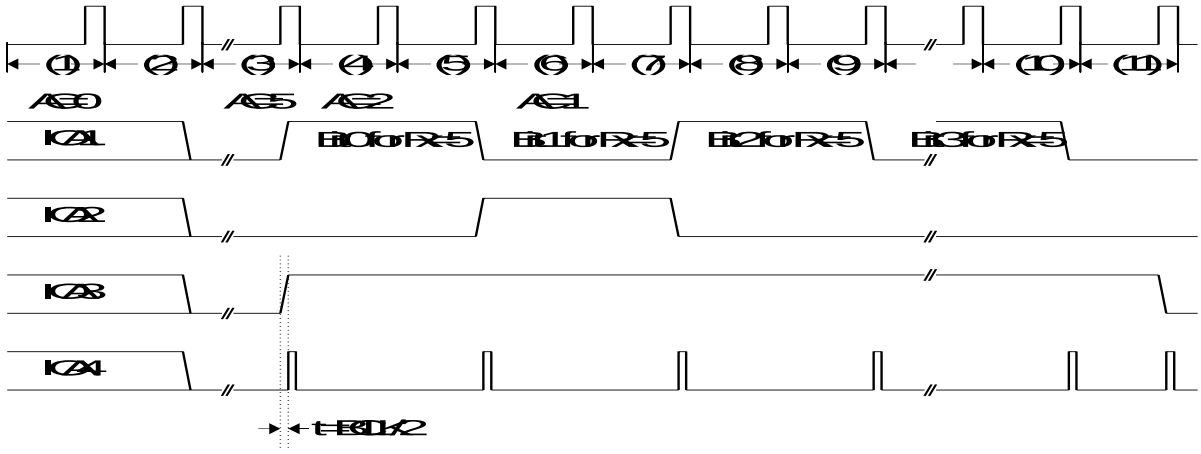
1. BIT0 and BIT1 of the port deliver RAM data.
2. BIT2 of the port delivers the constant data (D) in operand.
3. BIT3 of the port delivers a pulse.

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

(1) LDS 0AH, 0
(2) OPA 0AH
SPA 0FH

```
        :
        :
        LDS     1,5
(3)     OPAS    1,1     ; Bit 0 output, enable the serial output function
(4)     SR0     1       ; Shift bit 1 to bit 0
(5)     OPAS    1,1     ; Bit 1 output
(6)     SR0     1       ; Shift bit2 to bit 0
(7)     OPAS    1,1     ; Bit 2 output
(8)     SR0     1       ; Shift 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     ; Inactive the serial output function
```

The above program is illustrated by the timing chart below:



If the IOA1 pin is defined as the CX pin for the RFC function and the other pins (IOA2 ~ IOA3) are used as normal IO pins in mask option, the IOA1 function must be set as output mode in the begining of program to prevent the signal change on the CX pin getting into the IOA1 function within input mode. On the other hand, the IOA1 function cannot change the output signal within output mode because the output signal of IOA1 function will affect the counting of RFC counter through the CX pin when the RFC function is enabled.

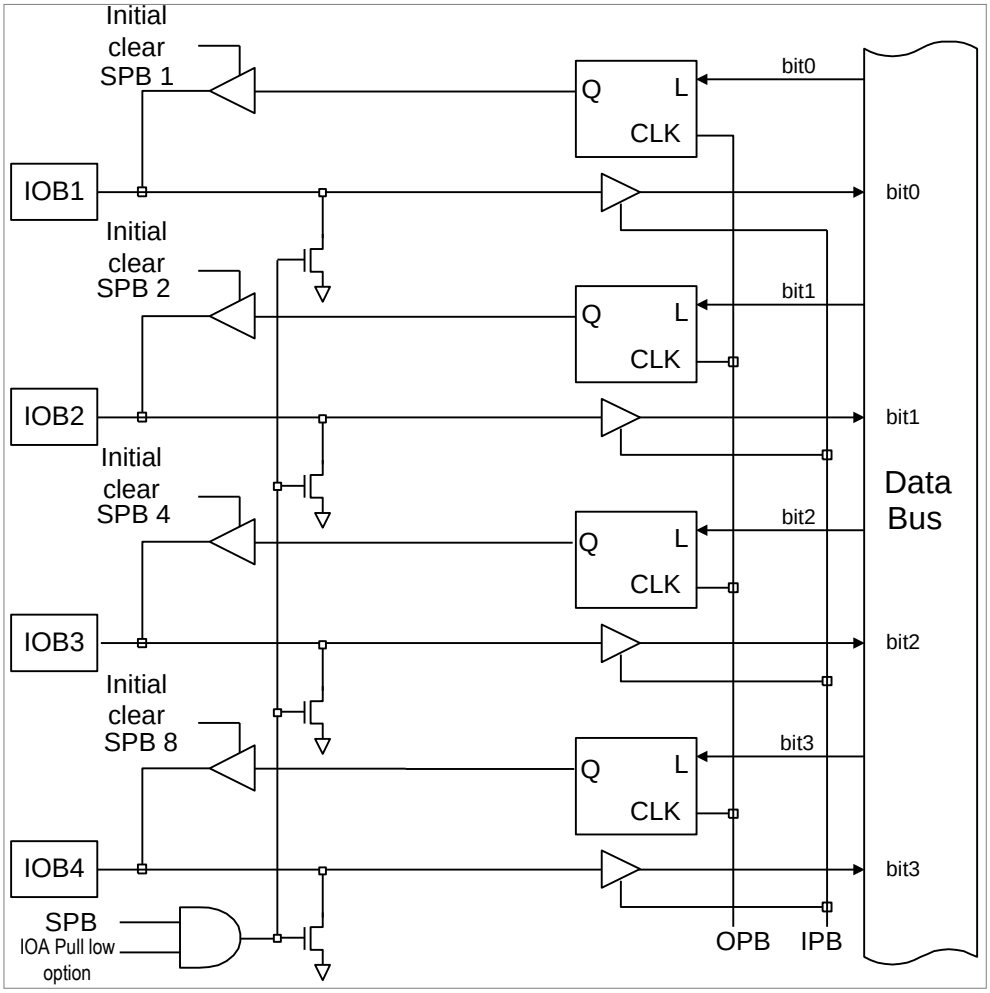
5-2. IOB PORT

IOB1 ~ IOB4 pins are MUXed with SEG28, SEG29, BZB / SEG30 and BZ / DC31 pins respectively by mask option.

MASK OPTION table:

Mask Option name	Selected item
SEG28/IOB1	(2) IOB1
SEG29/IOB2	(2) IOB2
SEG30/IOB3/BZB	(2) IOB3
DC31/IOB4/BZ	(2) IOB4

The following figure shows the organization of IOB port.



Note: The pins in the input mode should not be in floating, or a large current (straight-through current) will flow into the input buffer.

The default setting of the IOB port is input mode in the initial reset cycle, each bit of the port can be defined as input mode or output mode respectively by executing a SPB instruction. Executing an OPB

instruction can output the content of specified data memory to those pins which have 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, the OPB 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 changing the I/O mode.

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

5-3. IOC PORT

IOC1 ~ IOC4 pins are MUXed with KI1, KI2, KI3 and KI4 pins respectively by mask option.

MASK OPTION table:

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

The default setting of IOC port is input mode in the initial reset cycle, each bit of the port can be defined as input mode or output mode respectively by executing a SPC instruction. Executing an OPC instruction can output the content of specified data memory to the pins which has 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.

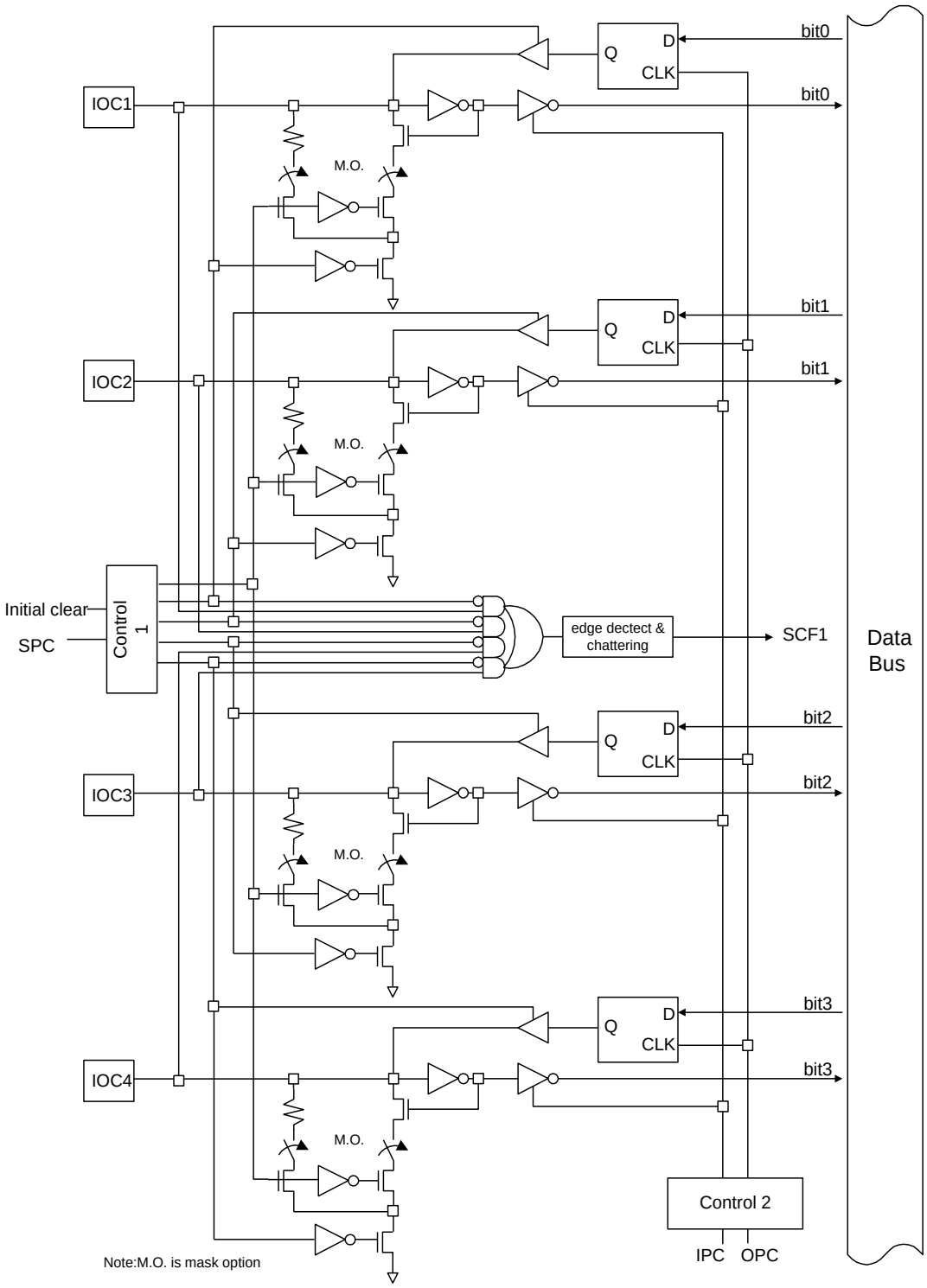
Before changing the I/O pins to output mode, the OPC instruction must be executed first to output the data to those output latches. It will prevent the chattering signal on the I/O pin when changing the I/O mode.

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.

The IOC port can select the pull-low device or low-level hold device for each pin in mask option and can be enabled/disabled by the software program. When the pull-low device and the low-level hold device are both enabled in mask option, a reset will enable the pull-low device and disable the low-level hold

device. Executing the SPC 10h instruction can also enable the pull-low device and disable the low-level hold device. Executing the SPC 0h instruction will disable the pull-low device and enable the low-level hold device.

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



This figure shows the organization of IOC port.

Note: The pins in input mode should not be in floating, or 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 the pull-low resistor.

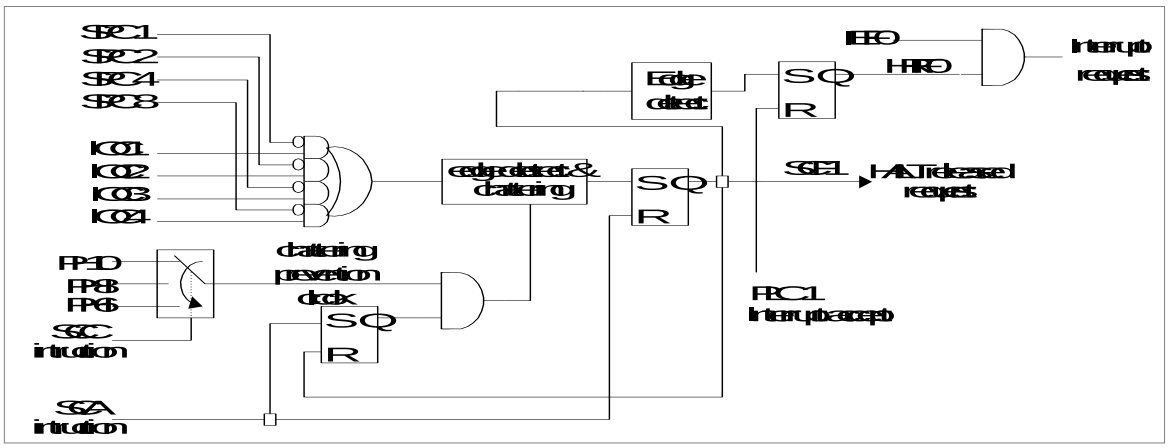
The Low-level-hold function option

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

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 (32 ms), PH8 (8 ms) or PH6 (2 ms) 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 (e.g. IOC1) changes from “L” level to “H” level or from “H” level to “L” level and the remaining pins (e.g. IOC2 to IOC4) are held at “L” level.

