

TM8952

4-Bit Microcontroller

Data Sheet

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

Version	Date	Description
V1.0	Feb, 2009	New release
V1.1	April, 2009	Revise the format of the document.
V1.2	Dec, 2011	Add Ordering Information table.

CONTENTS

AMENDMENT HISTORY.....	2
GENERAL DESCRIPTION.....	4
FEATURES	4
APPLICATIONS	5
BLOCK DIAGRAM.....	6
PAD ASSIGNMENT	6
PIN DESCRIPTION	7
RAM & INITIAL PAGE ASSIGNMENT.....	8
Configuration of LCD RAM Area	9
ABSOLUTE MAXIMUM RATINGS.....	11
POWER CONSUMPTION	12
ALLOWABLE OPERATING CONDITIONS	12
ALLOWABLE OPERATING FREQUENCY	12
INTERNAL RC FREQUENCY RANGE	13
ELECTRICAL CHARACTERISTICS.....	13
TYPICAL APPLICATION CIRCUIT.....	15
ORDERING INFORMATION	16
Appendix A: TM8952 Instruction Table.....	17
Symbol Description	35

GENERAL DESCRIPTION

The TM8952 is an embedded high-performance 4-bit microcontroller 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, LCD driver, look-up table, watchdog timer and key matrix scanning circuitry etc. on a signal chip.

FEATURES

1. Low power dissipation.
 - 1.5V/3V operating voltage range.
2. Powerful instruction set.
 - Binary addition, subtraction, BCD adjustment. BCD can be performed directly in addition/subtraction operations.
 - 4 bits x 4 bits Multiplier
 - Single-bit manipulation (set, reset, decision for branch).
 - Various conditional branches.
 - 16 initial working registers and manipulators. (can be extended to all RAM with Page Mode)
 - Look-up Table.
 - LCD driver data transfer.
3. ROM Capacity.

16K	x	16 bits.
● Instruction ROM Max. Capacity	16K	x 16 bits.
● Table ROM Max. Capacity	24K/12K	x 8/16 bits.
4. RAM Capacity.

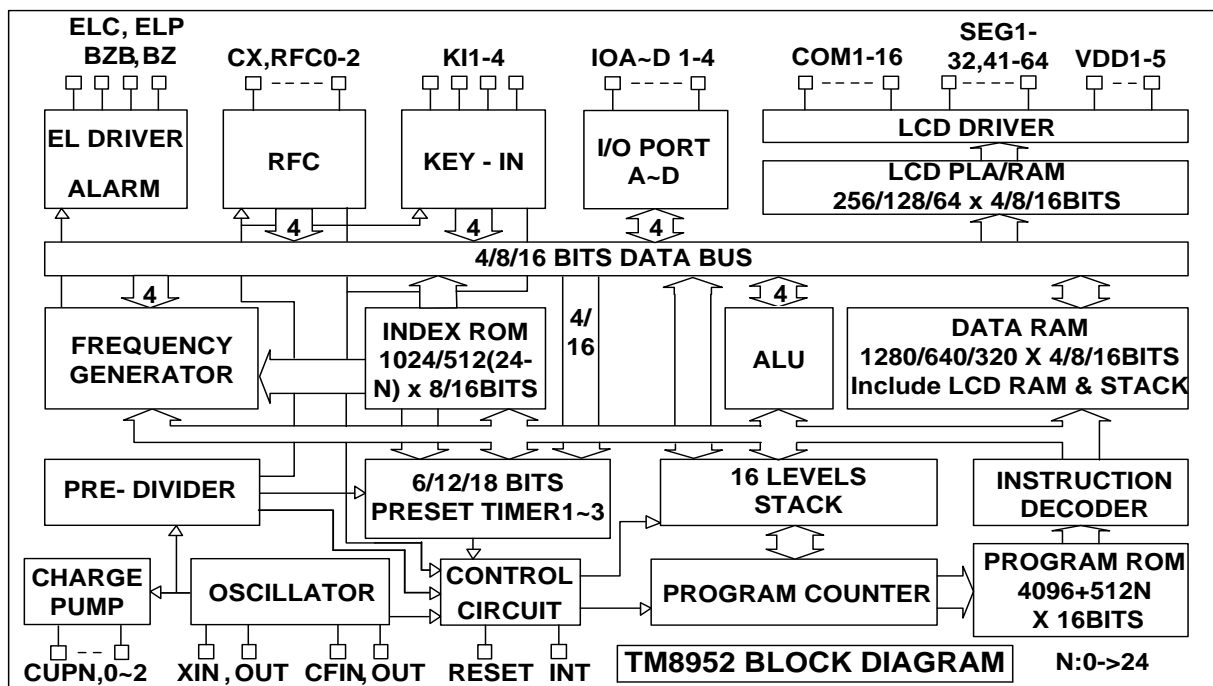
1280/640/320x	4/8/16 bits.
● LCD Max. Capacity	256/128/64 x 4/8/16 bits.
● STACK Max. Capacity	16 x 16 bits.
● HL/ZR store Max. Capacity	16/16 x 16 bits.
5. With direct/index addressing mode in data RAM access.
6. LCD driver output.
 - COM12~16 can be defined as SEG64~61 by mask option.
 - SEG41~60 can be defined as IOB1~4, KI~4, ELC, ELP, BZB, BZ, IOD1~4, IOC1~4 by mask option.
 - 1/4~1/16 Duty can be selected by mask option.
 - 1/3~1/5 Bias can be selected by mask option.
 - Single instruction to turn off all segments.
 - COM5~16 can be defined as CMOS or P_open drain type output port by mask option.
 - External regulator mode for VL1/2 by mask option.
7. Input/output ports.

- Port IOA 4 pins (with internal pull-low, input signal chattering prevention circuitry), and can be defined as CX, RFC0~2 by mask option.
 - Port IOB 4 pins (with internal pull-low), and can be defined as SEG41~44 by mask option.
 - Port IOC 4 pins (with internal pull-low, low-level-hold, input signal chattering prevention circuitry), and can be defined as SEG57~60 by mask option.
 - Port IOD 4 pins (with internal pull-low, input signal chattering prevention circuitry), and can be defined as SEG53~56 by mask option.
8. Interrupt function.
- External factors 5 (INT pin, 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 SEG49 and SEG50 by mask option.
10. Built-in Alarm, clock or single tone melody generator.
- BZB, BZ. Can be defined as SEG51 and SEG52 by mask option.
11. Built-in resistance to frequency converter.
- CX, RFC0~2. Can be defined as IOA1~4 by mask option.
12. Built-in key matrix scanning function.
- KO1~KO16(Shared with SEG1~16)
 - KI1~KI4(Can be defined as SEG45~48 by mask option)
13. Three 6-bit programmable timer with programmable clock source.
- Read the content at any time
 - Merge 2 or 3 timers into one 12-bit or 18-bit timer
 - Used as a counter for RFC
14. Watchdog timer.
15. Built-in voltage charge halver & pump circuit.
16. Dual clock operation
- Slow clock oscillation can be defined as X'tal or external RC type oscillator by mask option.
 - Fast clock oscillation can be defined as 3.58MHz ceramic resonator, internal R or external R type oscillator by mask option.
17. HALT function.
18. STOP function.

APPLICATIONS

- Timer / Calendar / Calculator / Thermometer

BLOCK DIAGRAM



PAD ASSIGNMENT

No	Name	No	Name	No	Name	No	Name
1	GND	27	COM11	53	SEG21	79	SEG55/IOD3
2	XIN	28	COM12	54	SEG22	80	SEG56/IOD4
3	XOUT	29	COM13/SEG64	55	SEG23	81	SEG57/IOC1
4	CFIN	30	COM14/SEG63	56	SEG24	82	SEG58/IOC2
5	CFOUT	31	COM15/SEG62	57	SEG25	83	SEG59/IOC3
6	BAK	32	COM16/SEG61	58	SEG26	84	SEG60/IOC4
7	VL1	33	SEG1(KO1)	59	SEG27	85	IOA1/CX
8	VBAT	34	SEG2(KO2)	60	SEG28	86	IOA2/RFC0
9	VL2	35	SEG3(KO3)	61	SEG29	87	IOA3/RFC1
10	VL3	36	SEG4(KO4)	62	SEG30	88	IOA4/RFC2
11	VL4	37	SEG5(KO5)	63	SEG31	89	INT
12	VL5	38	SEG6(KO6)	64	SEG32	90	TEST
13	CUPN	39	SEG7(KO7)	65	SEG41/IOB1	91	RESETB
14	CUP0	40	SEG8(KO8)	66	SEG42/IOB2		
15	CUP1	41	SEG9(KO9)	67	SEG43/IOB3		
16	CUP2	42	SEG10(KO10)	68	SEG44/IOB4		
17	COM1	43	SEG11(KO11)	69	SEG45/KI1		
18	COM2	44	SEG12(KO12)	70	SEG46/KI2		
19	COM3	45	SEG13(KO13)	71	SEG47/KI3		
20	COM4	46	SEG14(KO14)	72	SEG48/KI4		
21	COM5	47	SEG15(KO15)	73	SEG49/ELC		
22	COM6	48	SEG16(KO16)	74	SEG50/ELP		
23	COM7	49	SEG17	75	SEG51/BZB		
24	COM8	50	SEG18	76	SEG52/BZ		
25	COM9	51	SEG19	77	SEG53/IOD1		
26	COM10	52	SEG20	78	SEG54/IOD2		

