



十速

# TM89PL39

## *DATA SHEET*

*Rev 1.0*

**tenx** reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **Tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **Tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

---

## AMENDMENT HISTORY

<b>Version</b>	<b>Date</b>	<b>Description</b>
1.0		New Release.

# CONTENTS

<b>AMENDMENT HISTORY .....</b>	<b>2</b>
<b>CONTENTS.....</b>	<b>3</b>
<b>GENERAL DESCRIPTION .....</b>	<b>4</b>
<b>FEATURE.....</b>	<b>4</b>
<b>BLOCK DIAGRAM .....</b>	<b>7</b>
<b>PAD ASSIGNMENT.....</b>	<b>8</b>
<b>PIN DESCRIPTION .....</b>	<b>10</b>
<b>SERIAL PROGRAM/READ CONNECT PINS:.....</b>	<b>12</b>
<b>ELECTRICAL CHARACTERISTICS .....</b>	<b>17</b>
<b>TYPICAL APPLICATION CIRCUIT .....</b>	<b>26</b>
<b>Appendix A TM89PL39(L/H) Instruction Table .....</b>	<b>27</b>
<b>Symbol Description .....</b>	<b>63</b>
<b>MWM/MMW Rm Assignment .....</b>	<b>64</b>

## GENERAL DESCRIPTION

The TM89PL39(L/H) is a One Time Programmed ROM embedded high-performance 4/16-bit microcomputer with LCD driver. It contains all the necessary functions, such as 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, Resistance to Frequency Converter (RFC), EL panel driver, Serial Peripheral Interface (I2C/UART/SPI), LCD driver, look-up table, watchdog timer and key matrix scanning circuitry in a signal chip.

## FEATURE

### 1. Low power dissipation.

- 1.5V/3V operating voltage range.

TM89PL39L	1.5V Power Mode
TM89PL39H	3V Power Mode

### 2. Powerful instruction set.

- Binary addition, subtraction, BCD. BCD can be executed directly in addition, subtraction.
- 4 bits x 4 bits / 16bits x 16bits Multiplier.
- Single-bit manipulation (set, reset, decision for branch).
- Various conditional branches.
- 16 initial working registers and manipulators. (can be extended to all RAM by Page Mode)
- Table look-up.
- LCD driver data transfer.

### 3. ROM capacity(one/two/four program code). 63K /31K /15K x 16 bits.

- Instruction ROM Max. capacity(one/two/four program code) 63K /31K /15K x 16 bits.  
Program Address 8000~83FFh no use.
- Table ROM Max. capacity(one/two/four program code) 62K /32K /16K x 8 bits.
- Built-in Table ROM Word/Byte Write by Instruction in 3V Power Mode

### 4. RAM capacity. 8K /4K /2K x 4 /8 /16 bits.

- LCD Max. Capacity 256 / 128 / 64 x 4 /8 /16 bits.(for LCD Read/Write Mode , Data RAM only for LCD Write Only Mode)
- STACK Max. Capacity 16 x 16 bits.
- HL/ZR store Max. Capacity 16 / 16 x 16 bits.

### 5. With direct/index addressing mode in data RAM access.

### 6. LCD driver output.

- Max 1024 / 784 LCD dots by 16 / 8 common outputs and 64 / 98 segment outputs for >1/8 Duty / <=1/8 Duty code option.
- Max 64 segment outputs by SEG1~64 for 1/9~1/16Duty code option.
- Max 105 ~ 98 segment outputs by SEG1~64 & SEG1H~42H for 1/1 Duty ~ 1/8 Duty code option.
- 1/1 ~ 1/16 Duty can be selected by code option.

- No Bias & 1/2 ~1/4 Bias can be selected by code option.
- Single instruction to turn off all segments.
- COM1~16, SEG1~64, SEG1H~20H,23H~26H can be defined as CMOS or P\_open drain type output by code option(CUPN,0 pins no support LED & CMOS & P\_open drain type output for SEG64,63/ SEG21H,22H).
- COM1~16 pins can be mirrored to COM16~1 by code option.
- COM1~7,9~16 can be defined as SEG62~56,58~64 for  $\geq 1/9$  duty by code option.
- COM1~16 can be defined as SEG27H~42H for  $\leq 1/8$  duty by code option.
- CUPN,0 can be defined as SEG64,63 / SEG21H,22H (No Support LED & CMOS & P\_open drain type output) by code option.
- One of SEG40~43 & SEG1H~4H & SEG23H~26H can be defined as IOB1~4 by code option.
- SEG49~64 can be defined as IOI1~4,IOH1~4,IOG1~4,IOF1~4 by code option.
- SEG1H~4H can be defined as ELC, ELP, BZB, BZ by code option.
- SEG5H~20H can be defined as IOE/I1~4,IOD1~4,IOC1~4,IOA/H1~4 by code option.
- SEG23H~26H can be defined as KI1~4 by code option.
- Built-in regulator mode for VL1/2 by code option.
- Only “Lz” instruction can set LCD for LCD Write Only Mode.

#### 7. Input/output ports.

- Port IOA 4 pins (with internal pull-low, input signal chattering prevention circuitry), and can be defined as IOH/SEG17H~20H or CX,RFC0~2 / ELC,ELP,BZB,BZ / SDA,SCL,RXD,TXD by code option.
- Port IOB 4 pins (with internal pull-low), and can be defined as KI1~4 / SEG23H~26H or SEG40~43 or ELC,ELP,BZB,BZ / SEG1H~4H by code option.
- Port IOC 4 pins (with internal pull-low, low-level-hold, input signal chattering prevention circuitry), and can be defined as SEG13H~16H by code option.
- Port IOD 4 pins (with internal pull-low, input signal chattering prevention circuitry), and can be defined as SEG9H~12H by code option.
- Port IOE 4 pins (with internal pull-low), and can be defined as IOI/SEG5H~8H or RFC3~5,CX2 / ELC,ELP,BZB,BZ / SIOX,SIOY,SCK,SSB by code option.
- Port IOF 4 pins (with internal pull-low), and can be defined as SEG61H~64H by code option.
- Port IOG 4 pins (with internal pull-low), and can be defined as SEG57H~60H by code option.
- Port IOH 4 pins (with internal pull-low), and can be defined as SEG53H~56H or IOA/SEG17~20H by code option.
- Port IOI 4 pins (with internal pull-low), and can be defined as SEG49H~52H or IOE/SEG5~8H by code option.

#### 8. Interrupt function.

- External factors     5 (INT pin, SIO, Port IOA, IOC, IOD & KI input).
- Internal factors     5 (Pre-Divider, Timer1, Timer2, Timer3 & RFC).

#### 9. Built-in EL-light driver.

- ELC, ELP. Can be defined as SEG1H,2H / IOB1, 2 or CX,RFC0 / IOA1,2 / SDA,SDL or

RFC3,4/IOE1,2/MOSI,MISO by option.

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

- BZB, BZ. Can be defined as SEG3H,4H / IOB3, 4 or RFC2,3/IOA3,4/RXD, TXD or RFC5, CX2/IOE3,4/SCK/SSB by option.

**11. Built-in resistance to frequency converter.**

- CX, RFC0~2 Can be defined as SEG24~27/IOA1~4 or IOA1~4/ELC, ELP, BZB, BZ / SDA, SCL, RXD, TXD by code option.
- RFC3~5, CX2 Can be defined as IOE1~4/ELC, ELP, BZB, BZ/SIOX, SIOY, SCK, SSB by code option.

**12. Built-in key matrix scanning function.**

- KO1~KO16 (Shared with SEG1~16)
- KI1~KI4. Can be defined as SEG23H~26H/IOB1~4 by code option.

**13. Built-in Serial Interface (UART/SPI/I2C).**

- SDA, SCL, RXD, TXD Can be defined as CX, RFC0~2/IOA1~4/ELC, ELP, BZB, BZ by code option.
- SIOX, SIOY, SCK, SSB Can be defined as RFC3~5, CX2/IOE1~4/ELC, ELP, BZB, BZ by code option.

**14. Three 6-bit programmable timers with programmable clock source.**

- Read out the content in anytime
- Merge 2 or 3 timers into one 12-bit or 18-bit timer by STM instruction.
- Extend 1 timer as 12-bits timer by STE instruction.
- Used as a counter for RFC.

**15. Watchdog timer.**

**16. Built-in voltage charge halver & pump circuit.**

**17. Dual clock operation**

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

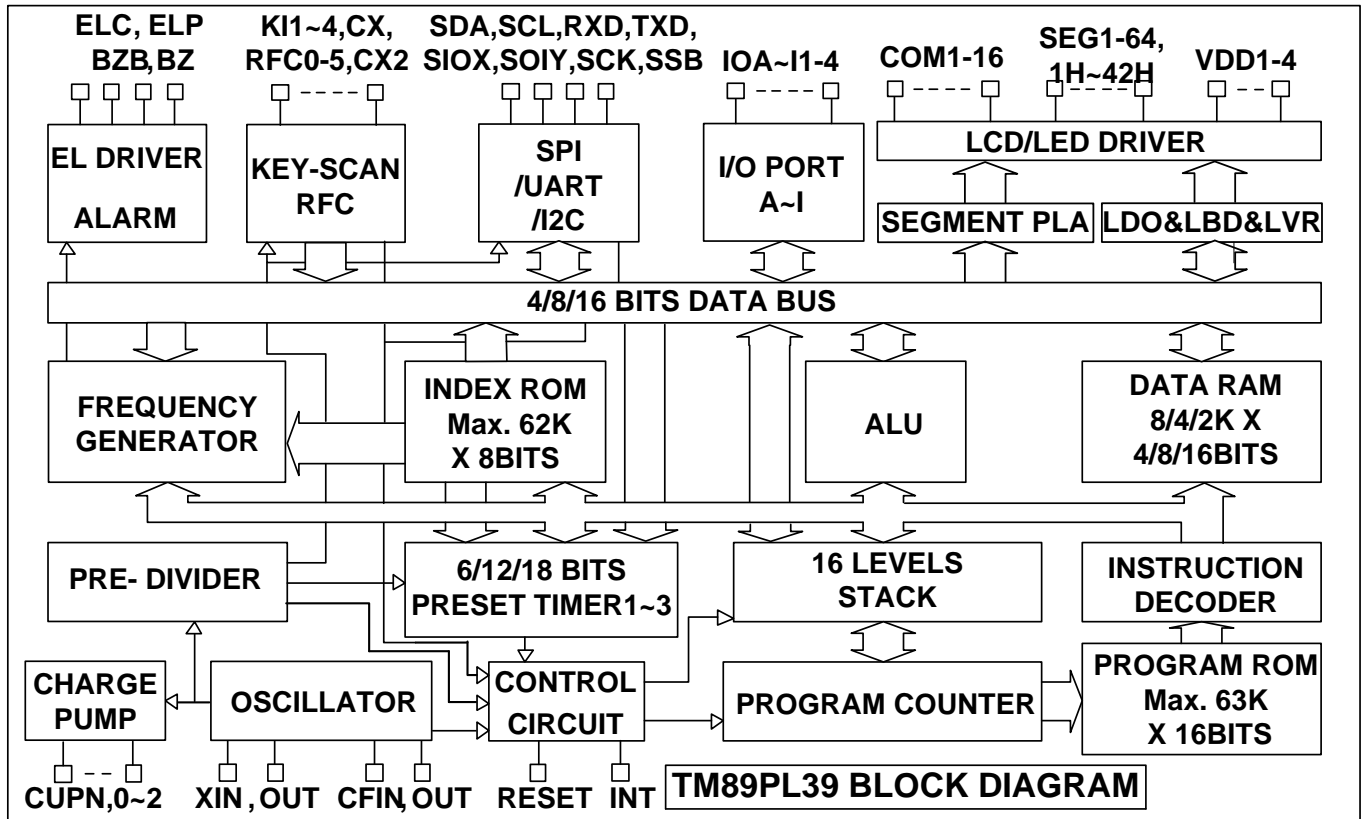
**18. HALT function.**

**19. STOP function.**

**20. Built-in Low Battery Detect.(2 type) & Comparator.**

**21. Built-in Low Voltage Reset(2 type).**

**BLOCK DIAGRAM**



**PAD ASSIGNMENT**

No	Name	No	Name
1	GND	66	SEG31
2	XIN	67	SEG32
3	XOUT	68	SEG33
4	CFIN	69	SEG34
5	CFOUT	70	SEG35
6	BAK	71	SEG36
7	VL1	72	SEG37
8	VBAT	73	SEG38
9	VL2	74	SEG39
10	VL3	75	SEG40
11	VL4	76	SEG41/IOB1'
12	CUPN/SEG(64/21H)	77	SEG42/IOB2'
13	CUP0/SEG(63/22H)	78	SEG43/IOB3'
14	CUP1	79	SEG44/IOB4'
15	CUP2	80	SEG45
16	IOB1/KI1/SEG23H	81	SEG46
17	IOB2/KI2/SEG24H	82	SEG47
18	IOB3/KI3/SEG25H	83	SEG48
19	IOB4/KI4/SEG26H	84	SEG49/IOI1
20	COM1/COM16^/SEG(62/27H)	85	SEG50/IOI2
21	COM2/COM15^/SEG(61/28H)	86	SEG51/IOI3
22	COM3/COM14^/SEG(60/29H)	87	SEG52/IOI4
23	COM4/COM13^/SEG(59/30H)	88	SEG53/IOH1
24	COM5/COM12^/SEG(58/31H)	89	SEG54/IOH2
25	COM6/COM11^/SEG(57/32H)	90	SEG55/IOH3
26	COM7/COM10^/SEG(56/33H)	91	SEG56/IOH4
27	COM8/COM9^/SEG34H	92	SEG57/IOG1
28	COM9/COM8^/SEG35H	93	SEG58/IOG2
29	COM10/COM7^/SEG(58/36H)	94	SEG59/IOG3
30	COM11/COM6^/SEG(59/37H)	95	SEG60/IOG4
31	COM12/COM5^/SEG(60/38H)	96	SEG61/IOF1
32	COM13/COM4^/SEG(61/39H)	97	SEG62/IOF2
33	COM14/COM3^/SEG(62/40H)	98	SEG63/IOF3
34	COM15/COM2^/SEG(63/41H)	99	SEG64/IOF4
35	COM16/COM1^/SEG(64/42H)	100	ELC/SEG1H/IOB1''
36	SEG1(KO1)	101	ELP/SEG2H/IOB2''
37	SEG2(KO2)	102	BZB/SEG3H/IOB3''
38	SEG3(KO3)	103	BZ/SEG4H/IOB4''
39	SEG4(KO4)	104	IOE1/SEG5H/IOI1'
40	SEG5(KO5)	105	IOE2/SEG6H/IOI2'
41	SEG6(KO6)	106	IOE3/SEG7H/IOI3'
42	SEG7(KO7)	107	IOE4/SEG8H/IOI4'
43	SEG8(KO8)	108	IOD1/SEG9H
44	SEG9(KO9)	109	IOD2/SEG10H
45	SEG10(KO10)	110	IOD3/SEG11H
46	SEG11(KO11)	111	IOD4 /SEG12H
47	SEG12(KO12)	112	IOC1/SEG13H
48	SEG13(KO13)	113	IOC2/SEG14H

49	SEG14(KO14)	114	IOC3/SEG15H
50	SEG15(KO15)	115	IOC4/SEG16H
51	SEG16(KO16)	116	IOA1/SEG17H/IOH1'
52	SEG17	117	IOA2/SEG18H/IOH2'
53	SEG18	118	IOA3/SEG19H/IOH3'
54	SEG19	119	IOA4/SEG20H/IOH4'
55	SEG20	120	CX/IOA1'/CP0/SDA
56	SEG21	121	RFC0/IOA2'/CP1/SCL
57	SEG22	122	RFC1/IOA3'/BZB/RXD
58	SEG23	123	RFC2/IOA4'/BZ/TXD
59	SEG24	124	RFC3/IOE1'/ELC/SIOX
60	SEG25	125	RFC4/IOE2'/ELP/SIOY
61	SEG26	126	RFC5/IOE3'/BZB/SCK
62	SEG27	127	CX2/IOE4'/BZ/SSB
63	SEG28	128	INT
64	SEG29	129	RESETB
65	SEG30		

**Symbol Description**

- '()' : Attached function
- '/' : Option function
- '^' : Mirror Pin Name
- ''' : Second group
- '''' : Third group

**PIN DESCRIPTION**

Name	I/O	Description
BAK	P	Positive Back-up voltage. If BAK=VL1/2 or VREG at BCF=0, connect a 0.1uF capacitor to GND. Positive voltage is need to BAK for Serial Program/Read Mode.
VBAT	P	Positive supply voltage. Positive voltage is need to VBAT for Serial Program/Read Mode.
VL1~4	P	LCD supply voltage. In 1.5V Power Mode & “LCD CHARGE PUMP MODE” option= “VL1 (NO REGULATOR”, can connect positive power output to VL1 to enhance driver for no IAP application. In 3V Power mode & “LCD CHARGE PUMP MODE” option= “VL1(NO REGULATOR)” or “VL2 (NO REGULATOR)”, can connect positive power to VL2 to enhance driver for no IAP application. If “LCD CHARGE PUMP MODE” option= “VDL (1.05V)” or “VDL (2.10V)”, connect Capacitors to VL1~4 For Capacitor Voltage Divider mode, can connect positive power to VL4 to enhance driver for no IAP application. High voltage is need to VL4 pin for Serial Program/Read Mode. VL1~4 pins build-in output voltage from VBAT by option & switch to VL1 pin output voltage from VBAT/VDL in IAP Mode. Don't connect VL2~4 pins to VBAT pin in Serial Program/Read Mode
RESETB	I	Input pin for external reset request signal, built-in internal pull-up resistor. Negative voltage is needed to RESET pin for Serial Program/Read Mode.
INT	I	Input pin for external INT request signal. <ul style="list-style-type: none"> <li>Falling edge or rising edge triggered is defined by option.</li> <li>Internal pull-down or pull-up resistor is defined by option.</li> </ul>
CUPN,01,2	O	Switching pins for supply the LCD driving voltage to the VL1~4 pins. <ul style="list-style-type: none"> <li>Connect the CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in 1/2 or 1/3 bias mode.</li> <li>Connect the CUP0 to CUP2 &amp; CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in 1/4 bias mode.</li> <li>Connect the CUPN to CUP0 &amp; CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in 1/5 bias mode or more efficient 1/3 &amp; 1/4 bias mode.</li> <li>CUPN,0 pins can be defined as SEG64,63(&gt;=1/9duty) or SEG21H,22H(&lt;=1/8duty) by code option, but no support LED mode &amp; CMOS &amp; P open drain type output.</li> </ul>
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. <ul style="list-style-type: none"> <li>The usage of 32 KHz Crystal oscillator or external RC oscillator is defined by option.</li> </ul>
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 option
COM1~16	O	Output pins for driving the common pins of the LCD panel. COM1~16 pins can be mirrored to COM16^~1^ by code option. COM1~16 can be defined as CMOS or P Open Drain type output by code option. COM10^~16^ can be defined as SEG56~62(>= 1/9duty) by code option. COM10~16 can be defined as SEG58~64(>= 1/9duty) by code option. COM1~16 can be defined as SEG27H~42H(<= 1/8duty) by code option. COM1(^) & SEG64 force same PSTB & DBUS code option for CMOS or P open drain type output. COM16^~10^/SEG27H~33H need follow COM8^~2^/SEG35H~41H support CMOS or P Open Drain type output by <=1/7~1/1duty. Set '0' for CMOS type output & '17' for Open Drain type output in .cfg file.

Name	I/O	Description
SEG1-64	O	Output pins for driving the LCD panel segment. SEG41~44 can be defined as IOB1~4 by code option. SEG49~52 can be defined as IOI1~4 by code option. SEG53~56 can be defined as IOH1~4 by code option. SEG57~60 can be defined as IOG1~4 by code option. SEG61~64 can be defined as IOF1~4 by code option. SEG1~64 can be defined as CMOS or P Open Drain type output by code option. SEG63,64 no support LED mode & CMOS & P open drain type output for code option with CUPN,0 pins. Set duty number of Lz & DBUS in .cfg file by '1'~'16' for "1/1DUTY"~"1/16DUTY" & '0' for CMOS type output & '17' for Open Drain type output.
SEG1H-42H	O	Output pins for driving the LCD panel segment if code option <= 1/8Duty. SEG1H~4H can be defined as ELC, ELP, BZB, BZ/IOB1~4 by code option. SEG5H~8H can be defined as IOE1~4/IOI1~4 by code option. SEG9H~12H can be defined as IOD1~4 by code option. SEG13H~16H can be defined as IOC1~4 by code option. SEG17~20H can be defined as IOA1~4/IOH1~4 by code option. SEG1H~20H,23~26H can be defined as COMS or Open Drain type output by option. SEG21H,22H no support LED mode & DC & P open drain type output for code option with CUPN,0 pins. Set duty number of Lz & DBUS in .cfg file by '9'~'16' for "1/1DUTY"~"1/8DUTY" & '18' for CMOS type output & '19' for Open Drain type output.
IOA1-4	I/O	Input/Output port A, and can be defined as SEG17H~20H or IOH1~4 or CX,RFC0~2/ELC, ELP, BZB, BZ/SDA,SCL,RXD,TXD by code option.
IOB1-4	I/O	Input/Output port B, and can be defined as KI1~4/SEG23H~26H or SEG41~44 or ELC, ELP,BZB,BZ/SEG1H~4H by code option.
IOC1-4	I/O	Input/Output port C, and can be defined as SEG13H~16H by code option.
IOD1~4	I/O	Input/Output port D, and can be defined as SEG9H~12H by option.
IOE1~4	I/O	Input/Output port E, and can be defined as SEG5H~8H or IOI1~4 or RFC3~5,CX2/ELC,ELP, BZB,BZ/MOSI/MISO/SCK/SSB by option.
IOF1-4	I/O	Input/Output port F, and can be defined as SEG61~64 by code option.
IOG1-4	I/O	Input/Output port G, and can be defined as SEG57~60 by code option.
IOH1-4	I/O	Input/Output port H, and can be defined as IOA1~4 or SEG53~56 by code option.
IOI1-4	I/O	Input/Output port I, and can be defined as IOE1~4 or SEG49~52 by code option.
CX,CX2 RRFC0~5	I O	2 input pins and 6 output pins for RFC application. CX,RFC0~2 and can be defined as IOA1~4/CP0,CP1, BZB, BZ/SDA,SCL,RXD,TXD by code option. RFC3~5,CX2 can be defined as IOE1~4/ELC,ELP,BZB,BZ/SIOX,SIOY,SCK,SSB by code option. CX,RR is Signal for Serial Program/Read Mode.
ELC/ELP	O	Output port for El panel driver, and can be defined as SEG1H,2H/IOB1, 2 or IOA1,2/SDA.SCL or RFC3,4/IOE1,2/SIOX,SIOY by code option.
BZB/BZ	O	Output port for alarm, clock or single tone melody generator, and can be defined as SEG3H,4H/IOB3, 4 or RFC1,2/IOA3,4/RXD,TXD or RFC5,CX2/IOE3,4/SCK,SSB by code option.
KO1~KO16	O	Output port for key matrix scanning, shared with SEG1~16.
KI1~4	I	Input port for key matrix scanning, and can be defined as IOB1~4/SEG23H~26H by option.
SDA	I/O	1 input/output data pins, 1 output synchronous clock pin for I2C application, and can be defined as. CX,RFC0/IOA1,2/CP0,1 by code option.
SCL	O	
RXD	I	1 input data pins, 1 input/output data pin for UART application, and can be defined as.
TXD	I/O	RFC1,2/IOA3,4/BZB,BZ by code option.
SIOX,SIOY,	I/O	2 input/output data pins, 1 input/output synchronous clock pin, and 1 input/output slave

Name	I/O	Description
SCK,SSB		select pin for SPI application, and can be defined as RFC3~5,CX2/IOE1~4/ELC,ELP, BZB,BZ by code option. SIOX,SIOY can be set as SDA,SCL/RXD,TXD pins for UART/I2C by instruction if IOA 1,2/3,4 not set to SDA,SCL/RXD,TXD.
CP0,1	I	2 input pins for comparator application.
GND	P	Negative supply voltage.