When the signal changes on the IOC port pins in input mode specified by the SCA instruction and stays in that state for at least two chattering clock (PH6, PH8, PH10) cycles, the control circuit that operates upon the input pins will transmit a halt release request signal (SCF1). At that time, the chattering prevention clock will stop due to the transmission of SCF1. SCF1 will 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 Flag (IEF0) of the port IOC is set, the interrupt will be accepted.

Since the IOC port is not available to hold the information of the signal on the input pins of IOC1 to IOC4, the input data on the port IOC should be read into the RAM immediately after the halt mode is released.

5-4. IOD PORT

The default setting of IOD port is input mode in the initial reset cycle, each bit of the port can be defined as input mode or output mode respectively by executing a SPD instruction. Executing an OPD instruction can output the content of specified data memory to the pins which has been defined as output mode.

Executing IPD instructions can store the signals applied to the IOD pins into the specified data memory. When the IOD pins are defined as output mode, executing an IPD instruction will store the data that stored in the output latches into the specified data memory.

Before changing the I/O pins to output mode, the OPD 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 changing the I/O mode.

IOD port has a built-in pull-low resistor for each pin which can be selected in mask option and can be enabled or disabled this resistor by executing a SPD instruction.

When the IOD pin is set to the output mode, the pull-low device will be disabled.

MASK OPTION table:

Pull-low function option

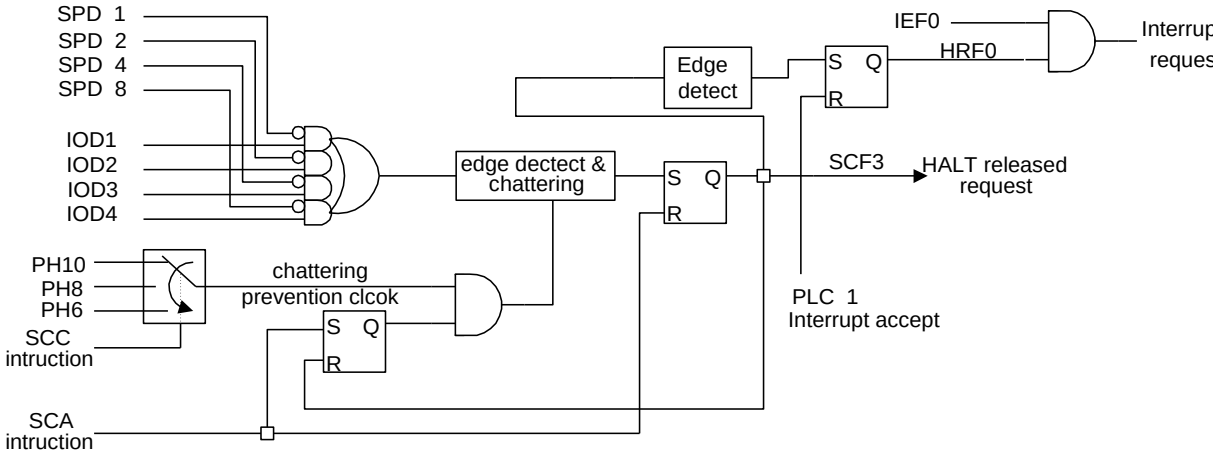
Mask Option name	Selected item
IOC PULL LOW RESISTOR	(1) USE
IOC PULL LOW RESISTOR	(2) NO USE

This figure shows the organization of IOD port.

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

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

The port IOD is capable of preventing the chattering signals (bounce) applied on the IOD1 to IOD4 pins. The de-bounce time can be selected as PH10 (32 ms), PH8 (8 ms) or PH6 (2 ms) by executing a SCC instruction. The default selection is PH10 after the reset cycle. The following figure shows the organization of chattering prevention circuitry.



This figure shows the organization of chattering prevention circuitry.

Note: The default prevention clock is PH10

The chattering prevention function will be invoked when the signal on the applicable pin (e.g. IOD1) changes from "L" level to "H" level or from "H" level to "L" level and the remaining pins (e.g. IOD2 to IOD4) are held at "L" level.

When the signal changes on the IOD port pins in input mode specified by the SCA instruction and stays in the state for at least two chattering clock (PH6, PH8, and PH10) cycles, the control circuit that operates upon the input pins will transmit the halt release request signal (SCF3). At that time, the chattering prevention clock will stop due to the transmission of SCF3. The SCF3 can be reset to 0 by executing a SCA instruction and the chattering prevention clock will be enabled at the same time. If the SCF3 has been set to 1, the halt release request flag 0 (HRF0) will be generated. In this case, if the interrupt enable mode (IEF0) of the port IOD is set, the interrupt will be accepted.

Since no flip-flop is available to hold the information of the signal on the input pins of IOD1 to IOD4, the input data on the port IOD should be stored into the RAM immediately after the halt mode is released.

6. EXTERNAL INT PIN

There are three kinds of input type can be selected in mask option for the INT pin: pull-up, pull-down, and high impedance. A signal change (either rising edge or falling edge in mask option) will set the halt release request flag 2 (HRF2). In this case, if the halt release enable flag (HEF2) is set, the start condition flag 2 will be set and a corresponding signal is delivered. If the INT pin Interrupt Enable Flag (IEF2) is set, the interrupt will be accepted.

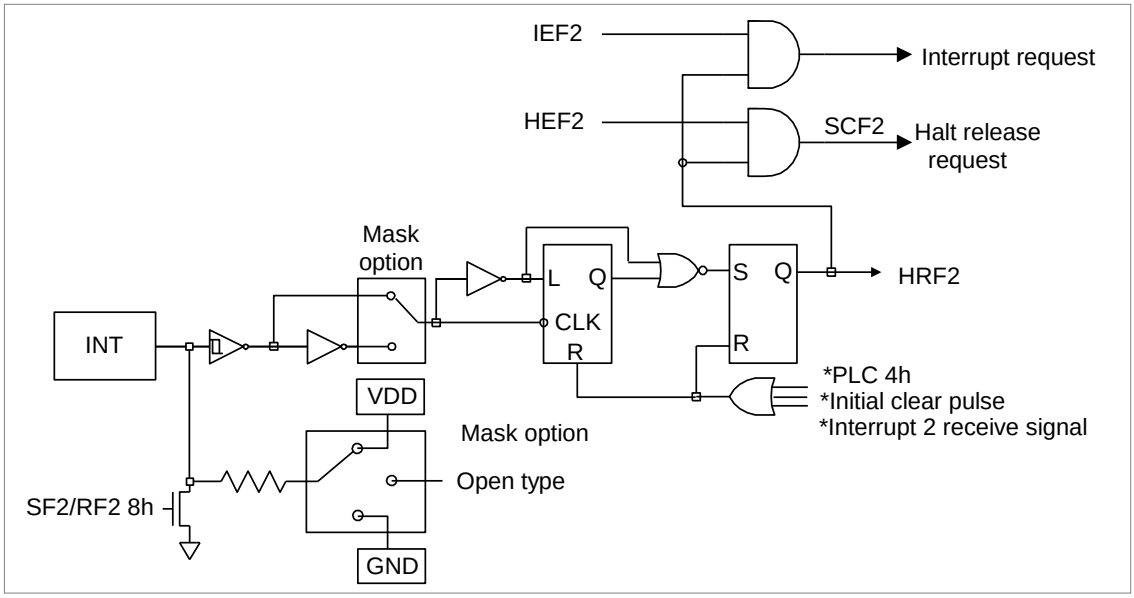
MASK OPTION table:

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

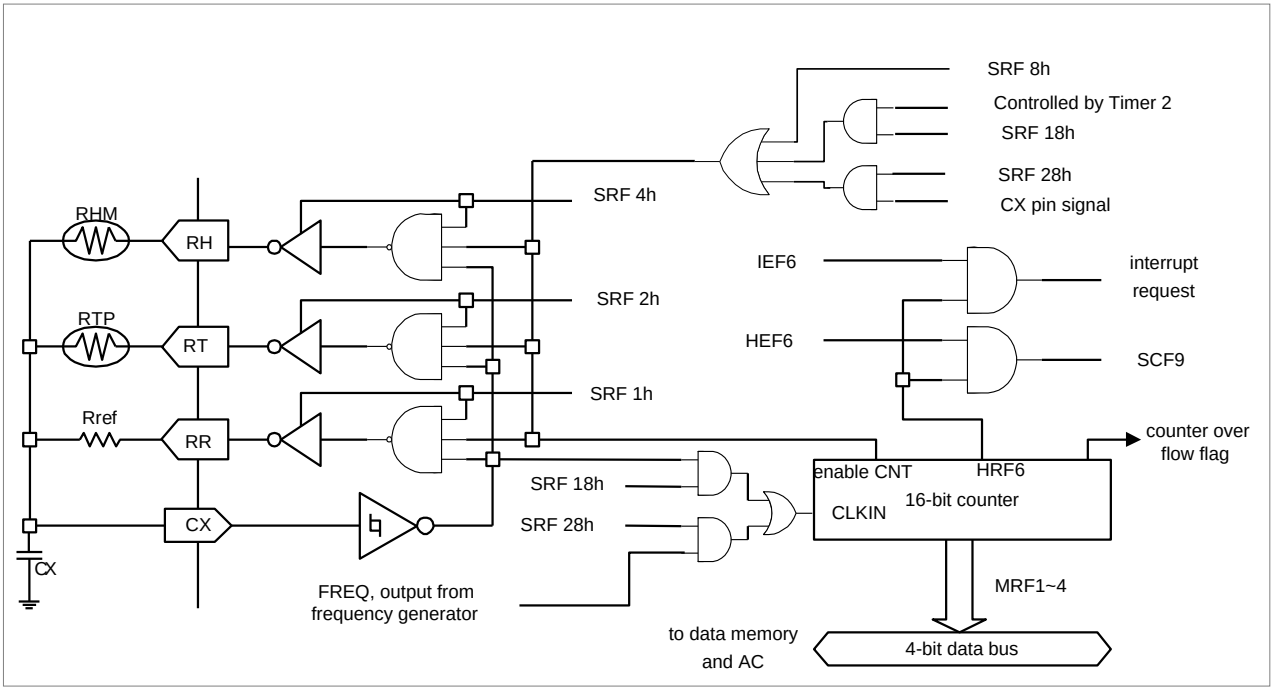


This figure shows the INT Pin Configuration

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

7. Resistor to Frequency Converter (RFC)

The Resistor to Frequency Converter (RFC) converts a specified resistance to a corresponding frequency.



This figure shows the block diagram of RFC.

Figure shows the block diagram of RFC.

RFC contains four external pins:

CX: the oscillation Schmitt trigger input pin

RR: the reference resistor 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 table:

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

7-1. RC Oscillation Network

The RFC circuitry can build up to 3 RC oscillation networks by connecting sensors or resistors between CX and one of RR, RT, or RH and CX pins. Only one RC oscillation network can be active at a time. When the oscillation network is built up (by executing SRF 1h, SRF 2h, and SRF 4h instructions to enable RR, RT, and RH networks, respectively), a clock signal with specified frequency corresponding to the resistance will be generated and counted by the 16-bit counter through the CX pin as the clock source.

How to build up the RC oscillation network:

1. Connect the resistor and capacitor on the RR, RT, RH and CX pins. 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 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 active until these instructions are executed. The output pin of RC oscillation network (one of the RR, RT, and RH pins) will output an "L" state before this 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 17th bit) for the 16-bit counter. This bit is the overflow flag (RFOVF) which can be checked a by MSD instruction, the 16-bit counter will stop counting 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 flag is only used as the 17th bit of the counter. There are 3 operation modes for the 16-bit counter. Each mode is described in the following sections:

7-2. Enable/Disable the Counter by Software

In this mode, the clock source of the 16-bit counter is received 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 will start to count the clocks 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 intend to count the number of clock from the CX pin for a time period, you can enable the 16-bit counter by executing a SRF 8 instruction and setting the timer1 to control the time period. The overflow flag (RFOVF) of the 16-bit counter will be checked during the time period. If the overflow flag is not set to 1, read the content of the counter directly; if the overflow flag has been

set to 1, the program is required 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.