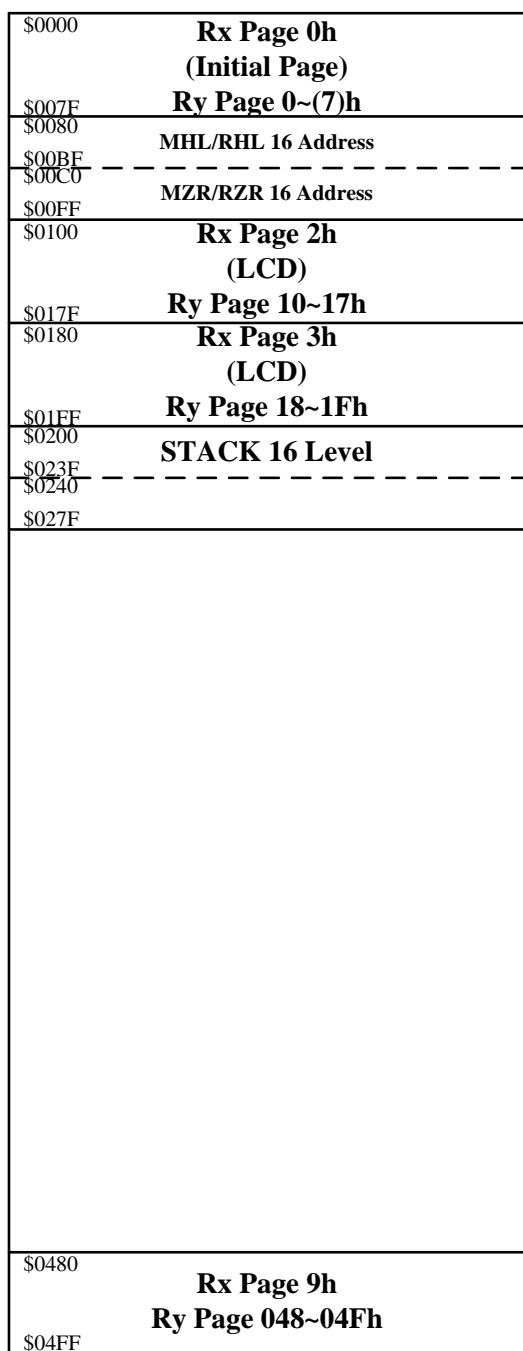
PIN DESCRIPTION

Name	I/O	Description
BAK	P	Positive Back-up voltage. In 3V power mode, connect a 0.1u capacitor to GND.
VBAT	P	Positive supply voltage.
VL1~5	P	LCD supply voltage. In 1.5V Power Mode or "External LCD Regulator for 3V Power Mode" mask option = "VL1", connect positive power or regulator output to VL1. In 3V Power mode, if "External LCD Regulator for 3V Power Mode" mask option = "No Use" or "VL2", connect positive power or regulator output to VL2.
RESETB	I	Input pin for external reset request signal, built-in internal pull-up resistor.
INT	I	Input pin for external INT request signal. . Falling edge or rising edge triggered is defined by mask option. . Internal pull-down or pull-up resistor is defined by mask option.
TEST		Test signal input pin.
CUPN,0,1,2	O	Switching pins for supplying the LCD driving voltage to the VL1~5 pins. . Connect the CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operates in 1/3 bias mode. . Connect the CUP0 to CUP2 & CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operates in 1/4 bias mode. . Connect the CUPN to CUP0 & CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operates in 1/5 bias mode or more efficient 1/3&1/4 bias mode.
XIN XOUT	I O	Low speed oscillator, generates clocks for time base functions (clock specified, LCD alternating frequency, Alarm signal frequency) and system clock oscillation. . The usage of 32KHz Crystal oscillator or external RC oscillator is defined by mask option.
CFIN CFOUT	I O	High speed oscillator, system clock oscillation for FAST clock only or DUAL clock operation. . The usage of 3.58MHz ceramic/resonator oscillator or external R type oscillator is defined by mask option
COM1~16	O	Output pins for driving the common pins of the LCD panel. COM5~16 can be defined as COMS or Open Drain type output port by mask option. COM13~16 can be defined as SEG64~61 by mask option.
SEG1-32, 41-64	O	Output pins for driving the LCD panel segment. SEG41~44 can be defined as IOB1~4 by mask option. SEG45~48 can be defined as KI1~4 by mask option. SEG49~52 can be defined as ELC, ELP, BZB, BZ by mask option. SEG53~56 can be defined as IOD1~4 by mask option. SEG57~60 can be defined as IOC1~4 by mask option. SEG61~64 can be defined as COM16~13 by mask option.
IOA1-4	I/O	Input / Output port A, and can be defined as CX, RFC0~2 by mask option.
IOB1-4	I/O	Input / Output port B, and can be defined as SEG41~44 by mask option.
IOC1-4	I/O	Input / Output port C, and can be defined as SEG57~60 by mask option.
IOD1~4	I/O	Input / Output port D, and can be defined as SEG53~56 by mask option.
CX,RFC0~2	I O	1 input pin and 3 output pins for RFC application, and can be defined as IOA1~4 by mask option.
ELC/ELP	O	Output port for the EI panel driver, and can be defined as SEG49 and SEG50 by mask option.
BZB/BZ	O	Output port for alarm, clock or the single tone melody generator, and can be defined as SEG51 and SEG52 by mask option.
KO1~KO16	O	Output port for the key matrix scanning, shared with SEG1~16.
KI1~4	I	Input port for the key matrix scanning, and can be defined as SEG45~48 by mask option.
GND	P	Negative supply voltage.

RAM & INITIAL PAGE ASSIGNMENT

Mode	Initial Page	Page Number	Instruction inserted for set other Page by Compiler
Rx (0000~04FFh) (0100~017Fh for LCD)	0000~007Fh(0)	10 (x128)	SRX X (X: 1~9h; X: 2, 3h for LCD)
Ry (0000~04FFh) (0100~017Fh for LCD)	0070~007Fh(7)	80(x16)	SRY X (X: 0~6, 8~04Fh; X: 10~1Fh for LCD)
Lz (000~7Fh)	0~1Fh(0)	4(x32)	SLZ X (X: 0~3h)