## SERIAL PROGRAM/READ CONNECT PINS:

VBAT, BAK, VL4, GND, RESETB, SCL,SDA

## RAM & INITIAL PAGE ASSIGNMENT

Mode	Initial Page	Page Number	Instruction inserted for set other Page by Compiler
Rx (0000~1FFFh)	0000~007Fh(0)	64 (x128)	SRX X (X : 1~3Fh ; X : 2,3h for LCD R/W Mode)
Ry (0000~1FFFh )	0070~007Fh(7)	512 (x16)	SRY X (X : 0~6,8~1FFh ; X : 10~1Fh for LCD R/W Mode)
Lz (00~7Fh)	0~1Fh(0)	4(x32)	SLZ X (X : 0~3h)

### 8K x 4 bits RAM

\$0000	<b>Rx Page 0h (Initial Page)</b>
\$007F	<b>Ry Page 0~(7)h</b>
\$0080	MHL/RHL 16 Address
\$00BF	-----
\$00C0	MZR/RZR 16 Address
\$00FF	-----
\$0100	<b>Rx Page 2h (LCD)</b>
\$017F	<b>Ry Page 10~17h</b>
\$0180	<b>Rx Page 3h (LCD)</b>
\$01FF	<b>Ry Page 18~1Fh</b>
\$0200	<b>STACK 16 Level</b>
\$023F	-----
\$0240	-----
\$027F	-----
\$1F80	<b>Rx Page 3Fh</b>
\$1FFF	<b>Ry Page 1F8~1FFh</b>

\$0,0000,10XX,XX00b <= IDBF3~0  
 \$0,0000,10XX,XX01b <= IDBF7~4  
 \$0,0000,10XX,XX10b <= IDBF11~8  
 \$0,0000,10XX,XX11b <= IDBF15~12  
  
 \$0,0000,11XX,XX00b <= ZRBF3~0  
 \$0,0000,11XX,XX01b <= ZRBF7~4  
 \$0,0000,11XX,XX10b <= ZRBF11~8  
 \$0,0000,11XX,XX11b <= ZRBF15~12

### 128 x 8 bits LCD

\$00(\$0101,0100)	<b>Lz Page 0h (Initial Page)</b>
\$1F(\$013F,013F)	-----
\$20(\$0141,0140)	<b>Lz Page 1h</b>
\$3F(\$017F,017E)	-----
\$40(\$0181,0180)	<b>Lz Page 2h</b>
\$5F(\$01BE,01BE)	-----
\$60(\$01C1,01C0)	<b>Lz Page 3h</b>
\$7F(\$01FF,01FE)	-----

### 16 level x 16 bits STACK

level	PC15~12	PC11~8	PC7~4	PC3~0
1	\$0203	\$0202	\$0201	\$0200
2	\$0207	\$0206	\$0205	\$0204
3	\$020B	\$020A	\$0209	\$0208
4	\$020F	\$020E	\$020D	\$020C
5	\$0213	\$0212	\$0211	\$0210
6	\$0217	\$0216	\$0215	\$0214
7	\$021B	\$021A	\$0219	\$0218
8	\$021F	\$021E	\$021D	\$021C
9	\$0223	\$0222	\$0221	\$0220
10	\$0227	\$0226	\$0225	\$0224
11	\$022B	\$022A	\$0229	\$0228
12	\$022F	\$022E	\$022D	\$022C
13	\$0233	\$0232	\$0231	\$0230
14	\$0237	\$0236	\$0235	\$0234
15	\$023B	\$023A	\$0239	\$0238
16	\$023F	\$023E	\$023D	\$023C

<Data of RAM Address : \$0100~\$01FF>

1. initial value = \$F/0 by LCD initial code option = On/Off for LCD Read/Write(share RAM) code option.
2. No initial value for LCD Write Only(No share RAM) code option.

### Configuration of RAM Area for LCD Read/Write Mode

Rx	Lz	BIT3~0	Rx	Lz	BIT3~0	Rx	Lz	BIT3~0	Rx	Lz	BIT3~0
0100H	00	DBUSD~A	0140H	20	DBUSD~A	0180H	40	DBUSD~A	01C0H	60	DBUSD~A
0101H		DBUSH~E	0141H		DBUSH~E	0181H		DBUSH~E	01C1H		DBUSH~E
0102H	01	DBUSD~A	0142H	21	DBUSD~A	0182H	41	DBUSD~A	01C2H	61	DBUSD~A
0103H		DBUSH~E	0143H		DBUSH~E	0183H		DBUSH~E	01C3H		DBUSH~E
0104H	02	DBUSD~A	0144H	22	DBUSD~A	0184H	42	DBUSD~A	01C4H	62	DBUSD~A
0105H		DBUSH~E	0145H		DBUSH~E	0185H		DBUSH~E	01C5H		DBUSH~E
0106H	03	DBUSD~A	0146H	23	DBUSD~A	0186H	43	DBUSD~A	01C6H	63	DBUSD~A
0107H		DBUSH~E	0147H		DBUSH~E	0187H		DBUSH~E	01C7H		DBUSH~E
0108H	04	DBUSD~A	0148H	24	DBUSD~A	0188H	44	DBUSD~A	01C8H	64	DBUSD~A
0109H		DBUSH~E	0149H		DBUSH~E	0189H		DBUSH~E	01C9H		DBUSH~E
010AH	05	DBUSD~A	014AH	25	DBUSD~A	018AH	45	DBUSD~A	01CAH	65	DBUSD~A
010BH		DBUSH~E	014BH		DBUSH~E	018BH		DBUSH~E	01CBH		DBUSH~E
010CH	06	DBUSD~A	014CH	26	DBUSD~A	018CH	46	DBUSD~A	01CCH	66	DBUSD~A
010DH		DBUSH~E	014DH		DBUSH~E	018DH		DBUSH~E	01CDH		DBUSH~E
010EH	07	DBUSD~A	014EH	27	DBUSD~A	018EH	47	DBUSD~A	01CEH	67	DBUSD~A
010FH		DBUSH~E	014FH		DBUSH~E	018FH		DBUSH~E	01CFH		DBUSH~E
0110H	08	DBUSD~A	0150H	28	DBUSD~A	0190H	48	DBUSD~A	01D0H	68	DBUSD~A
0111H		DBUSH~E	0151H		DBUSH~E	0191H		DBUSH~E	01D1H		DBUSH~E
0112H	09	DBUSD~A	0152H	29	DBUSD~A	0192H	49	DBUSD~A	01D2H	69	DBUSD~A
0113H		DBUSH~E	0153H		DBUSH~E	0193H		DBUSH~E	01D3H		DBUSH~E
0114H	0A	DBUSD~A	0154H	2A	DBUSD~A	0194H	4A	DBUSD~A	01D4H	6A	DBUSD~A
0115H		DBUSH~E	0155H		DBUSH~E	0195H		DBUSH~E	01D5H		DBUSH~E
0116H	0B	DBUSD~A	0156H	2B	DBUSD~A	0196H	4B	DBUSD~A	01D6H	6B	DBUSD~A
0117H		DBUSH~E	0157H		DBUSH~E	0197H		DBUSH~E	01D7H		DBUSH~E
0118H	0C	DBUSD~A	0158H	2C	DBUSD~A	0198H	4C	DBUSD~A	01D8H	6C	DBUSD~A
0119H		DBUSH~E	0159H		DBUSH~E	0199H		DBUSH~E	01D9H		DBUSH~E
011AH	0D	DBUSD~A	015AH	2D	DBUSD~A	019AH	4D	DBUSD~A	01DAH	6D	DBUSD~A
011BH		DBUSH~E	015BH		DBUSH~E	019BH		DBUSH~E	01DBH		DBUSH~E
011CH	0E	DBUSD~A	015CH	2E	DBUSD~A	019CH	4E	DBUSD~A	01DCH	6E	DBUSD~A
011DH		DBUSH~E	015DH		DBUSH~E	019DH		DBUSH~E	01DDH		DBUSH~E
011EH	0F	DBUSD~A	015EH	2F	DBUSD~A	019EH	4F	DBUSD~A	01DEH	6F	DBUSD~A
011FH		DBUSH~E	015FH		DBUSH~E	019FH		DBUSH~E	01DFH		DBUSH~E
0120H	10	DBUSD~A	0160H	30	DBUSD~A	01A0H	50	DBUSD~A	01E0H	70	DBUSD~A
0121H		DBUSH~E	0161H		DBUSH~E	01A1H		DBUSH~E	01E1H		DBUSH~E
0122H	11	DBUSD~A	0162H	31	DBUSD~A	01A2H	51	DBUSD~A	01E2H	71	DBUSD~A
0123H		DBUSH~E	0163H		DBUSH~E	01A3H		DBUSH~E	01E3H		DBUSH~E
0124H	12	DBUSD~A	0164H	32	DBUSD~A	01A4H	52	DBUSD~A	01E4H	72	DBUSD~A
0125H		DBUSH~E	0165H		DBUSH~E	01A5H		DBUSH~E	01E5H		DBUSH~E
0126H	13	DBUSD~A	0166H	33	DBUSD~A	01A6H	53	DBUSD~A	01E6H	73	DBUSD~A
0127H		DBUSH~E	0167H		DBUSH~E	01A7H		DBUSH~E	01E7H		DBUSH~E
0128H	14	DBUSD~A	0168H	34	DBUSD~A	01A8H	54	DBUSD~A	01E8H	74	DBUSD~A
0129H		DBUSH~E	0169H		DBUSH~E	01A9H		DBUSH~E	01E9H		DBUSH~E
012AH	15	DBUSD~A	016AH	35	DBUSD~A	01AAH	55	DBUSD~A	01EAH	75	DBUSD~A
012BH		DBUSH~E	016BH		DBUSH~E	01ABH		DBUSH~E	01EBH		DBUSH~E
012CH	16	DBUSD~A	016CH	36	DBUSD~A	01ACH	56	DBUSD~A	01ECH	76	DBUSD~A
012DH		DBUSH~E	016DH		DBUSH~E	01ADH		DBUSH~E	01EDH		DBUSH~E
012EH	17	DBUSD~A	016EH	37	DBUSD~A	01AEH	57	DBUSD~A	01EEH	77	DBUSD~A
012FH		DBUSH~E	016FH		DBUSH~E	01AFH		DBUSH~E	01EFH		DBUSH~E
0130H	18	DBUSD~A	0170H	38	DBUSD~A	01B0H	58	DBUSD~A	01F0H	78	DBUSD~A

Rx	Lz	BIT3~0	Rx	Lz	BIT3~0	Rx	Lz	BIT3~0	Rx	Lz	BIT3~0
0131H		DBUSH~E	0171H		DBUSH~E	01B1H		DBUSH~E	01F1H		DBUSH~E
0132H	19	DBUSD~A	0172H	39	DBUSD~A	01B2H	59	DBUSD~A	01F2H	79	DBUSD~A
0133H		DBUSH~E	0173H		DBUSH~E	01B3H		DBUSH~E	01F3H		DBUSH~E
0134H	1A	DBUSD~A	0174H	3A	DBUSD~A	01B4H	5A	DBUSD~A	01F4H	7A	DBUSD~A
0135H		DBUSH~E	0175H		DBUSH~E	01B5H		DBUSH~E	01F5H		DBUSH~E
0136H	1B	DBUSD~A	0176H	3B	DBUSD~A	01B6H	5B	DBUSD~A	01F6H	7B	DBUSD~A
0137H		DBUSH~E	0177H		DBUSH~E	01B7H		DBUSH~E	01F7H		DBUSH~E
0138H	1C	DBUSD~A	0178H	3C	DBUSD~A	01B8H	5C	DBUSD~A	01F8H	7C	DBUSD~A
0139H		DBUSH~E	0179H		DBUSH~E	01B9H		DBUSH~E	01F9H		DBUSH~E
013AH	1D	DBUSD~A	017AH	3D	DBUSD~A	01BAH	5D	DBUSD~A	01FAH	7D	DBUSD~A
013BH		DBUSH~E	017BH		DBUSH~E	01BBH		DBUSH~E	01FBH		DBUSH~E
013CH	1E	DBUSD~A	017CH	3E	DBUSD~A	01BCH	5E	DBUSD~A	01FCH	7E	DBUSD~A
013DH		DBUSH~E	017DH		DBUSH~E	01BDH		DBUSH~E	01FDH		DBUSH~E
013EH	1F	DBUSD~A	017EH	3F	DBUSD~A	01BEH	5F	DBUSD~A	01FEH	7F	DBUSD~A
013FH		DBUSH~E	017FH		DBUSH~E	01BFH		DBUSH~E	01FFH		DBUSH~E

## ABSOLUTE MAXIMUM RATINGS

GND=0V

Name	Symbol	Range	Unit
Maximum Supply Voltage	VBAT	-0.3 to 3.6	V
	VL1	-0.3 to 2.1	
	VL2	-0.3 to 3.6	
	VL3	-0.3 to 6.0	
	VL4	-0.3 to 6.0	
Maximum Input Voltage	Vin1	-0.3 to VBAT +0.3	
	Vin2	-0.3 to VL1/2 +0.3	
Maximum output Voltage	Vout1	-0.3 to VBAT +0.3	
	Vout2	-0.3 to VL1/2 +0.3	
	Vout3	-0.3 to VL3 +0.3	
	Vout4	-0.3 to VL4 +0.3	
Maximum Operating Temperature	Topg	-40 to +85	°C
Maximum Storage Temperature	Tstg	-40 to +125	

## ALLOWABLE OPERATING CONDITIONS

at #1: 1.5V Power Mode : Ta= -20°C to 70°C, GND=0V

at #2: 3V Power Mode (BCF=0: BAK<VBAT) , Ta= -20°C to 70°C, GND=0V

at #3: 3V Power Mode (BCF=0: BAK=VBAT) , Ta= -40°C to 85°C, GND=0V

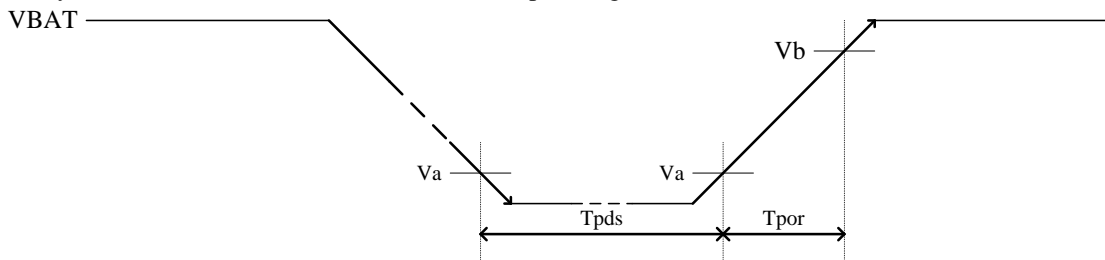
at #4: “3V BATTERY(BAK=VREG<=1.55V for BCF=0)” or “1.5V BATTERY(BAK=VREG for BCF=0)” : Ta= -20°C to 70°C, GND=0V

at #5: “3V BATTERY(BAK=VREG>1.55V for BCF=0)” : Ta= -20°C to 70°C, GND=0V

Crystal Mode condition: XIN&amp;XOUT with 15pF&amp;15pF match capacitor

Name	Symb.	Condition	Min.	Max.	Unit
Supply LCD Voltage	VL1		0.95	1.8	V
	VL2		1.8	3.6	
	VL3		1.8	5.4	
Oscillator Start-Up Voltage	BAK	Crystal Mode BCF=1, #1 BCF=0, #4	1.2		V
		Crystal Mode BCF=1, #2 BCF=0, #5	1.8		
Oscillator Sustain Voltage	BAK	Crystal Mode BCF=0, #1	1.1	1.8	V
		Crystal Mode BCF=0, #2	1.1	3.6	
		Crystal Mode BCF=0, #3	1.3	3.6	
Supply Voltage	BAK		1.2*1	3.6	V
	VBAT	1.5V Power Mode without regulator	1.2*1	1.8	
		1.5V Power Mode with regulator	1.35*1	1.8	
		3V Power Mode without regulator BAK=VBAT for BCF=0	1.8	3.6	
		3V Power Mode without regulator BAK=VL1 for BCF=0	2.4	3.6	
		3V Power Mode with regulator VL2=VDL	2.0	3.6	
		3V Power Mode with regulator VL1=VDL	1.8	3.6	
Input "H" Voltage	Vih1	I/O, INT, RESETB, CX/2, SIO	0.8xVBAT	VBAT	V
Input "L" Voltage	Vil1		0	0.2 x VBAT	
Input "H" Voltage	Vih2	OSCIN	0.8xBAK	BAK	V
Input "L" Voltage	Vil2		0	0.2 x BAK	
Operating Freq	Fopg1	Crystal Mode	32		KHz
	Fopg2	RC Mode	10	4096	
	Fopg3	CF Mode	1000	4096	
Power-down stable time before Power-on reset activation	Tpds	Va= 0.1xVBAT	1		S
Power-on reset activation power rise time	Tpor	Va/Vb=0.1/0.9 x VBAT VBAT>=1.2V		10	mS

\*1: Crystal mode need take care Oscillator Start-Up Voltage.



## ALLOWABLE OPERATING FREQUENCY

at Ta= -40°C to 85°C, GND=0V

Condition	Maximum Operating Frequency
BAK=1.2V	1MHz
BAK=1.8V	4 MHz
BAK=2.2V	6 MHz

## ELECTRICAL CHARACTERISTICS

### Power Consumption

at Ta= -40°C to 85°C, GND=0V

Halt Condition: BCF=0, 1/3Bias, 1/16Duty, LCD Alternating Frequency=PH5, Charge Pump Cycle=PH4, without loading.

Name	Sym.	Condition	Min.	Typ.	Max.	Unit
32768 X'tal / 32KHz Slow Internal RC HALT mode	IHALT1	1.5V Power Mode, VBAT=1.5V		5		uA
	IHALT2	3V Power Mode (BCF=0=> BAK=VL1), VBAT=3.0V		2		
	IHALT3	3V Power Mode (BCF=0=> BAK=VBAT), VBAT=3.0V		10		
STOP mode (VBAT=3.0V)	ISTOP1	Disable LVR1 & Regulator & LVR2			1	
	ISTOP2	Enable LVR1		0.2	1.5	
	ISTOP3	Enable Regulator or LVR2		1	3	

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

### Internal RC Frequency Range

at #1 : Ta=0°C to 50°C, GND= 0V

at #2 : Ta=-20°C to 70°C, GND= 0V

at #3 : Ta=-40°C to 85°C, GND= 0V

Option Mode	BAK	Min.	Typ.	Max.
SLOW-64KHz	1.5V,#1	-15%	65536Hz	+15%
	1.5V,#2	-20%	65536Hz	+20%
FAST-2MHz	1.5V,#3	0.6 MHz	1.6 MHz	2.6 MHz
	3.0V,#3	1.0 MHz	2.0 MHz	3.0 MHz
FAST-4MHz (Temperature Compensation)	1.8V,#1	-2%	4.00MHz	+2%
	1.8V,#2	-4%	4.00MHz	+2%
	1.8V,#3	-8%	4.00MHz	+2%

**Input Resistance**

at #1: VBAT=1.5V (1.5V Power Mode)

at #2: VBAT=3.0V (3V Power Mode)

at Ta= -40°C to 85°C, GND=0V

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
“L” Level Hold Tr. (IOC)	Rllh1	Vi=0.2VBAT, #1	10	40	100	KΩ
	Rllh2	Vi=0.2VBAT, #2	10	40	100	
INT,IOA~I,SCK(Normal) Pull-Down Tr.	Rnpd1	Vi=VBAT, #1	200	500	1000	
	Rnpd2	Vi=VBAT, #2	200	500	1000	
INT,IOA',IOE',SIO(Normal) Pull-up Tr.	Rnpu1	Vi=VBAT, #1	50	200	1000	
	Rnpu2	Vi=VBAT, #2	50	350	1000	
IOA',IOE',SIO(Small) Pull-up Tr.	Rspu1	Vi=GND, #1	10	40	100	
SIO(Small) Pull-Down Tr.	Rspd1	Vi=GND, #2	10	40	100	
RES Pull-Up R	Rres1	Vi=GND or VBAT, # 1	10	40	100	
	Rres2	Vi=GND or VBAT, #2	10	40	100	

**DC Output Characteristics**

at #1: VBAT=1.2V

at #2: VBAT=2.4V

at Ta= -40°C to 85°C, GND=0V

Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit	
Output “H” Voltage	Voh1a	Ioh= -100uA, #1	COM1~16 & SEG1~64,1H~16H,23H~42H (for DC/OD), IOB~G,IOI, ELC, ELP, BZB, BZ	1.0			V	
	Voh2a	Ioh= -1mA, #2		2.0				
Output “L” Voltage	Vol1a	Iol=200uA, #1				0.2		
	Vol2a	Iol=2mA, #2				0.4		
Output “H” Voltage	Voh1b	Ioh= -200uA, #1		IOA(°),IOH',SEG17H~20H (for DC/OD), IOE',RFC0~5,SIO,ELC*1,ELP*1,BZB1*,BZ*1 INT, CX(2)(Vol only)	1.0			
	Voh2b	Ioh= -3mA, #2			2.0			
Output “L” Voltage	Vol1b	Iol=400uA, #1				0.2		
	Vol2b	Iol=5mA, #2				0.4		
Output “L” Voltage	Vol2c	Iol=40mA, #2	COM1~16 (for LED)					0.6

\*1 : share pins with RFC.