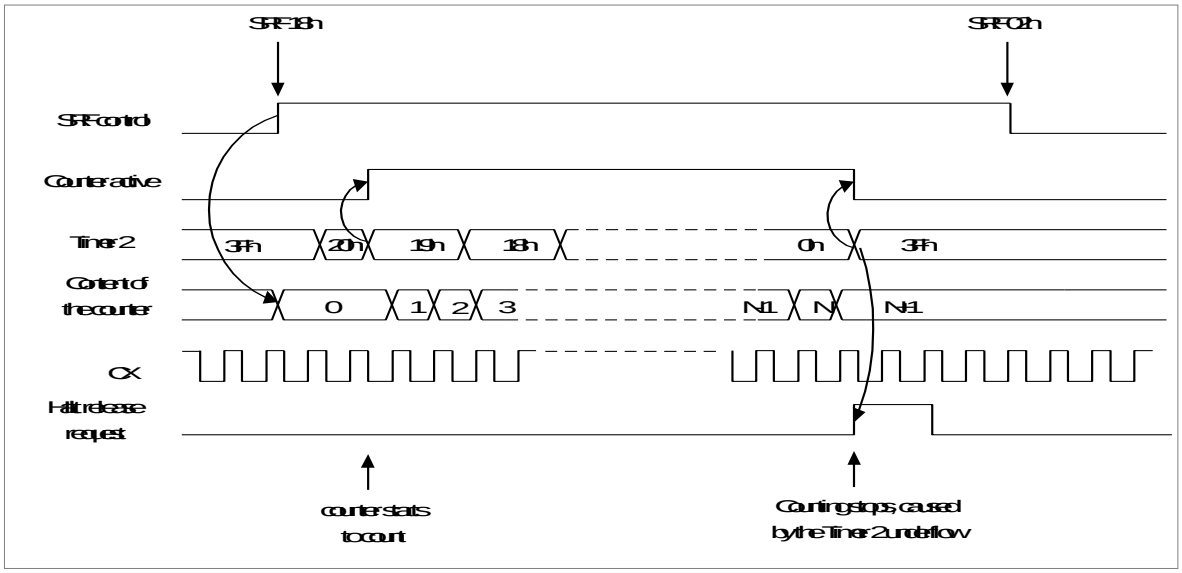
```
; Timer 1 is used to enable/disable the counter
        LDS     0, 0            ; Set the TMR1 clock source (PH9)
        LDS     1, 3            ; Initiate TMR1 setting value to 3F
        LDS     2, 0Fh
        SHE     2               ; Enable halt release by TMR1
RE_CNT:
        LDA     0
        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
```

7-3. Enable / Disable the Counter by Timer 2

In this mode, the clock source of the 16-bit counter is received from the CX pin and the 16-bit counter 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 falling edge of the clock has applied on the clock source 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 the 16-bit counter. For a detail description of the operation of TMR2, please refer to 2-13.

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



This 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 enables 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
```

7-4. Enable / Disable the Counter by CX Signal

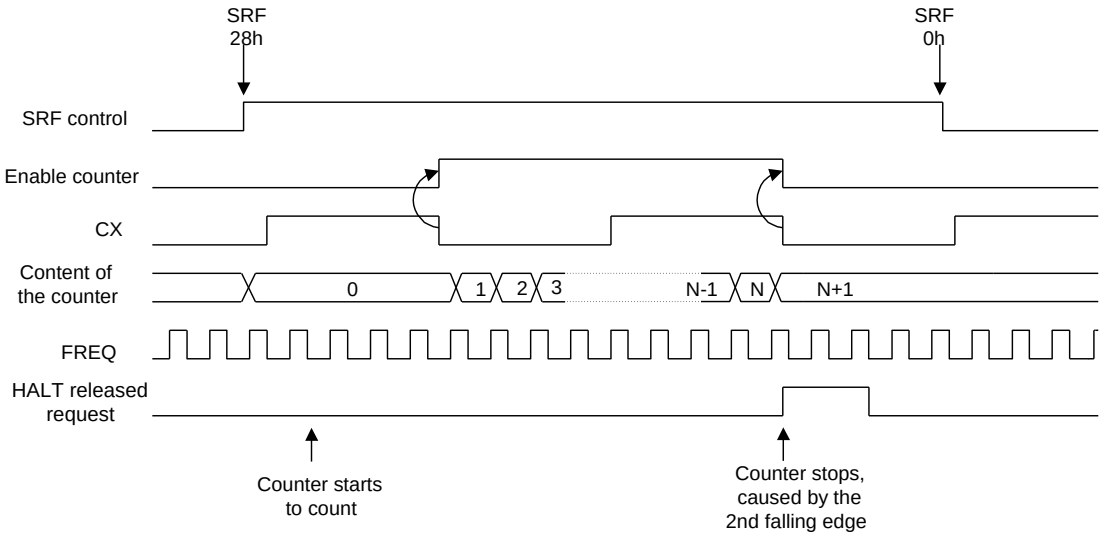
This is another usage for the 16-bit counter but it is not related to the RFC function. In the applications described in the previous section, the CX clock is used as the clock source for the 16-bit counter using the S/W or TMR2 to produce a time period to control the counter.

In this mode, however, the 16-bit counter operates differently.

The clock signal on the CX pin turns into the controlled signal to enable/disable the 16-bit counter and the clock source of the 16-bit counter coming from the output of the frequency generator (FREQ).

When the 16-bit counter is enabled, it will count the clock (FREQ) after the first rising edge signal applies to the CX pin. Once the second rising edge applies to the CX pin after the counter is enabled, 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 the status register 4 (STS4).

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



This figure 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 = (PH0/3) / 5
                ; the count value of the frequency generator is 5 and
                ; CK FREQ is 1/3 duty waveform.

                ; The setting value of the frequency generator is 5 and FREQ
```

```
                    ; has a 1/3 duty waveform.

SHE     40h         ; Enable the halt release caused by 16-bit counter
SRF     28h         ; Enable the counter controlled by the CX signal
HALT
PLC     40h         ; A halt release request is caused by the 2nd rising edge on CX
                    ; pin, and then clear the halt release request flag
MRF1    10h         ; Read the content of the counter
MRF2    11h
MRF3    12h
MRF4    13h
```

8. Key Matrix Scanning

The key matrix scanning function is made up of the 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.

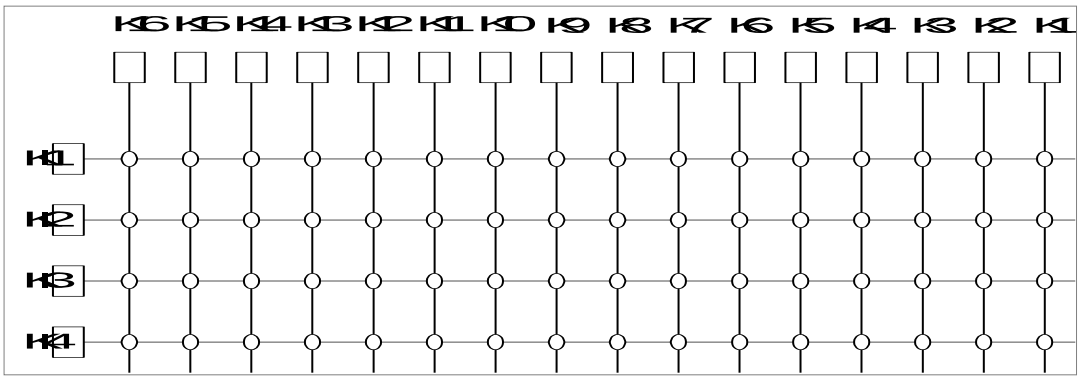
The input port of the key matrix circuitry is composed of KI1 ~ KI4 pins (these pins are muxed with IOC1 ~ IOC4 pins and selected in mask option).

MASK OPTION table:

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

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

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



This figure shows the key matrix.

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

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

X_6 = "0", when HEF5 is set to 1, the HALT release request (HRF5) will be set to 1 after the key is depressed on the key matrix, and then SCF8 will be set to 1.

"1", when HEF5 is set to 1, the HALT released request (HRF5) will be set to 1 after each scanning cycle regardless of key is depressed, and then SCF8 will be set to 1.

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

X_3 ~ X_0 = 0000, activates K1 column

X_3 ~ X_0 = 0001, activates K2 column

……..

X_3 ~ X_0 = 1110, activates K15 column

X_3 ~ X_0 = 1111, activates K16 column

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

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

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

X_3 = 0, activates K1 ~ K8 columns simultaneously

X_3 = 1, activates K9 ~ K16 columns simultaneously

X_2 ~ X_0 don't care.

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

X_3X_2 = 00, activates K1 ~ K4 columns simultaneously

X_3X_2 = 01, activates K5 ~ K8 columns simultaneously

X_3X_2 = 10, activates K9 ~ K12 columns simultaneously

X_3X_2 = 11, activates K13 ~ K16 columns simultaneously

X_1, X_0 don't care.

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

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

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

…….

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

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

X_0 is not a factor.

When KI1~4 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 to output mode before the key matrix scanning function is activated.

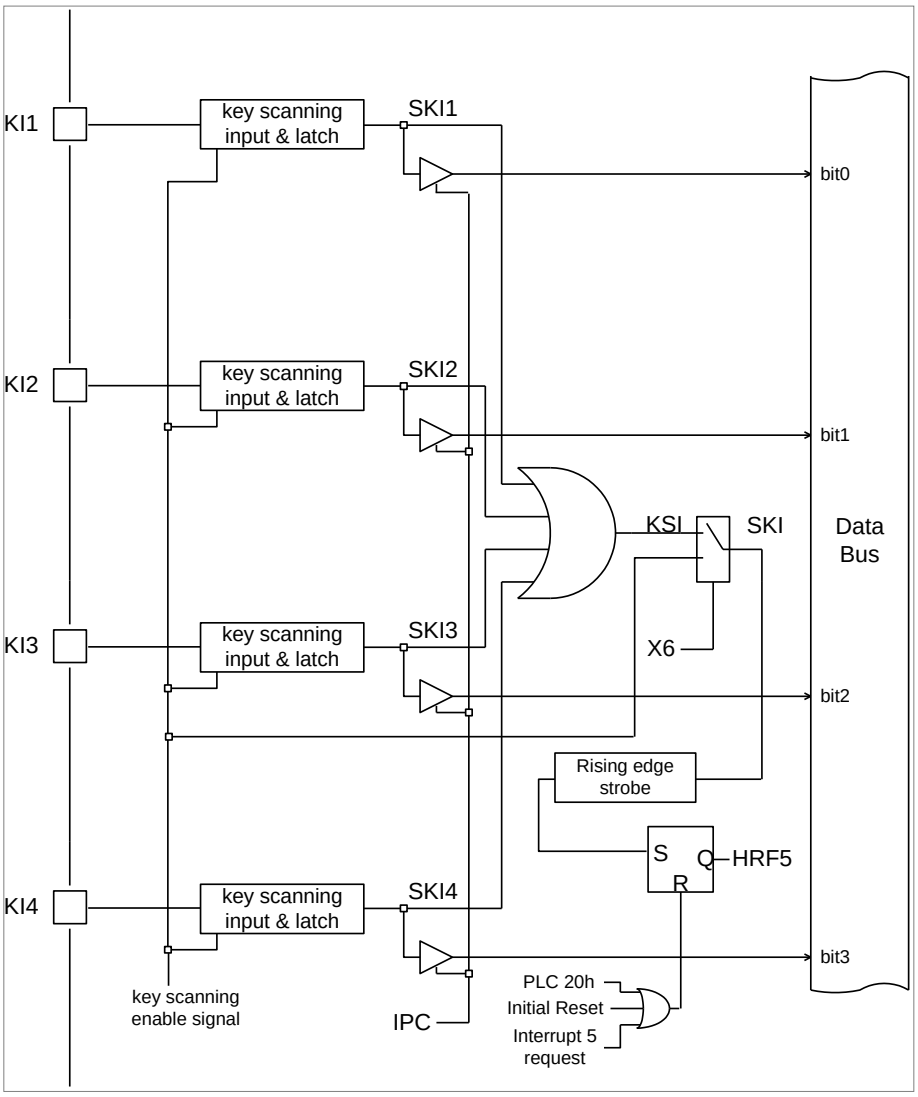
Fig 2-27 shows the organization of the Key matrix scanning input port. Once one of the KI1~4 pins detects the signal changes from "Hi-z" to "1", TM87P18M will set HRF5 to 1. If HEF5 has been set to 1 beforehand, it will cause SCF7 to be set and release the HALT mode. After the key scanning cycle finishes, the states of SKI1 ~ 4 pins will be stored into the output latch of the IOC port. Executing an IPC instruction can store these states into data RAM. Executing a PLC 20h instruction can clear the HRF5 flag.