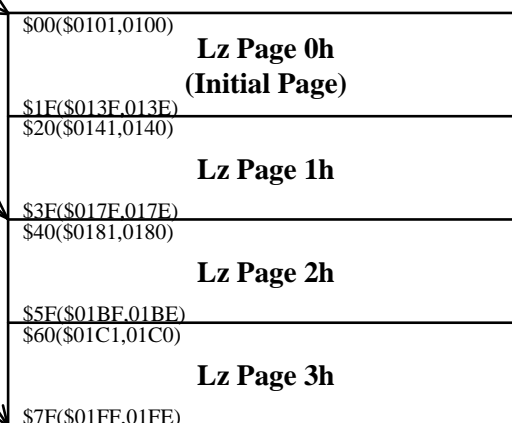
1.25K x 4bits RAM



\$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 8bits LCD



16 level x 16bits 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

Configuration of LCD RAM Area

ADDRESS		BIT3	BIT2	BIT1	BIT0	ADDRESS		BIT3	BIT2	BIT1	BIT0
		COM4	COM3	COM2	COM1			COM8	COM7	COM6	COM5
Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA	Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA
Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE	Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE
0100H	00	SEG1	SEG1	SEG1	SEG1	0140H	20	SEG1	SEG1	SEG1	SEG1
0101H		SEG2	SEG2	SEG2	SEG2	0141H		SEG2	SEG2	SEG2	SEG2
0102H	01	SEG3	SEG3	SEG3	SEG3	0142H	21	SEG3	SEG3	SEG3	SEG3
0103H		SEG4	SEG4	SEG4	SEG4	0143H		SEG4	SEG4	SEG4	SEG4
0104H	02	SEG5	SEG5	SEG5	SEG5	0144H	22	SEG5	SEG5	SEG5	SEG5
0105H		SEG6	SEG6	SEG6	SEG6	0145H		SEG6	SEG6	SEG6	SEG6
0106H	03	SEG7	SEG7	SEG7	SEG7	0146H	23	SEG7	SEG7	SEG7	SEG7
0107H		SEG8	SEG8	SEG8	SEG8	0147H		SEG8	SEG8	SEG8	SEG8
0108H	04	SEG9	SEG9	SEG9	SEG9	0148H	24	SEG9	SEG9	SEG9	SEG9
0109H		SEG10	SEG10	SEG10	SEG10	0149H		SEG10	SEG10	SEG10	SEG10
010AH	05	SEG11	SEG11	SEG11	SEG11	014AH	25	SEG11	SEG11	SEG11	SEG11
010BH		SEG12	SEG12	SEG12	SEG12	014BH		SEG12	SEG12	SEG12	SEG12
010CH	06	SEG13	SEG13	SEG13	SEG13	014CH	26	SEG13	SEG13	SEG13	SEG13
010DH		SEG14	SEG14	SEG14	SEG14	014DH		SEG14	SEG14	SEG14	SEG14
010EH	07	SEG15	SEG15	SEG15	SEG15	014EH	27	SEG15	SEG15	SEG15	SEG15
010FH		SEG16	SEG16	SEG16	SEG16	014FH		SEG16	SEG16	SEG16	SEG16
0110H	08	SEG17	SEG17	SEG17	SEG17	0150H	28	SEG17	SEG17	SEG17	SEG17
0111H		SEG18	SEG18	SEG18	SEG18	0151H		SEG18	SEG18	SEG18	SEG18
0112H	09	SEG19	SEG19	SEG19	SEG19	0152H	29	SEG19	SEG19	SEG19	SEG19
0113H		SEG20	SEG20	SEG20	SEG20	0153H		SEG20	SEG20	SEG20	SEG20
0114H	0A	SEG21	SEG21	SEG21	SEG21	0154H	2A	SEG21	SEG21	SEG21	SEG21
0115H		SEG22	SEG22	SEG22	SEG22	0155H		SEG22	SEG22	SEG22	SEG22
0116H	0B	SEG23	SEG23	SEG23	SEG23	0156H	2B	SEG23	SEG23	SEG23	SEG23
0117H		SEG24	SEG24	SEG24	SEG24	0157H		SEG24	SEG24	SEG24	SEG24
0118H	0C	SEG25	SEG25	SEG25	SEG25	0158H	2C	SEG25	SEG25	SEG25	SEG25
0119H		SEG26	SEG26	SEG26	SEG26	0159H		SEG26	SEG26	SEG26	SEG26
011AH	0D	SEG27	SEG27	SEG27	SEG27	015AH	2D	SEG27	SEG27	SEG27	SEG27
011BH		SEG28	SEG28	SEG28	SEG28	015BH		SEG28	SEG28	SEG28	SEG28
011CH	0E	SEG29	SEG29	SEG29	SEG29	015CH	2E	SEG29	SEG29	SEG29	SEG29
011DH		SEG30	SEG30	SEG30	SEG30	015DH		SEG30	SEG30	SEG30	SEG30
011EH	0F	SEG31	SEG31	SEG31	SEG31	015EH	2F	SEG31	SEG31	SEG31	SEG31
011FH		SEG32	SEG32	SEG32	SEG32	015FH		SEG32	SEG32	SEG32	SEG32
0120H	10	*1	*1	*1	*1	0160H	30	*1	*1	*1	*1
0121H		*1	*1	*1	*1	0161H		*1	*1	*1	*1
0122H	11	*1	*1	*1	*1	0162H	31	*1	*1	*1	*1
0123H		*1	*1	*1	*1	0163H		*1	*1	*1	*1
0124H	12	*1	*1	*1	*1	0164H	32	*1	*1	*1	*1
0125H		*1	*1	*1	*1	0165H		*1	*1	*1	*1
0126H	13	*1	*1	*1	*1	0166H	33	*1	*1	*1	*1
0127H		*1	*1	*1	*1	0167H		*1	*1	*1	*1
0128H	14	SEG41	SEG41	SEG41	SEG41	0168H	34	SEG41	SEG41	SEG41	SEG41
0129H		SEG42	SEG42	SEG42	SEG42	0169H		SEG42	SEG42	SEG42	SEG42
012AH	15	SEG43	SEG43	SEG43	SEG43	016AH	35	SEG43	SEG43	SEG43	SEG43
012BH		SEG44	SEG44	SEG44	SEG44	016BH		SEG44	SEG44	SEG44	SEG44
012CH	16	SEG45	SEG45	SEG45	SEG45	016CH	36	SEG45	SEG45	SEG45	SEG45
012DH		SEG46	SEG46	SEG46	SEG46	016DH		SEG46	SEG46	SEG46	SEG46
012EH	17	SEG47	SEG47	SEG47	SEG47	016EH	37	SEG47	SEG47	SEG47	SEG47
012FH		SEG48	SEG48	SEG48	SEG48	016FH		SEG48	SEG48	SEG48	SEG48
0130H	18	SEG49	SEG49	SEG49	SEG49	0170H	38	SEG49	SEG49	SEG49	SEG49
0131H		SEG50	SEG50	SEG50	SEG50	0171H		SEG50	SEG50	SEG50	SEG50
0132H	19	SEG51	SEG51	SEG51	SEG51	0172H	39	SEG51	SEG51	SEG51	SEG51
0133H		SEG52	SEG52	SEG52	SEG52	0173H		SEG52	SEG52	SEG52	SEG52
0134H	1A	SEG53	SEG53	SEG53	SEG53	0174H	3A	SEG53	SEG53	SEG53	SEG53
0135H		SEG54	SEG54	SEG54	SEG54	0175H		SEG54	SEG54	SEG54	SEG54
0136H	1B	SEG55	SEG55	SEG55	SEG55	0176H	3B	SEG55	SEG55	SEG55	SEG55

ADDRESS		BIT3	BIT2	BIT1	BIT0	ADDRESS		BIT3	BIT2	BIT1	BIT0		
		COM4	COM3	COM2	COM1			COM8	COM7	COM6	COM5		
Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA	Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA		
Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE	Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE		
0137H	1C	SEG56	SEG56	SEG56	SEG56	0177H	3C	SEG56	SEG56	SEG56	SEG56		
0138H		SEG57	SEG57	SEG57	SEG57	0178H		SEG57	SEG57	SEG57	SEG57		
0139H		SEG58	SEG58	SEG58	SEG58	0179H		SEG58	SEG58	SEG58	SEG58		
013AH	1D	SEG59	SEG59	SEG59	SEG59	017AH	3D	SEG59	SEG59	SEG59	SEG59		
013BH		SEG60	SEG60	SEG60	SEG60	017BH		SEG60	SEG60	SEG60	SEG60		
013CH	1E	SEG61	SEG61	SEG61	SEG61	017CH	3E	SEG61	SEG61	SEG61	SEG61		
013DH		SEG62	SEG62	SEG62	SEG62	017DH		SEG62	SEG62	SEG62	SEG62		
013EH	1F	SEG63	SEG63	SEG63	SEG63	017EH	3F	SEG63	SEG63	SEG63	SEG63		
013FH		SEG64	SEG64	SEG64	SEG64	017FH		SEG64	SEG64	SEG64	SEG64		
								C_DC8	C_DC7	C_DC6	C_DC5	C_OD8	C_OD7

ADDRESS		BIT3	BIT2	BIT1	BIT0	ADDRESS		BIT3	BIT2	BIT1	BIT0
		COM12	COM11	COM10	COM9			COM16	COM15	COM14	COM13
Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA	Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA
Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE	Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE
0180H	40	SEG1	SEG1	SEG1	SEG1	01C0H	60	SEG1	SEG1	SEG1	SEG1
0181H		SEG2	SEG2	SEG2	SEG2	01C1H		SEG2	SEG2	SEG2	SEG2
0182H	41	SEG3	SEG3	SEG3	SEG3	01C2H	61	SEG3	SEG3	SEG3	SEG3
0183H		SEG4	SEG4	SEG4	SEG4	01C3H		SEG4	SEG4	SEG4	SEG4
0184H	42	SEG5	SEG5	SEG5	SEG5	01C4H	62	SEG5	SEG5	SEG5	SEG5
0185H		SEG6	SEG6	SEG6	SEG6	01C5H		SEG6	SEG6	SEG6	SEG6
0186H	43	SEG7	SEG7	SEG7	SEG7	01C6H	63	SEG7	SEG7	SEG7	SEG7
0187H		SEG8	SEG8	SEG8	SEG8	01C7H		SEG8	SEG8	SEG8	SEG8
0188H	44	SEG9	SEG9	SEG9	SEG9	01C8H	64	SEG9	SEG9	SEG9	SEG9
0189H		SEG10	SEG10	SEG10	SEG10	01C9H		SEG10	SEG10	SEG10	SEG10
018AH	45	SEG11	SEG11	SEG11	SEG11	01CAH	65	SEG11	SEG11	SEG11	SEG11
018BH		SEG12	SEG12	SEG12	SEG12	01CBH		SEG12	SEG12	SEG12	SEG12
018CH	46	SEG13	SEG13	SEG13	SEG13	01CCH	66	SEG13	SEG13	SEG13	SEG13
018DH		SEG14	SEG14	SEG14	SEG14	01CDH		SEG14	SEG14	SEG14	SEG14
018EH	47	SEG15	SEG15	SEG15	SEG15	01CEH	67	SEG15	SEG15	SEG15	SEG15
018FH		SEG16	SEG16	SEG16	SEG16	01CFH		SEG16	SEG16	SEG16	SEG16
0190H	48	SEG17	SEG17	SEG17	SEG17	01D0H	68	SEG17	SEG17	SEG17	SEG17
0191H		SEG18	SEG18	SEG18	SEG18	01D1H		SEG18	SEG18	SEG18	SEG18
0192H	49	SEG19	SEG19	SEG19	SEG19	01D2H	69	SEG19	SEG19	SEG19	SEG19
0193H		SEG20	SEG20	SEG20	SEG20	01D3H		SEG20	SEG20	SEG20	SEG20
0194H	4A	SEG21	SEG21	SEG21	SEG21	01D4H	6A	SEG21	SEG21	SEG21	SEG21
0195H		SEG22	SEG22	SEG22	SEG22	01D5H		SEG22	SEG22	SEG22	SEG22
0196H	4B	SEG23	SEG23	SEG23	SEG23	01D6H	6B	SEG23	SEG23	SEG23	SEG23
0197H		SEG24	SEG24	SEG24	SEG24	01D7H		SEG24	SEG24	SEG24	SEG24
0198H	4C	SEG25	SEG25	SEG25	SEG25	01D8H	6C	SEG25	SEG25	SEG25	SEG25
0199H		SEG26	SEG26	SEG26	SEG26	01D9H		SEG26	SEG26	SEG26	SEG26
019AH	4D	SEG27	SEG27	SEG27	SEG27	01DAH	6D	SEG27	SEG27	SEG27	SEG27
019BH		SEG28	SEG28	SEG28	SEG28	01DBH		SEG28	SEG28	SEG28	SEG28
019CH	4E	SEG29	SEG29	SEG29	SEG29	01DCH	6E	SEG29	SEG29	SEG29	SEG29
019DH		SEG30	SEG30	SEG30	SEG30	01DDH		SEG30	SEG30	SEG30	SEG30
019EH	4F	SEG31	SEG31	SEG31	SEG31	01DEH	6F	SEG31	SEG31	SEG31	SEG31
019FH		SEG32	SEG32	SEG32	SEG32	01DFH		SEG32	SEG32	SEG32	SEG32
01A0H	50	*1	*1	*1	*1	01E0H	70	*1	*1	*1	*1
01A1H		*1	*1	*1	*1	01E1H		*1	*1	*1	*1
01A2H	51	*1	*1	*1	*1	01E2H	71	*1	*1	*1	*1
01A3H		*1	*1	*1	*1	01E3H		*1	*1	*1	*1
01A4H	52	*1	*1	*1	*1	01E4H	72	*1	*1	*1	*1
01A5H		*1	*1	*1	*1	01E5H		*1	*1	*1	*1
01A6H	53	*1	*1	*1	*1	01E6H	73	*1	*1	*1	*1
01A7H		*1	*1	*1	*1	01E7H		*1	*1	*1	*1
01A8H	54	SEG41	SEG41	SEG41	SEG41	01E8H	74	SEG41	SEG41	SEG41	SEG41