**Segment Driver Output Characteristics**

at #1: VL1=1.2V

at #2: VL2=2.4V

at #3: VL1=1.05V

at #4: VL2=2.10V

at #5: VL3=VL4=2.4V

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
------	-------	-----------	-----	------	------	------	-------

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.	
Static display Mode								
Output “H” Voltage	Voh12f	Ioh= -1uA, #1, #2	SEG-n	2.2			V	
	Voh34f	Ioh= -1uA, #3		1.90				
Output “L” Voltage	Vol12f	Iol=1uA, #1, #2				0.2		
	Vol34f	Iol=1uA, #3, #4				0.2		
Output “H” Voltage	Voh12g	Ioh=-10uA, #1, #2	COM-n	2.2				
	Voh34g	Ioh=-10uA, #3		1.90				
Output “L” Voltage	Vol12g	Iol=10uA, #1, #2				0.2		
	Vol34g	Iol=10uA, #3				0.2		
1/2 Bias display Mode								
Output “H” Voltage	Voh12f	Ioh= -1uA, #1, #2	SEG-n	2.2			V	
	Voh34f	Ioh= -1uA, #3		1.90				
Output “L” Voltage	Vol12f	Iol=1uA, #1, #2				0.2		
	Vol34f	Iol=1uA, #3, #4				0.2		
Output “H” Voltage	Voh12g	Ioh= -10uA, #1, #2	COM-n	2.2				
	Voh34g	Ioh= -10uA, #3		1.90				
Output “M1” Voltage	Vom112g	Iol/h= +/-10uA, #1, #2		1.0		1.4		
	Vom134g	Iol/h= +/-10uA, #3		0.85		1.25		
Output “L” Voltage	Vol12g	Iol=10uA, #1, #2			0.2			
	Vol34g	Iol=10uA, #3			0.2			
1/3 Bias display Mode								
Output “H” Voltage	Voh12h	Ioh= -1uA, #1, #2	SEG-n	3.4			V	
	Voh34h	Ioh= -1uA, #3, #4		2.95				
	Voh5h	Ioh= -1uA, #5		2.2				
Output “M1” Voltage	Vom112h	Iol/h= +/-1uA, #1, #2		1.0		1.4		
	Vom134h	Iol/h= +/-1uA, #3, #4		0.85		1.25		
	Vom15h	Iol/h= +/-1uA, #5		0.6		1.0		
Output “M2” Voltage	Vom212h	Iol/h= +/-1uA, #1, #2		2.2		2.6		
	Vom234h	Iol/h= +/-1uA, #3, #4		1.95		2.30		
	Vom25h	Iol/h= +/-1uA, #5		1.4		1.8		
Output “L” Voltage	Vol12h	Iol=1uA, #1, #2				0.2		
	Vol34h	Iol=1uA, #3, #4				0.2		
	Vol5h	Iol=1uA, #5				0.2		
Output “H” Voltage	Voh12i	Ioh= -10uA, #1, #2		COM-n	3.4			
	Voh34i	Ioh= -10uA, #3, #4			2.95			
	Voh5i	Ioh= -1uA, #5			2.2			
Output “M1” Voltage	Vom112i	Iol/h= +/-10uA, #1, #2	1.0			1.4		
	Vom134i	Iol/h= +/-10uA, #3, #4	0.85			1.25		
	Vom15i	Iol/h= +/-10uA, #5	0.6			1.0		
Output “M2” Voltage	Vom212i	Iol/h= +/-10uA, #1, #2	2.2			2.6		
	Vom234i	Iol/h= +/-10uA, #3, #4	1.90			2.30		
	Vom25i	Iol/h= +/-10uA, #5	1.4			1.8		
Output “L” Voltage	Vol12i	Iol=10uA, #1, #2				0.2		
	Vol34i	Iol=10uA, #3, #4				0.2		
	Vol5i	Iol=10uA, #5				0.2		
1/4 Bias display Mode								
Output “H” Voltage	Voh12h	Ioh= -1uA, #1, #2	SEG-n		4.6			V
	Voh34h	Ioh= -1uA, #3, #4			4.0			

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.
Output "M2" Voltage	Voh5h	Ioh= -1uA, #5		2.2			
	Vom212h	Iol/h= +/-1uA, #1, #2		2.2		2.6	
	Vom234h	Iol/h= +/-1uA, #3, #4		1.95		2.30	
Output "L" Voltage	Vol12h	Iol=1uA, #1, #2				0.2	
	Vol34h	Iol=1uA, #3, #4				0.2	
Output "H" Voltage	Voh12i	Ioh= -10uA, #1, #2	COM-n	4.6			
	Voh34i	Ioh= -10uA, #3, #4		4.0			
Output "M1" Voltage	Vom112i	Iol/h= +/-10uA, #1, #2		1.0		1.4	
	Vom134i	Iol/h= +/-10uA, #3, #4		0.85		1.25	
Output "M2" Voltage	Vom212i	Iol/h= +/-10uA, #1, #2		3.4		3.8	
	Vom234i	Iol/h= +/-10uA, #3, #4		2.95		3.35	
Output "L" Voltage	Vol12i	Iol=10uA, #1, #2				0.2	
	Vol34i	Iol=10uA, #3, #4				0.2	

### Regulator & Low-Battery-Detect Circuit Characteristics

at Ta= -20°C to 70°C, GND= 0V

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
VREG regulator output for BAK	VREG	VDX3~0=F	Typ. -4%	1.95	Typ. +4%	V
		VDX3~0=E		1.90		
		VDX3~0=D		1.85		
		VDX3~0=C		1.80		
		VDX3~0=B		1.75		
		VDX3~0=A		1.70		
		VDX3~0=9		1.65		
		VDX3~0=8		1.60		
		VDX3~0=7		1.55		
		VDX3~0=6		1.50		
		VDX3~0=5		1.45		
		VDX3~0=4		1.40		
		VDX3~0=3		1.35		
		VDX3~0=2		1.30		
		VDX3~0=1		1.25		
VDX3~0=0	1.20					
VDL regulator output for VL2 or BAK & for 3V Power Mode only & initial=2.10V	VDL	VDL3~0=F	Typ. -4%	2.55	Typ. +4%	V
		VDL3~0=E		2.50		
		VDL3~0=D		2.45		
		VDL3~0=C		2.40		
		VDL3~0=B		2.35		
		VDL3~0=A		2.30		
		VDL3~0=9		2.25		
		VDL3~0=8		2.20		
		VDL3~0=7		2.15		
		VDL3~0=6		2.10		
		VDL3~0=5		2.05		
		VDL3~0=4		2.00		
		VDL3~0=3		1.95		
		VDL3~0=2		1.90		
		VDL3~0=1		1.85		

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
VDL regulator output for VL1 & initial=1.05V	VDL	VDL3~0=0		1.80		
		VDL3~0=F	Typ. -4%	1.70	Typ. +4%	V
		VDL3~0=E		1.65		
		VDL3~0=D		1.60		
		VDL3~0=C		1.55		
		VDL3~0=B		1.50		
		VDL3~0=A		1.45		
		VDL3~0=9		1.40		
		VDL3~0=8		1.35		
		VDL3~0=7		1.30		
		VDL3~0=6		1.25		
		VDL3~0=5		1.20		
		VDL3~0=4		1.15		
		VDL3~0=3		1.10		
		VDL3~0=2		1.05		
		VDL3~0=1		1.00		
VDL3~0=0	0.95					
VREG Stable time, BAK with 0.1 uF Capacitor	TREG				600	mS
LBD voltage	VLBD	LBD4~0=1F	Typ. -5%	2.90	Typ. +5%	V
		LBD4~0=1E		2.85		
		LBD4~0=1D		2.80		
		LBD4~0=1C		2.75		
		LBD4~0=1B		2.70		
		LBD4~0=1A		2.65		
		LBD4~0=19		2.60		
		LBD4~0=18		2.55		
		LBD4~0=17		2.50		
		LBD4~0=16		2.45		
		LBD4~0=15		2.40		
		LBD4~0=14		2.35		
		LBD4~0=13		2.30		
		LBD4~0=12		2.25		
		LBD4~0=11		2.20		
		LBD4~0=10		2.15		
		LBD4~0=0F		2.10		
		LBD4~0=0E		2.05		
		LBD4~0=0D	2.00			
		LBD4~0=0C	Typ. -0.1	1.95	Typ. +0.1	V
		LBD4~0=0B		1.90		
		LBD4~0=0A		1.85		
		LBD4~0=09		1.80		
		LBD4~0=08		1.75		
		LBD4~0=07		1.70		
		LBD4~0=06		1.65		
		LBD4~0=05		1.60		
		LBD4~0=04		1.55		
		LBD4~0=03		1.50		
		LBD4~0=02		1.45		
		LBD4~0=01		1.40		

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
		LBD4~0=00		1.35		
Comparator Voltage for CMPMD1,0=X0	VCMP	LBD4~0=1F	Typ. -4%	2.90xN	Typ. +4%	V
		LBD4~0=1E		2.85xN		
		LBD4~0=1D		2.80xN		
		LBD4~0=1C		2.75xN		
		LBD4~0=1B		2.70xN		
		LBD4~0=1A		2.65xN		
		LBD4~0=19		2.60xN		
		LBD4~0=18		2.55xN		
		LBD4~0=17		2.50xN		
		LBD4~0=16		2.45xN		
		LBD4~0=15		2.40xN		
		LBD4~0=14		2.35xN		
		LBD4~0=13		2.30xN		
		LBD4~0=12		2.25xN		
		LBD4~0=11		2.20xN		
		LBD4~0=10		2.15xN		
		LBD4~0=0F		2.10xN		
		LBD4~0=0E		2.05xN		
		LBD4~0=0D		2.00xN		
		LBD4~0=0C		1.95xN		
		LBD4~0=0B		1.90xN		
		LBD4~0=0A		1.85xN		
		LBD4~0=09		1.80xN		
		LBD4~0=08		1.75xN		
		LBD4~0=07		1.70xN		
		LBD4~0=06		1.65xN		
		LBD4~0=05		1.60xN		
		LBD4~0=04		1.55xN		
LBD4~0=03	1.50xN					
LBD4~0=02	1.45xN					
LBD4~0=01	1.40xN					
LBD4~0=00	1.35xN					
						V
Comparator Voltage by CMPMD1,0=01	VCMP	-	-4%	0.95V	+4%	V
Comparator Voltage by CMPMD1,0=11	VCMP	-	-0.025	VREF	+0.025	V
VBAT for CMPMD1,0=00	VBAT		1.45			V
VBAT for CMPMD1,0=01	VBAT		1.35			V
VBAT for CMPMD1,0=10 (CMP- : CP0/1=0.70~0.74V)	VBAT		1.50			V
VBAT for CMPMD1,0=10 (CMP- : CP0/1=0.75~0.79V)	VBAT		1.55			V
VBAT for CMPMD1,0=10 (CMP- : CP0/1=0.80~0.84V)	VBAT		1.60			V
VBAT for CMPMD1,0=10 (CMP- : CP0/1>=0.85V)	VBAT		1.35			V
VBAT for CMPMD1,0=11	VBAT		1.20			V
CP0,CP1	VCPI		0.70			V
LBD/CMP circuit response time	TLBD				100	uS

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
LVR2 circuit response time	TLVR2				1	mS

Note1 : VREG & VDL & VCMP : at VBAT=3.0/1.5V & voltage of option < VBAT for TM89PL39H/L

Note2 : CMPMD1=0 : N=1 , CMPMD2,1=01 : N=CP1/0.95V , CMPMD2,1=11 : N=CP0/0.95V

### Low-Battery-Detect(detect VL2 Type) Circuit Characteristics

at Ta= -40°C to 85°C, GND=0V

Name	Symb.	Condition	Min.	Typ.	Max.
LBD Volatge For 3V Power Mode only & initial=2.45V (VL2=VBAT & check all LBF before ENLBD:1->0)	VLBD2A	V2LBD2~0=7	-8%	3.05V	+8%
		V2LBD2~0=6		2.95V	
		V2LBD2~0=5		2.85V	
		V2LBD2~0=4		2.75V	
		V2LBD2~0=3		2.65V	
		V2LBD2~0=2		2.55V	
		V2LBD2~0=1		2.45V	
		V2LBD2~0=0		2.35V	
LBD Volatge For 1.5V Power Mode only & initial=1.25V (CUPDRV=1 & CYC2~0=3 & ENCYC=1 & check all LBF be fore ENLBD:1->0 & variation of VL2 <2.5%)	VLBD2B	V2LBD2~0=7	-8%	1.55V	+8%
		V2LBD2~0=6		1.50V	
		V2LBD2~0=5		1.45V	
		V2LBD2~0=4		1.40V	
		V2LBD2~0=3		1.35V	
		V2LBD2~0=2		1.30V	
		V2LBD2~0=1		1.25V	
		V2LBD2~0=0		1.20V	
V2LBD circuit response time	TLBD2				100us

**Low-Voltage-Reset Circuit Characteristics**

at Ta= -40/20°C to 85/70°C, GND=0V for LVR1/2

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
LVR1 Reset Voltage	Vlvr1	3V Power Mode Only	1.20	1.80	2.40	V
LVR2 Reset Voltage	Vlvr2	Code option	Typ. -5%	2.90	Typ. +5%	V
				2.85		
2.80						
2.75						
2.70						
2.65						
2.60						
2.55						
2.50						
2.45						
LVR2 Reset Voltage	Vlvr2	Code option	Typ. -0.1	1.95	Typ. +0.1	V
				1.90		
				1.85		
				1.80		
				1.75		
				1.70		
				1.65		
				1.60		
				1.55		
				1.50		
				1.45		
1.40 *1						
1.35 *1						

\*1 : Ta= 0°C to 70°C, GND= 0V

**IAP**

at Ta=25°C, GND=0V

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
Program Time by Timer2	Tpgm		90	100	110	uS
FREQ for IAP Charge Pump	CUPIAP			500		KHz
BCLK for Program	Fpgm				500	
BAK for Program	Vpgm		3.0	3.3	3.6	V
BAK for Margin Read	Vmrd	Execute LDL/H(*) in IAP Mode	3.2	3.3	3.4	
VPP for Program by VL4	VPP	Spec. offset value see Note1	6.5	6.75	7	
VBAT-NoBias	BT0	Sent external voltage to VL4 pin in IAP Mode			16	bits
VBAT-1/2Bias(1C2P/2C4P) (VL1 Capacitor >=0.1uF)	BT2A1	VBAT = 3.3V & VL4: 0.1uF			1/2	
	BT2A2	VBAT = 3.3V & VL4: 1uF			2	
	BT2A3	VBAT = 3.3V & VL4: 3.3/2.2uF			4	
	BT2A4	VBAT = 3.3V & VL4: 10uF			8	
	BT2A5	VBAT = 3.3V & VL4: 47uF			16	
	BT2B1	VBAT = 3.4V & VL4: 0.1uF			4	
	BT2B2	VBAT = 3.4V & VL4: 1uF			8	
	BT2B3	VBAT = 3.4V & VL4: 3.3/2.2uF			16	
	BT2C1	VBAT = 3.5V & VL4: 0.1uF			8	
BT2C2	VBAT = 3.5V & VL4: 1uF			16		
VDL-1/3Bias(1C2P/2C4P) (VL1 Capacitor >=0.1uF)	BT31	VDL(2.30V) & VL4: 0.1uF			2/4	
	BT32	VDL(2.30V) & VL4: 1uF			8	
	BT33	VDL(2.30V) & VL4: 3.3/2.2uF			16	

1C2P : ONE CAPACTOR FOR CUP1-2

2C4P : TWO CAPACTOR FOR CUP1-2,CUPN-0

Note1 : VPP for Program by VL4 Spec need increase 0.01/0.02/0.03/0.05/0.1V to VL4 for care 1/2/4/8/16 bits  
Program current affect VL4 output to VPP voltage.

Note2 : Don't connect VBAT to VL1~4 for IAP Application.

Note3 : VL1&4 need connect Capacitors to GND for VBAT-1/2Bias option.

Note4 : VL1,2,4 need connect Capacitors to GND for VDL-1/3Bias option.

Note5 : Wait VL4 stable before execute each PTR for VBAT-1/2Bias & VDL-1/3Bias option.

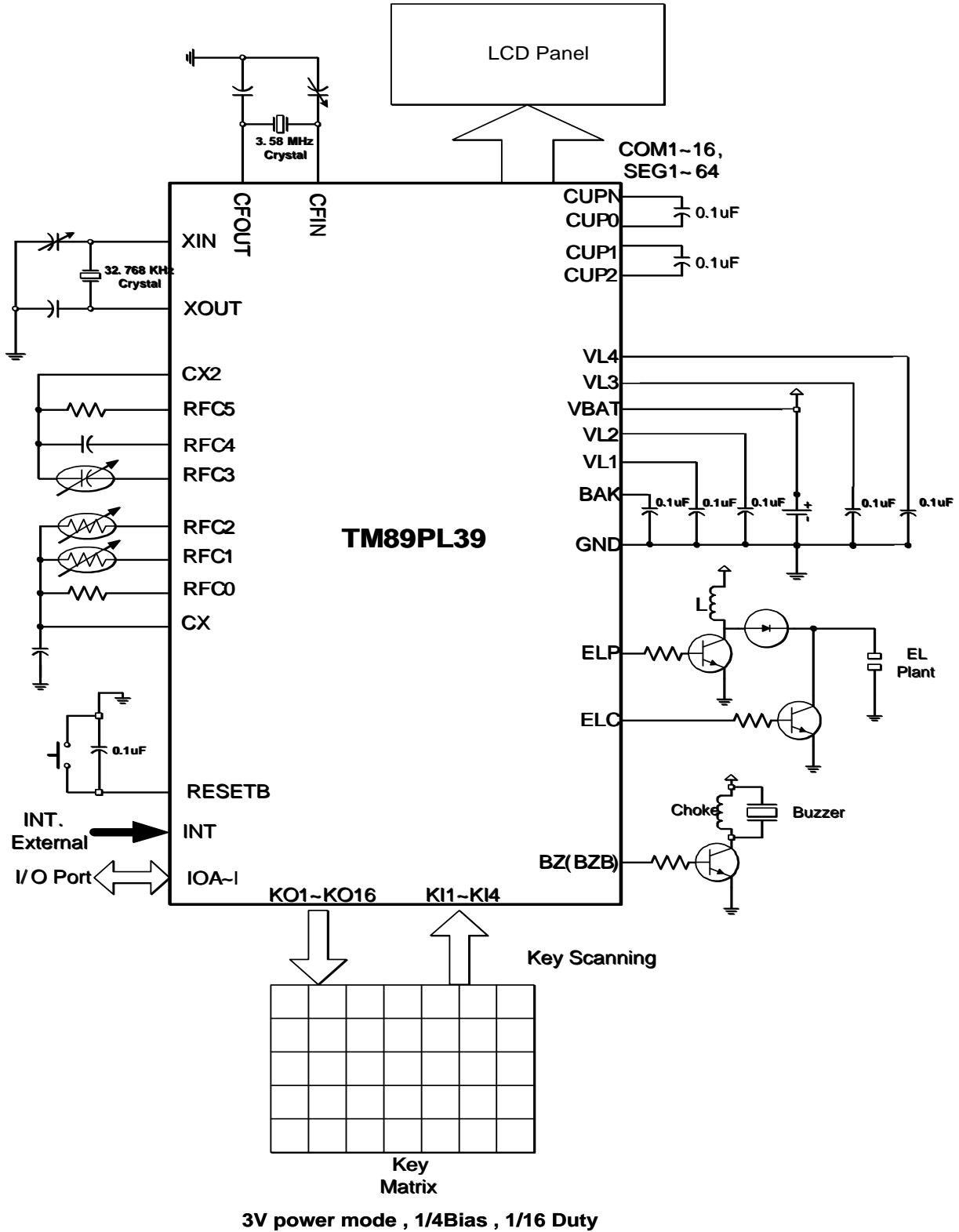
Note6 : System auto enable & change to VDL=1.80~2.55V by VDL3~0=0~Fh mode after enter IAP Mode for VDL-1/3Bias option.

Note7 : System auto enter HALT mode after execute PTR & control program time by Timer2 release(Tpgm=N x Ctm2 , N = timer2 set count value) & Timer2 enable count by execute PTR only in IAP Mode, so need execute TM2(X) to set program time before PTR in IAP Mode.

Note8 : If Margin Read check fail show program data can't hold > 10 years!

## TYPICAL APPLICATION CIRCUIT

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



**Appendix A TM89PL39(L/H) Instruction Table**

Instruction	Machine Code	Function	Flag/Remark
NOP	0000 0000 0000 0000	No Operation	
IDC& IDC% IDC\$	0000 0001 010B B010	'&,\$': HL ←HL+1 '%,\$': ZR ←ZR+1	'&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
IDC8& IDC8% IDC8\$	0000 0001 010B B100	'&,\$': HL ←HL+2 '%,\$': ZR ←ZR+2	'&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
IDCH& IDCH% IDCH\$	0000 0001 010B B110	'&,\$': HL ←HL+4 '%,\$': ZR ←ZR+4	'&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID LID& LID% LID\$	@ZR,@HL 0000 0001 011B B010	(@ZR) '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+1	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID LID& LID% LID\$	@HL,@ZR 0000 0001 011B B011	(@HL) '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+1	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID8 LID8& LID8% LID8\$	@ZR,@HL 0000 0001 011B B100	(@ZR)8 bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID8 LID8& LID8% LID8\$	@HL,@ZR 0000 0001 011B B101	(@HL)8 bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LIDH LIDH& LIDH% LIDH\$	@ZR,@HL 0000 0001 011B B110	(@ZR)16 bits '&,\$': HL ←HL+4 '%,\$': ZR ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LIDH LIDH& LIDH% LIDH\$	@HL,@ZR 0000 0001 011B B111	(@HL)16 bits '&,\$': HL ←HL+4 '%,\$': ZR ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCT LCT#	@ZR,Ry 0000 0001 100B YYYY	(@ZR)8 bits '#': ZR	' ': B4=0 '#': B4=1
LCB LCB#	@ZR,Ry 0000 0001 101B YYYY	(@ZR)8 bits '#': ZR	' ': B4=0 '#': B4=1
LCP LCP#	@ZR,Ry 0000 0001 110B YYYY	(@ZR)8 bits '#': ZR	' ': B4=0 '#': B4=1
LCD LCD& LCD% LCD\$	@ZR,@HL 0000 0001 111B B000	(@ZR)8 bits '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCDH LCDH& LCDH% LCDH\$	@ZR,@HL 0000 0001 111B B100	(@ZR)16 bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCT LCT& LCT% LCT\$	@ZR,@HL 0000 0001 111B B001	(@ZR)8 bits '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCB LCB& LCB% LCB\$	@ZR,@HL 0000 0001 111B B010	(@ZR)8 bits '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
		Blank Zero	

Instruction		Machine Code	Function		Flag/Remark
LCP LCP& LCP% LCP\$	@ZR,@HL	0000 0001 111B B011	(@ZR)8 bits '&,\$': HL '%,\$': ZR	← (@HL) & AC ← HL+1 ← ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCT	Lz,Ry	0000 001Z ZZZZ YYYY	(Lz)	← (7SEG ← (Ry))	
LCB	Lz,Ry	0000 010Z ZZZZ YYYY	(Lz)	← (7SEG ← (Ry))  Blank Zero	
LCP	Lz,Ry	0000 011Z ZZZZ YYYY	(Lz)	← (Ry) & AC	
LCD LCD#	Lz,@HL	0000 100Z ZZZZ B000	(Lz) '#': HL	← T(@HL) ← HL+1	' ': B3=0 '#': B3=1
LCT LCT#	Lz,@HL	0000 100Z ZZZZ B001	(Lz) '#': HL	← (7SEG ← (@HL)) ← HL+1	' ': B3=0 '#': B3=1
LCB LCB#	Lz,@HL	0000 100Z ZZZZ B010	(Lz) '#': HL	← (7SEG ← (@HL)) ← HL+1  Blank Zero	' ': B3=0 '#': B3=1
LCP LCP#	Lz,@HL	0000 100Z ZZZZ B011	(Lz) '#': HL	← (@HL) & AC ← HL+1	' ': B3=0 '#': B3=1
LCDX	D	0000 100D DD00 0100	Multi-Lz  D=000 : Multi-Lz=00H~0FH D=001 : Multi-Lz=10H~1FH D=010 : Multi-Lz=20H~2FH D=011 : Multi-Lz=30H~3FH D=100 : Multi-Lz=40H~4FH D=101 : Multi-Lz=50H~5FH D=110 : Multi-Lz=60H~6FH D=111 : Multi-Lz=70H~7FH	← T@HL	R/W Data no match for LCD R/W Mode
LCTX	D	0000 100D DD00 0101	Multi-Lz	← (7SEG ← @HL)	R/W Data no match for LCD R/W Mode
LCBX	D	0000 100D DD00 0110	Multi-Lz	← (7SEG ← @HL)  Blank Zero	R/W Data no match for LCD R/W Mode
LCPX	D	0000 100D DD00 0111	Multi-Lz	← @HL & AC	R/W Data no match for LCD R/W Mode
LCE LCE#	Lz,@HL	0000 100Z ZZZZ 110B	(Lz) '#': HL	← (@HL)8 bits ← HL+2	' ': B0=0 '#': B0=1
LCE LCE#	Lz,@ZR	0000 100Z ZZZZ 111B	(Lz) '#': ZR	← (@ZR)8 bits ← ZR+2	' ': B0=0 '#': B0=1
OPA	Rx	0000 1010 OXXX XXXX	Port(A) Port(A,B,C,D)	← (Rx) ← (Rx)H	If HMIOA=0 If HMIOA=1
OPAS	Rx,D	0000 1011 DXXX XXXX	A1,2,3,4	← (Rx)0,(Rx)1,D,Pulse	
OPB	Rx	0000 1100 OXXX XXXX	Port(B)	← (Rx)	
OPC	Rx	0000 1101 OXXX XXXX	Port(C)	← (Rx)	
OPD	Rx	0000 1110 OXXX XXXX	Port(D)	← (Rx)	
OPE	Rx	0000 1110 1XXX XXXX	Port(E)	← (Rx)	
SMUI	Rx	0000 1111 OXXX XXXX	MUI(H)	← (Rx)(H)	' ': B6=0 '#': B6=1
SMUI SMUI#	@HL	0000 1111 1B00 0000	MUI '#': HL	← (@HL) ← HL+1	
SMUI SMUI#	@ZR	0000 1111 1B10 0000	MUI '#': ZR	← (@ZR) ← ZR+1	
SMUIH SMUIH#	@HL	0000 1111 1B00 0100	MUIH '#': HL	← (@HL)H ← HL+4	
SMUIH	@ZR	0000 1111 1B10 0100	MUIH	← (@ZR)H	