Since the key matrix scanning function steals a part of the LCD driving waveforms as the scanning output signal, the scanning frequency is the same as the alternating clock frequency of the LCD. The formula for the key matrix scanning frequency is shown below:

The 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 32 Hz, and the duty cycle is 1/5 duty, the scanning frequency for the key matrix will be: 320 Hz (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 the internal IOC port.
; Set all the IOC pins as the output mode.
SPKX    10h         ; Generate a HALT release request when key is depressed
; Scan every column simultaneously in 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 the KI1~4 input latch state.
        JB0     ki1_release
        JB1     ki2_release
        JB2     ki3_release
        JB3     ki4_release
        .
        .
ki1_release:
        SPKX    40h             ; Check if the key depressed on K1 column.
        PLC     20h             ; Clear Flag HRF5 to avoid the false HALT release
        CALL    wait_scan_again ; Wait 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 the SEG16 scanning output.
        PLC     20h             ; Clear HRF5 to avoid the false HALT released
        CALL    wait_scan_again ; Wait for the time longer than the halt LCD clock
                                ; cycle to ensure scan again.
        IPC 10h                 ; Read the KI1 input latch state.
        JB0 kil_seg16
        ……
        ……
wait_scan_again:
        HALT
        PLC     20h
        RTS
```

CHAPTER 4 LCD/LED DRIVER OUTPUT

TM87P18M provides 32 segment output pins and 8 common output pins to drive LCD. COM5~COM8, SEG17~23, SEG40, SEG41 can also be used as DC output ports (the mask option).

MASK OPTION table:

Mask Option name	Selected item
COM5/DC5/OD5	(1) COM5
COM6/DC6/OD6	(1) COM6
COM7/DC7/OD7	(1) COM7
COM8/DC8/OD8	(1) COM8

MASK OPTION table:

Mask Option name	Selected item
SEG17~23,40,41/DC/OD	(1) SEG17~23,40,41

1. LCD LIGHTING SYSTEM IN TM87P18M

There are several settings for the LCD lighting systems that can be selected in mask option in TM87P18M, they are:

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

All these options for the lighting systems are combined into 2 kinds in mask options; the “LCD DUTY CYCLE” and the “BIAS”.

MASK OPTION table:

LCD duty cycle option

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

LCD bias option

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

The frame frequency for each lighting system is shown below. These frequencies can be selected in mask option. The entire LCD frame frequencies in the following tables are based on the slow clock source is 32768 Hz.

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

Mask Option name	Selected item	Frequency
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	0 Hz (LCD is not used)

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

Mask Option name	Selected item	Frequency
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	0 Hz (LCD is not used)

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

Mask Option name	Selected item	Frequency
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	0 Hz (LCD is not used)

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

Mask Option name	Selected item	Frequency
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	0 Hz (LCD is not used)

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

Mask Option name	Selected item	Frequency
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	0 Hz (LCD is not used)

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

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

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

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

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

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

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

2. DC OUTPUT

TM87P18M allows the LCD driver output pins (COM5 ~ COM8 and SEG17 ~ SEG23, 40, 41) to be defined as CMOS type DC output or P open-drain DC output ports in mask option. It is also possible to use some LCD driver output pins as DC output and the rest of the LCD driver output pins as LCD drivers. Refer to 4-3-4 for details.

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

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

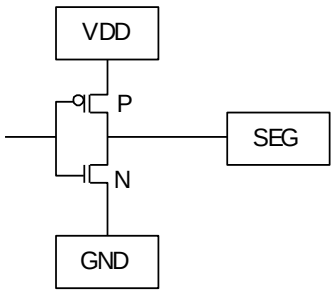


Figure 4-1 CMOS Output Type

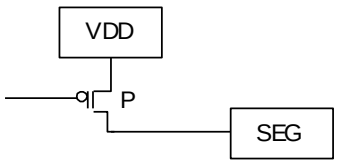


Figure 4-2 P Open-Drain Output Type

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

For example, when the LCD lighting system is specified as 1/5 duty, the COM pad used for LCD driver must be COM1 ~ COM5. Only COM6 ~ COM8 pads can be defined as DC output ports.

3. SEGMENT CIRCUIT FOR LCD DISPLAY

3-1. PRINCIPLE OF OPERATION OF LCD DRIVER SECTION

Fig. 4-3-1 below illustrates how the LCD driver module operates when the LCD-related instructions are executed.

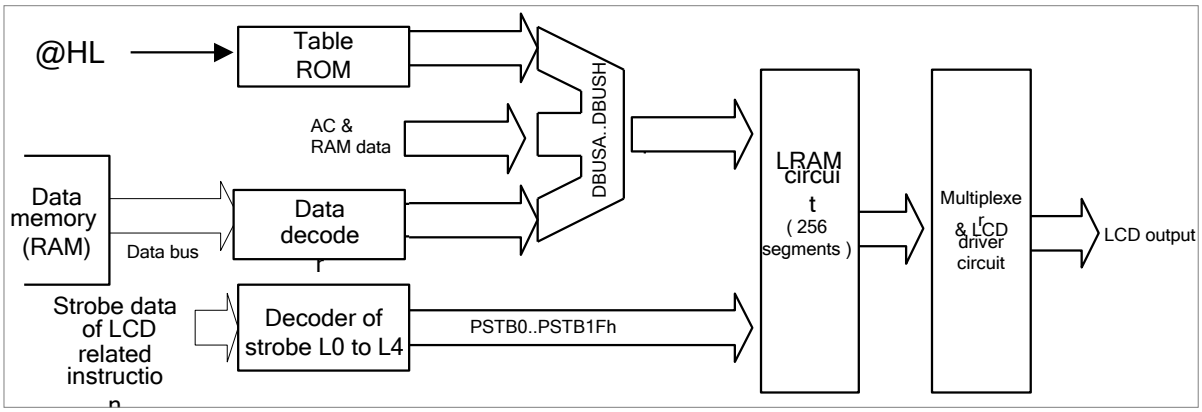


Figure 4-3-1 Principal Drawing of LCD Driver Section

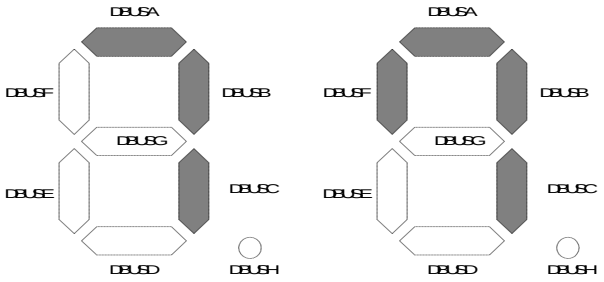
The LCD driver section consists of the following units:

- Data decoder to decode data supplied from RAM or table ROM
- Latch circuit to store LCD lighting information
- L0 to L4 decoder to decode the Lz-specified data in LCD-related instructions which specifies the strobe of the latch circuit
- Multiplexer to select 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty, 1/6 duty, 1/7 duty, and 1/8 duty.
- LCD driver circuitry
- Segment LRAM circuit connected between data decoder, L0 to L5 decoder and latch circuit.

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

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

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



DBUSF=0 DBUSF=1

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

MASK OPTION table:

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

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

The LCP instruction transfers content of the RAM (Rx) and accumulator (AC) to "DBUSA" ~ "DBUSH" directly by passing the data decoder.

The LCD instruction transfers the table ROM data (T@HL) to "DBUSA" ~ "DBUSH" directly bypassing the data decoder.

Table 4-3-2 The bit mapping table of LCP and LCD instructions

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

If we define that a “pixel” is a pattern on LCD panel corresponding to a specified segment and common, TM87P18M can drive a LCD panel which contains up to 256 (32 SEGs and 8 COMs) pixels. Each pixel needs a “pixel latch” to store its display information (ON or OFF), so there are total 256 pixel latches in latch circuitry. The input data of the pixel latch comes from DBUS data and the strobe signal comes from PSTB signal.

The segment LRAM determines the connection between DBUS data the data input of a latch circuit, and so does the connection between PSTB signals and the strobe signal. The connection is performed 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 of LCD panel’s pixel is very flexible.

Among the 256 signals obtainable by combining the data “DBUSA” to “DBUSH” with the address PSTB 0h to PSTB 1Fh, any one of 256 signals (corresponding to the number of latch circuits incorporated in the hardware) can be selected by programming the aforementioned segment LRAM. Table 4-3-3 shows the selectable PSTB 0h to PSTB 1Fh in mask option.

Table 4-3-3 Strobe Signal for LCD Latch in Segment LRAM and Strobe in the LCT Instruction

Strobe signal for LCD latch	Strobe in LCT, LCB, LCP, LCD instructions The values of Lz in ”LCT Lz, Q": *
PSTB0	0H
PSTB1	1H
PSTB2	2H
PSTB3	3H
PSTB4	4H
PSTB5	5H
……	……
PSTB1Ah	1AH
PSTB1Bh	1BH
PSTB1Ch	1CH
PSTB1Dh	1DH
PSTB1Eh	1EH
PSTB1Fh	1FH

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

The LCD pattern (pixels) can be turned off without changing the DBUS data. The execution of the SF2 4h instruction can turn off all the patterns on the LCD panel simultaneously. The execution of the RF2 4h instruction can turn on the panel. These two instructions will not affect the content stored in the latch circuitry. When executing the RF2 4h instruction to turn off the LCD, the program can still execute LCT, LCB, LCP and LCD instructions to update the content in the latch circuitry. The new data will be displayed on the LCD panel while the panel is turned on again.

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

3-2. Relative Instructions

1. Lz, Ry

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

2. LCB Lz, Ry

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

3. LCD Lz, @HL

Transfers the table ROM data specified by @HL directly to "DBUSA" through "DBUSH" without passing through the data decoder. The mapping table is shown in table 4-3-4.

4. LCP Lz, Ry

The data in 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 table 4-3-4.

5. LCT Lz, @HL

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

6. LCB Lz, @HL

Decodes the content specified in index RAM (@HL) and stores the DBUS data into the LCD 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 into the latch circuit specified by Lz. The mapping table is shown below:

Table 4-3-4 The mapping table of LCP and LCD instructions

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

8. SF2 4h

Turns off the LCD display.

9. RF2 4h

Turns on the LCD display.

3-3. THE CONFIGURATION of LCD RAM Area

	< 1/8 Duty >									
SEG	Lz	COM1	COM2	COM3	COM4	Lz	COM5	COM6	COM7	COM8
SEG1	00H	DBUSA	DBUSB	DBUSC	DBUSD	10H	DBUSA	DBUSB	DBUSC	DBUSD
SEG2		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG3	01H	DBUSA	DBUSB	DBUSC	DBUSD	11H	DBUSA	DBUSB	DBUSC	DBUSD
SEG4		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG5	02H	DBUSA	DBUSB	DBUSC	DBUSD	12H	DBUSA	DBUSB	DBUSC	DBUSD
SEG6		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG7	03H	DBUSA	DBUSB	DBUSC	DBUSD	13H	DBUSA	DBUSB	DBUSC	DBUSD
SEG8		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG9	04H	DBUSA	DBUSB	DBUSC	DBUSD	14H	DBUSA	DBUSB	DBUSC	DBUSD
SEG10		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG11	05H	DBUSA	DBUSB	DBUSC	DBUSD	15H	DBUSA	DBUSB	DBUSC	DBUSD
SEG12		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG13	06H	DBUSA	DBUSB	DBUSC	DBUSD	16H	DBUSA	DBUSB	DBUSC	DBUSD
SEG14		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG15	07H	DBUSA	DBUSB	DBUSC	DBUSD	17H	DBUSA	DBUSB	DBUSC	DBUSD
SEG16		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG17	08H	DBUSA	DBUSB	DBUSC	DBUSD	18H	DBUSA	DBUSB	DBUSC	DBUSD
SEG18		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG19	09H	DBUSA	DBUSB	DBUSC	DBUSD	19H	DBUSA	DBUSB	DBUSC	DBUSD
SEG20		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG21	0AH	DBUSA	DBUSB	DBUSC	DBUSD	1AH	DBUSA	DBUSB	DBUSC	DBUSD