ADDRESS		BIT3	BIT2	BIT1	BIT0	ADDRESS		BIT3	BIT2	BIT1	BIT0
		COM12	COM11	COM10	COM9			COM16	COM15	COM14	COM13
Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA	Rx(even)	Lz	DBUSD	DBUSC	DBUSB	DBUSA
Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE	Rx(odd)		DBUSH	DBUSG	DBUSF	DBUSE
01A9H	55	SEG42	SEG42	SEG42	SEG42	01E9H	75	SEG42	SEG42	SEG42	SEG42
01AAH		SEG43	SEG43	SEG43	SEG43	01EAH		SEG43	SEG43	SEG43	SEG43
01ABH		SEG44	SEG44	SEG44	SEG44	01EBH		SEG44	SEG44	SEG44	SEG44
01ACH	56	SEG45	SEG45	SEG45	SEG45	01ECH	76	SEG45	SEG45	SEG45	SEG45
01ADH		SEG46	SEG46	SEG46	SEG46	01EDH		SEG46	SEG46	SEG46	SEG46
01AEH	57	SEG47	SEG47	SEG47	SEG47	01EEH	77	SEG47	SEG47	SEG47	SEG47
01AFH		SEG48	SEG48	SEG48	SEG48	01EFH		SEG48	SEG48	SEG48	SEG48
01B0H	58	SEG49	SEG49	SEG49	SEG49	01F0H	78	SEG49	SEG49	SEG49	SEG49
01B1H		SEG50	SEG50	SEG50	SEG50	01F1H		SEG50	SEG50	SEG50	SEG50
01B2H	59	SEG51	SEG51	SEG51	SEG51	01F2H	79	SEG51	SEG51	SEG51	SEG51
01B3H		SEG52	SEG52	SEG52	SEG52	01F3H		SEG52	SEG52	SEG52	SEG52
01B4H	5A	SEG53	SEG53	SEG53	SEG53	01F4H	7A	SEG53	SEG53	SEG53	SEG53
01B5H		SEG54	SEG54	SEG54	SEG54	01F5H		SEG54	SEG54	SEG54	SEG54
01B6H	5B	SEG55	SEG55	SEG55	SEG55	01F6H	7B	SEG55	SEG55	SEG55	SEG55
01B7H		SEG56	SEG56	SEG56	SEG56	01F7H		SEG56	SEG56	SEG56	SEG56
01B8H	5C	SEG57	SEG57	SEG57	SEG57	01F8H	7C	SEG57	SEG57	SEG57	SEG57
01B9H		SEG58	SEG58	SEG58	SEG58	01F9H		SEG58	SEG58	SEG58	SEG58
01BAH	5D	SEG59	SEG59	SEG59	SEG59	01FAH	7D	SEG59	SEG59	SEG59	SEG59
01BBH		SEG60	SEG60	SEG60	SEG60	01FBH		SEG60	SEG60	SEG60	SEG60
01BCH	5E	SEG61	SEG61	SEG61	SEG61	01FCH	7E	SEG61	SEG61	SEG61	SEG61
01BDH		SEG62	SEG62	SEG62	SEG62	01FDH		SEG62	SEG62	SEG62	SEG62
01BEH	5F	SEG63	SEG63	SEG63	SEG63	01FEH	7F	SEG63	SEG63	SEG63	SEG63
01BFH		SEG64	SEG64	SEG64	SEG64	01FFH		SEG64	SEG64	SEG64	SEG64
		C_DC12	C_DC11	C_DC10	C_DC9	C_DC16		C_DC15	CDC14	C_DC13	
		C_OD12	C_OD11	C_OD10	C_OD9	C_OD16	C_OD15	COD14	C_OD13		

*1: Data RAM Only.

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	V
	VL2	-0.3 to 3.6	V
	VL3	-0.3 to 6.0	V
	VL4	-0.3 to 6.0	V
	VL5	-0.3 to 6.0	V
Maximum Input Voltage	Vin1	-0.3 to VBAT+0.3	V
	Vin2	-0.3 to VL1/2+0.3	V
Maximum output Voltage	Vout1	-0.3 to VBAT+0.3	V
	Vout2	-0.3 to VL1/2+0.3	V
	Vout3	-0.3 to VL3+0.3	V
	Vout4	-0.3 to VL4+0.3	V
	Vout5	-0.3 to VL5+0.3	V
Maximum Operating Temperature	Topg	-40 to +80	°C
Maximum Storage Temperature	Tstg	-40 to +125	°C

POWER CONSUMPTION

At Ta = -40 °C to 80°C, GND = 0V
Halt Conduction: BCF=0, 1/3Bias, 1/16Duty, LCD Alternating Frequency = PH5,

Charge Pump Cycle = PH4,

Only operates in 32.768KHz Crystal Oscillator operating, without loading.

Name	Sym.	Power Mode	Min.	Typ.	Max.	Unit
HALT mode	IHALT1	1.5V Mode, VBAT=1.5V		5		uA
	IHALT2	3V Mode(BCF=0=>BAK=VL1), VBAT=3.0V		2		uA
	IHALT2	3V Mode(BCF=0=>BAK=VBAT), VBAT=3.0V		10		uA
STOP mode	ISTOP				1	uA

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

ALLOWABLE OPERATING CONDITIONS

At Ta = -40 °C to 80°C, GND = 0V

Name	Symb.	Condition	Min.	Max.	Unit
Supply Voltage	VBAT		1.2	3.6	
	VL1		0.95	1.8	V
	VL2		2.0	3.6	V
	VL3		3.0	6.0	V
	VL4		3.0	6.0	V
	VL5		3.0	6.0	V
Oscillator Start-Up Voltage	VDDDB	Crystal Mode BCF=1	1.3		V
Oscillator Sustain Voltage	VDDDB	Crystal Mode BCF=0		1.1	V
Supply Voltage	VBAT	1.5V Power Mode	1.2	1.8	V
Supply Voltage	VBAT	3V Power Mode	2.2	3.6	V
Input "H" Voltage	Vih1	I/O,DC	VBAT-0.6	VBAT+0.6	V
Input "L" Voltage	Vil1		-0.6	0.6	V
Input "H" Voltage	Vih2	CX	VL2-0.6	VL2+0.6	V
Input "L" Voltage	Vil2		-0.6	0.6	V
Input "H" Voltage	Vih3	OSCIN	0.8xBAK	BAK	V
Input "L" Voltage	Vil3		0	0.2xBAK	V
Operating Freq	Fopg1	Crystal Mode	32		KHZ
	Fopg2	RC Mode	10	4096	KHZ
	Fopg3	CF Mode	1000	4096	KHZ

ALLOWABLE OPERATING FREQUENCY

At Ta = -40 °C to 80°C, GND = 0V

Condition	Max, Operating Frequency
BAK=1.2V	1MHz
BAK=2.2V	4MHz

INTERNAL RC FREQUENCY RANGE

At Ta = -40°C to 80°C, GND = 0V

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

ELECTRICAL CHARACTERISTICS

Input Resistance

A t#1: VBAT = 1.5V (1.5V Power Mode)

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

At Ta = -40°C to 80°C, GND = 0V

Name	Symb.	Condition	Min.	Typ.	Max.	Unit
“L” Level Hold Tr(IOC)	Rllh1	Vi=0.2VBAT,#1	10	40	100	Kohm
	Rllh2	Vi=0.2VBAT,#2	10	40	100	Kohm
IOC Pull-Down Tr	Rmad1	Vi=VBAT,#1	200	500	1000	Kohm
	Rmad2	Vi=VBAT,#2	200	500	1000	Kohm
INT Pull-up Tr	Rintu1	Vi=VBAT,#1	50	200	1000	Kohm
	Rintu2	Vi=VBAT,#2	50	350	1000	Kohm
INT Pull-Down Tr	Rintd1	Vi=GND,#1	200	500	1000	Kohm
	Rintd2	Vi=GND,#2	200	500	1000	Kohm
RES Pull-Up R	Rres1	Vi=GND or VBAT,#1	10	40	100	Kohm
	Rres2	Vi=GND or VBAT,#2	10	40	100	Kohm

DC Output Characteristics

At #1: VBAT = 1.2V

At #2: VBAT = 2.4V

At Ta = -40°C to 80°C, GND = 0V

Name	Symb.	Condition	Port	Min.	Typ.	Max.	Unit
Output “H” Voltage	Voh1a	Ioh=-100uA,#1	COM5~16 IOB~D	1.0			V
	Voh2a	Ioh=-1mA,#2		2.0			V
Output “L” Voltage	Vol1a	Iol=200uA,#1	ELC,ELP BZB,BZ			0.2	V
	Vol2a	Iol=2mA,#2				0.4	V
Output “H” Voltage	Voh1b	Ioh=-200uA,#1	IOA, RFC0~2, INT/CX(Vol only)	1.0			V
	Voh2b	Ioh=-3mA,#2		2.0			V
Output “L” Voltage	Vol1b	Iol=400uA,#1				0.2	V
	Vol2b	Iol=5mA,#2				0.4	V

