



十速

TM89P51M

DATA SHEET

Rev V1.2

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, 2011 | New release |
| V1.1 | Dec, 2011 | Add Ordering Information table. |
| V1.2 | Jun, 2014 | Fix application circuit : Reset PIN circuit |

CONTENTS

| | |
|---|----|
| AMENDMENT HISTORY | 2 |
| CONTENTS..... | 3 |
| GENERAL DESCRIPTION | 4 |
| FEATURES | 5 |
| APPLICATION..... | 7 |
| BLOCK DIAGRAM | 7 |
| PAD DIAGRAM..... | 7 |
| PAD ASSIGNMENT | 8 |
| PIN DESCRIPTION | 9 |
| RAM & INITIAL PAGE ASSIGNMENT | 11 |
| CONFIGURATION OF LCD RAM AREA | 12 |
| ABSOLUTE MAXIMUM RATINGS | 15 |
| POWER CONSUMPTION | 15 |
| ALLOWABLE OPERATING CONDITIONS..... | 16 |
| ALLOWABLE OPERATING FREQUENCY | 16 |
| INTERNAL RC FREQUENCY RANGE | 17 |
| ELECTRICAL CHARACTERISTICS | 17 |
| DC OUTPUT CHARACTERISTICS | 18 |
| SEGMENT DRIVER OUTPUT CHARACTERISTICS | 19 |
| TYPICAL APPLICATION CIRCUIT | 20 |
| ORDERING INFORMATION | 21 |
| APPENDIX A TM89P51M INSTRUCTION TABLE..... | 22 |
| SYMBOL DESCRIPTION..... | 41 |

GENERAL DESCRIPTION

TM89P51M is a One Time PROM embedded high-performance 4-bit microcomputer with LCD driver. It contains all the necessary functions, such as 4-bit parallel processing ALU, ROM, RAM, I/O ports, timer, clock generator, dual clock operation, Resistance to Frequency Converter (RFC), EL panel driver, LCD driver, look-up table, watchdog timer and key matrix scanning circuitry in 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 executed directly in addition, subtraction operation.
 - 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 by Page Mode)
 - Table look-up.
 - LCD driver data transfer.
3. ROM capacity. 8K x 16 bits.
 - Instructon ROM Max. capacity 8K x 16 bits.
 - Table ROM Max. capacity 8K/4K x 8/16 bits.
4. RAM capacity. 960/480/240 x 4/8/16 bits.
 - LCD Max. capacity 192/96/48 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.
 - Max 432 LCD dots by 9 common outputs and 48 segment outputs.
 - SEG41~60 can be defined as IOB1~4, KI1~4, ELC, ELP, BZB, BZ, IOD1~4, IOC1~4 by option.
 - 1/4~1/19 Duty can be selected by option.
 - 1/3 ~1/4 Bias can be selected by option.
 - Single instruction to turn off all segments.
 - COM5~9 can be defined as CMOS or P_open drain type output by option.
 - External regulator mode for VL1/2 by 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 option.
- Port IOB 4 pins (with internal pull-low), and can be defined as SEG41~44 by 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 option.
- Port IOD 4 pins (with internal pull-low, input signal chattering prevention circuitry), and can be defined as SEG53~56 by 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, 50 by option.

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

- BZB, BZ. Can be defined as SEG51, 52 by option.

11. Built-in resistance to frequency converter.

- CX, RFC0~2. Can be defined as IOA1~4 by option.

12. Built-in key matrix scanning function.

- KO1~KO16 (Shared with SEG1~16)
- KI1~KI4. Can be defined as SEG45~48 by option.

13. Three 6-bit programmable timers with programmable clock source.

- Read out the content in anytime
- Merged 2 or 3 timers as 12-bit or 18-bit timer
- Used as counter for RFC