SEG22		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG23	0BH	DBUSA	DBUSB	DBUSC	DBUSD	1BH	DBUSA	DBUSB	DBUSC	DBUSD
SEG24		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG25	0CH	DBUSA	DBUSB	DBUSC	DBUSD	1CH	DBUSA	DBUSB	DBUSC	DBUSD
SEG26		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG27	0DH	DBUSA	DBUSB	DBUSC	DBUSD	1DH	DBUSA	DBUSB	DBUSC	DBUSD
SEG28		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG29	0EH	DBUSA	DBUSB	DBUSC	DBUSD	1EH	DBUSA	DBUSB	DBUSC	DBUSD
SEG30		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH
SEG40	0FH	DBUSA	DBUSB	DBUSC	DBUSD	1FH	DBUSA	DBUSB	DBUSC	DBUSD
SEG41		DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH

LZ					LZ				
20H	**SEG17 DC/OD**	**SEG18 DC/OD**	**SEG19 DC/OD**	**SEG20 DC/OD**	21H	**COM5 DC5/OD5**	**COM6 DC6/OD6**	**COM7 DC7/OD7**	**COM8 DC8/OD8**
	DBUSA	DBUSB	DBUSC	DBUSD		DBUSA	DBUSB	DBUSC	DBUSD
	SEG21 DC/OD	**SEG22 DC/OD**	**SEG23 DC/OD**			**DC9/OD9**	**DC31/OD31**	**SEG40 DC/OD**	**SEG41 DC/OD**
	DBUSE	DBUSF	DBUSG	DBUSH		DBUSE	DBUSF	DBUSG	DBUSH

※ Duty 1/2: COM1~COM2, Duty 1/3: COM1~COM3…, Duty 1/8: COM1~COM8.

4. LED DRIVER OUTPUT

If the LED mode option is selected in the mask option, TM87P18M will switch the LCD driver to the LED driver. TM87P18M provides 32 segment pins (SEG) and 8 common pins (COM) to drive a LED module with 256 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, 1/2~1/8 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

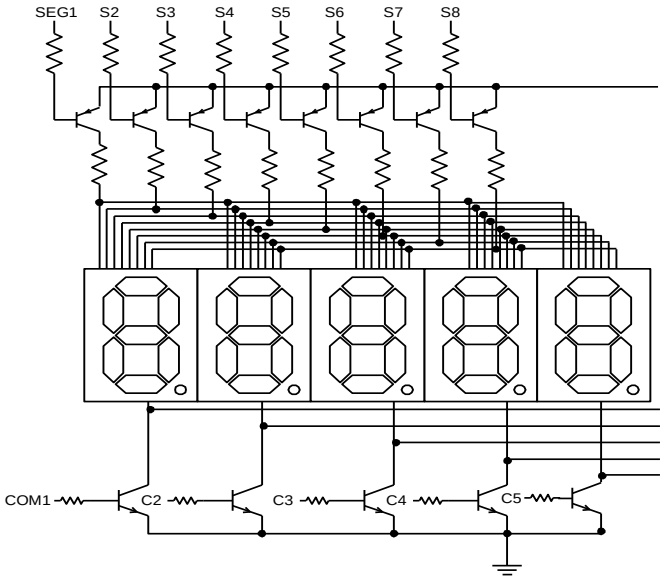
Mask Option name	Selected item
LCD/LED ACTIVE TYPE	(2) LED HIGH ACTIVE

When COM pins drives the low active LED panel

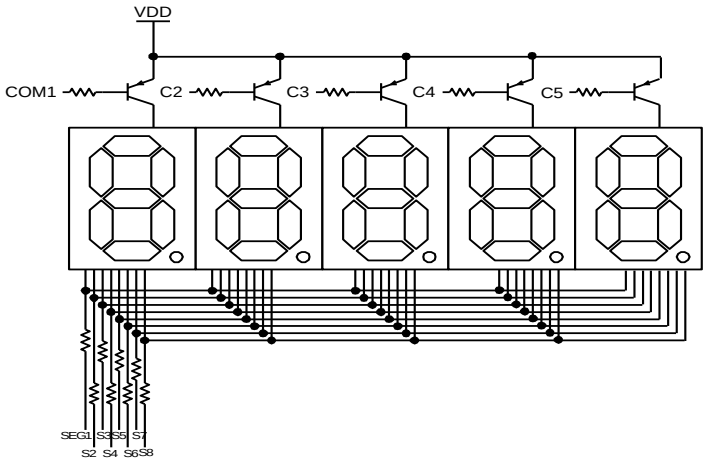
Mask Option name	Selected item
LCD/LED ACTIVE TYPE	(3) LED LOW ACTIVE

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 40 mA 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 32768 Hz).

The LED alternating frequency in 1/2 duty mode

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

The LED alternating frequency in 1/3 duty mode

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

The LED alternating frequency in 1/4 duty mode

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

The LED alternating frequency in 1/5 duty mode

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

The LED alternating frequency in 1/6 duty mode

LED duty cycle	1/5 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	42 Hz	84 Hz	171 Hz

The LED alternating frequency in 1/7 duty mode

LED duty cycle	1/5 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	36Hz	73Hz	146 Hz

The LED alternating frequency in 1/8 duty mode

LED duty cycle	1/5 duty		
Mask option	Slow	Typ.	Fast
LED alternating frequency	56Hz	113Hz	226 Hz

LED Lighting System and Maximum Number of Driving LED Segments

LED Lighting System	Maximum Number of Driving LED Segments
Static	32
Duplex	64
1/3duty	96
1/4duty	128
1/5duty	160
1/6duty	192
1/7duty	224
1/8duty	256

TM87P18M allows some SEG pins to be the DC output ports and the remaining of the SEG pins to be the LED driver outputs.

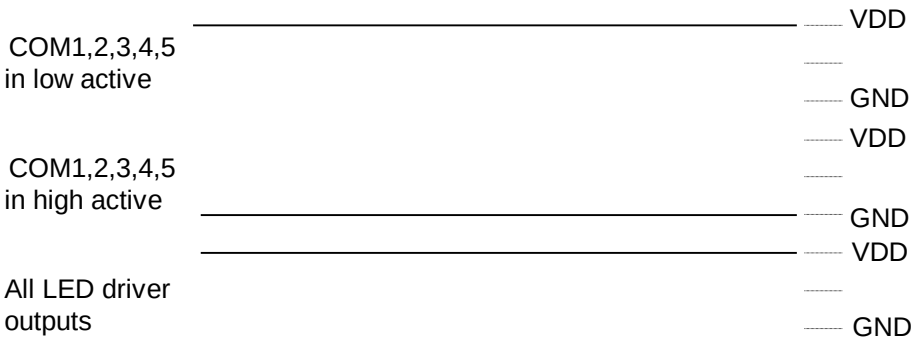
When a SEG pin is defined as the DC output port, the output data will remain intact even if the MCU enters the STOP mode or the LED turn-off mode is active.

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.

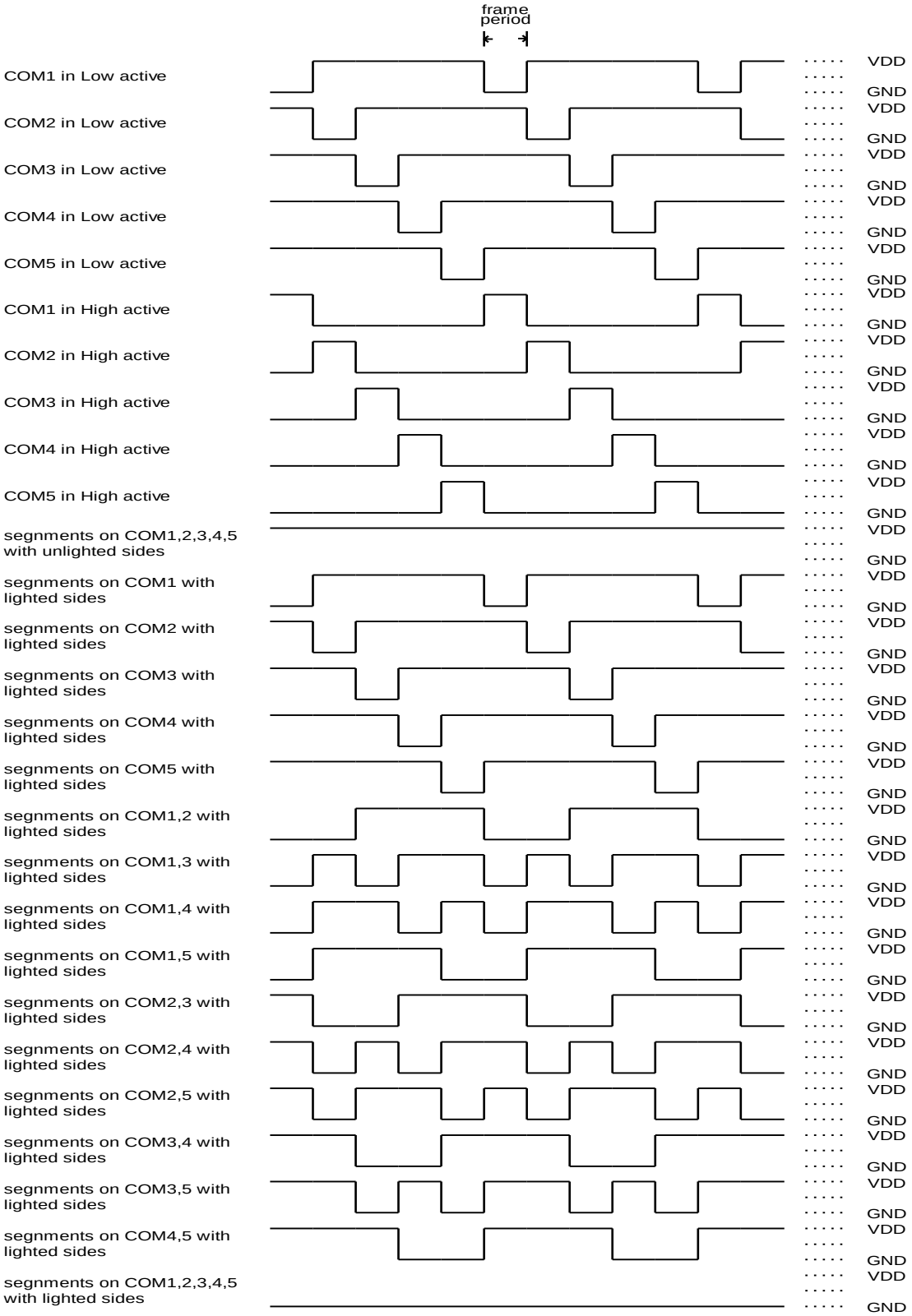
The waveform on the COM output and LED driver output for each LED lighting system are shown below.

Example. 1/5 DUTY LIGHTING SYSTEM FOR LED DRIVER

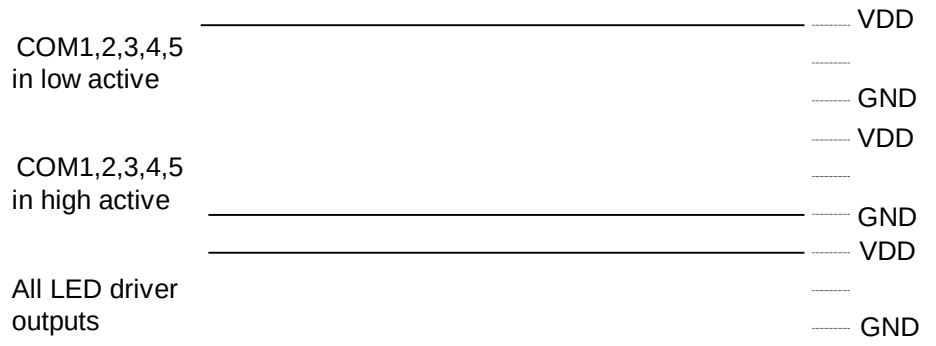
(i) Initial reset cycle (lighting)



(ii) Normal operation mode



(iii) Display Turned Off



(iv) Stop mode

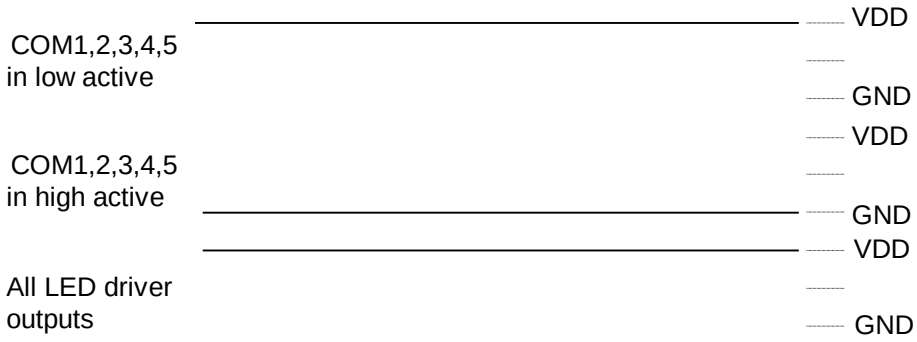


Figure 4-4-5 1/5 duty LED Waveform

Chapter 5 Detail Explanation of TM87P18M Instructions

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

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

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

- Lz represents the address of the LCD pixel latch which is configured in the segment LRAM; the address range specified by Lz is from 00H to 1FH.

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.
Lz : 00 ~ 1FH, Ry : 0 ~ 7H.

LCB Lz, Ry

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

Description: The content of working register contents, specified by Ry, are loaded to the LCD latch, specified by Lz, through the data decoder.
If the content of Ry is "0", the output of the data decoder will consist entirely of "0"s.
Lz : 00 ~ 1FH, Ry : 0 ~ 7H.

LCP Lz, Ry

Function: LCD latch [Lz] ← [Ry],AC

Description: The content of working register contents, specified by Ry, and the contents of AC are loaded to the LCD latch, specified by Lz.
Lz : 00 ~ 1FH, Ry : 0 ~ 7H.

Table 5-2 The mapping table of LCD latches with the contents of AC and Ry.

	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

LCD Lz, @HL

Function: LCD latch [Lz] ← TAB[@HL]

Description: @HL indicates an index address of table ROM.

The content of table ROM, specified by @HL, are loaded to the LCD latch, specified by Lz, directly. Refer to Table 5-2.

Lz : 00 ~ 1FH.

LCT Lz, @HL

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

Description: The content of index RAM, specified by @HL, are loaded to the LCD latch, specified by Lz, through the data decoder. Refer to Table 5-2.

Lz : 00 ~ 1FH.

LCB Lz, @HL

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

Description: The contents of index RAM, specified by @HL, are loaded to the LCD latch, specified by Lz, through the data decoder. Refer to Table 5-2.

If the content of @HL is "0", the output of the data decoder will consist entirely of "0"s.

Lz : 00 ~ 1FH.

LCP Lz, @HL

Function: LCD latch [Lz] ← [@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. Refer to Table 5-2.

Lz : 00 ~ 1FH.

SPA X

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

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

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

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

Description: Content of Rx is output to IOA port. D is output to IOA3, pulse is output to IOA4.

D = 0 or 1

IPA Rx
Function: Rx, AC ← (IOA)

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

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

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

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enable the pull-low device on IOB1~IOB4 simultaneously	X4=0	Disable the pull-low device on IOB1~IOB4 simultaneously
X3=1	IOB4 as output mode	X3=0	IOB4 as input mode
X2=1	IOB3 as output mode	X2=0	IOB3 as input mode
X1=1	IOB2 as output mode	X1=0	IOB2 as input mode
X0=1	IOB1 as output mode	X0=0	IOB1 as input mode

OPB Rx
Function: I/OB ← (Rx)

Description: The contents of Rx are output to I/OB port.

IPB Rx
Function: Rx, AC ← (IOB)

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

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

Description: Sets the I/O mode and turns on/off the pull-low device. The input pull-low device will be enabled when the I/O pin is set as input mode.

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

Bit pattern	Setting	Bit pattern	Setting
X4=1	Enables all of the pull-low and disables the low-level hold devices	X4=0	Disables all of the pull-low and enables the low-level hold devices
X3=1	IOC4 as output mode	X3=0	IOC4 as input mode
X2=1	IOC3 as output mode	X2=0	IOC3 as input mode
X1=1	IOC2 as output mode	X1=0	IOC2 as input mode
X0=1	IOC1 as output mode	X0=0	IOC1 as input mode

OPC Rx

Function: I/OC ← (Rx)

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

IPC Rx

Function: Rx, AC ← (IOC)

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

SPD X

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

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

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

OPD Rx

Function: I/OD ← [Rx]

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

IPD Rx

Function: [Rx], AC ← [I/OD]

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

SPKX X

Function: Sets the Key matrix scanning output state.

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

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

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

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

X_3 ~ X_0 = 0000, activates the K1 column

X_3 ~ X_0 = 0001, activates the K2 column

……..

X_3 ~ X_0 = 1110, activates the K15 column

X_3 ~ X_0 = 1111, activates the K16 column

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

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

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

X_3 = 0, activates the K1 ~ K8 columns simultaneously

X_3 = 1, activates the K9 ~ K16 columns simultaneously

(X_2 ~ X_0 are not a factor)

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

X_3X_2 = 00, activates the K1 ~ K4 columns simultaneously

X_3X_2 = 01, activates the K5 ~ K8 columns simultaneously

X_3X_2 = 10, activates the K9 ~ K12 columns simultaneously

X_3X_2 = 11, activates the K13 ~ K16 columns simultaneously

(X_1, X_0 are not a factor)

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

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

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

…….

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

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

(X_0 is not a factor)

SPK Rx

Function: Sets the Key matrix scanning output state.

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

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

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

SPK @HL

Function: Sets the Key matrix scanning output state.

Description: When SEG1~16 is(are) used for LCD driver pin(s), sets the content of table ROM([@HL]) to specify the key matrix scanning output state for each SEGn pin in the scanning interval.

The bit setting is the same as the SPKX instruction. The bit pattern of the table ROM corresponding to SPKX is shown below:

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

ALM X

Function: Sets buzzer output frequency.

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

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

X8	X7	X6	Clock Source (Higher Frequency)
1	1	1	FREQ*
1	0	0	DC1
0	1	1	PH3 (4 KHz)
0	1	0	PH4 (2 KHz)
0	0	1	PH5 (1 KHz)
0	0	0	DC0

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

Bit	Clock Source(Lower Frequency)
X5	PH15 (1 Hz)
X4	PH14 (2 Hz)
X3	PH13 (4 Hz)
X2	PH12 (8 Hz)
X1	PH11 (16 Hz)
X0	PH10 (32 Hz)

Notes: **1. FREQ is the output of the frequency generator.**

2. When the buzzer output does not need the envelope waveform, X5 ~ X0 should be set to 0.

3. The frequency inside is based on the PH0 is 32768 Hz.