Segment Driver Output Characteristics

At #1: VL1 = 1.2V

At #2: VL2 = 2.4V

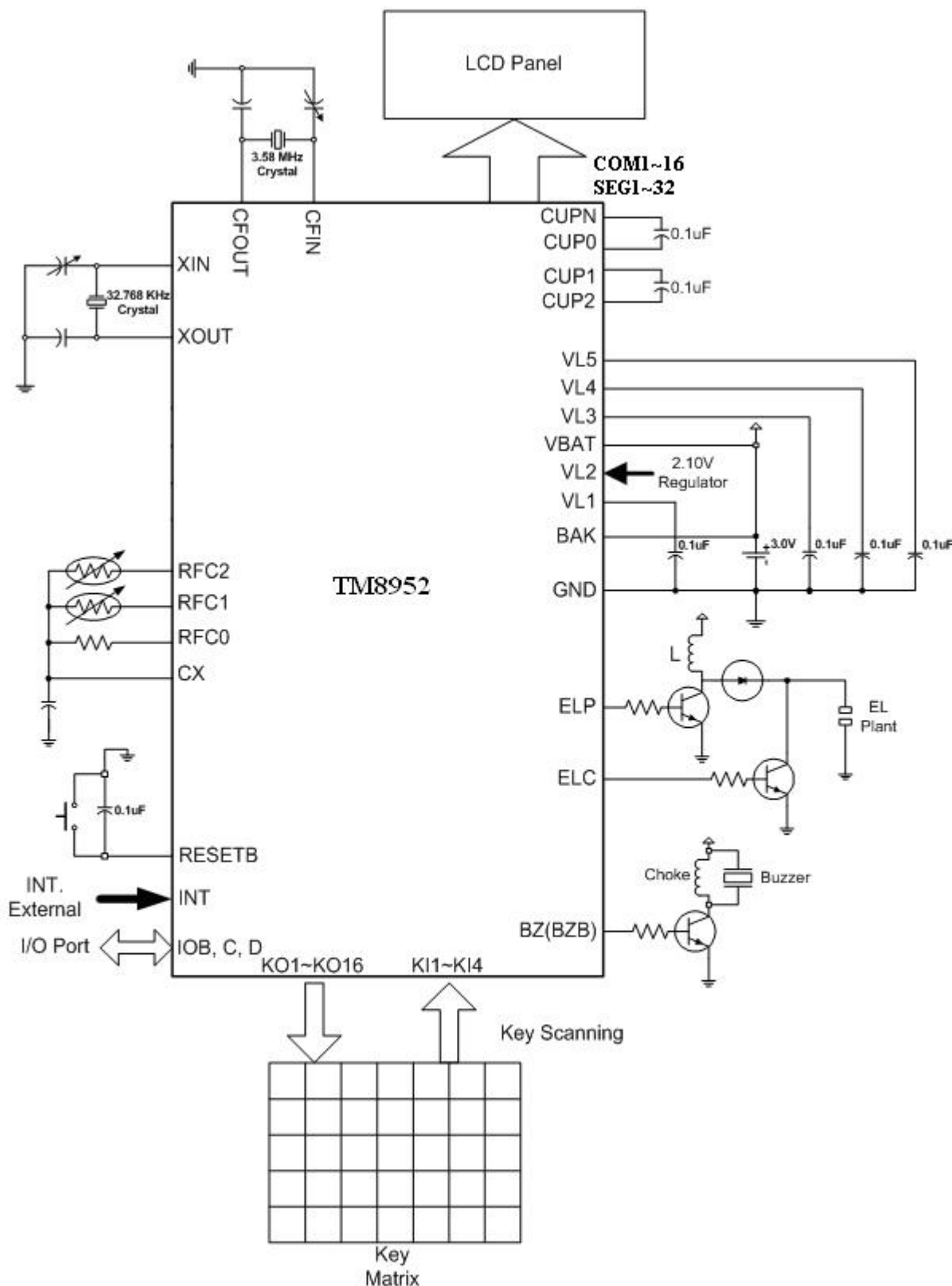
At #3: VL1 = 1.05V

At #4: VL2 = 2.10V

Name	Symb.	Condition	For	Min.	Typ.	Max.	Unit.	
1/3 Bias display Mode								
Output "H" Voltage	Voh12h	loh=-1uA,#1,#2	SEG-n	3.4			V	
	Voh34h	loh=-1uA,#3,#4		2.95			V	
Output "M1" Voltage	Vom112h	lol/h=+/-1uA,#1,#2		1.0		1.4	V	
	Vom134h	lol/h=+/-1uA,#3,#4		0.85		1.25	V	
Output "M2" Voltage	Vom212h	lol/h=+/-1uA,#1,#2		2.2		2.6	V	
	Vom234h	lol/h=+/-1uA,#3,#4		1.95		2.30	V	
Output "L" Voltage	Vol12h	lol=1uA,#1,#2				0.2	V	
	Vol34h	lol=1uA,#3,#4				0.2	V	
Output "H" Voltage	Voh12i	loh=-10uA,#1,#2		COM-n	3.4			V
	Voh34i	loh=-10uA,#3,#4			2.95			V
Output "M1" Voltage	Vom112i	lol/h=+/-10uA,#1,#2			1.0		1.4	V
	Vom134i	lol/h=+/-10uA,#3,#4			0.85		1.25	V
Output "M2" Voltage	Vom212i	lol/h=+/-10uA,#1,#2	2.2			2.6	V	
	Vom234i	lol/h=+/-10uA,#3,#4	1.90			2.30	V	
Output "L" Voltage	Vol12i	lol=10uA,#1,#2				0.2	V	
	Vol34i	lol=10uA,#3,#4				0.2	V	
1/4 Bias display Mode								
Output "H" Voltage	Voh12j	loh=-1uA,#1,#2	SEG-n		4.6			V
	Voh34j	loh=-1uA,#3,#4			4.00			V
Output "M2" Voltage	Vom212j	lol/h=+/-1uA,#1,#2			2.2		2.6	V
	Vom234j	lol/h=+/-1uA,#3,#4		1.90		2.30	V	
Output "L" Voltage	Vol12j	lol=1uA,#1,#2				0.2	V	
	Vol12j	lol=1uA,#3,#4				0.2	V	
Output "H" Voltage	Voh12k	loh=-10uA,#1,#2	COM-n	4.6			V	
	Voh34k	loh=-10uA,#3,#4		4.00			V	
Output "M1" Voltage	Vom112k	lol/h=+/-10uA,#1,#2		1.0		1.4	V	
	Vom134k	lol/h=+/-10uA,#3,#4		0.85		1.25	V	
Output "M3" Voltage	Vom312k	lol/h=+/-10uA,#1,#2		3.4		3.8	V	
	Vom334k	lol/h=+/-10uA,#3,#4		2.95		3.35	V	
Output "L" Voltage	Vol12k	lol=10uA,#1,#2				0.2	V	
	Vol34k	lol=10uA,#3,#4				0.2	V	
1/5 Bias display Mode								
Output "H" Voltage	Voh12l	loh=-1uA,#1,#2		SEG-n	5.80			V
	Voh34l	loh=-1uA,#3,#4			5.05			V
Output "M2" Voltage	Vom212l	lol/h=+/-1uA,#1,#2			2.20		2.60	V
	Vom234l	lol/h=+/-1uA,#3,#4	1.90			2.30	V	
Output "M3" Voltage	Vom312l	lol/h=+/-1uA,#1,#2	3.40			3.80	V	
	Vom334l	lol/h=+/-1uA,#3,#4	2.95			3.35	V	
Output "L" Voltage	Vol12l	lol=1uA,#1,#2				0.2	V	
	Vol34l	lol=1uA,#3,#4				0.2	V	
Output "H" Voltage	Voh12m	loh=-10uA,#1,#2	COM-n		5.80			V
	Voh34m	loh=-10uA,#3,#4			5.05			V
Output "M1" Voltage	Vom112m	lol/h=+/-10uA,#1,#2			1.00		1.40	V
	Vom134m	lol/h=+/-10uA,#3,#4			0.85		1.25	V
Output "M4" Voltage	Vom412m	lol/h=+/-10uA,#1,#2		4.60		5.00	V	
	Vom434m	lol/h=+/-10uA,#3,#4		4.00		4.40	V	
Output "L" Voltage	Vol12m	lol=10uA,#1,#2				0.2	V	
	Vol34m	lol=10uA,#3,#4				0.2	V	

TYPICAL APPLICATION CIRCUIT

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



3V power mode (BAK = VBAT for BCF = 0), 1/5 Bias, 1/16 Duty

ORDERING INFORMATION

The ordering information:

Ordering number	Package
TM8952-COD	Wafer / Dice with code

Appendix A: TM8952 Instruction Table

Instruction	Machine Code	Function	Flag/Remark
NOP	0000 0000 0000 0000	No Operation	NOP
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	(@ZR) '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+1	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID LID& LID% LID\$	@HL,@Z R	(@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	(@ZR)8bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LID8 LID8& LID8% LID8\$	@HL,@Z R	(@HL)8bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LIDH LIDH& LIDH% LIDH\$	@ZR,@ HL	(@ZR)16bits '&,\$': HL '%,\$': ZR ←HL+4 ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LIDH LIDH& LIDH% LIDH\$	@HL,@Z R	(@HL)16 bits '&,\$': HL '%,\$': ZR ←HL+4 ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCT LCT#	@ZR,Ry	(@ZR)8bits '#': ZR ←(7SEG ← (Ry)) ← ZR+2	' ': B4=0 '#': B4=1
LCB LCB#	@ZR,Ry	(@ZR)8bits '#': ZR ←(7SEG ← (Ry)) ← ZR+2	' ': B4=0 '#': B4=1
LCP LCP#	@ZR,Ry	(@ZR)8bits '#': ZR ←(Ry) & AC ← ZR+2	' ': B4=0 '#': B4=1
LCD LCD& LCD% LCD\$	@ZR,@ HL	(@ZR)8bits '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCDH LCDH& LCDH% LCDH\$	@ZR,@ HL	(@ZR)16bits '&,\$': HL ←HL+2 '%,\$': ZR ←ZR+4	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11
LCT LCT& LCT%	@ZR,@ HL	(@ZR)8bits '&,\$': HL ←HL+1 '%,\$': ZR ←ZR+2	' ': B4,3=00 '&': B4,3=01 '%': B4,3=10