Instruction		Machine Code	Function		Flag/Remark
SMUIH#			'#': ZR	← ZR+4	
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 FRQ#	D,@HL	0001 01DD 0000 B000	FREQ '#': HL	← T(@HL) ← HL+1	' ': B3=0 '#': B3=1
FRQX	D,X	0001 10DD XXXX XXXX	FREQ	← X	
MVL	Rx	0001 1100 0XXX XXXX	IDBF0~3	← (Rx)	
MRL	Rx	0001 1100 1XXX XXXX	ZRBF0~3	← (Rx)	
MVH	Rx	0001 1101 0XXX XXXX	IDBF4~7	← (Rx)	
MRH	Rx	0001 1101 1XXX XXXX	ZRBF4~7	← (Rx)	
MVU	Rx	0001 1110 0XXX XXXX	IDBF8~11	← (Rx)	
MRU	Rx	0001 1110 1XXX XXXX	ZRBF8~11	← (Rx)	
MVV	Rx	0001 1111 0XXX XXXX	IDBF12~15	← (Rx)	If HMIDV=0
			IDBF0~15	← (Rx)H	If HMIDV=1
MRV	Rx	0001 1111 1XXX XXXX	ZRBF12	← (Rx)0	If HMIDV=0
			ZRBF0~12	← (Rx)H	If HMIDV=1
ADC ADC*	Rx	0010 000B 0XXX XXXX	AC(H) BCD(AC(H)) '*': (Rx)(H)	← (Rx)(H) + AC(H) + CF ← If RXDA=1 ← AC(H)	CF , CFN3~0 ' ': B8=0 '*': B8=1
ADC ADC# ADC* ADC*# ADCM ADCM# ADCM* ADCM*#	@HL	0010 000B 1B00 B000	AC '*': (@HL) '#': HL	← (@HL) + ? + CF ← AC ← HL+1	' ': B8=0 '*': B8=1 ' ': B6=0 '#': B6=1
ADC ADC# ADC* ADC*# ADCM ADCM# ADCM* ADCM*#	@HL,DA	0010 000B 1B01 B000	AC '*': (@HL) '#': HL	← BCD((@HL)+?+CF) ← AC ← HL+1	'ADC' : ? = AC 'ADCM' : ? = MU/MUI by ALUMUI=0/1
ADC ADC# ADC* ADC*# ADCM ADCM# ADCM* ADCM*#	@ZR	0010 000B 1B10 B000	AC '*': (@ZR) '#': ZR	← (@ZR) + ? + CF ← AC ← ZR+1	
ADC ADC# ADC* ADC*# ADCM ADCM# ADCM* ADCM*#	@ZR,DA	0010 000B 1B11 B000	AC '*': (@ZR) '#': ZR	← BCD((@ZR)+?+CF) ← AC ← ZR+1	

Instruction	Machine Code	Function	Flag/Remark	
ADCH ADCH# ADCH* ADCH*# ADCMH ADCMH# ADCMH* ADCMH*#	@HL 0010 000B 1B00 B100	ACH ‘*’: (@HL)H ‘#’: HL	← (@HL)H + ?H + CF ← ACH ← HL+4	‘ADCH’: ? = ACH ‘ADCMH’: ? = MUH
ADCH ADCH# ADCH* ADCH*# ADCMH ADCMH# ADCMH* ADCMH*#	@HL,DA 0010 000B 1B01 B100	ACH ‘*’: (@HL)H ‘#’: HL	← BCD((@HL)H+?+CF) ← ACH ← HL+4	
ADCH ADCH# ADCH* ADCH*# ADCMH ADCMH# ADCMH* ADCMH*#	@ZR 0010 000B 1B10 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	← (@ZR)H + ? + CF ← ACH ← ZR+4	
ADCH ADCH# ADCH* ADCH*# ADCMH ADCMH# ADCMH* ADCMH*#	@ZR,DA 0010 000B 1B11 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	← BCD((@ZR)H+?+CF) ← ACH ← ZR+4	
ADC ADC& ADC% ADC\$ ADC^ ADC^& ADC^% ADC^\$	@HL,@ZR 0010 0000 1B0B 001B	AC AC,(@HL) ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	← (@HL)+(@ZR)+CF ← (@ZR)+AC+CF :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1 ‘^’: B4=0 ‘^’: B4=1 ‘ ’: B6,0=00 ‘&’: B6,0=10/01 for @HL/ZR,@ZR/HL ‘%’: B6,0=01/10 for @HL/ZR,@ZR/HL ‘\$’: B6,0=11
ADC* ADC*& ADC*% ADC*\$ ADC^* ADC^*& ADC^*% ADC^*\$	@HL,@ZR 0010 0001 1B0B 001B	AC,(@HL) (AC,(@HL)) ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	← (@HL)+(@ZR)+CF (← (@ZR)+AC+CF) :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADC ADC& ADC% ADC\$	@ZR,@HL 0010 0000 1B1B 001B	AC AC,(ZR) ‘^’	← (@ZR)+(@HL)+CF ← (@HL)+AC+CF :BCD	If ALUHLZR=0 If ALUHLZR=1

Instruction	Machine Code	Function	Flag/Remark	
ADC^ ADC^& ADC^% ADC^\$		'&,\$': HL '%,\$': ZR	← HL+1 ← ZR+1	
ADC* ADC*& ADC*% ADC*\$ ADC^* ADC^*& ADC^*% ADC^*\$	@ZR,@HL 0010 0001 1B1B 001B	AC,(ZR) (AC,@ZR)) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+(@HL)+CF (←(@HL)+AC+CF) :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADCH ADCH& ADCH% ADCH\$ ADCH^ ADCH^& ADCH^% ADCH^\$ ADCH^*	@HL,@ZR 0010 0000 1B0B 011B	ACH ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)H+CF ←(@ZR)H+ACH+CF :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADCH* ADCH*& ADCH*% ADCH*\$ ADCH^*& ADCH^*% ADCH^*\$	@HL,@ZR 0010 0001 1B0B 011B	ACH,(@HL)H (ACH,@HL)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)H+CF (←(@ZR)H+ACH+CF) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADCH ADCH& ADCH% ADCH\$ ADCH^ ADCH^& ADCH^% ADCH^\$	@ZR,@HL 0010 0000 1B1B 011B	ACH ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+(@HL)H+CF ←(@HL)H+ACH+CF :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADCH* ADCH*& ADCH*% ADCH*\$ ADCH^*& ADCH^*% ADCH^*\$	@ZR,@HL 0010 0001 1B1B 011B	ACH,(@ZR)H (ACH,@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+ACH+CF (←(@HL)H+ACH+CF) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADCM ADCM& ADCM% ADCM\$ ADCM^ ADCM^& ADCM^% ADCM^\$	@HL,@ZR 0010 0000 1B0B 101B	AC,(@HL) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+?+CF :BCD ← HL+1 ← ZR+1	Don't care ALUHLZR ? ? = MU/MUI by ALUMUI=0/1 '^': B4=0 '^': B4=1
ADCM ADCM& ADCM%	@ZR,@HL 0010 0000 1B1B 101B	AC,(@ZR) '^' '&,\$': HL	←(@HL)+?+CF :BCD ← HL+1	'^': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for

Instruction	Machine Code	Function	Flag/Remark
ADCM\$ ADCM^ ADCM^& ADCM^% ADCM^\$		'%, \$': ZR ← ZR+1	@HL/ZR, @ZR/HL '\$': B6,0=11
ADCMH ADCMH& ADCMH% ADCMH\$ ADCMH^ ADCMH^& ADCMH^% ADCMH^\$	@HL, @ZR 0010 0000 1B0B 111B	ACH, (@HL)H '^': '&, \$': HL '%, \$': ZR ← (@ZR)H+?+CF :BCD ← HL+4 ← ZR+4	? = MUH/MUIH by ALUMUI=0/1
ADCMH ADCMH& ADCMH% ADCMH\$ ADCMH^ ADCMH^& ADCMH^% ADCMH^\$	@ZR, @HL 0010 0000 1B1B 111B	ACH, (@ZR)H '^': '&, \$': HL '%, \$': ZR ← (@HL)H+?+CF :BCD ← HL+4 ← ZR+4	
SBC SBC*	Rx 0010 001B 0XXX XXXX	AC(H) BCD(AC(H)) '*': (Rx)(H) ← (Rx)(H) + ACB(H) + CF ← If RXDA=1 ← AC(H)	CF, CFN3~0 '*': B8=0
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@HL 0010 001B 1B00 B000	AC '*': (@HL) '#': HL ← (@HL) + ? + CF ← AC ← HL+1	'*': B8=1 '#': B6=0 '#': B6=1
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@HL, DA 0010 001B 1B01 B000	AC '*': (@HL) '#': HL ← BCD((@HL)+?+CF) ← AC ← HL+1	'SBC' : ? = ACB 'SBCM' : ? = MUB/MUIB by ALUMUI=0/1 'SBC(H)' : B3=0 'SBCM(H)' : B3=1
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@ZR 0010 001B 1B10 B000	AC '*': (@ZR) '#': ZR ← (@ZR) + ? + CF ← AC ← ZR+1	
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@ZR, DA 0010 001B 1B11 B000	AC '*': (@ZR) '#': ZR ← BCD((@ZR)+?+CF) ← AC ← ZR+1	

Instruction	Machine Code	Function	Flag/Remark
SBCH SBCH# SBCH* SBCH*# SBCM# SBCM# SBCM* SBCM*#	@HL 0010 001B 1B00 B100	ACH '*': (@HL)H '#': HL	← (@HL)H + ?H + CF ← ACH ← HL+4  'SBCH' : ? = ACBH 'SBCM#': ? = MUBH/MUI BH by ALUMUI=0/1
SBCH SBCH# SBCH* SBCH*# SBCM# SBCM# SBCM* SBCM*#	@HL,DA 0010 001B 1B01 B100	ACH '*': (@HL)H '#': HL	← BCD((@HL)H+? +CF) ← ACH ← HL+4
SBCH SBCH# SBCH* SBCH*# SBCM# SBCM# SBCM* SBCM*#	@ZR 0010 001B 1B10 B100	ACH '*': (@ZR)H '#': ZR	← (@ZR)H + ? + CF ← ACH ← ZR+4
SBCH SBCH# SBCH* SBCH*# SBCM# SBCM# SBCM* SBCM*#	@ZR,DA 0010 001B 1B11 B100	ACH '*': (@ZR)H '#': ZR	← BCD((@ZR)H+?+CF) ← ACH ← ZR+4
SBC SBC& SBC% SBC\$ SBC^ SBC^& SBC^% SBC^\$	@HL,@ZR 0010 0010 1B0B 001B	AC AC,(@HL)  '^' '&,\$': HL '%,\$': ZR	← (@HL)+(@ZR)B+CF ← (@ZR)+ACB+CF  :BCD ← HL+1 ← ZR+1  If ALUHLZR=0 If ALUHLZR=1  ' ': B4=0 '^': B4=1  ' ': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL '\$': B6,0=11
SBC* SBC*& SBC*% SBC*\$ SBC^* SBC^*& SBC^*% SBC^*\$	@HL,@ZR 0010 0011 1B0B 001B	AC,(@HL) (AC,(@HL))  '^' '&,\$': HL '%,\$': ZR	← (@HL)+(@ZR)B+CF (← (@ZR)+ACB+CF)  :BCD ← HL+1 ← ZR+1  If ALUHLZR=0 (If ALUHLZR=1)
SBC SBC& SBC% SBC\$	@ZR,@HL 0010 0010 1B1B 001B	AC AC,(ZR)  '^'	← (@ZR)+(@HL)B+CF ← (@HL)+ACB+CF  :BCD

Instruction	Machine Code	Function	Flag/Remark	
SBC^ SBC^& SBC^% SBC^\$		'&,\$': HL '%,\$': ZR	← HL+1 ← ZR+1	
SBC* SBC*& SBC*% SBC*\$ SBC^* SBC^*& SBC^*% SBC^*\$	@ZR,@HL 0010 0011 1B1B 001B	AC,(ZR) (AC,(ZR)) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+(@HL)B+CF (←(@HL)+ACB+CF) :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
SBCH SBCH& SBCH% SBCH\$ SBCH^ SBCH^& SBCH^% SBCH^\$ SBCH^*	@HL,@ZR 0010 0010 1B0B 011B	ACH ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)BH+CF ←(@ZR)H+ACBH+CF :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
SBCH* SBCH*& SBCH*% SBCH*\$ SBCH^*& SBCH^*% SBCH^*\$	@HL,@ZR 0010 0011 1B0B 011B	ACH,(@HL)H (ACH,(@HL)) '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)BH+CF (←(@ZR)H+ACBH+CF) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
SBCH SBCH& SBCH% SBCH\$ SBCH^ SBCH^& SBCH^% SBCH^\$	@ZR,@HL 0010 0010 1B1B 011B	ACH ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+(@HL)BH+CF ←(@HL)H+ACBH+CF :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
SBCH* SBCH*& SBCH*% SBCH*\$ SBCH^* SBCH^*& SBCH^*% SBCH^*\$	@ZR,@HL 0010 0011 1B1B 011B	ACH,(@ZR)H (ACH,(@ZR)H) '^' '&,\$': HL '%,\$': ZR	←(@HL)H+ACBH+CF (←(@HL)H+ACBH+CF) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
SBCM SBCM& SBCM% SBCM\$ SBCM^ SBCM^& SBCM^% SBCM^\$	@HL,@ZR 0010 0010 1B0B 101B	AC,(@HL) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+?+CF :BCD ← HL+1 ← ZR+1	Don't care ALUHLZR ? ? = MUB/MUIB by ALUMUI=0/1 '^': B4=0 '^': B4=1
SBCM SBCM& SBCM% SBCM\$	@ZR,@HL 0010 0010 1B1B 101B	AC,(@ZR) '^' '&,\$': HL '%,\$': ZR	←(@HL)+?+CF :BCD ← HL+1 ← ZR+1	'^': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL

Instruction	Machine Code	Function	Flag/Remark	
SBCM^ SBCM^& SBCM^% SBCM^\$			'\$' : B6,0=11	
SBCM <sup>H</sup> SBCM <sup>H</sup> & SBCM <sup>H</sup> % SBCM <sup>H</sup> \$ SBCM <sup>H</sup> ^ SBCM <sup>H</sup> ^& SBCM <sup>H</sup> ^% SBCM <sup>H</sup> ^\$	@HL,@ZR  0010 0010 1B0B 111B	ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+?+CF :BCD ← HL+4 ← ZR+4  ? = MUBH/MUIBH by ALUMUI=0/1	
SBCM <sup>H</sup> SBCM <sup>H</sup> & SBCM <sup>H</sup> % SBCM <sup>H</sup> \$ SBCM <sup>H</sup> ^ SBCM <sup>H</sup> ^& SBCM <sup>H</sup> ^% SBCM <sup>H</sup> ^\$	@ZR,@HL  0010 0010 1B1B 111B	ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+?+CF :BCD ← HL+4 ← ZR+4	
ADD ADD*	Rx  0010 010B 0XXX XXXX	AC(H) BCD(AC(H)) '*': (Rx)(H)	← (Rx)(H) + AC(H) ← If RXDA=1 ← AC(H)	CF , CFN3~0
ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#	@HL  0010 010B 1B00 B000	AC '*': (@HL) '#': HL	← (@HL) + ? ← AC ← HL+1	'*': B8=0 '*': B8=1  '*': B6=0 '#': B6=1
ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#	@HL,DA  0010 010B 1B01 B000	AC '*': (@HL) '#': HL	←BCD((@HL)+?) ← AC ← HL+1	'ADD(H)': B3=0 'ADDM(H)': B3=1
ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#	@ZR  0010 010B 1B10 B000	AC '*': (@ZR) '#': ZR	← (@ZR) + ? ← AC ← ZR+1	
ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#	@ZR,DA  0010 010B 1B11 B000	AC '*': (@ZR) '#': ZR	←BCD((@ZR)+?) ← AC ← ZR+1	

Instruction		Machine Code	Function		Flag/Remark
ADDH ADDH# ADDH* ADDH** ADDMH ADDMH# ADDMH* ADDMH**	@HL	0010 010B 1B00 B100	ACH ‘*’: (@HL)H ‘#’: HL	←(@HL)H + ?H ← ACH ← HL+4	‘ADDH’: ? = ACH ‘ADDMH’: ? = MUH/MUIH by ALUMUI=0/1
ADDH ADDH# ADDH* ADDH** ADDMH ADDMH# ADDMH* ADDMH**	@HL,DA	0010 010B 1B01 B100	ACH ‘*’: (@HL)H ‘#’: HL	←BCD((@HL)H+?) ← ACH ← HL+4	
ADDH ADDH# ADDH* ADDH** ADDMH ADDMH# ADDMH* ADDMH**	@ZR	0010 010B 1B10 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	← (@ZR)H + ? ← ACH ← ZR+4	
ADDH ADDH# ADDH* ADDH** ADDMH ADDMH# ADDMH* ADDMH**	@ZR,DA	0010 010B 1B11 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	←BCD((@ZR)H+?) ← ACH ← ZR+4	
ADD ADD& ADD% ADD\$ ADD^ ADD^& ADD^% ADD^\$	@HL,@ZR	0010 0100 1B0B 001B	AC AC,(@HL)  ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	←(@HL)+(@ZR) ←(@ZR)+AC  :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1  ‘^’: B4=0 ‘^’:B4=1  ‘ ’: B6,0=00 ‘&’: B6,0=10/01 for @HL/ZR,@ZR/HL ‘%’: B6,0=01/10 for @HL/ZR,@ZR/HL ‘\$’: B6,0=11
ADD* ADD*& ADD*% ADD*\$ ADD^* ADD^*& ADD^*% ADD^*\$	@HL,@ZR	0010 0101 1B0B 001B	AC,(@HL) (AC,(@HL))  ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	←(@HL)+(@ZR) (←(@ZR)+AC)  :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADD ADD& ADD% ADD\$	@ZR,@HL	0010 0100 1B1B 001B	AC AC,(ZR)  ‘^’	←(@ZR)+(@HL) ←(@HL)+AC  :BCD	If ALUHLZR=0 If ALUHLZR=1

Instruction	Machine Code	Function	Flag/Remark	
ADD^ ADD^& ADD^% ADD^\$		'&,\$': HL '%,\$': ZR	← HL+1 ← ZR+1	
ADD* ADD*& ADD*% ADD*\$ ADD^* ADD^*& ADD^*% ADD^*\$	@ZR,@HL 0010 0101 1B1B 001B	AC,(ZR) (AC,(ZR)) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+(@HL) (←(@HL)+AC) :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADDH ADDH& ADDH% ADDH\$ ADDH^ ADDH^& ADDH^% ADDH^\$ ADDH^*	@HL,@ZR 0010 0100 1B0B 011B	ACH ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)H ←(@ZR)H+ACH :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADDH* ADDH*& ADDH*% ADDH*\$ ADDH^*& ADDH^*% ADDH^*\$	@HL,@ZR 0010 0101 1B0B 011B	ACH,(@HL)H (ACH,(@HL)H) '^' '&,\$': HL '%,\$': ZR	←(@HL)H+(@ZR)H (←(@ZR)H+ACH) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADDH ADDH& ADDH% ADDH\$ ADDH^ ADDH^& ADDH^% ADDH^\$	@ZR,@HL 0010 0100 1B1B 011B	ACH ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+(@HL)H ←(@HL)H+ACH :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADDH* ADDH*& ADDH*% ADDH*\$ ADDH^*& ADDH^*% ADDH^*\$	@ZR,@HL 0010 0101 1B1B 011B	ACH,(@ZR)H (ACH,(@ZR)H) '^' '&,\$': HL '%,\$': ZR	←(@HL)H+ACH (←(@HL)H+ACH) :BCD ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADDM ADDM& ADDM% ADDM\$ ADDM^ ADDM^& ADDM^% ADDM^\$	@HL,@ZR 0010 0100 1B0B 101B	AC,(@HL) '^' '&,\$': HL '%,\$': ZR	←(@ZR)+? :BCD ← HL+1 ← ZR+1	Don't care ALUHLZR ? ? = MU/MUI by ALUMUI=0/1 '^': B4=0 '^': B4=1
ADDM ADDM& ADDM% ADDM\$	@ZR,@HL 0010 0100 1B1B 101B	AC,(@ZR) '^' '&,\$': HL '%,\$': ZR	←(@HL)+? :BCD ← HL+1 ← ZR+1	'^': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL

Instruction	Machine Code	Function	Flag/Remark	
ADDM^ ADDM^& ADDM^% ADDM^\$			'\$': B6,0=11	
ADDMH ADDMH& ADDMH% ADDMH\$ ADDMH^ ADDMH^& ADDMH^% ADDMH^\$	@HL,@ZR 0010 0100 1B0B 111B	ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+? :BCD ← HL+4 ← ZR+4  ? = MUH/MUIH by ALUMUI=0/1	
ADDMH ADDMH& ADDMH% ADDMH\$ ADDMH^ ADDMH^& ADDMH^% ADDMH^\$	@ZR,@HL 0010 0100 1B1B 111B	ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+? :BCD ← HL+4 ← ZR+4	
SUB SUB*	Rx 0010 011B 0XXX XXXX	AC(H) BCD(AC(H))	← (Rx)(H) + ACB(H) + 1 ← If RXDA=1	CF , CFN3~0
SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*#	@HL 0010 011B 1B00 B000	AC '*': (@HL) '#': HL	← AC(H) ← (@HL) + ? + 1 ← AC ← HL+1	'*': B8=0 '*': B8=1  '*': B6=0 '#': B6=1
SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*#	@HL,DA 0010 011B 1B01 B000	AC '*': (@HL) '#': HL	←BCD((@HL)+?+1) ← AC ← HL+1	'SUB' : ? = ACB 'SUBM' : ? = MUB/MUIB by ALUMUI=0/1  'SUB(H)' : B3=0 'SUBM(H)' : B3=1
SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*#	@ZR 0010 011B 1B10 B000	AC '*': (@ZR) '#': ZR	← (@ZR) + ? + 1 ← AC ← ZR+1	
SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*#	@ZR,DA 0010 011B 1B11 B000	AC '*': (@ZR) '#': ZR	←BCD((@ZR)+?+1) ← AC ← ZR+1	