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.

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.

MRW# @HL, Rx

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

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

The content of the index register (@HL) will be incremented automatically after executing this instruction.

MWR Rx, Ry

Function: AC, Rx ← (Ry)

Description: The content of 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.

MWR# Rx, @HL

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

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

The content of the index register (@HL) will be incremented automatically after executing this instruction.

SR0 Rx

Function: Rxn, ACn ← Rx(n+1),AC(n+1)

Rx3, AC3 ← 0

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

The result is loaded to the AC.

0 Rx3 Rx2 Rx1 Rx0

SR1 Rx

Function: Rxn, ACn ← Rx(n+1),AC(n+1)

Rx3, AC3 ← 1

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

1 Rx3 Rx2 Rx1 Rx0

SL0 Rx

Function: Rxn, ACn ← Rx(n-1),AC(n-1)

Rx0, AC0 ← 0

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

Rx3 Rx2 Rx1 Rx0 0

SL1 Rx

Function: Rxn, ACn ← Rx(n-1),AC(n-1)

Rx0, AC0 ← 1

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

Rx3 Rx2 Rx1 Rx0 1

MRA Rx

Function: CF ← (Rx)3

Description: Bit3 of the content of Rx is loaded to Carry Flag (CF).

MAF Rx

Function: AC,Rx ← CF

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

Bit 3 .... CF

Bit 2 .... (AC)=0, zero flag

Bit 1 .... (No Use)

Bit 0 .... (No Use)

3. OPERATION INSTRUCTIONS

INC* Rx
Function: Rx,AC ← (Rx)+1
Description: Adds 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: Adds 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.

INC*# @HL
Function: [@HL],AC ← R[@HL]+1, @HL HL + 1
Description: Adds 1 to the content of @HL; the result is loaded to the data memory @HL and AC. The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The Carry Flag (CF) will be affected.
• @HL indicates an index address of data memory.

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

DEC* @HL
Function: R@HL, AC ← (R@HL)-1
Description: Substrates 1 from the content of 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*# @HL
Function: R@HL, AC ← R[@HL] -1, @HL HL + 1
Description: Substrates 1 from the content of @HL; the result is loaded to the data memory @HL and AC. The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The Carry Flag (CF) will be affected.
• @HL indicates an index address of data memory.

ADC Rx
Function: AC ← (Rx)+(AC)+CF
Description: 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# @HL

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

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

ADC* Rx

Function: AC, Rx ← (Rx)+(AC)+CF

Description: 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.

ADC*# @HL

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

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

SBC Rx

Function: AC ← (Rx)+ (AC)B+CF

Description: 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# @HL

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

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

SBC* Rx

Function: AC, Rx ← (Rx)+(AC)B+CF

Description: 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.

SBC*# @HL

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

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

ADD Rx

Function: AC ← [Rx]+AC

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

ADD @HL

Function: AC ← [@HL]+AC

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

ADD# @HL
Function: AC ← [@HL]+AC, @HL HL + 1
Description: Binary-adds the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The Carry Flag (CF) will be affected.

ADD* Rx
Function: AC, [Rx] ← [Rx]+AC
Description: Binary-adds the contents of Rx and AC; the result is loaded to AC and the data memory Rx.
* The Carry Flag (CF) will be affected.

ADD* @HL
Function: AC,[@HL] ← [@HL]+AC
Description: Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.
* The Carry Flag (CF) will be affected.

ADD*# @HL
Function: AC,[@HL] ← [@HL]+AC, @HL HL + 1
Description: Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The Carry Flag (CF) will be affected.

SUB Rx
Function: AC ← [Rx]+ (AC)B+1
Description: Binary-subtracts the content of AC from the content of Rx; the result is loaded to AC.
* The Carry Flag (CF) will be affected.

SUB @HL
Function: AC ← [@HL]+ (AC)B+1
Description: Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC.
. @HL indicates an index address of data memory.
* The Carry Flag (CF) will be affected.

SUB# @HL
Function: AC ← [@HL]+ (AC)B+1, @HL HL + 1
Description: Binary-subtracts the content of AC from the content of @HL; the result is loaded to AC. The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.
* The Carry Flag (CF) will be affected.

SUB* Rx

Function: AC,[Rx] ← [Rx]+ (AC)B+1

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

SUB* @HL

Function: AC, [@HL] ← [@HL]+ (AC)B+1

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

SUB*# @HL

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

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

ADN Rx

Function: AC ← [Rx]+AC

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

ADN @HL

Function: AC ← [@HL]+AC

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

ADN# @HL

Function: AC ← [@HL]+AC, @HL HL + 1

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

ADN* Rx

Function: AC, [Rx] ← [Rx]+AC

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

ADN* @HL
Function: AC, [@HL] ← [@HL]+AC
Description: Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
* The result will not affect the Carry Flag (CF).
. @HL indicates an index address of data memory.

ADN*# @HL
Function: AC, [@HL] ← [@HL]+AC, @HL HL + 1
Description: Binary-adds the contents of @HL and AC; the result is loaded to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction.
* The result will not affect the Carry Flag (CF).
. @HL indicates an index address of data memory.

AND Rx
Function: AC ← [Rx] & AC
Description: Binary-ANDs the contents of Rx and AC; the result is loaded to AC.

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

AND# @HL
Function: AC ← [@HL] & AC, @HL HL + 1
Description: Binary-ANDs the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

AND* Rx
Function: AC, [Rx] ← [Rx] & AC
Description: Binary-ANDs the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

AND* @HL
Function: AC, [@HL] ← [@HL] & AC
Description: Binary-ANDs the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

AND*# @HL
Function: AC, [@HL] ← [@HL] & AC, @HL HL + 1
Description: Binary-ANDs the contents of @HL and AC; the result is loaded to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

EOR Rx
Function: AC ← [Rx] AC
Description: Exclusive-Ors the contents of Rx and AC; the result is loaded to AC.

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

EOR# @HL
Function: AC ← [@HL] AC, @HL HL + 1
Description: Exclusive-Ors the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

EOR* Rx
Function: AC, Rx ← [Rx] AC
Description: Exclusive-Ors the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

EOR* @HL
Function: AC, [@HL] ← [@HL] AC
Description: Exclusive-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

EOR*# @HL
Function: AC, [@HL] ← [@HL] AC, @HL HL + 1
Description: Exclusive-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

OR Rx
Function: AC ← [Rx] | AC
Description: Binary-Ors the contents of Rx and AC; the result is loaded to AC.

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

OR# @HL
Function: AC ← [@HL] | AC, @HL HL + 1
Description: Binary-Ors the contents of @HL and AC; the result is loaded to AC.
The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

OR* Rx
Function: AC, Rx ← [Rx] | AC
Description: Binary-Ors the contents of Rx and AC; the result is loaded to AC and the data memory Rx.

OR* @HL
Function: AC,[@HL] ← [@HL] | AC
Description: Binary-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL.
. @HL indicates an index address of data memory.

OR*# @HL
Function: AC,[@HL] ← [@HL] | AC, @HL HL + 1
Description: Binary-Ors the contents of @HL and AC; the result is loaded to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction.
. @HL indicates an index address of data memory.

ADCI Ry, D
Function: AC ← [Ry]+D+CF
Description: D represents the immediate data.
Binary-ADDs the contents of Ry, D and CF; the result is loaded to AC.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

ADCI* Ry, D
Function: AC,[Ry] ← [Ry]+D+CF
Description: D represents the immediate data.
Binary-ADDs the contents of Ry, D and CF; the result is loaded to AC and the working register Ry.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

SBCI Ry, D
Function: AC ← [Ry]+#(D)+CF
Description: D represents the immediate data.
Binary-subtracts the CF and immediate data D from the working register Ry; the result is loaded to AC.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

SBCI* Ry, D
Function: AC,[Ry] ← [Ry]+#(D)+CF
Description: D represents the immediate data.
Binary-subtracts the CF and immediate data D from the working register Ry; the result is loaded to AC and the working register Ry.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

ADDI Ry, D
Function: AC ← [Ry]+D
Description: D represents the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

ADDI* Ry, D
Function: AC,[Ry] ← [Ry]+D
Description: D represents the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC and the working register Ry.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

SUBI Ry, D
Function: AC ← [Ry]+#(D)+1
Description: D represents the immediate data.
Binary-subtracts the immediate data D from the working register Ry; the result is loaded to AC.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

SUBI* Ry, D
Function: AC,[Ry] ← [Ry]+#(Y)+1
Description: D represents the immediate data.
Binary-subtracts the immediate data D from the working register Ry; the result is loaded to AC and the working register Ry.
* The Carry Flag (CF) will be affected.
D = 0H ~ FH

ADNI Ry, D
Function: AC ← [Ry]+D
Description: D represents the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC.
* The result will not affect the Carry Flag (CF).
D = 0H ~ FH

ADNI* Ry, D
Function: AC, [Ry] ← [Ry]+D
Description: D represents the immediate data.
Binary-ADDs the contents of Ry and D; the result is loaded to AC and the working register Ry.
* The result will not affect the Carry Flag (CF).
D = 0H ~ FH

ANDI Ry, D
Function: AC ← [Ry] & D
Description: D represents the immediate data.
Binary-ANDs the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

ANDI* Ry, D