Instruction	Machine Code	Function	Flag/Remark
LCT\$			'\$' : B4,3=11
LCB LCB& LCB% LCB\$	@ZR,@ HL	0000 0001 111B B010 (@ZR)8bits '&,\$': HL '%,\$': ZR	← (7SEG ← (@HL)) ←HL+1 ←ZR+2 Blank Zero
LCP LCP& LCP% LCP\$	@ZR,@ HL	0000 0001 111B B011 (@ZR)8bits '&,\$': HL '%,\$': ZR	← (@HL) & AC ←HL+1 ←ZR+2
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
LCT LCT#	Lz,@HL	0000 100Z ZZZZ B001 (Lz) '#': HL	← (7SEG ← (@HL)) ← HL+1
LCB LCB#	Lz,@HL	0000 100Z ZZZZ B010 (Lz) '#': HL	← (7SEG ← (@HL)) ← HL+1 Blank Zero
LCP LCP#	Lz,@HL	0000 100Z ZZZZ B011 (Lz) '#': HL	← (@HL) & AC ← HL+1
LCE LCE#	Lz,@HL	0000 100Z ZZZZ 110B (Lz) '#': HL	← (@HL)8bits ← HL+2
LCE LCE#	Lz,@ZR	0000 100Z ZZZZ 111B (Lz) '#': ZR	← (@ZR)8bits ← ZR+2
OPA	Rx	0000 1010 0XXX XXXX Port(A)	← (Rx)
OPAS	Rx,D	0000 1011 DXXX XXXX A1,2,3,4	← (Rx)0,(Rx)1,D,Pulse
OPB	Rx	0000 1100 0XXX XXXX Port(B)	← (Rx)
OPC	Rx	0000 1101 0XXX XXXX Port(C)	← (Rx)
OPD	Rx	0000 1110 0XXX XXXX Port(D)	← (Rx)
SMUI	Rx	0000 1111 0XXX XXXX MUI	← (Rx)
SMUI SMUI#	@HL	0000 1111 1B00 0000 MUI '#': HL	← (@HL) ← HL+1
SMUI SMUI#	@ZR	0000 1111 1B10 0000 MUI '#': ZR	← (@ZR) ← ZR+1
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
FRQX	D,X	0001 10DD XXXX XXXX FREQ	← X
MVL	Rx	0001 1100 0XXX XXXX IDBF0~3	← (Rx)

Instruction		Machine Code	Function		Flag/Remark
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	ZRBF8~10	← (Rx)0~2	
MVV	Rx	0001 1111 0XXX XXXX	IDBF12~14	← (Rx)0~2	
ADC ADC*	Rx	0010 000B 0XXX XXXX	AC *': (Rx)	← (Rx) + AC + CF ← AC	CF
ADC ADC# ADC* ADC*# ADCM ADCM# ADCM* ADCM*#	@HL	0010 000B 1B00 B000	AC *': (@HL) #': HL	← (@HL) + ? + CF ← AC ← HL+1	'ADC' : ? = AC 'ADCM' : ? = MU ' : 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' : B3=0 'ADCM' : B3=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	
SBC SBC*	Rx	0010 001B 0XXX XXXX	AC *': (Rx)	← (Rx) + ACB + CF ← AC	CF
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@HL	0010 001B 1B00 B000	AC *': (@HL) #': HL	← (@HL) + ? + CF ← AC ← HL+1	'SBC' : ? = ACB 'SBCM' : ? = MUB ' : B8=0 *': B8=1 ' : B6=0 #': B6=1 'SBC' : B3=0

Instruction	Machine Code	Function	Flag/Remark	
SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*#	@HL,DA 0010 001B 1B01 B000	AC ‘*’: (@HL) ‘#’: HL	←BCD((@HL)+?+CF) ← AC ← HL+1	‘SBCM’: 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	
ADD ADD* ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#	Rx @HL @HL,DA @ZR @ZR,DA	AC ‘*’: (Rx) AC ‘*’: (@HL) ‘#’: HL AC ‘*’: (@HL) ‘#’: HL AC ‘*’: (@ZR) ‘#’: ZR AC	← (Rx) + AC ← AC ← (@HL) + ? ← AC ← HL+1 ←BCD((@HL)+?) ← AC ←HL+1 ← (@ZR) + ? ← AC ←ZR+1 ←BCD((@ZR)+?)	CF ‘ADD’: ? = AC ‘ADDM’: ? = MU ‘’: B8=0 ‘*’: B8=1 ‘’: B6=0 ‘#’: B6=1 ‘ADD’: B3=0 ‘ADDM’: B3=1

Instruction		Machine Code	Function		Flag/Remark
ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*#			'*': (@ZR) '#': ZR	← AC ← ZR+1	
SUB SUB*	Rx	0010 011B 0XXX XXXX	AC '*': (Rx)	← (Rx) + ACB + 1 ← AC	CF
SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*#	@HL	0010 011B 1B00 B000	AC '*': (@HL) '#': HL	← (@HL) + ? + 1 ← AC ← HL+1	'SUB' : ? = ACB 'SUBM' : ? = MUB ' ': 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' : B3=0 'SUBM' : 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	
ADN ADN*	Rx	0010 100B 0XXX XXXX	AC '*': (Rx)	← (Rx) + AC ← AC	' ': B8=0 '*': B8=1
ADN ADN# ADN* ADN*#	@HL	0010 100B 1B00 0000	AC '*': (@HL) '#': HL	← (@HL) + AC ← AC ← HL+1	' ': B6=0 '#': B6=1
ADN ADN# ADN* ADN*#	@ZR	0010 100B 1B10 0000	AC '*': (@ZR) '#': ZR	← (@ZR) + AC ← AC ← ZR+1	

Instruction		Machine Code	Function		Flag/Remark
AND AND*	Rx	0010 101B 0XXX XXXX	AC '*': (Rx)	← (Rx) AND AC ← AC	' ': B8=0 '*': B8=1
AND AND# AND* AND*#	@HL	0010 101B 1B00 0000	AC '*': (@HL) '#': HL	← (@HL) AND AC ← AC ← HL+1	' ': B6=0 '#': B6=1
AND AND# AND* AND*#	@ZR	0010 101B 1B10 0000	AC '*': (@ZR) '#': ZR	← (@ZR) AND AC ← AC ← ZR+1	
EOR EOR*	Rx	0010 110B 0XXX XXXX	AC '*': (Rx)	← (Rx) EOR AC ← AC	' ': B8=0 '*': B8=1
EOR EOR# EOR* EOR*#	@HL	0010 110B 1B00 0000	AC '*': (@HL) '#': HL	← (@HL) EOR AC ← AC ← HL+1	' ': B6=0 '#': B6=1
EOR EOR# EOR* EOR*#	@ZR	0010 110B 1B10 0000	AC '*': (@ZR) '#': ZR	← (@ZR) EOR AC ← AC ← ZR+1	
OR OR*	Rx	0010 111B 0XXX XXXX	AC '*': (Rx)	← (Rx) OR AC ← AC	' ': B8=0 '*': B8=1
OR OR# OR* OR*#	@HL	0010 111B 1B00 0000	AC '*': (@HL) '#': HL	← (@HL) OR AC ← AC ← HL+1	' ': B6=0 '#': B6=1
OR OR# OR* OR*#	@ZR	0010 111B 1B10 0000	AC '*': (@ZR) '#': ZR	← (@ZR) OR AC ← AC ← ZR+1	
ADCI	Ry,D	0011 0000 DDDD YYYY	AC	← (Ry) + D + CF	CF
ADCI*	Ry,D	0011 0001 DDDD YYYY	AC,(Ry)	← (Ry) + D + CF	CF
SBCI	Ry,D	0011 0010 DDDD YYYY	AC	← (Ry) + DB + CF	CF
SBCI*	Ry,D	0011 0011 DDDD YYYY	AC,(Ry)	← (Ry) + DB + CF	CF
ADDI	Ry,D	0011 0100 DDDD YYYY	AC	← (Ry) + D	CF
ADDI*	Ry,D	0011 0101 DDDD YYYY	AC,(Ry)	← (Ry) + D	CF
SUBI	Ry,D	0011 0110 DDDD YYYY	AC	← (Ry) + DB + 1	CF
SUBI*	Ry,D	0011 0111 DDDD YYYY	AC,(Ry)	← (Ry) + DB + 1	CF
ADNI	Ry,D	0011 1000 DDDD YYYY	AC	← (Ry) + D	
ADNI*	Ry,D	0011 1001 DDDD YYYY	AC,(Ry)	← (Ry) + D	
ANDI	Ry,D	0011 1010 DDDD YYYY	AC	← (Ry) AND D	
ANDI*	Ry,D	0011 1011 DDDD YYYY	AC,(Ry)	← (Ry) AND D	
EORI	Ry,D	0011 1100 DDDD YYYY	AC	← (Ry) EOR D	
EORI*	Ry,D	0011 1101 DDDD YYYY	AC,(Ry)	← (Ry) EOR D	
ORI	Ry,D	0011 1110 DDDD YYYY	AC	← (Ry) OR D	
ORI*	Ry,D	0011 1111 DDDD YYYY	AC,(Ry)	← (Ry) OR D	
INC*	Rx	0100 0000 0XXX XXXX	AC,(Rx)	← (Rx) + 1	' ': B6=0 '#': B6=1