Instruction		Machine Code	Function		Flag/Remark
SUBH SUBH# SUBH* SUBH*# SUBMH SUBMH# SUBMH* SUBMH*#	@HL	0010 011B 1B00 B100	ACH ‘*’: (@HL)H ‘#’: HL	← (@HL)H + ?H + 1 ← ACH ← HL+4	‘SBCH’: ? = ACBH ‘SBCMH’: ? = MUBH/MUI BH by ALUMUI=0/1
SUBH SUBH# SUBH* SUBH*# SUBMH SUBMH# SUBMH* SUBMH*#	@HL,DA	0010 011B 1B01 B100	ACH ‘*’: (@HL)H ‘#’: HL	← BCD((@HL)H+?+1) ← ACH ← HL+4	
SUBH SUBH# SUBH* SUBH*# SUBMH SUBMH# SUBMH* SUBMH*#	@ZR	0010 011B 1B10 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	← (@ZR)H + ? + 1 ← ACH ← ZR+4	
SUBH SUBH# SUBH* SUBH*# SUBMH SUBMH# SUBMH* SUBMH*#	@ZR,DA	0010 011B 1B11 B100	ACH ‘*’: (@ZR)H ‘#’: ZR	← BCD((@ZR)H+?+1) ← ACH ← ZR+4	
SUB SUB& SUB% SUB\$ SUB^ SUB^& SUB^% SUB^\$	@HL,@ZR	0010 0110 1B0B 001B	AC AC,(@HL) ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	← (@HL)+(@ZR)B ← (@ZR)+ACB :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1 ‘^’: B4=0 ‘^’: B4=1 ‘ ’: B6,0=00 ‘&’: B6,0=10/01 for @HL/ZR,@ZR/HL ‘%’: B6,0=01/10 for @HL/ZR,@ZR/HL ‘\$’: B6,0=11
SUB* SUB*& SUB*% SUB*\$ SUB^* SUB^*& SUB^*% SUB^*\$	@HL,@ZR	0010 0111 1B0B 001B	AC,(@HL) (AC,(@HL)) ‘^’ ‘&,\$’: HL ‘%,\$’: ZR	← (@HL)+(@ZR)B (← (@ZR)+ACB) :BCD ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
SUB SUB& SUB% SUB\$	@ZR,@HL	0010 0110 1B1B 001B	AC AC,(ZR) ‘^’	← (@ZR)+(@HL)B ← (@HL)+ACB :BCD	If ALUHLZR=0 If ALUHLZR=1

Instruction	Machine Code	Function	Flag/Remark
SUB^ SUB^& SUB^% SUB^\$		'&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	
SUB* SUB*& SUB*% SUB*\$ SUB^* SUB^*& SUB^*% SUB^*\$	@ZR,@HL 0010 0111 1B1B 001B	AC,(ZR) (AC,(ZR)) ←(@ZR)+(@HL)B (←(@HL)+ACB) '^' :BCD '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
SUBH SUBH& SUBH% SUBH\$ SUBH^ SUBH^& SUBH^% SUBH^\$ SUBH^*	@HL,@ZR 0010 0110 1B0B 011B	ACH ACH,(@HL) ←(@HL)H+(@ZR)BH ←(@ZR)H+ACBH '^' :BCD '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
SUBH* SUBH*& SUBH*% SUBH*\$ SUBH^*& SUBH^*% SUBH^*\$	@HL,@ZR 0010 0111 1B0B 011B	ACH,(@HL)H (ACH,(@HL)) ←(@HL)H+(@ZR)BH (←(@ZR)H+ACBH) '^' :BCD '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
SUBH SUBH& SUBH% SUBH\$ SUBH^ SUBH^& SUBH^% SUBH^\$	@ZR,@HL 0010 0110 1B1B 011B	ACH ACH,(@ZR)H ←(@ZR)H+(@HL)BH ←(@HL)H+ACBH '^' :BCD '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
SUBH* SUBH*& SUBH*% SUBH*\$ SUBH^*& SUBH^*% SUBH^*\$	@ZR,@HL 0010 0111 1B1B 011B	ACH,(@ZR)H (ACH,(@ZR)H) ←(@HL)H+ACBH (←(@HL)H+ACBH) '^' :BCD '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
SUBM SUBM& SUBM% SUBM\$ SUBM^ SUBM^& SUBM^% SUBM^\$	@HL,@ZR 0010 0110 1B0B 101B	AC,(@HL) '^' :BCD '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	Don't care ALUHLZR ? ? = MUB/MUIB by ALUMUI=0/1 '^': B4=0 '^': B4=1
SUBM SUBM& SUBM% SUBM\$	@ZR,@HL 0010 0110 1B1B 101B	AC,(@ZR) '^' :BCD '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	'^': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL

Instruction	Machine Code	Function	Flag/Remark
SUBM^ SUBM^& SUBM^% SUBM^\$			'\$': B6,0=11
SUBMH SUBMH& SUBMH% SUBMH\$ SUBMH^ SUBMH^& SUBMH^% SUBMH^\$	@HL,@ZR	0010 0110 1B0B 111B ACH,(@HL)H '^' '&,\$': HL '%,\$': ZR	←(@ZR)H+? :BCD ← HL+4 ← ZR+4 ? ? = MUBH/MUIBH by ALUMUI=0/1
SUBMH SUBMH& SUBMH% SUBMH\$ SUBMH^ SUBMH^& SUBMH^% SUBMH^\$	@ZR,@HL	0010 0110 1B1B 111B ACH,(@ZR)H '^' '&,\$': HL '%,\$': ZR	←(@HL)H+? :BCD ← HL+4 ← ZR+4
ADN ADN*	Rx	0010 100B 0XXX XXXX AC(H) '*': (Rx)(H)	← (Rx)(H) + AC(H) ← AC(H) '*': B8=0 '*': B8=1
ADN ADN# ADN* ADN*# ADNM ADNM# ADNM* ADNM*#	@HL	0010 100B 1B00 B000 AC '*': (@HL) '#': HL	← (@HL) + ? ← AC ← HL+1 '*': B6=0 '#': B6=1 'ADN' : ? = AC 'ADNM' : ? = MUI
ADN ADN# ADN* ADN*# ADNM ADNM# ADNM* ADNM*#	@ZR	0010 100B 1B10 B000 AC '*': (@ZR) '#': ZR	← (@ZR) + ? ← AC ← ZR+1
ADNH ADNH# ADNH* ADNH*# ADNMH ADNMH# ADNMH* ADNMH*#	@HL	0010 100B 1B00 B100 ACH '*': (@HL)H '#': HL	←(@HL)H + ? ← ACH ← HL+4 'ADNH' : ? = ACH 'ADNMH' : ? = MUIH
ADNH ADNH# ADNH* ADNH*# ADNMH ADNMH# ADNMH* ADNMH*#	@ZR	0010 100B 1B10 B100 ACH '*': (@ZR)H '#': ZR	← (@ZR)H + ? ← ACH ← ZR+4
ADN	@HL,@ZR	0010 1000 0B00 001B AC	←(@HL)+(@ZR) If ALUHLZR=0

Instruction	Machine Code	Function	Flag/Remark
ADN& ADN% ADN\$		AC,(@HL) ← (@ZR)+AC  '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	If ALUHLZR=1  ' ': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL '\$': B6,0=11
ADN* ADN*& ADN*% ADN*\$	@HL,@ZR 0010 1001 0B00 001B	AC,(@HL) (AC,(@HL)) ←(@HL)+(@ZR) (← (@ZR)+AC)  '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADN ADN& ADN% ADN\$	@ZR,@HL 0010 1000 1B10 001B	AC AC,(@ZR) ←(@ZR)+(@HL) ←(@HL)+AC  '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	If ALUHLZR=0 If ALUHLZR=1
ADN* ADN*& ADN*% ADN*\$	@ZR,@HL 0010 1001 1B10 001B	AC,(@ZR) (AC,(@ZR)) ←(@ZR)+(@HL) (← (@HL)+AC)  '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ADNH ADNH& ADNH% ADNH\$	@HL,@ZR 0010 1000 1B00 011B	ACH ACH,(@HL)H ←(@HL)H+(@ZR)H ← (@ZR)H+ACH  '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADNH* ADNH*& ADNH*% ADNH*\$	@HL,@ZR 0010 1001 1B00 011B	ACH,(@HL)H (ACH,(@HL)H) ←(@HL)H+(@ZR)H (← (@ZR)H+ACH)  '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADNH ADNH& ADNH% ADNH\$	@ZR,@HL 0010 1000 1B10 011B	ACH ACH,(@ZR)H ←(@ZR)H+(@HL)H ← @HL)H+ACH  '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ADNH* ADNH*& ADNH*% ADNH*\$	@ZR,@HL 0010 1001 1B10 011B	ACH,(@ZR)H (ACH,(@ZR)H) ←(@ZR)H+(@HL)H (← @HL)H+ACH)  '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ADNM ADNM& ADNM% ADNMS	@HL,@ZR 0010 1000 1B00 101B	AC,(@HL) '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	Don't care ALUHLZR
ADNM ADNM& ADNM% ADNMS	@ZR,@HL 0010 1000 1B10 101B	AC,(@ZR) '&,\$': HL ← HL+1 '%,\$': ZR ← ZR+1	
ADNMH ADNMH& ADNMH% ADNMH\$	@HL,@ZR 0010 1000 1B00 111B	ACH,(@HL)H '&,\$': HL ← HL+4 '%,\$': ZR ← ZR+4	
ADNMH	@ZR,@HL 0010 1000 1B10 111B	ACH,(@ZR)H ←(@HL)H+MUI	

Instruction		Machine Code	Function		Flag/Remark
ADNMH& ADNMH% ADNMH\$			'&,\$': HL '%,\$': ZR	← HL+4 ← ZR+4	
AND AND*	Rx	0010 101B 0XXX XXXX	AC(H) '*': (Rx)(H)	← (Rx)(H) AND AC(H) ← AC(H)	' ': B8=0 '*': B8=1
AND AND# AND* AND*# ANDM ANDM# ANDM* ANDM*#	@HL	0010 101B 1B00 B000	AC '*': (@HL) '#': HL	← (@HL) AND ? ← AC ← HL+1	' ': B6=0 '#': B6=1  'ADN' : ? = AC 'ADNM' : ? = MUI  'ADN(H)' : B3=0 'ADNM(H)' : B3=1
AND AND# AND* AND*# ANDM ANDM# ANDM* ANDM*#	@ZR	0010 101B 1B10 B000	AC '*': (@ZR) '#': ZR	← (@ZR) AND ? ← AC ← ZR+1	
ANDH ANDH# ANDH* ANDH*# ANDMH ANDMH# ANDMH* ANDMH*#	@HL	0010 101B 1B00 B100	ACH '*': (@HL)H '#': HL	← (@HL)H AND ? ← ACH ← HL+4	'ADNH' : ? = ACH 'ADNMH' : ? = MUIH
ANDH ANDH# ANDH* ANDH*# ANDMH ANDMH# ANDMH* ANDMH*#	@ZR	0010 101B 1B10 B100	ACH '*': (@ZR)H '#': ZR	← (@ZR)H AND ? ← ACH ← ZR+4	
AND AND& AND% AND\$	@HL,@ZR	0010 1010 0B00 001B	AC AC,(@HL)  '&,\$': HL '%,\$': ZR	← (@HL) AND (@ZR) ← (@ZR) AND AC  ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1  ' ': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL '\$': B6,0=11
AND* AND*& AND*% AND*\$	@HL,@ZR	0010 1011 0B00 001B	AC,(@HL) (AC,(@HL))  '&,\$': HL '%,\$': ZR	← (@HL) AND (@ZR) (← (@ZR) AND AC)  ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
AND AND& AND% AND\$	@ZR,@HL	0010 1010 1B10 001B	AC AC, (@ZR)  '&,\$': HL	← (@ZR) AND (@HL) ← (@HL) AND AC  ← HL+1	If ALUHLZR=0 If ALUHLZR=1

Instruction		Machine Code	Function		Flag/Remark
AND* AND*& AND*% AND*\$	@ZR,@HL	0010 1011 1B10 001B	'%,\$': ZR AC, (@ZR) (AC, (@ZR))	← ZR+1 ← (@ZR) AND (@HL) (← (@HL) AND AC)	If ALUHLZR=0 (If ALUHLZR=1)
ANDH ANDH& ANDH% ANDH\$	@HL,@ZR	0010 1010 1B00 011B	'&,\$': HL '%,\$': ZR ACH ACH,(@HL)H	← HL+1 ← ZR+1 ← (@HL)H AND (@ZR)H ← (@ZR)H AND ACH	If ALUHLZR=0 If ALUHLZR=1
ANDH* ANDH*& ANDH*% ANDH*\$	@HL,@ZR	0010 1011 1B00 011B	'&,\$': HL '%,\$': ZR ACH,(@HL)H (ACH,(@HL)H)	← HL+4 ← ZR+4 ← (@HL)H AND (@ZR)H (← (@ZR)H AND ACH)	If ALUHLZR=0 (If ALUHLZR=1)
ANDH ANDH& ANDH% ANDH\$	@ZR,@HL	0010 1010 1B10 011B	'&,\$': HL '%,\$': ZR ACH ACH,(@ZR)H	← HL+4 ← ZR+4 ← (@ZR)H AND (@HL)H ← @HL)H AND ACH	If ALUHLZR=0 If ALUHLZR=1
ANDH* ANDH*& ANDH*% ANDH*\$	@ZR,@HL	0010 1011 1B10 011B	'&,\$': HL '%,\$': ZR ACH,(@ZR)H (ACH,(@ZR)H)	← HL+4 ← ZR+4 ← (@ZR)H AND (@HL)H (← @HL)H AND ACH)	If ALUHLZR=0 (If ALUHLZR=1)
ANDM ANDM& ANDM% ANDM\$	@HL,@ZR	0010 1010 1B00 101B	'&,\$': HL '%,\$': ZR AC,(@HL)	← HL+1 ← ZR+1 ← (@ZR) AND MUI	Don't care ALUHLZR
ANDM ANDM& ANDM% ANDM\$	@ZR,@HL	0010 1010 1B10 101B	'&,\$': HL '%,\$': ZR AC,(@ZR)	← HL+1 ← ZR+1 ← (@HL) AND MUI	
ANDMH ANDMH& ANDMH% ANDMH\$	@HL,@ZR	0010 1010 1B00 111B	'&,\$': HL '%,\$': ZR ACH,(@HL)H	← HL+4 ← ZR+4 ← (@ZR)H AND MUI	
ANDMH ANDMH& ANDMH% ANDMH\$	@ZR,@HL	0010 1010 1B10 111B	'&,\$': HL '%,\$': ZR ACH,(@ZR)H	← HL+4 ← ZR+4 ← (@HL)H AND MUI	
EOR EOR*	Rx	0010 110B 0XXX XXXX	'*': (Rx)(H) AC(H)	← (Rx)(H) EOR AC(H) ← AC(H)	'*': B8=0 '*': B8=1
EOR EOR# EOR* EOR** EORM EORM# EORM* EORM*#	@HL	0010 110B 1B00 B000	'#': HL '*': (@HL) AC	← HL+1 ← AC ← (@HL) EOR ?	'*': B6=0 '#': B6=1  'ADN' : ? = AC 'ADNM' : ? = MUI  'ADN(H)' : B3=0 'ADNM(H)' : B3=1

Instruction		Machine Code	Function		Flag/Remark
EOR EOR# EOR* EOR** EORM EORM# EORM* EORM**	@ZR	0010 110B 1B10 B000	AC ** : (@ZR) # : ZR	← (@ZR) EOR ? ← AC ← ZR+1	
EORH EORH# EORH* EORH** EORMH EORMH# EORMH* EORMH**	@HL	0010 110B 1B00 B100	ACH ** : (@HL)H # : HL	← (@HL)H EOR ? ← ACH ← HL+4	'ADNH' : ? = ACH 'ADNMH' : ? = MUIH
EORH EORH# EORH* EORH** EORMH EORMH# EORMH* EORMH**	@ZR	0010 110B 1B10 B100	ACH ** : (@ZR)H # : ZR	← (@ZR)H EOR ? ← ACH ← ZR+4	
EOR EOR& EOR% EOR\$	@HL,@ZR	0010 1100 0B00 001B	AC AC,(@HL)  '&,\$': HL '%,\$': ZR	← (@HL) EOR(@ZR) ← (@ZR) EORAC  ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1  ' ': B6,0=00 '&': B6,0=10/01 for @HL/ZR,@ZR/HL '%': B6,0=01/10 for @HL/ZR,@ZR/HL '\$': B6,0=11
EOR* EOR*& EOR**% EOR*\$	@HL,@ZR	0010 1101 0B00 001B	AC,(@HL) (AC,(@HL))  '&,\$': HL '%,\$': ZR	← (@HL) EOR (@ZR) (← (@ZR) EORAC)  ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
EOR EOR& EOR% EOR\$	@ZR,@HL	0010 1100 1B10 001B	AC AC,(@ZR)  '&,\$': HL '%,\$': ZR	← (@ZR) EOR (@HL) ← (@HL) EOR AC  ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1
EOR* EOR*& EOR**% EOR*\$	@ZR,@HL	0010 1101 1B10 001B	AC,(@ZR) (AC,(@ZR))  '&,\$': HL '%,\$': ZR	← (@ZR) EOR (@HL) (← (@HL) EOR AC)  ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
EORH EORH& EORH% EORH\$	@HL,@ZR	0010 1100 1B00 011B	ACH ACH,(@HL)H  '&,\$': HL '%,\$': ZR	← (@HL)H EOR (@ZR)H ← (@ZR)H EOR ACH  ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
EORH*	@HL,@ZR	0010 1101 1B00 011B	ACH,(@HL)H	← (@HL)H EOR (@ZR)H	If ALUHLZR=0

Instruction	Machine Code	Function	Flag/Remark	
EORH*& EORH*% EORH*\$		ACH,(@HL)H '&,\$': HL '%,\$': ZR	(← (@ZR)H EOR ACH) ← HL+4 ← ZR+4	(If ALUHLZR=1)
EORH EORH& EORH% EORH\$	@ZR,@HL	0010 1100 1B10 011B ACH ACH,(@ZR)H '&,\$': HL '%,\$': ZR	←(@ZR)H EOR (@HL)H ←@HL)H EOR ACH ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
EORH* EORH*& EORH*% EORH*\$	@ZR,@HL	0010 1101 1B10 011B ACH,(@ZR)H (ACH,(@ZR)H) '&,\$': HL '%,\$': ZR	←(@ZR)H EOR (@HL)H ←@HL)H EOR ACH ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
EORM EORM& EORM% EORM\$	@HL,@ZR	0010 1100 1B00 101B AC,(@HL) '&,\$': HL '%,\$': ZR	←(@ZR) EOR MUI ← HL+1 ← ZR+1	Don't care ALUHLZR
EORM EORM& EORM% EORM\$	@ZR,@HL	0010 1100 1B10 101B AC,(@ZR) '&,\$': HL '%,\$': ZR	←(@HL) EOR MUI ← HL+1 ← ZR+1	
EORMH EORMH& EORMH% EORMH\$	@HL,@ZR	0010 1100 1B00 111B ACH,(@HL)H '&,\$': HL '%,\$': ZR	←(@ZR)H EOR MUI ← HL+4 ← ZR+4	
EORMH EORMH& EORMH% EORMH\$	@ZR,@HL	0010 1100 1B10 111B ACH,(@ZR)H '&,\$': HL '%,\$': ZR	←(@HL)H EOR MUI ← HL+4 ← ZR+4	
OR OR*	Rx	0010 111B 0XXX XXXX AC(H) '*': (Rx)(H)	← (Rx)(H) OR AC(H) ← AC(H)	'*': B8=0 '*': B8=1
OR OR# OR* OR*# ORM ORM# ORM* ORM*#	@HL	0010 111B 1B00 B000 AC '*': (@HL) '#': HL	← (@HL) OR ? ← AC ← HL+1	'*': B6=0 '#': B6=1  'ADN' : ? = AC 'ADNM' : ? = MUI
OR OR# OR* OR*# ORM ORM# ORM* ORM*#	@ZR	0010 111B 1B10 B000 AC '*': (@ZR) '#': ZR	← (@ZR) OR ? ← AC ← ZR+1	'ADN(H)' : B3=0 'ADNM(H)' : B3=1
ORH ORH# ORH* ORH*# ORMH ORMH#	@HL	0010 111B 1B00 B100 ACH '*': (@HL)H '#': HL	←(@HL)H OR ? ← ACH ← HL+4	'ADNH' : ? = ACH 'ADNMH' : ? = MUIH

Instruction	Machine Code	Function	Flag/Remark	
ORMH* ORMH*#				
ORH ORH# ORH* ORH*# ORMH ORMH# ORMH* ORMH*#	@ZR	0010 111B 1B10 B100 ACH ‘*’: (@ZR)H ‘#’: ZR	← (@ZR)H OR ? ← ACH ← ZR+4	
OR OR& OR% OR\$	@HL,@ZR	0010 1110 0B00 001B AC AC,(@HL) ‘&,\$’: HL ‘%,\$’: ZR	←(@HL) OR(@ZR) ← (@ZR) ORAC ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1 ‘ ’: B6,0=00 ‘&’: B6,0=10/01 for @HL/ZR,@ZR/HL ‘%’: B6,0=01/10 for @HL/ZR,@ZR/HL ‘\$’: B6,0=11
OR* OR*& OR*% OR*\$	@HL,@ZR	0010 1111 0B00 001B AC,(@HL) (AC,(@HL)) ‘&,\$’: HL ‘%,\$’: ZR	←(@HL) OR (@ZR) (← (@ZR) ORAC) ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
OR OR& OR% OR\$	@ZR,@HL	0010 1110 1B10 001B AC AC, (@ZR) ‘&,\$’: HL ‘%,\$’: ZR	←(@ZR) OR (@HL) ←(@HL) OR AC ← HL+1 ← ZR+1	If ALUHLZR=0 If ALUHLZR=1
OR* OR*& OR*% OR*\$	@ZR,@HL	0010 1111 1B10 001B AC, (@ZR) (AC, (@ZR)) ‘&,\$’: HL ‘%,\$’: ZR	←(@ZR) OR (@HL) (←(@HL) OR AC) ← HL+1 ← ZR+1	If ALUHLZR=0 (If ALUHLZR=1)
ORH ORH& ORH% ORH\$	@HL,@ZR	0010 1110 1B00 011B ACH ACH,(@HL)H ‘&,\$’: HL ‘%,\$’: ZR	←(@HL)H OR (@ZR)H ← (@ZR)H OR ACH ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ORH* ORH*& ORH*% ORH*\$	@HL,@ZR	0010 1111 1B00 011B ACH,(@HL)H (ACH,(@HL)H) ‘&,\$’: HL ‘%,\$’: ZR	←(@HL)H OR (@ZR)H (← (@ZR)H OR ACH) ← HL+4 ← ZR+4	If ALUHLZR=0 (If ALUHLZR=1)
ORH ORH& ORH% ORH\$	@ZR,@HL	0010 1110 1B10 011B ACH ACH,(@ZR)H ‘&,\$’: HL ‘%,\$’: ZR	←(@ZR)H OR (@HL)H ←@HL)H OR ACH ← HL+4 ← ZR+4	If ALUHLZR=0 If ALUHLZR=1
ORH* ORH*& ORH*% ORH*\$	@ZR,@HL	0010 1111 1B10 011B ACH,(@ZR)H (ACH,(@ZR)H) ‘&,\$’: HL	←(@ZR)H OR (@HL)H (←@HL)H OR ACH) ← HL+4	If ALUHLZR=0 (If ALUHLZR=1)