Function: AC,[Ry] ← [Ry] & D
Description: D represents the immediate data.
Binary-ANDs the contents of Ry and D; the result is loaded to AC and the working register Ry.
D = 0H ~ FH

EORI Ry, D
Function: AC ← [Ry] EOR D
Description: D represents the immediate data.
Exlusive-Ors the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

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

ORI Ry, D
Function: AC ← [Ry] | D
Description: D represents the immediate data.
Binary-Ors the contents of Ry and D; the result is loaded to AC.
D = 0H ~ FH

ORI* Ry, D
Function: AC,[Ry] ← [Ry] | D
Description: D represents the immediate data.
Binary-Ors the contents of Ry and D; the result is loaded to AC and the working register Ry.
D = 0H ~ FH

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.

STA# @HL

Function: [@HL] ← AC, @HL HL + 1

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

LDS Rx, D

Function: AC,Rx ← D

Description: Immediate data D is loaded to the AC and 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.

LDA# @HL

Function: AC ← [@HL] , @HL HL + 1

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

LDH Rx, @HL

Function: Rx, AC ← 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 16-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 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 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. CPU CONTROL INSTRUCTIONS

NOP

Function: no operation

Description: no operation

HALT

Function: Enters halt mode

Description: The following 3 conditions cause the halt mode to be released.
1) An interrupt is accepted.

2) The signal change specified by the SCA instruction is applied to IOC.
3) The halt release condition specified by SHE instruction is met.
When an interrupt is accepted to release the halt mode, the halt mode returns by executing the RTS instruction after completion of interrupt service.

STOP

Function: Enters stop mode and stops all oscillators

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

SCA X

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

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

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

X7~5, X3~0 is reserved

SIE* X

Function: Set/Reset interrupt enable flag

Description:

X0=1	The IEF0 is set so that interrupt 0 (Signal change at port IOC specified by SCA) is accepted.
X1=1	The IEF1 is set so that interrupt 1 (underflow from timer 1) is accepted.
X2=1	The IEF2 is set so that interrupt 2 (the signal change at the INT pin) is accepted.
X3=1	The IEF3 is set so that interrupt 3 (overflow from the predivider) is accepted.
X4=1	The IEF4 is set so that interrupt 4 (underflow from timer 2) is accepted.
X6=1	The IEF6 is set so that interrupt 6 (overflow from the RFC counter) is 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 is released by TMR1 underflow.
X2=1	The HEF2 is set so that the halt mode is released by signal changed on INT pin.
X3=1	The HEF3 is set so that the halt mode is released by predivider overflow.
X4=1	The HEF4 is set so that the halt mode is released by TMR2 underflow.
X6=1	The HEF6 is set so that the halt mode is 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 is released by the signal change on IOC port.
X5=1	The SRF5 is set so that the stop mode is released by the signal change on INT pin.

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 ← SCF3, SCF1, SCF2, BCF

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

Bit 3	Bit 2	Bit 1	Bit 0
Start Condition Flag 3 (SCF 3)	Start Condition Flag 2 (SCF2)	Start Condition Flag 1 (SCF1)	Backup flag (BCF)
Halt release caused by the IOD port	Halt release caused by SCF4,5,6,7,8,9	Halt release caused by the IOC port	The Backup mode status in TM87P18M

MSC Rx

Function: AC, Rx ← SCF4…7

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

Bit 3	Bit 2	Bit 1	Bit 0
Start Condition Flag 7 (SCF7)	The content of 15th stage of the predivider	Start Condition Flag 5 (SCF5)	Start Condition Flag 4 (SCF4)
Halt release caused by predivider overflow		Halt release caused by TM1 underflow	Halt release caused by INT pin

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 meaning of bit 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 "L" applied on KI1~4 in scanning interval

MSD Rx

Function: Rx, AC ← WDF, CSF, RFOVF

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

Bit 3	Bit 2	Bit 1	Bit 0
NA	The overflow flag of 16-bit counter of RFC (RFVOF)	Watchdog timer enable flag (WDF)	System Clock Selection Flag (CSF)

6. INDEX ADDRESS INSTRUCTIONS

MVU Rx

Function: [@U] ← (Rx)

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

MVH Rx

Function: (@H) ← (Rx)

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

MVL Rx

Function: (@L) ← (Rx)

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

CPHL X

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

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

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

The comparison bit pattern is shown below:

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

7. DECIMAL ARITHMETIC INSTRUCTIONS

DAA

Function: AC ← BCD(AC)

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

DAA* Rx

Function: AC, Rx ← BCD(AC)

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

DAA* @HL

Function: AC,R@HL ← BCD(AC)

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

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

DAA*# @HL

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

Description: Converts the content of AC to binary format, and then restores to AC and the data memory specified by @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction. When this instruction is executed, the AC must be the result of any added instruction.
* The Carry Flag (CF) will be affected.

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

DAS

Function: AC ← BCD[AC]

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

DAS* Rx

Function: AC, Rx ← BCD(AC)

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

DAS* @HL

Function: AC, @HL ← BCD[AC]

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

DAS*# @HL

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

Description: Converts the content of AC to binary format, and then restores to AC and the data memory @HL. The content of the index register (@HL) will be incremented automatically after executing this instruction. When this instruction is executed, the AC must be the result of any subtracted instruction.
* The Carry Flag (CF) will be affected.

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

8. JUMP INSTRUCTIONS

JB0 X

Function: Program counter jumps to X if AC0=1.

Description: If bit0 of AC is 1, jump occurs.
If 0, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JB1 X

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

Description: If bit1 of AC is 1, jump occurs.
If 0, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JB2 X

Function: Program counter jumps to X if AC2=1.

Description: If bit2 of AC is 1, jump occurs.
If 0, the PC is increased by 1.

The range of X is from 000H to 7FFH or 800H to FFFH.

JB3 X

Function: Program counter jumps to X if AC3=1.
Description: If bit3 of AC is 1, jump occurs.
If 0, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JNZ X

Function: Program counter jumps to X if (AC) != 0.
Description: If the content of AC is not 0, jump occurs.
If 0, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JNC X

Function: Program counter jumps to X if CF=0.
Description: If the content of CF is 0, jump occurs.
If 1, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JZ X

Function: Program counter jumps to X if (AC)=0.
Description: If the content of AC is 0, jump occurs.
If 1, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JC X

Function: Program counter jumps to X if CF=1.
Description: If the content of CF is 1, jump occurs.
If 0, the PC is increased by 1.
The range of X is from 000H to 7FFH or 800H to FFFH.

JMP X

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

CALL X

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

RTS

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

9. MISCELLANEOUS INSTRUCTIONS

SCC X

Function: Setting the clock source for IOC, IOD 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 comes from the system clock (BCLK).	X6=0	The clock source comes from the 0. Refer to section 3-3-4 for 0.

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

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 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 initial value. The following table shows the preset data and the duty cycle setting:

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

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

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

FRQX D, X

Function: Frequency generator ← D, X

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

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

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

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

1. FRQ D, Rx

   The content of Rx and AC as preset data N.

2. FRQ D, @HL

   The content of tables TOM specified by index address buffer as preset data N.

3. FRQX D, X

   The data of operand in the instruction are 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	PH9
0	1	PH3
1	0	PH15
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	PH9
0	1	PH3
1	0	PH15
1	1	FREQ

TMSX X

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

Description: The data specified by X(X8 ~ X0) is loaded to timer 1 to start the timer.
The following table shows the bit pattern for this instruction:

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

The clock source setting for timer 1

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

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	PH9
0	1	PH3
1	0	PH15
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	PH9
0	1	PH3
1	0	PH15
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	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

SF X

Function: Sets flag

Description: Description of each flag
X0: "1" The CF is set to 1.
X1: "1" The chip enters backup mode and BCF is set to 1.
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 $X_7$00X_4 00X_1X_0

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.
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

X4: “1” Enables low battery detected function

X3: “1” Enables INT powerful pull-low

X2: "1" Disables the LCD 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 flag

Description: Description of each flag

X4: “1” Disables low battery detected function

X3: “1” Disables INT powerful pull-low