Instruction		Machine Code	Function		Flag/Remark
INC* INC*#	@HL	0100 0000 1B00 0000	AC,(@HL) ‘#’: HL	← (@HL) + 1 ← HL+1	CF
INC* INC*#	@ZR	0100 0000 1B10 0000	AC,(@ZR) ‘#’: ZR	← (@ZR) + 1 ← ZR+1	
DEC* DEC*#	Rx	0100 0001 0XXX XXXX	AC,(Rx)	← (Rx) - 1	‘‘ : B6=0 ‘#’ : B6=1 CF
DEC* DEC*#	@HL	0100 0001 1B00 0000	AC,(@HL) ‘#’: HL	← (@HL) - 1 ← HL+1	
DEC* DEC*#	@ZR	0100 0001 1B10 0000	AC,(@ZR) ‘#’: ZR	← (@ZR) - 1 ← ZR+1	
MULH MULH#	Rx	0100 0010 0XXX XXXX	MU AC	←(Hex. : H) MUI x (Rx) ←(Hex. : L) MUI x (Rx)	‘‘ : B6=0 ‘#’ : B6=1
MULH MULH#	@HL	0100 0010 1B00 0000	MU AC ‘#’: HL	←(Hex. : H) MUI x (@HL) ←(Hex. : L) MUI x (@HL) ← HL+1	
MULH MULH#	@ZR	0100 0010 1B10 0000	MU AC ‘#’: ZR	←(Hex. : H) MUI x (@ZR) ←(Hex. : L) MUI x (@ZR) ← ZR+1	
MULD MULD#	Rx	0100 0011 0XXX XXXX	MU AC	←(Dec. : H) MUI x (Rx) ←(Dec. : L) MUI x (Rx)	‘‘ : B6=0 ‘#’ : B6=1
MULD MULD#	@HL	0100 0011 1B00 0000	MU AC ‘#’: HL	←(Dec. : H) MUI x (@HL) ←(Dec. : L) MUI x (@HL) ← HL+1	
MULD MULD#	@ZR	0100 0011 1B10 0000	MU AC ‘#’: ZR	←(Dec. : H) MUI x (@ZR) ←(Dec. : L) MUI x (@ZR) ← ZR+1	
IPA	Rx	0100 0110 0XXX XXXX	AC,(Rx)	← Port(A)	
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)	
LDS LDS#	@HL,D	0100 1001 1B00 DDDD	AC, (@HL) ‘#’: HL	← D ← HL+1	
LDS LDS#	@ZR,D	0100 1001 1B10 DDDD	AC,(@ZR) ‘#’: ZR	← D ← ZR+1	‘‘ : B6=0 ‘#’ : B6=1
MAF	Rx	0100 1010 0XXX XXXX	AC,(Rx)	← STS1	B3 : CF B2 : ZERO B1 : No Use B0 : SCF11(CX)
RTM2L	Rx	0100 1010 1XXX XXXX	AC,(Rx)	← TM2(0~3)	
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)

Instruction		Machine Code	Function		Flag/Remark
					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)	← TM3(0~3)	
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 : No Use B0 : CX
MKI	Rx	0100 1111 1XXX XXXX	AC,(Rx)	← STS5(KI4~1)	
SR0	Rx	0101 0000 0XXX XXXX	ACn, (Rx)n AC3, (Rx)3	← (Rx)(n+1) ← 0	
SR1	Rx	0101 0000 1XXX XXXX	ACn, (Rx)n AC3, (Rx)3	← (Rx)(n+1) ← 1	
SL0	Rx	0101 0001 0XXX XXXX	ACn, (Rx)n AC0, (Rx)0	← (Rx)(n-1) ← 0	
SL1	Rx	0101 0001 1XXX XXXX	ACn, (Rx)n AC0, (Rx)0	← (Rx)(n-1) ← 1	
RRC	Rx	0101 0010 0XXX XXXX	ACn, (Rx)n AC3, (Rx)3 CF	← (Rx)(n+1) ← CF ← (Rx)0	CF ‘ ‘ : B6=0 ‘#’ : B6=1
RRC RRC#	@HL	0101 0010 1B00 0000	ACn, (@HL)n AC3, (@HL)3 CF ‘#’: HL	← (@HL)(n+1) ← CF ← (@HL)0 ← HL+1	
RRC RRC#	@ZR	0101 0010 1B10 0000	ACn, (@ZR)n AC3, (@ZR)3 CF ‘#’: ZR	← (@HL)(n+1) ← CF ← (@ZR)0 ← ZR+1	
RLC	Rx	0101 0011 0XXX XXXX	ACn, (Rx)n AC0, (Rx)0 CF	← (Rx)(n-1) ← CF ← (Rx)3	CF ‘ ‘ : B6=0 ‘#’ : B6=1
RLC RLC#	@HL	0101 0011 1B00 0000	ACn, (@HL)n AC0, (@HL)0 CF ‘#’: HL	← (@HL)(n-1) ← CF ← (@HL)3 ← HL+1	
RLC RLC#	@ZR	0101 0011 1B10 0000	ACn, (@ZR)n AC0, (@ZR)0 CF ‘#’: ZR	← (@HL)(n-1) ← CF ← (@ZR)3 ← ZR+1	
DAA		0101 0100 0000 0000	AC	← BCD(CF,AC) for add.	CF
DAA*	Rx	0101 0101 0XXX XXXX	AC,(Rx)	← BCD(CF,AC) for add.	‘ ‘ : B6=0 ‘#’ : B6=1

Instruction		Machine Code	Function		Flag/Remark
DAA* DAA*#	@HL	0101 0101 1B00 0000	AC,(@HL) ‘#’: HL	← BCD(CF,AC) for add. ← HL+1	
DAA* DAA*#	@ZR	0101 0101 1B10 0000	AC,(@ZR) ‘#’: ZR	← BCD(CF,AC) for add. ← ZR+1	
DAS		0101 0110 0000 0000	AC	← BCD(CF,AC) for sub.	CF
DAS* DAS*#	Rx	0101 0111 0XXX XXXX	AC,(Rx)	← BCD(CF,AC) for sub.	‘‘ : B6=0 ‘#’ : B6=1
DAS* DAS*#	@HL	0101 0111 1B00 0000	AC,(@HL) ‘#’: HL	← BCD(CF,AC) for sub. ← HL+1	
DAS* DAS*#	@ZR	0101 0111 1B10 0000	AC,(@ZR) ‘#’: ZR	← BCD(CF,AC) for sub. ← ZR+1	
LDS	Rx,D	0101 1DDD DXXX XXXX	AC,(Rx)	← D	
LDH LDH*	Rx,@HL	0110 000B 0XXX XXXX	AC,(Rx) *’: HL	← H(T(@HL)) ← HL + 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)	← RFC3-0	
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(8bits) ← HL+2	#0
LDS8 LDS8#	@ZR,D	0110 011B DDDD DDDD	(@ZR) ‘#’: ZR	← D(8bits) ← ZR+2	‘‘ : B8=0 ‘#’ : B8=1
STA	Rx	0110 1000 0XXX XXXX	(Rx)	← AC	‘‘ : 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	
MMH	Rx	0110 1001 0XXX XXXX	AC,(Rx)	← MU	‘‘ : B6=0 ‘#’ : B6=1
MMH MMH#	@HL	0110 1001 1B00 0000	AC,(@HL) ‘#’: HL	← MU ← HL+1	
MMH MMH#	@ZR	0110 1001 1B10 0000	AC,(@ZR) ‘#’: ZR	← MU ← ZR+1	
MRW MRW#	@ZR,Rx	0110 1010 BXXX XXXX	AC,(@ZR) ‘#’: ZR	← (Rx) ← ZR+1	‘‘ : B7=0 ‘#’ : B7=1
MWR MWR#	Rx,@ZR	0110 1011 BXXX XXXX	AC,(Rx) ‘#’: ZR	← (@ZR) ← ZR+1	‘‘ : B7=0 ‘#’ : B7=1
LDA	Rx	0110 1100 0XXX XXXX	AC	← (Rx)	‘‘ : 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	
MRA	Rx	0110 1101 0XXX XXXX	CF	← (Rx)3	

Instruction		Machine Code	Function		Flag/Remark
MHL	X	0110 1101 10XX XX00	IDBF0~14	← (Rx)16bits	Rx : 0 0000 10XX XX00b~ 0 0000 10XX XX11b
RHL	X	0110 1101 10XX XX01	(Rx)16bits (Rx)15=0	← IDBF0~14	Rx : 0 0000 10XX XX00b~ 0 0000 10XX XX11b
MZR	X	0110 1101 10XX XX10	ZRBF0~10	← (Rx)16bits	Rx : 0 0000 11XX XX00b~ 0 0000 11XX XX11b
RZR	X	0110 1101 10XX XX11	(Rx)16bits (Rx)15~11=0	← ZRBF0~10	Rx : 0 0000 11XX XX00b~ 0 0000 11XX XX11b
MRW MRW#	@HL,Rx	0110 1110 BXXX XXXX	AC,(@HL) '#': HL	← (Rx) ←HL+1	' ': B7=0 '#': B7=1
MWR MWR#	Rx,@HL	0110 1111 BXXX XXXX	AC,(Rx) '#': HL	← (@HL) ←HL+1	' ': B7=0 '#': B7=1
MRW	Ry,Rx	0111 0YYY YXXX XXXX	AC,(Ry)	← (Rx)	
MWR	Rx,Ry	0111 1YYY YXXX XXXX	AC,(Rx)	← (Ry)	
JB0	X	1000 0XXX XXXX XXXX	PC	← X	if AC0 = 1
JB1	X	1000 1XXX XXXX XXXX	PC	← X	if AC1 = 1
JB2	X	1001 0XXX XXXX XXXX	PC	← X	if AC2 = 1
JB3	X	1001 1XXX XXXX XXXX	PC	← X	if AC3 = 1
JNZ	X	1010 0XXX XXXX XXXX	PC	← X	if AC ≠ 0
JNC	X	1010 1XXX XXXX XXXX	PC	← X	if CF = 0
JZ	X	1011 0XXX XXXX XXXX	PC	← X	if AC = 0
JC	X	1011 1XXX XXXX XXXX	PC	← X	if CF = 1
CALL	X	1100 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 0000 0XXX XXXX	X6~0	:Set Ry Page (0~6,8~ 04Fh) for Rx 0~04FFh	Insert by Compiler
ERY	X	1101 0010 0XXX XXXX	X6~0	:Enable Ry Page (0~6,8~ 04Fh) for Rx 0~04FFh 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 the hardware till all page settings are cleared by "CLPG".
SRX	X	1101 0100 0000 XXXX	X3~0	:Set Rx Page (1~9h)	Insert by Compiler
ERX	X	1101 0110 0000 XXXX	X3~0	:Enable Rx Page (1~9h) Set & Lock	1. Before executing "CLPG (X0=1)", Page still can be changed by