Instruction		Machine Code	Function		Flag/Remark
ORM ORM& ORM% ORM\$	@HL,@ZR	0010 1110 1B00 101B	'%,\$': ZR AC,@HL '&,\$': HL '%,\$': ZR	← ZR+4 ←(@ZR) OR MUI ← HL+1 ← ZR+1	Don't care ALUHLZR
ORM ORM& ORM% ORM\$	@ZR,@HL	0010 1110 1B10 101B	AC,@ZR '&,\$': HL '%,\$': ZR	←(@HL) OR MUI ← HL+1 ← ZR+1	
ORMH ORMH& ORMH% ORMH\$	@HL,@ZR	0010 1110 1B00 111B	ACH,@HL)H '&,\$': HL '%,\$': ZR	←(@ZR)H OR MUI ← HL+4 ← ZR+4	
ORMH ORMH& ORMH% ORMH\$	@ZR,@HL	0010 1110 1B10 111B	ACH,@ZR)H '&,\$': HL '%,\$': ZR	←(@HL)H OR MUI ← HL+4 ← ZR+4	
ADCI	Ry,D	0011 0000 DDDD YYYY	AC	← (Ry) + D + CF	CF, CFN3~0
ADCI*	Ry,D	0011 0001 DDDD YYYY	AC,(Ry)	← (Ry) + D + CF	CF, CFN3~0
SBCI	Ry,D	0011 0010 DDDD YYYY	AC	← (Ry) + DB + CF	CF, CFN3~0
SBCI*	Ry,D	0011 0011 DDDD YYYY	AC,(Ry)	← (Ry) + DB + CF	CF, CFN3~0
ADDI	Ry,D	0011 0100 DDDD YYYY	AC	← (Ry) + D	CF, CFN3~0
ADDI*	Ry,D	0011 0101 DDDD YYYY	AC,(Ry)	← (Ry) + D	CF, CFN3~0
SUBI	Ry,D	0011 0110 DDDD YYYY	AC	← (Ry) + DB + 1	CF, CFN3~0
SUBI*	Ry,D	0011 0111 DDDD YYYY	AC,(Ry)	← (Ry) + DB + 1	CF, CFN3~0
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 OXXX XXXX	AC(H),(Rx)(H)	← (Rx)(H) + 1	CF , CFN3~0
INC* INC*#	@HL	0100 0000 1B00 0000	AC,@HL) '#': HL	← (@HL) + 1 ← HL+1	' ': B6=0 '#': B6=1
INC* INC*#	@ZR	0100 0000 1B10 0000	AC,@ZR) '#': ZR	← (@ZR) + 1 ← ZR+1	
INCH* INCH*#	@HL	0100 0000 1B00 0100	ACH,@HL)H '#': HL	← (@HL)H + 1 ← HL+4	
INCH* INCH*#	@ZR	0100 0000 1B10 0100	ACH,@ZR)H '#': ZR	← (@ZR)H + 1 ← ZR+4	
DEC*	Rx	0100 0001 OXXX XXXX	AC(H),(Rx)(H)	← (Rx)(H) - 1	CF , CFN3~0
DEC* DEC*#	@HL	0100 0001 1B00 0000	AC,@HL) '#': HL	← (@HL) - 1 ← HL+1	' ': B6=0 '#': B6=1
DEC* DEC*#	@ZR	0100 0001 1B10 0000	AC,@ZR) '#': ZR	← (@ZR) - 1 ← ZR+1	
DECH*	@HL	0100 0001 1B00 0100	ACH,@HL)H	← (@HL)H - 1	

Instruction		Machine Code	Function		Flag/Remark
DECH*#			'#': HL	← HL+4	
DECH* DECH*#	@ZR	0100 0001 1B10 0100	ACH,(@ZR)H '#': ZR	← (@ZR)H - 1 ← ZR+4	
MULH  MULH MULH#	Rx  @HL	0100 0010 0XXX XXXX  0100 0010 1B00 0000	MU(H) AC(H)  MU AC '#': HL	←(Hex. : H)MUI(H)x(Rx)(H) ←(Hex. : L)MUI(H)x(Rx)(H)  ←(Hex. : H) MUI x (@HL) ←(Hex. : L) MUI x (@HL) ←HL+1	' ': B6=0 '#': B6=1
MULH MULH#	@ZR	0100 0010 1B10 0000	MU AC '#': ZR	←(Hex. : H) MUI x (@ZR) ←(Hex. : L) MUI x (@ZR) ←ZR+1	
MULHH MULHH#	@HL	0100 0010 1B00 0100	MUH ACH '#': HL	←(Hex. :H) MUIH x (@HL)H ←(Hex. :L) MUIH x (@HL)H ← HL+4	
MULHH MULHH#	@ZR	0100 0010 1B10 0100	MUH ACH '#': ZR	←(Hex. :H) MUIH x (@ZR)H ←(Hex. :L) MUIH x (@ZR)H ← ZR+4	
MULD  MULD MULD#	Rx  @HL	0100 0011 0XXX XXXX  0100 0011 1B00 0000	MU(H) AC(H)  MU AC '#': HL	←(Dec. : H)MUI(H)x(Rx)(H) ←(Dec. : L)MUI(H)x(Rx)(H)  ←(Dec. : H) MUI x (@HL) ←(Dec. : L) MUI x (@HL) ←HL+1	' ': B6=0 '#': B6=1
MULD MULD#	@ZR	0100 0011 1B10 0000	MU AC '#': ZR	←(Dec. : H) MUI x (@ZR) ←(Dec. : L) MUI x (@ZR) ←ZR+1	
MULDH MULDH#	@HL	0100 0011 1B00 0100	MUH ACH '#': HL	←(Dec. :H) MUIH x (@HL)H ←(Dec. :L) MUIH x (@HL)H ← HL+4	
MULDH MULDH#	@ZR	0100 0011 1B10 0100	MUH ACH '#': ZR	←(Dec. :H) MUIH x (@ZR)H ←(Dec. :L) MUIH x (@ZR)H ← ZR+4	
MWM	Rm, Ry	0100 0100 MMMM YYYY	Rm	← Ry	
MMW	Ry, Rm	0100 0101 MMMM YYYY	AC, Ry	← Rm	
IPA	Rx	0100 0110 0XXX XXXX	AC,(Rx) ACH,(Rx)H	← Port(A) ← Port(A,B,C,D)	If HMIOA=0 If HMIOA=1
IPB	Rx	0100 0110 1XXX XXXX	AC,(Rx)	← Port(B)	
IPC	Rx	0100 0111 0XXX XXXX	AC,(Rx)	← Port(C)	
LSP	Rx	0100 0111 1XXX XXXX	AC,(Rx)	← STACK Point	
IPD	Rx	0100 1000 0XXX XXXX	AC,(Rx)	← Port(D)	
IPE	Rx	0100 1000 1XXX XXXX	AC,(Rx)	← Port(E)	
LDS LDS#	@HL,D	0100 1001 1B00 DDDD	AC, (@HL) '#': HL	← D ← HL+1	' ': B6=0 '#': B6=1
LDS LDS#	@ZR,D	0100 1001 1B10 DDDD	AC,(@ZR) '#': ZR	← D ← ZR+1	
MAF	Rx	0100 1010 0XXX XXXX	AC,(Rx)	← STS1	B3 : CF B2 : ZERO B1 : SCF12(CX2) B0 : SCF11(CX)
RTM2L	Rx	0100 1010 1XXX XXXX	AC,(Rx)	← TM2(0~3)	If HMRM=0

Instruction		Machine Code	Function		Flag/Remark
			ACH,(Rx)H	←TM2,TM1 ,TM3(0~3)	If HMRTM=1
MSB	Rx	0100 1011 0XXX XXXX	AC,(Rx)	← STS2	B3 : SCF3(DPT) B2 : SCF2(HRx) B1 : SCF1(CPT) B0 : BCF
RTM21	Rx	0100 1011 1XXX XXXX	AC,(Rx)	← TM2(4,5),1(0,1)	
MSC	Rx	0100 1100 0XXX XXXX	AC,(Rx)	← STS3	B3 : SCF7(PDV) B2 : PH15 B1 : SCF5(TM1) B0 : SCF4(INT)
RTM1H	Rx	0100 1100 1XXX XXXX	AC,(Rx)	← TM1(2~5)	
MCX	Rx	0100 1101 0XXX XXXX	AC,(Rx)	← STS3X	B3 : No Use B2 : SCF0(APT) B1 : SCF6(TM2) B0 : SCF8(SKI)
RTM3L	Rx	0100 1101 1XXX XXXX	AC,(Rx) ACH,(Rx)H	← TM3(0~3) ←TM3,TM1 ,TM2(0~3)	If HMRTM=0 If HMRTM=1
MSD	Rx	0100 1110 0XXX XXXX	AC,(Rx)	← STS4	B3 : No Use B2 : RFOVF B1 : WDF B0 : CSF
RTM31	Rx	0100 1110 1XXX XXXX	AC,(Rx)	← TM3(4,5), 1(0,1)	
MDX	Rx	0100 1111 0XXX XXXX	AC,(Rx)	← STS4X	B3 : SCF10(TM3) B2 : INT B1 : CX2 B0 : CX
MKI	Rx	0100 1111 1XXX XXXX	AC,(Rx)	← STS5(KI4~1)	
SR0	Rx	0101 0000 0XXX XXXX	AC(H)n,(Rx)(H)n AC(H)x,(Rx)(H)x	← (Rx)(H)(n+1) ← 0	'x'='3' for RXHM=0 'x'='f' for RXHM=1
SR1	Rx	0101 0000 1XXX XXXX	AC(H)n,(Rx)(H)n AC(H)x,(Rx)(H)x	← (Rx)(H)(n+1) ← 1	'x'='3' for RXHM=0 'x'='f' for RXHM=1
SL0	Rx	0101 0001 0XXX XXXX	AC(H)n,(Rx)(H)n AC(H)0,(Rx)(H)0	← (Rx)(H) (n-1) ← 0	
SL1	Rx	0101 0001 1XXX XXXX	AC(H)n,(Rx)(H)n AC(H)0,(Rx)(H)0	← (Rx)(n-1) ← 1	
RRC	Rx	0101 0010 0XXX XXXX	AC(H)n,(Rx)(H)n AC(H)x,(Rx)(H)x CF	← (Rx)(H) (n+1) ← CF ← (Rx)(H)0	CF , CFN3~0
RRC RRC#	@HL	0101 0010 1B00 0000	ACn, (@HL)n AC3,(@HL)3 CF '#': HL	← (@HL)(n+1) ← CF ← (@HL)0 ← HL+1	'x'='3' 'for RXHM=0 'x'='f' for RXHM=1  ' ': B6=0 '#': B6=1
RRC RRC#	@ZR	0101 0010 1B10 0000	ACn, (@ZR)n AC3, (@ZR)3 CF '#': ZR	← (@HL)(n+1) ← CF ← (@ZR)0 ← ZR+1	
RRCH RRCH#	@HL	0101 0010 1B00 0100	ACHn, (@HL)Hn ACHf,(@HL)Hf CF '#': HL	← (@HL)H(n+1) ← CF ← (@HL)H0 ← HL+4	
RRCH RRCH#	@ZR	0101 0010 1B10 0100	ACHn, (@ZR)Hn ACHf, (@ZR)Hf CF '#': ZR	← (@HL)H(n+1) ← CF ← (@ZR)H0 ← ZR+4	
RLC	Rx	0101 0011 0XXX XXXX	AC(H)n,(Rx)(H)n AC(H)0,(Rx)(H)0 CF	← (Rx)(H) (n-1) ← CF ← (Rx)(H)x	CF , CFN3~0
RLC RLC#	@HL	0101 0011 1B00 0000	ACn, (@HL)n AC0,(@HL)0	← (@HL)(n-1) ← CF	'x'='3' 'for RXHM=0 'x'='f' for RXHM=1  ' ': B6=0

Instruction		Machine Code	Function		Flag/Remark
RLC RLC#	@ZR	0101 0011 1B10 0000	CF '#': HL	← (@HL)3 ← HL+1	'#': B6=1
RLCH RLCH#	@HL	0101 0011 1B00 0100	ACn, (@ZR)n AC0, (@ZR)0 CF '#': ZR	← (@HL)(n-1) ← CF ← (@ZR)3 ← ZR+1	
RLCH RLCH#	@ZR	0101 0011 1B10 0100	ACHn, (@HL)Hn ACH0, (@HL)H0 CF '#': HL	← (@HL)H(n-1) ← CF ← (@HL)Hf ← HL+4	
RLCH RLCH#	@ZR	0101 0011 1B10 0100	ACHn, (@ZR)Hn ACH0, (@ZR)H0 CF '#': ZR	← (@HL)H(n-1) ← CF ← (@ZR)Hf ← ZR+4	
DAA DAAH DAA* DAA*# DAA* DAA*# DAAH* DAAH*# DAAH* DAAH*#	Rx  @HL  @ZR  @HL  @ZR	0101 0100 0000 0000 0101 0100 0000 0100 0101 0101 0XXX XXXX 0101 0101 1B00 0000 0101 0101 1B10 0000 0101 0101 1B00 0100 0101 0101 1B10 0100	AC  ACH AC(H),(Rx) (H)	← BCD(CF,AC) for add.  ← BCD(CF,ACH) for add. ← BCD(CF,AC(H)) for add.	CF , CFN3~0  ' ': B6=0 '#': B6=1
DAS DASH DAS* DAS*# DAS* DAS*# DASH* DASH*# DASH* DASH*#	Rx,D  @HL  @ZR  @HL  @ZR	0101 0110 0000 0000 0101 0110 0000 0100 0101 0111 0XXX XXXX 0101 0111 1B00 0000 0101 0111 1B10 0000 0101 0111 1B00 0100 0101 0111 1B10 0100	AC  ACH AC(H),(Rx) (H)	← BCD(CF,AC) for sub.  ← BCD(CF,AC) for sub. ← BCD(CF,AC(H)) for sub.	CF , CFN3~0  ' ': B6=0 '#': B6=1
LDS	Rx,D	0101 1DDD DXXX XXXX	AC,(Rx)	← D	
LDH LDH*	Rx,@HL	0110 000B 0XXX XXXX	AC,(Rx) '*': HL  ACH,(Rx)H '*': HL	← H(T(@HL)) ← HL + 1  ← T(@HL) 16 bits ← HL + 2	If HMLDh=0  If HMLDh=1  ' ': B8=0 '*': B8=1
LDL LDL*	Rx,@HL	0110 001B 0XXX XXXX	AC,(Rx) '*': HL	← L(T(@HL)) ← HL + 1	' ': B8=0 '*': B8=1
MRF1	Rx	0110 0000 1XXX XXXX	AC,(Rx) AC,(Rx)15~0	← RFC3-0 ← RFC15~0	If HMMRF=0 If HMMRF=1

Instruction		Machine Code	Function		Flag/Remark
MRF2	Rx	0110 0001 1XXX XXXX	AC,(Rx)	← RFC7-4	
MRF3	Rx	0110 0010 1XXX XXXX	AC,(Rx)	← RFC11-8	
MRF4	Rx	0110 0011 1XXX XXXX	AC,(Rx)	← RFC15-12	
LDS8 LDS8#	@HL,D	0110 010B DDDD DDDD	(@HL) ‘#’: HL	← D(8 bits) ← HL+2	‘ ’: B8=0 ‘#’: B8=1
LDS8 LDS8#	@ZR,D	0110 011B DDDD DDDD	(@ZR) ‘#’: ZR	← D(8 bits) ← ZR+2	‘ ’: B8=0 ‘#’: B8=1
STA	Rx	0110 1000 0XXX XXXX	(Rx)(H)	← AC(H)	‘ ’: B6=0 ‘#’: B6=1
STA STA#	@HL	0110 1000 1B00 0000	(@HL) ‘#’: HL	← AC ← HL+1	
STA STA#	@ZR	0110 1000 1B10 0000	(@ZR) ‘#’: ZR	← AC ← ZR+1	
STAH STAH#	@HL	0110 1000 1B00 0100	(@HL)H ‘#’: HL	← ACH ← HL+4	
STAH STAH#	@ZR	0110 1000 1B10 0100	(@ZR)H ‘#’: ZR	← ACH ← ZR+4	
MMH	Rx	0110 1001 0XXX XXXX	AC(H),(Rx)(H)	← ?(H)	? = MU/MUI by ALUMUI=0/1
MMH MMH#	@HL	0110 1001 1B00 0000	AC,(@HL) ‘#’: HL	← ? ← HL+1	‘ ’: B6=0 ‘#’: B6=1
MMH MMH#	@ZR	0110 1001 1B10 0000	AC,(@ZR) ‘#’: ZR	← ? ← ZR+1	
MMHH MMHH#	@HL	0110 1001 1B00 0100	ACH,(@HL)H ‘#’: HL	← ?H ← HL+4	
MMHH MMHH#	@ZR	0110 1001 1B10 0100	ACH,(@ZR)H ‘#’: ZR	← ?H ← ZR+4	
MRW MRW#	@ZR,Rx	0110 1010 BXXX XXXX	AC(H),(@ZR)(H) ‘#’: ZR	← (Rx)(H) ← ZR+N	‘ ’: B7=0 ‘#’: B7=1  ‘N’=’1’ for RXHM=0 ‘N’=’4’ for RXHM=1
MWR MWR#	Rx,@ZR	0110 1011 BXXX XXXX	AC(H),(Rx) (H) ‘#’: ZR	← (@ZR) (H) ← ZR+N	‘ ’: B7=0 ‘#’: B7=1  ‘N’=’1’ for RXHM=0 ‘N’=’4’ for RXHM=1
LDA	Rx	0110 1100 0XXX XXXX	AC(H)	← (Rx)(H)	‘ ’: B6=0 ‘#’: B6=1
LDA LDA#	@HL	0110 1100 1B00 0000	AC ‘#’: HL	← (@HL) ← HL+1	
LDA LDA#	@ZR	0110 1100 1B10 0000	AC ‘#’: ZR	← (@ZR) ← ZR+1	
LDAH LDAH#	@HL	0110 1100 1B00 0100	ACH ‘#’: HL	← (@HL)H ← HL+4	
LDAH LDAH#	@ZR	0110 1100 1B10 0100	ACH ‘#’: ZR	← (@ZR)H ← ZR+4	
MRA	Rx	0110 1101 0XXX XXXX	CF	← (Rx)3	
MHL	X	0110 1101 10XX XX00	IDBF0~15	← (Rx)16 bits	Rx : 0 0000 10XX XX00b~

Instruction		Machine Code	Function		Flag/Remark
					0 0000 10XX XX11b
RHL	X	0110 1101 10XX XX01	(Rx)16 bits	← IDBF0~15	Rx : 0 0000 10XX XX00b~ 0 0000 10XX XX11b
MZR	X	0110 1101 10XX XX10	ZRBF0~12	← (Rx)16 bits	Rx : 0 0000 11XX XX00b~ 0 0000 11XX XX11b
RZR	X	0110 1101 10XX XX11	(Rx)16 bits	← ZRBF0~12	Rx : 0 0000 11XX XX00b~ 0 0000 11XX XX11b
MRW MRW#	@HL,Rx	0110 1110 BXXX XXXX	AC(H),(@HL)(H) ‘#’ : HL	← (Rx)(H) ← HL+N	‘ ’ : B7=0 ‘#’ : B7=1  ‘N’=’1’ for RXHM=0 ‘N’=’4’ for RXHM=1
MWR MWR#	Rx,@HL	0110 1111 BXXX XXXX	AC(H),(Rx)(H) ‘#’ : HL	← (@HL)(H) ← HL+N	‘ ’ : B7=0 ‘#’ : B7=1  ‘N’=’1’ for RXHM=0 ‘N’=’4’ for RXHM=1
MRW	Ry,Rx	0111 0YYY YXXX XXXX	AC(H),(Ry)(H)	← (Rx)(H)	
MWR	Rx,Ry	0111 1YYY YXXX XXXX	AC(H),(Rx)(H)	← (Ry)(H)	
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 0XXX XXXX XXXX	STACK PC	← PC + 1 ← X	
JMP	X	1100 1XXX XXXX XXXX	PC	← X	
CPZR	X	1101 0101 XXXX XXXX	(PC+1)	← force “NOP” if X7~0=ZRBF7~0	Can’t set ERX/ERY/ ELZ/CLPG to PC+1
SRY	X	1101 000X XXXX XXXX	X8~0	:Set Ry Page (0~6,8~ 1FFh ) for Rx 0~17FFh	Inserted by Compiler
ERY	X	1101 001X XXXX XXXX	X8~0	:Enable Ry Page (0~6,8~ 1FFh ) for Rx 0~1FFFh Set & Lock	1. Before executing “CLPG (X1=1)”, Page still can be changed by executing “ERY” again. 2. Between jump location, can’t have any “ERY/ERX/ELZ/ CLPG” instructions. 3. Interrupt will be masked by hardware till all page settings are cleared by “CLPG”.
SRX	X	1101 0100 00XX XXXX	X6~0	:Set Rx Page (1~3Fh)	Inserted by Compiler
ERX	X	1101 0110 00XX XXXX	X6~0	:Enable Rx Page (1~3Fh) Set & Lock	1. Before executing “CLPG (X0=1)”, Page still can be changed by executing “ERX” again.