X2: "1" Enables the LCD segment output.

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

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

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

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

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

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

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

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

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

Appendix A TM87P18M Instruction Table

Instruction		Machine Code	Function		Flag/Remark
NOP		0000 0000 0000 0000	No Operation		
LCT	Lz,Ry	0000 001Z ZZZZ ZYYY	Lz	← (7SEG ← Ry)	(Ry=70H~77H)
LCB	Lz,Ry	0000 010Z ZZZZ ZYYY	Lz	← (7SEG ← Ry) Blank Zero	(Ry=70H~77H)
LCP	Lz,Ry	0000 011Z ZZZZ ZYYY	Lz	← Ry & AC	(Ry=70H~77H)
LCD	Lz,@HL	0000 100Z ZZZZ Z000	Lz	← T@HL	
LCT	Lz,@HL	0000 100Z ZZZZ Z001	Lz	← (7SEG ← @HL)	
LCB	Lz,@HL	0000 100Z ZZZZ Z010	Lz	← (7SEG ← @HL) Blank Zero	
LCP	Lz,@HL	0000 100Z ZZZZ Z011	Lz	← @HL & AC	
OPA	Rx	0000 1010 0XXX XXXX	Port(A)	← Rx	
OPAS	Rx,D	0000 1011 DXXX XXXX	A1,2,3,4	← Rx0,Rx1,D,Pulse	
OPB	Rx	0000 1100 0XXX XXXX	Port(B)	← Rx	
OPC	Rx	0000 1101 0XXX XXXX	Port(C)	← Rx	
OPD	Rx	0000 1110 0XXX XXXX	Port(D)	← 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	IDBF0~3	← Rx	
MVH	Rx	0001 1101 0XXX XXXX	IDBF4~7	← Rx	
MVU	Rx	0001 1110 0XXX XXXX	IDBF8~11	← Rx	
ADC	Rx	0010 0000 0XXX XXXX	AC	← Rx + AC + CF	CF
ADC	@HL	0010 0000 1000 0000	AC	← @HL + AC + CF	CF
ADC#	@HL	0010 0000 1100 0000	AC HL	← @HL + AC + CF ←HL+1	CF
ADC*	Rx	0010 0001 0XXX XXXX	AC,Rx	← Rx + AC + CF	CF
ADC*	@HL	0010 0001 1000 0000	AC,@HL	← @HL + AC + CF	CF
ADC*#	@HL	0010 0001 1100 0000	AC,@HL HL	← @HL + AC + CF ←HL+1	CF
SBC	Rx	0010 0010 0XXX XXXX	AC	← Rx + ACB + CF	CF
SBC	@HL	0010 0010 1000 0000	AC	← @HL + ACB + CF	CF
SBC#	@HL	0010 0010 1100 0000	AC HL	← @HL + ACB + CF ←HL+1	CF
SBC*	Rx	0010 0011 0XXX XXXX	AC,Rx	← Rx + ACB + CF	CF
SBC*	@HL	0010 0011 1000 0000	AC,@HL	← @HL + ACB + CF	CF
SBC*#	@HL	0010 0011 1100 0000	AC,@HL HL	← @HL + ACB + CF ←HL+1	CF
ADD	Rx	0010 0100 0XXX XXXX	AC	← Rx + AC	CF
ADD	@HL	0010 0100 1000 0000	AC	← @HL + AC	CF
ADD#	@HL	0010 0100 1100 0000	AC HL	← @HL + AC ←HL+1	CF
ADD*	Rx	0010 0101 0XXX XXXX	AC,Rx	← Rx + AC	CF
ADD*	@HL	0010 0101 1000 0000	AC,@HL	← @HL + AC	CF
ADD*#	@HL	0010 0101 1100 0000	AC,@HL HL	← @HL + AC ←HL+1	CF
SUB	Rx	0010 0110 0XXX XXXX	AC	← Rx + ACB + 1	CF
SUB	@HL	0010 0110 1000 0000	AC	← @HL + ACB + 1	CF
SUB#	@HL	0010 0110 1100 0000	AC	← @HL + ACB + 1	CF

			HL	←HL+1	
SUB*	Rx	0010 0111 0XXX XXXX	AC,Rx	← Rx + ACB + 1	CF
SUB*	@HL	0010 0111 1000 0000	AC,@HL	← @HL + ACB + 1	CF
SUB*#	@HL	0010 0111 1100 0000	AC,@HL HL	← @HL + ACB + 1 ←HL+1	CF
ADN	Rx	0010 1000 0XXX XXXX	AC	← Rx + AC	
ADN	@HL	0010 1000 1000 0000	AC	← @HL + AC	
ADN#	@HL	0010 1000 1100 0000	AC HL	← @HL + AC ←HL+1	
ADN*	Rx	0010 1001 0XXX XXXX	AC,Rx	← Rx + AC	
ADN*	@HL	0010 1001 1000 0000	AC,@HL	← @HL + AC	
ADN*#	@HL	0010 1001 1100 0000	AC,@HL HL	← @HL + AC ←HL+1	
AND	Rx	0010 1010 0XXX XXXX	AC	← Rx AND AC	
AND	@HL	0010 1010 1000 0000	AC	← @HL AND AC	
AND#	@HL	0010 1010 1100 0000	AC HL	← @HL AND AC ←HL+1	
AND*	Rx	0010 1011 0XXX XXXX	AC,Rx	← Rx AND AC	
AND*	@HL	0010 1011 1000 0000	AC,@HL	← @HL AND AC	
AND*#	@HL	0010 1011 1100 0000	AC,@HL HL	← @HL AND AC ←HL+1	
EOR	Rx	0010 1100 0XXX XXXX	AC	← Rx EOR AC	
EOR	@HL	0010 1100 1000 0000	AC	← @HL EOR AC	
EOR#	@HL	0010 1100 1100 0000	AC HL	← @HL EOR AC ←HL+1	
EOR*	Rx	0010 1101 0XXX XXXX	AC,Rx	← Rx EOR AC	
EOR*	@HL	0010 1101 1000 0000	AC,@HL	← @HL EOR AC	
EOR*#	@HL	0010 1101 1100 0000	AC,@HL HL	← @HL EOR AC ←HL+1	
OR	Rx	0010 1110 0XXX XXXX	AC	← Rx OR AC	
OR	@HL	0010 1110 1000 0000	AC	← @HL OR AC	
OR#	@HL	0010 1110 1100 0000	AC HL	← @HL OR AC ←HL+1	
OR*	Rx	0010 1111 0XXX XXXX	AC,Rx	← Rx OR AC	
OR*	@HL	0010 1111 1000 0000	AC,@HL	← @HL OR AC	
OR*#	@HL	0010 1111 1100 0000	AC,@HL HL	← @HL OR AC ←HL+1	
ADCI	Ry,D	0011 0000 DDDD YYYY	AC	← Ry + D + CF	
ADCI*	Ry,D	0011 0001 DDDD YYYY	AC,Ry	← Ry + D + CF	
SBCI	Ry,D	0011 0010 DDDD YYYY	AC	← Ry + DB + CF	
SBCI*	Ry,D	0011 0011 DDDD YYYY	AC,Ry	← Ry + DB + CF	
ADDI	Ry,D	0011 0100 DDDD YYYY	AC	← Ry + D	
ADDI*	Ry,D	0011 0101 DDDD YYYY	AC,Ry	← Ry + D	
SUBI	Ry,D	0011 0110 DDDD YYYY	AC	← Ry + DB + 1	
SUBI*	Ry,D	0011 0111 DDDD YYYY	AC,Ry	← Ry + DB + 1	
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,@HL	← @HL + 1	CF
INC*#	@HL	0100 0000 1100 0000	AC,@HL	← @HL + 1	CF

			HL	←HL+1	
DEC*	Rx	0100 0001 0XXX XXXX	AC,Rx	← Rx – 1	CF
DEC*	@HL	0100 0001 1000 0000	AC,@HL	← @HL - 1	CF
DEC*#	@HL	0100 0001 1100 0000	AC,@HL HL	← @HL - 1 ←HL+1	CF
IPA	Rx	0100 0010 0XXX XXXX	AC,Rx	← Port(A)	
IPB	Rx	0100 0100 0XXX XXXX	AC,Rx	← Port(B)	
IPC	Rx	0100 0111 0XXX XXXX	AC,Rx	← Port(C)	
IPD	Rx	0100 1000 0XXX XXXX	AC,Rx	← Port(D)	
MAF	Rx	0100 1010 0XXX XXXX	AC,Rx	← STS1	B3 : CF B2 : ZERO B1 : (No use) B0 : (No use)
MSB	Rx	0100 1011 0XXX XXXX	AC,Rx	← STS2	B3 : SCF3(DPT) B2 : SCF2(HRx) B1 : SCF1(CPT) B0 : BCF
MSC	Rx	0100 1100 0XXX XXXX	AC,Rx	← STS3	B3 : SCF7(PDV) B2 : PH15 B1 : SCF5(TM1) B0 : SCF4(INT)
MCX	Rx	0100 1101 0XXX XXXX	AC,Rx	← STS3X	B3 : SCF9(RFC) B2 : (unused) B1 : SCF6(TM2) B0 : SCF8(SKI)
MSD	Rx	0100 1110 0XXX XXXX	AC,Rx	← STS4	B3 : (No use) B2 : FROVF B1 : WDF B0 : CSF
SR0	Rx	0101 0000 0XXX XXXX	ACn, Rxn AC3, Rx3	← Rx(n+1) ← 0	
SR1	Rx	0101 0001 0XXX XXXX	ACn, Rxn AC3, Rx3	← Rx(n+1) ← 1	
SL0	Rx	0101 0010 0XXX XXXX	ACn, Rxn AC0, Rx0	← Rx(n-1) ← 0	
SL1	Rx	0101 0011 0XXX XXXX	ACn, Rxn AC0, Rx0	← 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,@HL	← BCD(AC)	CF
DAA*#	@HL	0101 0101 1100 0000	AC,@HL HL	← BCD(AC) ←HL+1	CF
DAS		0101 0110 0000 0000	AC	← BCD(AC)	CF
DAS*	Rx	0101 0111 0XXX XXXX	AC,Rx	← BCD(AC)	CF
DAS*	@HL	0101 0111 1000 0000	AC,@HL	← BCD(AC)	CF
DAS*#	@HL	0101 0111 1100 0000	AC,@HL HL	← BCD(AC) ←HL+1	CF
LDS	Rx,D	0101 1DDD DXXX XXXX	AC,Rx	← D	
LDH	Rx,@HL	0110 0000 0XXX XXXX	AC,Rx	← H(T@HL)	
LDH*	Rx,@HL	0110 0001 0XXX XXXX	AC,Rx HL	← H(T@HL) ← HL + 1	
LDL	Rx,@HL	0110 0010 0XXX XXXX	AC,Rx	← L(T@HL)	
LDL*	Rx,@HL	0110 0011 0XXX XXXX	AC,Rx HL	← L(T@HL) ← HL + 1	
MRF1	Rx	0110 0100 0XXX XXXX	AC,Rx	← RFC3-0	
MRF2	Rx	0110 0101 0XXX XXXX	AC,Rx	← RFC7-4	
MRF3	Rx	0110 0110 0XXX XXXX	AC,Rx	← RFC11-8	

MRF4	Rx	0110 0111 0XXX XXXX	AC,Rx	← RFC15-12	
STA	Rx	0110 1000 0XXX XXXX	Rx	← AC	
STA	@HL	0110 1000 1000 0000	@HL	← AC	
STA#	@HL	0110 1000 1100 0000	@HL HL	← AC ←HL+1	
LDA	Rx	0110 1100 0XXX XXXX	AC	← Rx	
LDA	@HL	0110 1100 1000 0000	AC	← @HL	
LDA#	@HL	0110 1100 1100 0000	AC HL	← @HL ←HL+1	
MRA	Rx	0110 1101 0XXX XXXX	CF	← Rx3	
MRW	@HL,Rx	0110 1110 0XXX XXXX	AC,@HL	← Rx	
MRW#	@HL,Rx	0110 1110 1XXX XXXX	AC,@HL HL	← Rx ←HL+1	
MWR	Rx,@HL	0110 1111 0XXX XXXX	AC,Rx	← @HL	
MWR#	Rx,@HL	0110 1111 1XXX XXXX	AC,Rx HL	← @HL ←HL+1	
MRW	Ry,Rx	0111 0YYY YXXX XXXX	AC,Ry	← Rx	
MWR	Rx,Ry	0111 1YYY YXXX XXXX	AC,Rx	← Ry	
JB0	X	1000 0XXX XXXX XXXX	PC	← X	if AC0 = 1
JB1	X	1000 1XXX XXXX XXXX	PC	← X	if AC1 = 1
JB2	X	1001 0XXX XXXX XXXX	PC	← X	if AC2 = 1
JB3	X	1001 1XXX XXXX XXXX	PC	← X	if AC3 = 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 PXXX XXXX XXXX	STACK PC	← PC + 1 ← X	
JMP	X	1101 PXXX XXXX XXXX	PC	← X	
TMS	Rx	1110 0000 0XXX XXXX	AC3,2 = 11 AC3,2 = 10 AC3,2 = 01 AC3,2 = 00 AC1,0,PB3~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
TMS	@HL	1110 0001 0000 0000	TD7,6 = 11 TD7,6 = 10 TD7,6 = 01 TD7,6 = 00 TD5~0	: Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
TMSX	X	1110 001X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100X8 ,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
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	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3	

			X8,7,6=000 X5~0	: Ctm = PH9 : Set Timer2 Value	
SHE	X	1110 1000 0XXX XXX0	X6 X5 X4 X3 X2 X1	: Enable HEF6 : Enable HEF5 : Enable HEF4 : Enable HEF3 : Enable HEF2 : Enable HEF1	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 C, DPT
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 X000	X7 X5 X4 X3	: Enable SRF7(key_s) : Enable SRF5(INT) : Enable SRF4(C port) : Enable SRF3(D port)	
FAST		1110 1110 0000 0000	SCLK	: High Speed Clock	
SLOW		1110 1110 1000 0000	SCLK	: Low Speed Clock	
CPHL	X	1110 1111 XXXX XXXX	(PC+1)	← force “NOP” if X7~0=IDBF7~0	
SPK	Rx	1111 0000 0XXX XXXX	KO1~16	← Rx & AC	
SPK	@HL	1111 0001 0000 0000	KO1~16	← T @HL	
SPKX	X	1111 0010 XXXX XXXX	 X6=1 X6=0 X7,5,4=000 X7,5,4=001 X7,5,4=010 X7,5,4=10X X7,5,4=110 X7,5,4=111	: KEY_S is released by scanning cycle : KEY_S is released by normal key scanning : Set one of KO1~16 =1 by X3~0 : Set all = 1 : Set all Hi-z : Set eight of KO1~16 =1 by X3 X3=0 => KO1~8 X3=1 => KO9~16 : Set four of KO1~16 =1 by X3,2 X3,2=00 => KO1~4 X3,2=01 => KO5~8 X3,2=10 => KO9~12 X3,2=11 => KO13~16 : Set two of KO1~16 =1 by X3,2,1 X3~1=000=>KO1,2 X3~1=001=>KO3,4 X3~1=010=>KO5,6 X3~1=011=>KO7,8	

				X3~1=100=>KO9,10 X3~1=101=>KO11,12 X3~1=110=>KO13,14 X3~1=111=>KO15,16	
RTS		1111 0100 0000 0000	PC	← STACK (CALL Return)	
SCC	X	1111 0100 1X0X XXXX	X6 = 1 X6 = 0 X4 = 1 X3 = 1 X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cfq = BCLK : Cfq = PH0 : Set P(C) Cch : Set P(D) Cch : Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1111 0101 000X X000	X4 X3	: Enable SEF4(C1-4) : Enable SEF3(D1-4)	
SPA	X	1111 0101 100X XXXX	X4 X3~0	: Set A4-1 Pull-Low : Set A4-1 I/O	1:Pull low 1:Output, 0: Input
SPB	X	1111 0101 101X XXXX	X4 X3~0	: Set B4-1 Pull-Low : Set B4-1 I/O	1:Pull low 1:Output, 0: Input
SPC	X	1111 0101 110X XXXX	X4 X3-0	: Set C4-1 Pull-Low / Low-Level-Hold : Set C4-1 I/O	1:Pull low, 0:LLH 1:Output, 0: Input
SPD	X	1111 0101 111X XXXX	X4 X3-0	: Set D4-1 Pull-Low : Set D4-1 I/O	1:Pull low 1:Output, 0: Input
SF	X	1111 0110 X00X 00XX	X7 X4 X1 X0	: Reload 1 Set : WDT Enable : BCF Set : CF Set	
RF	X	1111 0111 X00X 00XX	X7 X4 X1 X0	:Reload 1 Reset : WDT Reset : BCF Reset : CF Reset	
ALM	X	1111 110X 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	
SF2	X	1111 1110 0000 XXXX	X3 X2 X1 X0	: Enable INT powerful Pull-low : Close all Segments : Dis-ENX Set : Reload 2 Set	
RF2	X	1111 1110 1000 XXXX	X3 X2 X1 X0	: Disable INT powerful Pull-low : Release Segments : Dis-ENX Reset : Reload 2 Reset	
HALT		1111 1111 0000 0000	Halt Operation		
STOP		1111 1111 1000 0000	Stop Operation		

Symbol Description

Symbol	Description	Symbol	Description
( )	Content of Register	D	Immediate Data
AC	Accumulator	(D)B	Complement of Immediate Data
(AC)n	Content of Accumulator (bit n)	PC	Program Counter
(AC)B	Complement of content of Accumulator	CF	Carry Flag
X	Address of program or control data	ZERO	Zero Flag
Rx	Address X of data RAM	WDF	Watch-Dog Timer Enable Flag
(Rx)n	Bit n content of Rx	7SEG	7 segment decoder for LCD
Ry	Address Y of working register	BCLK	System clock for instruction
R@HL	Address of data RAM specified by @HL	IEFn	Interrupt Enable Flag
BCF	Backup 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		