Instruction		Machine Code	Function		Flag/Remark
					executing "ERX" 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".
SLZ	X	1101 0100 0100 00XX	X1~0	:Set Lz Page(1~3h)	Insert 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 1000 0XXX	X2~0	:Set POM BANK X=00~07H	Insert 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= 1010/1000 0111~0100 0011~0000 SD11~6 SD5~0	: Set Counter 17~12 (Timer3) Value :Ctm=INT/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 if TM1 is merged into TM2
T1XH		1101 0111 0100 0000	SD15~12= 1010/1000 0111~0100 0011~0000 SD11~6	:Ctm=INT/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	← (Rx)16bits	Add. "SETDAT SD" to the next address

Instruction		Machine Code	Function		Flag/Remark
			SD12~0 (Rx)15~12=1010/1000 0111~0100 0011~0000 (Rx)11~6	:Rx Address (SD1,0=00b) for 16bits data mode :Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 5~0 (Timer1) Value	SD:0000~04FFH
T2XH		1101 0111 0100 0100	SD15~12=1010/1000 0111~0100 0011~0000 SD11~6 SD5~0	:Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value : Set Counter 5~0 (Timer2) Value	Add. "SETDAT SD" to the next address if TM1 is merged into TM2
T2RH		1101 0111 0100 0110	Timer2 SD12~0 (Rx)15~12=1010/1000 0111~0100 0011~0000 (Rx)11~6 (Rx)5~0	← (Rx)16bits :Rx Address (SD1,0=00b) for 16bits data mode :Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value : Set Counter 5~0 (Timer2) Value	Add. "SETDAT SD" to the next address SD:0000~04FFH if TM1 is merged into TM2
T3XH		1101 0111 0100 1000	SD15~12=1010/1000 0111~0100 0011~0000 SD11~6 SD5~0	:Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value : Set Counter 5~0 (Timer3) Value	Add. "SETDAT SD" to the next address if TM1 is merged into TM3
T3RH		1101 0111 0100 1010	Timer3 SD12~0 (Rx)15~12=1010/1000	← (Rx)16bits :Rx Address (SD1,0=00b) for 16bits data mode :Ctm=INT/CX	Add. "SETDAT SD" to the next address SD:0000~04FFH

Instruction		Machine Code	Function		Flag/Remark
			0111~0100 0011~0000	:Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/	
			(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(16bits)	Add. "SETDAT SD" to the next address
			D=1	: KEY_S release by scanning cycle	
			D=0	: KEY_S release by normal key scanning	
			SD15~0	: Enable KO16~1=1	
SPKRH	D	1101 0111 0100 111D	KO16~1	← (Rx)16bits	Add. "SETDAT SD" to the next address
			D=1	: KEY_S release by scanning cycle	SD:0000~04FFH
			D=0	: KEY_S release by normal key scanning	
			(Rx)15~0	: Enable KO16~1=1	
LDSH LDSH#	@HL	1101 0111 0101 100B	(@HL)16bits '#': HL	← SETDAT SD(16bits) ← HL+4	Add. "SETDAT SD" to the next address ' ': B0=0 '#': B0=1
LDSH LDSH#	@ZR	1101 0111 0101 101B	(@ZR)16bits '#': ZR	← SETDAT SD(16bits) ← ZR+4	
SHLX		1101 0111 0101 1100	IDBF14~0	← SETDAT SD(16bits) SD : 0000~5FFFH	Add. "SETDAT SD" to the next address
SZRX		1101 0111 0101 1101	ZRBF10~0	← SETDAT SD(16bits) SD : 0000~04FFH	Add. "SETDAT SD" to the next address
CPHLH		1101 0111 0101 1110	(PC+2)	← force "NOP" if SETDAT SD(16bits) = IDBF14~0	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(16bits) = ZRBF10~0	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(16bits) ← PC+X+2	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(16bits) ← PC+X+2	Add. "SETDAT SD" to the next X(<16) Addresses range

Instruction		Machine Code	Function		Flag/Remark
T1TH T1TH#	@HL	1101 0111 1000 B000	Timer1 #: HL TD15~12= 1010/1000 0111~0100 0011~0000 TD11~6	← T(@HL)16bits ← HL+2 :Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ : Set Counter 11~6 (Timer1) Value	' ': B3=0 #: B3=1
T2TH T2TH#	@HL	1101 0111 1001 B000	Timer2(2->1) #: HL TD15~12= 1010/1000 0111~0100 0011~0000 TD11~6 TD5~0	← T(@HL)16bits ← HL+2 :Ctm=INT/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 if TM1 is merged into TM2
T3TH T3TH#	@HL	1101 0111 1010 B000	Timer3(3->1) #: HL TD15~12= 1010/1000 0111~0100 0011~0000 TD11~6 TD5~0	← T(@HL)16bits ← HL+2 :Ctm=INT/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 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)16bits ← HL+2 : KEY_S release by scanning cycle : KEY_S release by normal key scanning : Enable KO16~1=1	' ': 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	
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)0~2 AC3/(Rx)3=0	← ZRBF8~10	
RVV	Rx	1101 1011 0XXX XXXX	AC,(Rx)0~2	← IDBF12~14	

Instruction		Machine Code	Function		Flag/Remark
			AC3/(Rx)3=0		
TM3	Rx	1101 1100 0XXX XXXX	Timer3	← (Rx) & AC	
TM3 TM3#	@HL	1101 1101 0000 B000	Timer3 ‘#’: HL	← T(@HL) ← HL+1	‘‘’: B3=0 ‘#’: 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 TMS#	@HL	1110 0001 0000 B000	Timer1 ‘#’: HL	← T(@HL) ← HL+1	‘‘’: B3=0 ‘#’: 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 TM2#	@HL	1110 0101 0000 B000	Timer2 ‘#’: HL	← T(@HL) ← HL+1	‘‘’: B3=0 ‘#’: 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 TMR1
SIE*	X	1110 1001 XXXX XXXX	X7 X6 X5 X4 X3 X2 X1 X0	: Enable IEF7 : Enable IEF6 : Enable IEF5 : Enable IEF4 : Enable IEF3 : Enable IEF2 : Enable IEF1 : Enable IEF0	TMR3 RFC KEY_S TMR2 PDV INT TMR1 A,C,DPT

Instruction		Machine Code	Function		Flag/Remark
PLC	X	1110 101X XXXX XXXX	X8 X7-0	: Reset PH15~11 : Reset HRF7-0	
SRF	X	1110 1100 0000 0XXX	X2~0	:Enable RFC2~0 Output	
SRE	X	1110 1101 0X0X X000	X6 X4 X3	:Enable SRF6(A port) :Enable SRF4(C port) :Enable SRF3(D port)	
FAST		1110 1110 0000 0000	SCLK	: High Speed Clock	
SLOW		1110 1110 1000 0000	SCLK	: Low Speed Clock	
CPHL	X	1110 1111 XXXX XXXX	(PC+1)	← force "NOP" if X7~0=IDBF7~0	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 release by scanning cycle : KEY_S release by normal key scanning : Set one of KO1~16 =1 by X3~0 : Set all = 1 : Set all Hi-z : Set eight of KO1~16 =1 by X3 X3=0 => KO1~8 X3=1 => KO9~16 : Set four of KO1~16 =1 by X3,2 X3,2=00 => KO1~4 X3,2=01 => KO5~8 X3,2=10 => KO9~12 X3,2=11 => KO13~16 : Set two of KO1~16 =1 by X3,2,1 X3~1=000=>KO1,2 X3~1=001=>KO3,4 X3~1=010=>KO5,6 X3~1=011=>KO7,8 X3~1=100=>KO9,10 X3~1=101=>KO11,12 X3~1=110=>KO13,14 X3~1=111=>KO15,16	
RTS		1111 0100 0000 0000	PC	← STACK (CALL Return)	
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
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)
DISTM	X	1111 0100 0011 1XXX	X2~0	: Disable TM3~1 Count	

Instruction		Machine Code	Function		Flag/Remark
SCNT	X	1111 0100 010X XX00	X4=0 X4=1 X3,2=00 X3,2=01 X3,2=10 X3,2=11	Counter clock source for not CX control Mode : CX : FREQ Control Mode : Softwave : TM2 : CX – One Cycle : CX – High Level	initial
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 = BCLK : 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)	
SCX	X	1111 0101 0110 000X	X0	: Enable SEF0(CX)	
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 X6 X5 X4 X2 X1 X0	: Reload 1 Reset : 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	BCLK/8 BCLK/4 BCLK/2 BCLK	

Instruction		Machine Code	Function		Flag/Remark
			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	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 00XX XXXX	X5 X3 X2 X1 X0	:Enable CX Counter :Enable INT powerful Pull- low : Close all Segments : Dis-ENX Set : Reload 2 Set	
RF2	X	1111 1110 10XX XXXX	X5 X3 X2 X1 X0	:Disable CX Counter :Disable INT powerful Pull- low : Release Segments : Dis-ENX Reset : Reload 2 Reset	
HALT		1111 1111 0000 0000	Halt Operation		
STOP		1111 1111 1000 0000	Stop Operation		

Symbol Description

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

(@HL)/(@ZR): 4bits data type for Data RAM

(@HL)8bits/(@ZR)8bits: 8bits data type for Data RAM

(@HL)16bits/(@ZR)16bits: 16bits data type for Data RAM

T(@HL): 8 bits data type for Index ROM

T(@HL)16bits: 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