Instruction		Machine Code	Function		Flag/Remark
					2. Between jump location, can't have any "ERY/ERX/ELZ/CLPG" instructions 3. Interrupt will be masked by hardware till all page settings are cleared by "CLPG".
SLZ	X	1101 0100 0100 00XX	X1~0	:Set Lz Page(1~3h)	Inserted by Compiler
ELZ	X	1101 0110 0100 00XX	X1~0	:Enable Lz Page(1~3h) Set & Lock	1. Before executing "CLPG (X2=1)", Page still can be changed by executing "ELZ" again. 2. Between jump location, can't have any "ERY/ERX/ELZ/CLPG" instructions. 3. Interrupt will be masked by hardware till all page settings are cleared by "CLPG".
SPBK	X	1101 0110 100X XXXX	X4~0	:Set POM BANK X=00~1FH	Inserted by Compiler
CLPG	X	1101 0110 1010 1XXX	X2=1 X1=1 X0=1	:Release Lz Page Lock :Release Ry Page Lock :Release Rx Page Lock	
T2M3X	X	1101 0111 00XX XXXX	X5~0  SD15~12= 1011~1000 0111~0100 0011~0000  SD11~6  SD5~0	: Set Counter 17~12 (Timer3) Value  :Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/  : Set Counter 11~6 (Timer1) Value  : Set Counter 5~0 (Timer2) Value	If TM3 is merged into TM2  Add "SETDAT SD" to the next address XCLK only for SIO  if TM1 is merged into TM2
T1XH		1101 0111 0100 0000	SD15~12= 1010~1000 0111~0100 0011~0000  SD11~6	:Ctm=INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/  : Set Counter 5~0 (Timer1) Value	Add "SETDAT SD" to the next address
T1RH		1101 0111 0100 0010	Timer1  SD12~0  (Rx)15~12= 1010~1000 0111~0100 0011~0000	← (Rx)16 bits  :Rx Address (SD1,0=00b) for 16bits data mode  :Ctm=INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	Add "SETDAT SD" to the next address  SD:0000~1FFFH

Instruction		Machine Code	Function		Flag/Remark
			(Rx)11~6	: Set Counter 5~0 (Timer1) Value	
T2XH		1101 0111 0100 0100	SD15~12= 1011~1000 0111~0100 0011~0000	:Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	Add "SETDAT SD" to the next address XCLK only for SIO
			SD11~6	: Set Counter 11~6 (Timer1) Value	if TM1 is merged into TM2
			SD5~0	: Set Counter 5~0 (Timer2) Value	
T2RH		1101 0111 0100 0110	Timer2	← (Rx)16 bits	Add "SETDAT SD" to the next address
			SD12~0	:Rx Address (SD1,0=00b) for 16 bits data mode	SD:0000~1FFFH
			(Rx)15~12= 1011~1000 0111~0100 0011~0000	:Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	XCLK only for SIO
			(Rx)11~6	: Set Counter 11~6 (Timer1) Value	if TM1 is merged into TM2
			(Rx)5~0	: Set Counter 5~0 (Timer2) Value	
T3XH		1101 0111 0100 1000	SD15~12= 1011~1000 0111~0100 0011~0000	:Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	Add "SETDAT SD" to the next address XCLK only for SIO
			SD11~6	: Set Counter 11~6 (Timer1) Value	if TM1 is merged into TM3
			SD5~0	: Set Counter 5~0 (Timer3) Value	
T3RH		1101 0111 0100 1010	Timer3	← (Rx)16 bits	Add "SETDAT SD" to the next address
			SD12~0	:Rx Address (SD1,0=00b) for 16 bits data mode	SD:0000~1FFFH
			(Rx)15~12= 1011~1000 0111~0100 0011~0000	:Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	XCLK only for SIO
			(Rx)11~6	: Set Counter 11~6 (Timer1) Value	if TM1 is merged into TM3
			(Rx)5~0	: Set Counter 5~0 (Timer3) Value	
SPKXH	D	1101 0111 0100 110D	KO16~1	← SETDAT SD(16 bits)	Add "SETDAT SD" to the next address
			D=1	: KEY_S released by scanning cycle	
			D=0	: KEY_S released by normal	

Instruction		Machine Code	Function		Flag/Remark
				key scanning	
SPKRH	D	1101 0111 0100 111D	SD15~0 KO16~1	: Enable KO16~1=1 ← (Rx)16 bits	Add "SETDAT SD" to the next address
			D=1 D=0	: KEY_S released by scanning cycle : KEY_S released by normal key scanning	SD:0000~1FFFH
			(Rx)15~0	: Enable KO16~1=1	
SPAXH	D	1101 0111 0101 000D	D=1 D=0	: Set IOA,B,C,D Pull-Low : Diable IOA,B,C,D Pull-Low	Add "SETDAT SD" to the next address
			SD15~12 SD11~8 SD7~4 SD3~0	: Set IOD4-1 I/O : Set IOC4-1 I/O : Set IOB4-1 I/O : Set IOA4-1 I/O	
SPARH	D	1101 0111 0101 001X	D=1 D=0	: Set IOA,B,C,D Pull-Low : Diable IOA,B,C,D Pull-Low	Add "SETDAT SD" to the next address
			SD12~0	: Rx Address (SD1,0=00b) for 16 bits data mode	SD:0000~1FFFH
			(Rx)15~12 (Rx)11~8 (Rx)7~4 (Rx)3~0	: Set IOD4-1 I/O : Set IOC4-1 I/O : Set IOB4-1 I/O : Set IOA4-1 I/O	
LDSH LDSH#	@HL	1101 0111 0101 100B	(@HL)16 bits #: HL	← SETDAT SD(16 bits) ← HL+4	Add "SETDAT SD" to the next address ' ': B0=0 #: B0=1
LDSH LDSH#	@ZR	1101 0111 0101 101B	(@ZR)16 bits #: ZR	← SETDAT SD(16 bits) ← ZR+4	
SHLX		1101 0111 0101 1100	IDBF14~0	← SETDAT SD(16 bits) SD : 0000~F7FFH	Add "SETDAT SD" to the next address
SZRX		1101 0111 0101 1101	ZRBF12~0	← SETDAT SD(16 bits) SD : 0000~1FFFH	Add "SETDAT SD" to the next address
CPHLH		1101 0111 0101 1110	(PC+2)	← force "NOP" if SETDAT SD(16 bits) = IDBF15~0	Add "SETDAT SD" to the next address Can't set ERX/ERY/ELZ/CLPG to PC+2
CPZRH		1101 0111 0101 1111	(PC+2)	← force "NOP" if SETDAT SD(16 bits) = ZRBF12~0	Add "SETDAT SD" to the next address Can't set ERX/ERY/ELZ/CLPG to PC+2
CAC	X	1101 0111 0110 XXXX	X>=AC: STACK PC(SET) PC  X<AC: PC	← PC+X+2 ← PC+AC+1 ← SETDAT SD(16 bits)	Add "SETDAT SD" to the next X(<16) Addresses range
JAC	X	1101 0111 0111 XXXX	X>=AC: PC(SET) PC  X<AC: PC	← PC+AC+1 ← SETDAT SD(16 bits)	Add "SETDAT SD" to the next X(<16) Addresses range
T1TH	@HL	1101 0111 1000 B000	Timer1	← T(@HL)16 bits	' ': B3=0

Instruction		Machine Code	Function		Flag/Remark
T1TH#			‘#’: HL TD15~12=1010~1000 0111~0100 0011~0000 TD11~6	← HL+2 :Ctm=INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value	‘#’: B3=1
T2TH T2TH#	@HL	1101 0111 1001 B000	Timer2(2->1) ‘#’: HL TD15~12=1011~1000 0111~0100 0011~0000 TD11~6 TD5~0	← T(@HL)16 bits ← HL+2 :Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value : Set Counter 5~0 (Timer2) Value	‘ ’: B3=0 ‘#’: B3=1 XCLK only for SIO if TM1 is merged into TM2
T3TH T3TH#	@HL	1101 0111 1010 B000	Timer3(3->1) ‘#’: HL TD15~12=1011~1000 0111~0100 0011~0000 TD11~6 TD5~0	← T(@HL)16 bits ← HL+2 :Ctm=XCLK/INT/CX2/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value : Set Counter 5~0 (Timer3) Value	‘ ’: B3=0 ‘#’: B3=1 XCLK only for SIO if TM1 is merged into TM3
SPKTH SPKTH#	D,@HL	1101 0111 1011 B00X	KO16~1 ‘#’: HL D=1 D=0 TD15~0	← T(@HL)16 bits ← HL+2 : KEY_S released by scanning cycle : KEY_S released by normal key scanning : Enable KO16~1=1	‘ ’: B3=0 ‘#’: B3=1
SPATH SPATH#	D,@HL	1101 0111 1100 B00X	A16~1 ‘#’: HL D=1 D=0 TD15~12 TD11~8 TD7~4 TD3~0	← T(@HL)16 bits ← HL+2 : Enable IOA,B,C,D Pull-Low : Diable IOA,B,C,D Pull-Low : Set IOD4-1 I/O : Set IOC4-1 I/O : Set IOB4-1 I/O : Set IOA4-1 I/O	‘ ’: B3=0 ‘#’: B3=1
RVL	Rx	1101 1000 0XXX XXXX	AC,(Rx)	← IDBF0~3	
RRL	Rx	1101 1000 1XXX XXXX	AC,(Rx)	← ZRBF0~3	
RVH	Rx	1101 1001 0XXX XXXX	AC,(Rx)	← IDBF4~7	

Instruction		Machine Code	Function		Flag/Remark
RRH	Rx	1101 1001 1XXX XXXX	AC,(Rx)	← ZRBF4~7	
RVU	Rx	1101 1010 0XXX XXXX	AC,(Rx)	← IDBF8~11	
RRU	Rx	1101 1010 1XXX XXXX	AC,(Rx)	← ZRBF8~11	
RVV	Rx	1101 1011 0XXX XXXX	AC,(Rx)	← IDBF12~15	If HMIDV=0
			ACH,(Rx)H	← IDBF0~15	If HMIDV=1
RRV	Rx	1101 1011 1XXX XXXX	AC,(Rx)0	← ZRBF12	If HMIDV=0
			ACH,(Rx)H	← ZRBF0~12	If HMIDV=1
TM3	Rx	1101 1100 0XXX XXXX	Timer3	← (Rx) & AC	
TM3	@HL	1101 1101 0000 B000	Timer3	← T(@HL)	' ': B3=0
TM3#			'#': HL	← HL+1	'#': B3=1
TM3X	X	1101 111X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer3 Value	
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 B000	Timer1	← T(@HL)	' ': B3=0
TMS#			'#': HL	← HL+1	'#': B3=1
TMSX	X	1110 001X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer1 Value	
TM2	Rx	1110 0100 0XXX XXXX	Timer2	← (Rx) & AC	
TM2	@HL	1110 0101 0000 B000	Timer2	← T(@HL)	' ': B3=0
TM2#			'#': HL	← HL+1	'#': B3=1
TM2X	X	1110 011X XXXX XXXX	X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0	: Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ : Ctm = PH15 : Ctm = PH3 : Ctm = PH9 : Set Timer2 Value	
SHE	X	1110 1000 X0XX XXX0	X7 X5 X4 X3 X2 X1	: Enable HEF7 : Enable HEF5 : Enable HEF4 : Enable HEF3 : Enable HEF2 : Enable HEF1	TMR3 KEY_S TMR2 PDV INT/SIO TMR1
SIE*	X	1110 1001 XXXX XXXX	X7 X6 X5 X4 X3	: Enable IEF7 : Enable IEF6 : Enable IEF5 : Enable IEF4 : Enable IEF3	TMR3 RFC KEY_S TMR2 PDV

Instruction		Machine Code	Function		Flag/Remark
			X2 X1 X0	: Enable IEF2 : Enable IEF1 : Enable IEF0	INT/SIO TMR1 A,C,DPT
PLC	X	1110 101X XXXX XXXX	X8 X7-0	: Reset PH15~11 : Reset HRF7-0	
SRF	X	1110 1100 00XX XXXX	X5~0	:Enable RFC5~0 Output	
SRE	X	1110 1101 0X0X X000	X6 X4 X3	:Enable SRF6(A port) :Enable SRF4(C port) :Enable SRF3(D port)	
FAST	(X)	1110 1110 0000 0XXX	B/SCLK X=7 X=6 X=5 X=4 X=3 X=2 X=1 X=0 or None	: High Speed Clock : B/SCLK=FTOSC/128 : B/SCLK=FTOSC/64 : B/SCLK=FTOSC/32 : B/SCLK=FTOSC/16 : B/SCLK=FTOSC/8 : B/SCLK=FTOSC/4 : B/SCLK=FTOSC/2 : B/SCLK=FTOSC	
SLOW	(X)	1110 1110 1000 0XXX	B/SCLK X=7 X=6 X=5 X=4 X=3 X=2 X=1 X=0 or None	: Low Speed Clock : B/SCLK=SLOSC/128 : B/SCLK=SLOSC/64 : B/SCLK=SLOSC/32 : B/SCLK=SLOSC/16 : B/SCLK=SLOSC/8 : B/SCLK=SLOSC/4 : B/SCLK=SLOSC/2 : B/SCLK=SLOSC	
CPHL	X	1110 1111 XXXX XXXX	(PC+1)	← force “NOP” if X7~0=IDBF7~0	Can't set ERX/ERY/ ELZ/CLPG to PC+1
SPK	Rx	1111 0000 0XXX XXXX	KO1~16	← (Rx) & AC	
SPK SPK#	@HL	1111 0001 0000 B000	KO1~16 HL	← T (@HL) ← HL+1	' ': B3=0 '#': B3=1
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 released by scanning cycle : KEY_S 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	

Instruction		Machine Code	Function		Flag/Remark
				X3~1=101=>KO11,12 X3~1=110=>KO13,14 X3~1=111=>KO15,16	
RTS		1111 0100 0000 0000	PC	← STACK (CALL Return)	
MRI MRI#	X	1111 0100 0001 B0XX	X1,0=00 X1,0=01 X1,0=10 X1,0=11 ‘#’: HL	: (RILH)3~0 ←(@HL) : (RILH)7~4 ←(@HL) : (RILH)11~8 ←(@HL) : (RILH)15~12 ←(@HL) : ← HL+1	‘ ‘: B3=0 ‘#’: B3=1
MRIH MRIH#	@HL	1111 0100 0001 010B	RILH ‘#’: HL	← (@HL)16 bits : ← HL+4	‘ ‘: B0=0 ‘#’: B0=1
MRIH MRIH#	@ZR	1111 0100 0001 011B	RILH ‘#’: ZR	← (@ZR)16 bits : ← ZR+4	
SBZ	X	1111 0100 0010 00XX	X1=0 X1=1  X0=0 X0=1	<set BZB Pad> :BZB :FREQB only  <set BZ Pad> :BZ :FREQ only	Initial  Initial
SWPWR	X	1111 0100 0010 01XX	X1,0=0X X1,0=10  X1,0=11	: Power Mode by Code Option : Switch to 3.0V Power Mode BCF=0 : BAK<VBAT : Switch to 1.5V Power Mode	Initial
SRP	X	1111 0100 0010 1XXX	X2  X1  X0	: Enable Timer3 repeat set if RL3=1 : Enable Timer2 repeat set if RL2=1 : Enable Timer1 repeat set if RL1=1	
STM	X	1111 0100 0011 00XX	X1,0=00 X1,0=01 X1,0=10 X1,0=11	:TM1,2,3 is independent : TM2(2=>1) : TM3(3=>1) : TM2(2=>1=>3)	Initial(X1,0=00)
SIAP	X	1111 0100 0011 01XX	X1=0 X1=1  X0=0 X0=1	: Byte Program Mode : Word Program Mode  : Disable IAP Mode : Enable IAP Mode	
DISTM	X	1111 0100 0011 1XXX	X2~0	: Disable TM3~1 Count	
SCNT	X	1111 0100 01XX XXXX	X5=0 X5=1  X4=0 X4=1  X3,2=00 X3,2=01 X3,2=10  X3,2=11  X1,0=00 X1,0=01 X1,0=10	: Enable CX set : Enable CX2 set  Counter clock source for not CX/2 control Mode : CX/2 : FREQ  Control Mode : Softwave : TM2 : CX/2 – One Cycle/Rising Edge(by code option) : CX/2 – High Level  Counter for CX/2 : RFC : TM1 : TM2	initial  initial

Instruction		Machine Code	Function		Flag/Remark
			X1,0=11	: TM3	
SCC	X	1111 0100 1XXX XXXX	X6 = 1 X6 = 0 X5 = 1 X4 = 1 X3 = 1 X2,1,0=001 X2,1,0=010 X2,1,0=100	: Cfq = XCLK : Cfq = PH0 : Set P(A) Cch : Set P(C) Cch : Set P(D) Cch : Cch = PH10 : Cch = PH8 : Cch = PH6	
SCA	X	1111 0101 00XX X000	X5 X4 X3	: Enable SEF5(A1-4) : Enable SEF4(C1-4) : Enable SEF3(D1-4)	
SPE	X	1111 0101 010X XXXX	X4 X3~0	: Set E4-1 Pull-Low : Set E4-1 I/O	
SCX	X	1111 0101 0110 00XX	X1 X0	: Enable SEF1(CX2) : Enable SEF0(CX)	
SXCLK	X	1111 0101 0110 01XX	X1 X0=0 X0=1	: Force clock source of RFC 16bits counter to XCLK : set XCLK=BCLK : set XCLK=SLOSC/FTOSC @ SLOW/FAST Mode	
STE	X	1111 0101 0110 1XXX	X2  X1,0=00 X1,0=01 X1,0=10 X1,0=11	: Replace timer1 read of RTM by extension bits & replace timer2 read of RTM by timer1 bits if X1,0=01  : TM1,2,3 is normal 6bits : TM1=>12bits(STM:00) : TM2=>12bits(STM:X0) : TM3=>12bits(STM:0X)	
ST3OV	X	1111 0101 0111 01XX	X1 X0	: Mask one cycle of Ctm2 when Timer3 overflow : Mask one cycle of Cfq when Timer3 overflow	
ADJ	X	1111 0101 0111 1XXX	X2  X1,0=00 X1,0=01 X1,0=10 X1,0=11	: 0/1 to inc./dec. one PH0 cycle for PDV when Timer overflow set by X1,0  : Normal PDV : Adjust PDV by TM1 : Adjust PDV by TM2 : Adjust PDV by TM3	
SPA	X	1111 0101 100X XXXX	X4 X3~0	: Set A4-1 Pull-Low : Set A4-1 I/O	
SPB	X	1111 0101 101X XXXX	X4 X3~0	: Set B4-1 Pull-Low : Set B4-1 I/O	
SPC	X	1111 0101 110X XXXX	X4 X3~0	: Set C4-1 Pull-Low / Low-Level-Hold : Set C4-1 I/O	
SPD	X	1111 0101 111X XXXX	X4 X3~0	: Set D4-1 Pull-Low : Set D4-1 I/O	
SF	X	1111 0110 XXXX XXXX	X7 X6 X5  X4 X3 X2 X1 X0	: Reload 1 Set : Reload 3 Set : Enable all Timer Counter update & latch : WDT Enable : HALT after EL : EL LIGHT On : BCF Set : CF Set	
RF	X	1111 0111 XXXX 0XXX	X7	: Reload 1 Reset	

Instruction		Machine Code	Function		Flag/Remark
			X6 X5 X4 X2 X1 X0	: Reload 3 Reset : Disable all Timer Counter latch : WDT Reset : EL LIGHT Off : BCF Reset : CF Reset	
ELC	X	1111 10XX XXXX XXXX	(ELP) X8,7,6=111 X8,7,6=110 X8,7,6=101 X8,7,6=100 X8,7,6=011 X8,7,6=000  X9,5,4=101 X9,5,4=100 X9,5,4=x11 X9,5,4=x10 X9,5,4=001 X9,5,4=000  (ELC) X3,2=11 X3,2=10 X3,2=01 X3,2=00 X1,0=11 X1,0=10 X1,0=01 X1,0=00	BCLK/8 BCLK/4 BCLK/2 BCLK FREQB PH0  2/3 3/4 1/1 1/2 1/3 1/4  PH5 PH6 PH7 PH8 1/1 1/2 1/3 1/4	
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 0XXX XXXX	X6 X5 X3  X2 X1 X0	: Enable CX2 Counter : Enable CX Counter : Enable INT powerful Pull-low  : Close all Segments : Dis-ENX Set : Reload 2 Set	
RF2	X	1111 1110 1XXX XXXX	X6 X5 X3  X2 X1 X0	: Disable CX2 Counter : Disable CX Counter : Disable INT powerful Pull-low  : Release Segments : Dis-ENX Reset : Reload 2 Reset	
HALT		1111 1111 0000 0000	Halt Operation		
PTR		1111 1111 0111 1111	T(@HL)	←RILH	
STOP		1111 1111 1000 0000	Stop Operation		

## Symbol Description

AC	: Accumulator	D	: Immediate Data
ACB	: Invert of Accumulator	DB	: Invert of Immediate Data
ACn	Accumulator bit n	PC	: Program Counter
X	: Address or Set data	CF	: Carry Flag
Rx	: Address of Data RAM	ZERO	: Zero Flag
(Rx)n	: Bit n of (Rx)	WDF	: Watch-Dog Timer Enable Flag
Ry	: Address of the working register	PDV	: Pre-Divider
BCF	: Back-up Flag	BCLK	: System clock stop only in STOP condition
IEFn	: Interrupt Enable Flag	SEFn	: Switch Enable Flag
HRFn	: HALT Release Flag	SRFn	: STOP Release Enable Flag
HEFn	: HALT Release Enable Flag	SCFn	: Start Condition Flag
TMR	: Timer Overflow Release Flag	Cch	: Clock Source of Chattering Detector
Ctm	: Clock Source of Timer	Cfq	: Clock Source of Frequency Generator
Lz	: Address of LCD data	FREQ	: Frequency Generator setting Value
RFOVF	: RFC Overflow Flag	( )	: Content of Address Register
MUI	: Multiplication Input Register	MU	: Multiplication High Nibble Result
Register			
@HL	: Address assigned by Index Register	HL	: Index Register
@ZR	: Address assigned by Index2 Register	ZR	: Index2 Register
H(T@HL)	: High Nibble of Index ROM data	L(T@HL)	: Low Nibble of Index ROM Data
IDBF	: Content of Index Register	ZRBF	: Content of Index2 Register
T@HL	: Index ROM address assigned by Index Register		
CSF	: Clock Source Flag	DA	: BCD for result
FTOSC	: Fast Oscillation clock	SLOSC	: Slow Oscillation clock
(H)	: Change to 16bits Mode by execute "MWM \$d,Ry" instruction to set RXHM=1		
ACH	: 16bits data type for Accumulator	ACBH	: 16bits data type for Invert of Accumulator
MUIH	: 16bits data type for Multiplication Input Register		
MUH	: 16bits data type for Multiplication High Word Result Register		
( )H	: 16 bits data type for content of Address Register		

Force AC/MU/MUI bit15~4=0 @4bits mode.

(@HL)/(@ZR): 4 bits data type for Data RAM  
 (@HL)8 bits/(@ZR)8 bits: 8 bits data type for Data RAM  
 (@HL)16 bits/(@ZR)16 bits: 16 bits data type for Data RAM  
 T(@HL): 8 bits data type for Index ROM  
 T(@HL)16 bits: 16 bits data type for Index ROM  
 HL+1 / ZR+1: +1 to index address bit0  
 HL+2 / ZR+2: mask index address bit0, and +1 to bit1  
 HL+4 / ZR+4: mask index address bit1&0, and +1 to bit2

## MWM/MMW Rm Assignment

RMPG1~0=0(initial page)

Rm	R/W	Instruction	BIT3	BIT2	BIT1	BIT0
0	R	MMW Ry, 0	IPF3	IPF2	IPF1	IPF0
	W	MWM 0, Ry	OPF3	OPF2	OPF1	OPF0
1	W	MWM 1, Ry	SPF3	SPF2	SPF1	SPF0
2	R	MMW Ry, 2	IPG3	IPG2	IPG1	IPG0
	W	MWM 2, Ry	OPG3	OPG2	OPG1	OPG0
3	W	MWM 3, Ry	SPG3	SPG2	SPG1	SPG0
4	R	MMW Ry, 4	IPH3	IPH2	IPH1	IPH0
	W	MWM 4, Ry	OPH3	OPH2	OPH1	OPH0
5	W	MWM 5, Ry	SPH3	SPH2	SPH1	SPH0
6	R	MMW Ry, 6	IPI3	IPI2	IPI1	IPI0
	W	MWM 6, Ry	OPI3	OPI2	OPI1	OPI0
7	W	MWM 7, Ry	SPI3	SPI2	SPI1	SPI0
8	W	MWM 8, Ry	SPLI	SPLH	SPLG	SPLF
9	R	MMW Ry, 9	LVR2F	LBD4	-	LBF
	W	MWM 9, Ry	CLRLVR2/ LVR2RLS		0	ENLBD
A	R	MMW Ry, A	VDL3	VDL2	VDL1	VDL0
	W	MWM A, Ry				
B	R	MMW Ry, B	LBD3	LBD2	LBD1	LBD0
	W	MWM B, Ry				
C	R	MMW Ry, C	LBD2CMP	CMPMD2	CMPMD1	CMPMD0
	W	MWM C, Ry				
D	R	MMW Ry, D	-	-	RMPG1	RMPG0
	W	MWM D, Ry	0	0		
E	Don't Use					
F	R	MMW Ry, F	VDX3	VDX2	VDX1	VDX0
	W	MWM F, Ry				

IPF3~0 : IOF4~1 input Data.  
 OPF3~0 : set IOF4~1 output data.  
 SPF3~0 : 1 for set IOF4~1 to Output(initial value = 0000).  
 SPLF : 1 for enable IOF4~1 Pull-low(initial value = 1).

IPG3~0 : IOG~1 input Data.  
 OPG3~0 : set IOG~1 output data.  
 SPG3~0 : 1 for set IOG4~1 to Output(initial value = 0000).  
 SPLG : 1 for enable IOG~1 Pull-low(initial value = 1).

IPH3~0 : IOH4~1 input Data.  
 OPH3~0 : set IOH4~1 output data.  
 SPH3~0 : 1 for set IOH4~1 to Output(initial value = 0000).

SPLH : 1 for enable IOH4~1 Pull-low(initial value = 1).

IPI3~0 : IOI~1 input Data.

OPI3~0 : set IOI~1 output data.

SPI3~0 : 1 for set IOI4~1 to Output(initial value = 0000).

SPLI : 1 for enable IOI~1 Pull-low(initial value = 1).

LBF : Low Battery Detect Flag.

ENLBD : 1 for generating pulse to enable Low Battery Detect.

LVR2F : Reset by LVR2 Flag & can be clear by CLRLVR2=1 if set LVR2 to LVR by code option.  
LBD/SWPWR state if set LVR2 to auto-LBD/SWPWR by code option(SWPWR instruction will be disable if set LVR2 to auto-SWPWR, LVR2F=0/1 for 3/1.5V Power Mode).

CLRLVR2 : Clear LVR2F if set LVR2 to LVR by code option.

LVR2RLS : Enable LVR2F 1<->0 Halt/Interrupt Release by share with INT if set LVR2 to auto-LBD/SWPWR by code option.

VDL3~0 : Set VDL regulator output voltage.  
±50mV for one step voltage.  
<VDL for VL2>

Initial value=6h for VDL=2.10V.

VDL=2.55/1.80V for VL3~0=F/0h.

<VDL for VL1>

Initial value=2h for VDL=1.05V.

VDL=1.70/0.95V for VL3~0=F/0h.

LBD4~0 : Set Low Battery Detect Voltage.

±50mV for one step voltage.

Initial value=15/00h for LBD=2.40/1.35V at 3/1.5V Power Mode.

LBD=2.90/1.35V for LBD4~0=1F/00h.

Share to set LVR2 voltage if set LVR2 to auto-LBD by code option.

VDX3~0 : Set VDX regulator output voltage.

Initial value=0h for VREG=1.20V.

VDX=1.95/1.20V for VDX3~0=F/0h.

LBD2CMP : 1 for change LBD4~0 & LBF & ENLBD for Comparator.

CMPMD2~0 : set CP0,1 input pins & mode for Comparator.

CMPMD2~0	CMP+	CMP- (VREF)
000	CP0	Voltage set by LBD4~0
001		VBG(0.95V)
010		Voltage set by LBD4~0 x CP1/VBG
011		CP1
100(initial)	CP1	Voltage set by LBD4~0
101		VBG(0.95V)
110		Voltage set by LBD4~0 x CP0/VBG
111		CP0

LBF=0/1 if voltage of CMP(+) >/< voltage of CMP(-) .

RMPG1~0 : set RM page.

RMPG1~0=1

Rm	R/W	Instruction	BIT3	BIT2	BIT1	BIT0
0~1	Don't Use					
2	R	MMW Ry, 2	SDAT3	SDAT2	SDAT1	SDAT0
	W	MWM 2, Ry				
3	R	MMW Ry, 3	SDAT7	SDAT6	SDAT5	SDAT4
	W	MWM 3, Ry				
4	W	MWM 4, Ry	SCR3	SCR2	SCR1	SCR0
5	<W> I2C UART SPI	MWM 5, Ry	<SCTL3> I2CEN UARTEN SPIEN	<SCTL2> STRSTP REN MSTR	<SCTL1> START RENRLS BIDIROE	<SCTL0> STOP TB8 LSBF
6	<R> I2C UART  SPI	MMW Ry, 6	<SSTA3> ACTF RCVOVF /TXOF RCVOVF	<SSTA2> ACKF RCVBF /TXSF RCVBF	<SSTA1> EAF RB8F  SSI	SBUSY
	<W> I2C UART SPI	MWM 6, Ry	<SUTI3> SDOCHK TXDCHK TXOCHK	<SUTI2> ACKOSTP CLRXF CLRXF	<SUTI1> WTACKI0 RXCK SSOE	<SUTI0> MKACKI TXCK FSNSS0
7	<R> I2C UART SPI	MMW Ry, 7	<SSCF3> IRQF RXSF MSF	<SSCF2> EROF RXBF WCOF/ER OF	<SSCF1> TRBF TXBF TRBF	INTF
	<W> I2C UART SPI	MWM 7, Ry	<SSRE3> IRQFEN RXSFEN MSFEN	<SSRE2> EROFEN RXBFEN WCOFEN /EROFEN	<SSRE1> TRBFEN TXBFEN TRBFEN	INTFEN
8	W	MWM 8, Ry	SIOTYP2	SIOTYP1	DSIOPHL	
9~B	Don't Use					
C	W	MWM C, Ry	<SIO3> 0 0 1	<SIO2> 0 1 CPOL	<SIO1> 0 TRB8 CPHA	<SIO0> 1 UART1W SPBIR
D	R	MMW Ry, D	-	-	RMPG1	RMPG0
	W	MWM D, Ry	0	0		
E~F	Don't Use					

SDAT7~0 : SIO Transmit and Receive Data.  
Read SDAT7~4 register will clear RCVOVF & RCVBF.  
Write data to SDAT7~4 register will start a transfer in master mode.

SCR3~0 : SIO clock source(no initial value).  
SCR3=0 : enable set SXCLK Clock Rate by SCR2~0  
000 => XCLK/2(XCLK/4 for I2C)  
001 => XCLK/4(XCLK/8 for I2C)  
010 => XCLK/8(XCLK/16 for I2C)  
011 => XCLK/16(XCLK/32 for I2C)  
100 => XCLK/32(XCLK/64 for I2C)  
101 => XCLK/64(XCLK/128 for I2C)

110 => XCLK/128(XCLK/256 for I2C)  
 111 => XCLK/256(XCLK/2 for I2C)  
 SCR3,2=10(11) : enable set TX&RX(set RX independently in UART mode) clock by SCR1,0 ( SPI & I2C don't care RX clock)  
 00 => SXCLK(baud rate = Fsxclk for I2C , Fsxclk/16 for UART , Fsxclk for SPI)  
 01 => FREQ(baud rate = Ffreq/2 for I2C , Ffreq/16 for UART , Ffreq for SPI)  
 10 => TM2(baud rate = Ftm2ov/2 for I2C , Ftm2ov/16 for UART , Ftm2ov for SPI)  
 11 => TM3(baud rate = Ftm3ov/2 for I2C , Ftm3ov/16 for UART , Ftm3ov for SPI)  
 SCR3,2=11(RX) is no use for I2C & SPI.  
 tm2ov : timer2 overflow rate = F(Ctm2) / (timer2 set value+1)  
 tm3ov : timer3 overflow rate = F(Ctm3) / (timer3 set value+1)

If no use FREQ/TM2/TM3 for SCR of SIO & active for other function, set SCR before enter SIO Mode to sure save current.  
 When set SCR=FREQ, disable FREQ for other function before enter SIO mode to save current & control FREQ by SIO, execute FRQ(X) instruction to set Cfq=XCLK for synchronizing with instruction timing & initial state before enable SIO mode.  
 Before set SCR=TM2/TM3, set Ctm2/Ctm3=XCLK to force TMRL2/3=1 & synchronizing with instruction timing & control TM2/3 by SIO

<SCTL3~0> : enable & options for SIO(initial value = 0000)  
 <SCTL0> : 1 for set I2C(STOP) if I2CEN=1.  
 used for UART(TB8) to set 9<sup>th</sup> transmit data bit in the mode of TRB8=1  
 1/0 for set SPI(LSBF) to set data transfer first by LSB/MSB  
 <SCTL1> : 1 for set I2C(START) if I2CEN=1.  
 for set UART(RENRLS)=1 to disable receive interrupt received if “stop bit” / “9<sup>th</sup> data bit” is 0 in the mode of TRB8=0/1.  
 for set SPI(BIDIROE)=1 to enable MOSI/MISO output for Master/Slave Mode in the Bidirectional mode of operation (SPBIR=1).  
 <SCTL2> : 1 for set I2C(STRSTP) if I2CEN=1.  
 for set UART(REN)=1 to enable receive by RXD/TXD pin when UART1W=0/1.  
 for set SPI(MSTR)=1/0 to enter Master/Slave mode  
 <SCTL3> : 1 for enable I2C/UART/SPI mode

<SCTL2~0> for I2C (SCTL3=1) :

STRSTP	START	STOP	Description
X	0	0	Nothing
0	0	1	Sent STOP bit after ACKI/ACKO bit by WDATH/RDATH for write/read mode.
0	1	0	Sent START bit before MSB of data by WDATH
0	1	1	Sent START bit before MSB of data by WDATH & sent STOP bit after ACKI/ACKO bit for LSB of data =0/1.
1	0	1	Direct sent STOP bit
1	1	0	Direct sent START bit & force to Write Mode(don't care LSB=1) after write SDAT7~4.
1	1	1	Direct sent START & STOP bits

SBUSY : SIO BUSY Flag(Read Only & set by H/W when a SIO transfer in in progress)  
 <SSTA1> : for I2C(EAF) to read external active flag by IRQ or others master active.(Need to sure IRQ output delay > 1/2 cycle of SCL after SDA:0->1 @ SCL=1 for SIOTYP=11 mode)  
 for UART RB8 Flag if set TRB8=1.  
 for SPI(SSI) to read SSB input invert state.  
 ACKF : for I2C ACKI Flag, and ACKF=1 if ACKI fail. H/W auto-sent stop bit & set EROF=1 by EROFEN=1 if ACKF=1 & MKACKI=0.  
 ACTF : for I2C Active Flag, and ACTF=1 from start to stop.

- RCVBF : for UART/SPI Receive Buffer Full Flag(Set by H/W at the end of a data transfer, and set SCTL3=0 or write CLRBF=1 or read the register of SDAT7~4 will clear this flag). In UART mode, enable this flag by set TXDCHK=0.
- RCVOVF : for UART/SPI Receive Buffer Overrun Flag(Set by H/W at the end of a data transfer, and set SCTL3=0 or write CLROVF=1 or read the register of SDAT7~4 will clear this flag). In UART mode, enable this flag by set TXDCHK=0.
- TXSF : for UART TXD Start Flag (Set by H/W if TXD have falling edge happen when TX in standby state & REN=0 in UART1W=1 mode, and set SCTL3=0 or write CLRXF=1 or write the register of SDAT7~4 will clear this flag.), only enable this flag by set TXDCHK=1 in UART mode.
- TXOF : for UART TXD Output Fail Flag(Set by H/W if output data & start bit don't match with TXD pin, and set SCTL3=0 or write CLRXF=1 or write the register of SDAT7~4 will clear this flag.), only enable this flag by set TXDCHK=1 in UART mode.
- <SUTI3~0> : Utility option & function for SIO(initial value = 0000)
- <SUTI0> : for set I2C(MKACKI)=1 to disable generate STOP bit & EROF if ACKI=High, but still make ACKF=1.  
for UART(TXCK), set 1 to enable transmit clock always, set TXCK=1 can make all transmit on same trace of Baud Rate if RX count of target IC can't auto-reset by each START bit, TXCK=1 also can avoid error TXOF by race between TX start and SIO clock if RX and TX set same clock source.  
for SPI(FSNSS0), set 1 to force SSB pin output low always if SSOE=1.
- <SUTI1> : for set I2C(WTACKI0)=1 to wait ACKI=0.  
for set UART(RXCK)=1 to enable RX clock always for reduce error START to make RXSF=1 by majority detect.  
1 for SPI(SSOE) enable SSB output in master mode.
- <SUTI2> : for set I2C(ACKOSTP)=1 to generate STOP bit replace ACKO bit.  
for set UART(CLRXF)=1 to clear RCVBF/TXSF & RCVOVF/TXOF  
for set SPI(CLRXF)=1 to clear RCVBF & RCVOVF.
- <SUTI3> : for set I2C(SDOCHK)=1 to enable EROF function .  
for set UART(TXDCHK)=1 to change enable RCVBF/RCVOVF to TXSF/TXOF function, set TXCK=1 to avoid error TXOF=1 if RX and TX have same clock source & one condition of RXCK=1 and SBUSY=1 by RX at start TX.  
for set SPI(TXOCHK)=1 to change WCOF to EROF function .
- INTF : INT pin interrupt/halt release flag
- <SSCF1> : I2C&SPI(TRBF) interrupt/halt release flag for the end of byte data transfer. Read/write SDAT7~4 or set SCTL3=0 or set SSRE1=0 will clear this flag  
UART(TXBF) transmit interrupt/halt release flag for the end of byte data transmit, disable by TXOF=1 or TXSF=1. Write SDAT7~4 or set SCTL3=0 or set SSRE1=0 will clear this flag  
Enable SSCF1 by set SSRE1=1
- <SSCF2> : I2C(EROF) transmit error interrupt/halt release flag for transmit data or ACKO(ACKOSTP=0) no match to SDA. Write SDAT7~4 or set SCTL3=0 or set SSRE2=0 will clear this flag.  
UART(RXBF) receive interrupt/halt release flag for the end of byte data receive. Read SDAT7~4 or set SCTL3=0 or set SSRE2=0 will clear this flag.  
SPI(WCOF) write collision interrupt/halt release flag by set TXOCHK=0 for writing data to SDAT7~4 when SBUSY=1. Write SDAT7~4 or set SCTL3=0 or set SSRE2=0 will clear this flag.  
SPI(EROF) transmit error interrupt/halt release flag by set TXOCHK=1 for transmit data no match to MOSI/MISO output pin. Write SDAT7~4 or set SCTL3=0 or set SSRE2=0 will clear this flag.  
Enable SSCF2 by set SSRE2=1.
- <SSCF3> : for I2C(IRQF) IRQ(Interrupt Request) interrupt/halt release flag for SDA pin receive falling edge at SCL pin = High when ACTF=0. Write SDAT7~4 or set SCTL3=0 or set SSRE3=0 will clear this flag.  
for UART(RXSF) receive start interrupt/halt release flag for start bit of receive . Read SDAT7~4

or set SCTL3=0 or set SSRE3=0 will clear this flag.

for SPI(MSF) is used as :

- mode fault interrupt/halt release flag with falling edge of SSB in master mode & SSOE=0, auto change to slave mode immediately by H/W.
- other master active start&stop interrupt/halt release flag with falling & rising edge of SSB in master mode & SSOE=1
- slave SSB start&stop interrupt/halt release flag with falling&rising edge of SSB at SBUSY=0 in slave mode.
- slave fault interrupt/halt release flag with rising edge of SSB at SBUSY=1 in slave mode. Auto disable output immediately by H/W. Clear SBUSY & re-work by re-enable SPI.
- slave SCK start interrupt/halt release flag with first edge of SCK at SBUSY=0 in slave mode & SSB pin no use.

set SCTL3=0 or set SSRE3=0 will clear this flag.

Enable SSCF3 by set SSRE3=1

INTFEN : 1 for enable INT pin interrupt/halt release(initial value = 1).

<SSRE3~1> : SSRE3~1 is forced to 0 when SCTL3=0(initial value = 000).

<SSRE1> : 1 for enable I2C/SPI(TRBF) interrupt/halt release at the end of byte data transfer.

1 for enable UART(TXBF) interrupt/halt release at the end of byte data transmit.

<SSRE2> : 1 for enable I2C(EROFEN) interrupt/halt release if transmit data or ACKO(ACKOSTP=0) no match to SDA

1 for enable UART(RXBF) interrupt/halt release at the end of byte data receive.

1 for enable SPI(WCOFEN) interrupt/halt release(by set TXOCHK=0) if write data to SDAT7~4 when SBUSY=1.

1 for enable SPI(EROFEN) interrupt/halt release(by set TXOCHK=1) if transmit data no match to MOSI/MISO output pin.

<SSRE3> : 1 for I2C(IRQEN) to enable interrupt/halt release by SDA pin receive falling edge at SCL pin = High when ACTF=0.

1 for UART(RXSF) to enable interrupt/halt release by start bit of receive.

1 for SPI(MSFEN) to enable interrupt/halt release by edge of SSB or first edge of SCK if SSB no use.

SIOTYP2,1: I2C/UART/SPI pins input/output type.

SIOTYP2,1		00(initial)	01	10	11
I2C	SDA(I/O) ,SCL(O)	NOD(I/O) ,NOD(O) (standard & for Multi-Master)	NOD(I/O) ,NOD(O) (for Multi-Master)	DC(O)/NOD(I) ,DC(O) (For Single-Master)	DC(I/O) ,DC(O) (For Single-Master)
UART	TXD(I/O) ,RXD(I)	NPOD(O)/NOD(I) ,NOD(I) (standard)	NOD(I/O) ,NOD(I) (For Multi-Master)	DC(O)/NOD(I) ,NOD(I) (For Single-Master & Multi-Slave)	DC(I/O) ,DC(I) (For Single-Master & Single-Slave)
SPI		(standard & for Sub-Master)	(For Multi-Master)	(For Single-Master & Multi-Slave)	(For Single-Master & Single-Slave)
(Master)	MISO(I) ,MOSI(I/O) ,SCK(O) ,SSB(I/O)	NOD(I) ,NOD(I/O) ,NOD/POD(O) ,DC(I/O)	NOD(I) ,NOD(I/O) ,NOD/POD(O) ,NOD(I/O)	NOD(I) ,NOD(I/O) ,DC(O) ,DC(I/O)	DC(I) ,DC(I/O) ,DC(O) ,DC(I/O)
(Slave)	MISO(I/O) ,MOSI(I) ,SCK(I) ,SSB(I)	NOD(I/O) ,NOD(I) ,NOD/POD(I) ,DC(I)	NOD(I/O) ,NOD(I) ,NOD/POD(I) ,NOD(I)	NOD(I/O) ,NOD(I) ,DC(I) DC(I)	DC(I/O) ,DC(I) ,DC(I) ,DC(I)*SPI-1

POD : P-type Normal Open Drain.

NOD : N-type Normal Open Drain.

NPOD : N-type Pseudo Open Drain.  
DC : CMOS output.

DSIOPHL : 1 for disable I2C/UART/SPI Pull-High/Low Resistance(initial value=0). Control Pull-High/Low Resistance turn on/off by hardware if DSIOPHL=0.

<SIO3~0> : I2C/UART/SPI select & Option (initial value = 0000)  
SIO3~0=0001 : for I2C Mode(Master Only)  
SIO3~2=01 : for UART Mode & set TRB8/UART1W by SIO1/0  
TRB8 : 0/1 for 8/9bits UART.  
UART1W : 0/1 for transmit & receive by TXD&RXD / TXD only  
In SIOTYP=11 mode, before set REN=1 need to sure input-High start to avoid error by floating.

SIO3=1 : for SPI Mode & set CPOL/CPHA/SPBIR by SIO2/1/0  
CPOL : 0/1 for SCK is low/high in idle mode  
CPHA : 0/1 for Data sampled on first/second edge of SCK period  
SPBIR : 1 for enable Bidirectional mode of operation(SPBIR=1 only enable MOSI/MISO(=>MOMI/SISO) one pin active in Master/Slave mode. In this mode, input is always enable, but output is enable by BIDIROE=1).

RMPG1~0=2

Rm	R/W	Instruction	BIT3	BIT2	BIT1	BIT0
0~8	Don't Use					
9	R	MMW Ry, 9	-	-	MKVDL	V2LBF
	W	MWM 9, Ry	0	0		ENV2LBD
A	R	MMW Ry, A	CUPDRV	CYC2	CYC1	CYC0
	W	MWM A, Ry				
B	R	MMW Ry, B	ENCYC	V2LBD2	V2LBD1	V2LBD0
	W	MWM B, Ry				
C	Don't Use					
D	R	MMW Ry, D	-	-	RMPG1	RMPG0
	W	MWM D, Ry	0	0		
E~F	Don't Use					

MKVDL : 1 for disable VDL outut & VL2/1 output voltage of VBAT level for execute VL2LBD @ 3.0/1.5V Power Mode.

V2LBF : Low Battery by Detect VL2 Flag.

ENV2LBD : 1 for enable Low Battery by Detect VL2 & hold value of VL2LBF after ENV2LBD:1->0.

V2LBD2~0 : Set Low Battery Detect Voltage.  
±100/50mV for one step voltage at 3/1.5V Power Mode.  
Initial value=1 for LBD=2.45/1.25V at 3/1.5V Power Mode.

ENCYC : 1 for enable change CUP1,2 driver by CUPDRV & Charge Pump Cycle Time by set CYC2~0.

CYC2~0 : set Charge Pump Cycle Time(initial value = 3 & must change CYC2~0 @ ENCYC=0).

CYC2~0=000 : 2 cycle of XCLK for charge pump cycle time.

CYC2~0=001 : 4 cycle of FREQ for charge pump cycle time.

CYC2~0=010 : 1 cycle of PH2 for charge pump cycle time.

CYC2~0=011 : 1 cycle of PH1 for charge pump cycle time.

CYC2~0=100 : 1 cycle of PH6 for charge pump cycle time.

CYC2~0=101 : 1 cycle of PH5 for charge pump cycle time.

CYC2~0=110 : 1 cycle of PH4 for charge pump cycle time.

CYC2~0=111 : 1 cycle of PH3 for charge pump cycle time.

CUPDRV : 0/1 for set CUP1/2 to Normal/Big Driver.

RMPG1~0=3

Rm	R/W	Instruction	BIT3	BIT2	BIT1	BIT0
0	R	MMW Ry, 0	LIDAC	LDSHAC	ADNIDP	RXDA
	W	MWM 0, Ry				
1	R	MMW Ry, 1	RXHM		HMLDH	HMIDV
	W	MWM 1, Ry				
2	R	MMW Ry, 2		HMIOA	HMMRF	HMRTM
	W	MWM 2, Ry				
3	R	MMW Ry, 3			ALUHLZR	ALUMUI
	W	MWM 3, Ry				
4~5	Don't Use					
6	R	MMW Ry, 6	CFN3	CFN2	CFN1	CFN0
	W	MWM 6, Ry				
7	R	MMW Ry, 7	ZFN3	ZFN2	ZFN1	ZFN0
8~C	Don't Use					
D	R	MMW Ry, D	-	-	RMPG1	RMPG0
	W	MWM D, Ry	0	0		
E~F	Don't Use					

RXDA : 1 for enable BCD convert when execute ADC/SBC/ADD/SUB instructions(initial value = 0).  
 ADNIDP : 1 for change 4 nibbles of ADN(H) @ 16bits Mode to independence mode(initial value = 0).  
 LDSHAC : 1 for enable sent "SETDAT SD(16 bits)" to ACH when execute "LCDH(#) @HL/ZR instruction.  
 LIDAC : 1 for enable sent RAM data to AC(H) when execute "LID(H)" instruction.

HMIDV : 1 for enable "MVV/MRV/RVV/RRV Rx" change to 16bits Mode (initial value = 0).

HMLDH : 1 for enable "LDH(\*) Rx,@HL" change to 16bits Mode (initial value = 0).

RXHM : 1 for enable some instructions of Rx change to 16bits Mode:

ADC/SBC/ADD/SUB/AND/AND/EOR/OR Rx

INC\*/DEC\* Rx

SMUI/MMH/MULH/MULD Rx

SR0/SR1/SL0/SL1/RRC/RLC Rx

LDA/STA Rx

MRW Ry/@HL/@ZR,Rx

MWR Rx,Ry/@HL/@ZR

HMRTM : 1 for enable "RTM2L/RTM3L Rx" change to 16bits Mode (RTM3L,1H,2L/2L,1H,21,2L)  
 (initial value = 0).

HMMRF : 1 for enable "MRF1 Rx" change to 16bits Mode (initial value = 0).

HMIOA : 1 for enable "IPA/OPA Rx" hange to 16bits Mode(I/OPD,C,B,A) (initial value = 0).

ALUMUI : 1 for change MU to MUI of ADCM(H)/SBCM(H)/ADDM(H)/SUBM(H) instruction.

ALUHLZR : 1 for change function of ADC(H)/SBC(H)/ADD(H)/SUB(H)/ADN(H)/AND(H)/EOR(H)/ORH  
 @HL/ZR,@ZR/HL instruction to" AC(H),(@HL/ZR)(H) ← result of (@ZR/HL)(H) with AC(H).

CFN3~0 : Carry Flag output for Nibble3~0 , Write Mode for DAA/S(RXHM=1),DAA/SH only.

ZFN3~0 : ZERO Flag output for Nibble3~0.