14. Watchdog timer.**15. Built-in voltage charge halver & pump circuit.****16. Dual clock operation**

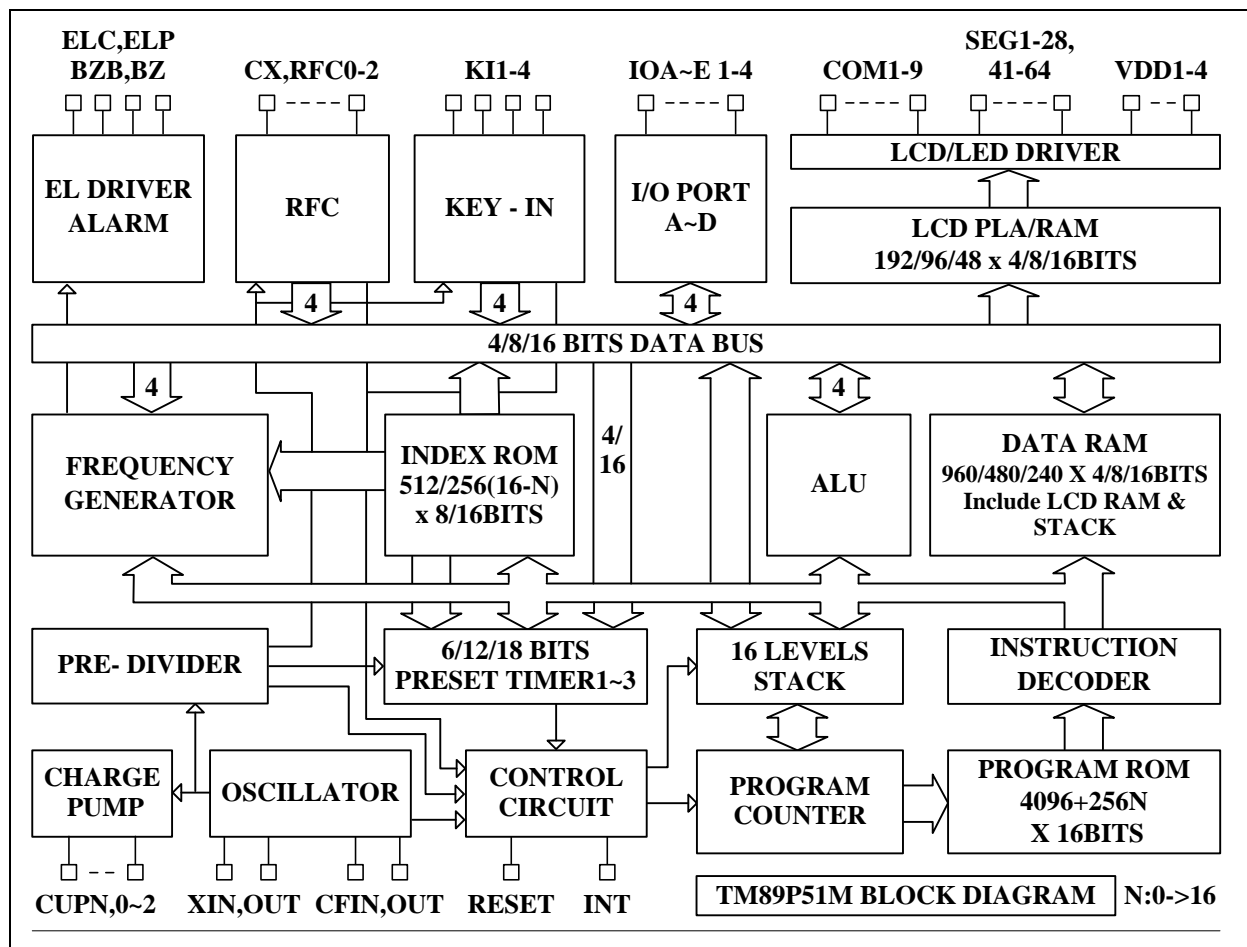
- slow clock oscillation can be defined as X'tal or external RC type oscillator by option.
- fast clock oscillation can be defined as 3.58 MHz ceramic resonator, internal R or external R type oscillator by option.

17. HALT function.**18. STOP function.**

APPLICATION

- Timer / Calendar / Calculator / Thermometer

BLOCK DIAGRAM



PAD DIAGRAM

The substrate of chip should be connected to GND.

PAD ASSIGNMENT

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

PIN DESCRIPTION

| Name | I/O | Description |
|---------------|--------|--|
| BAK | P | Positive Back-up voltage. At 3V power mode, connect a 0.1u capacitor to GND. Positive voltage is needed to BAK for Serial Program/Read Mode. |
| VBAT | P | Positive supply voltage. Positive voltage is needed to VBAT for Serial Program/Read Mode. |
| VL1~4 | P | LCD supply voltage. In 1.5V Power Mode or “External LCD Regulator for 3V Power Mode” option = “VL1”, connect positive power or regulator output to VL1. In 3V Power mode, if “External LCD Regulator for 3V Power Mode” option = “No Use” or “VL2”, connect positive power or regulator output to VL2. In 1.5V Power Mode, connect VL2 to VPP (but VL2 need >=2.1V). Positive voltage is needed to VL4 for Serial Program/Read Mode. |
| RESETB | I | Input pin for external reset request signal, built-in internal pull-up resistor. Signal for Serial Program/Read Mode. |
| INT | I | Input pin for external INT request signal. - Falling edge or rising edge triggered is defined by option. - Internal pull-down or pull-up resistor is defined by option. Signal for Serial Program/Read Mode. |
| VPP | | High voltage is needed to VPP for Serial Program/Read Mode. In 1.5V Power Mode, connect VL2 to VPP. In 3.0V Power Mode, connect VBAT to VPP or Floating. |
| CUPN,0,1,2 | O | Switching pins for supply the LCD driving voltage to the VL1~4 pins. - Connect the CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in 1/3 bias mode. - Connect the CUP0 to CUP2 & CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in 1/4 bias mode. - Connect the CUPN to CUP0 & CUP1 to CUP2 pins with non-polarized electrolytic capacitors when chip operated in more efficient 1/3 & 1/4 bias mode. |
| 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. - The usage of 32 KHz Crystal oscillator or external RC oscillator is defined by option. |
| 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 |
| COM1~9 | O | Output pins for driving the common pins of the LCD panel. COM5~9 can be defined as COMS or Open Drain type output by option. |
| SEG1-28,41-60 | O | Output pins for driving the LCD panel segment. SEG41~44 can be defined as IOB1~4 by option. SEG45~48 can be defined as KI1~4 by option. SEG49~52 can be defined as ELC, ELP, BZB, BZ by option. SEG53~56 can be defined as IOD1~4 by option. SEG57~60 can be defined as IOC1~4 by option. |
| IOA1-4 | I/O | Input/output port A, and can be defined as CX, RFC0~2 by option. |
| IOB1-4 | I/O | Input/output port B, and can be defined as SEG41~44 by option. |
| IOC1-4 | I/O | Input/output port C, and can be defined as SEG57~60 by option. |
| IOD1~4 | I/O | Input/output port D, and can be defined as SEG53~56 by option. |
| CX,RFC0~2 | I O | 1 input pin and 3 output pins for RFC application, and can be defined as IOA1~4 by option. |

| Name | I/O | Description |
|----------|-----|--|
| ELC/ELP | O | Output port for EI panel driver, and can be defined as SEG49,50 by option. |
| BZB/BZ | O | Output port for alarm, clock or single tone melody generator, and can be defined as SEG51, 52 by 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 SEG45~48 by option. |
| GND | P | Negative supply voltage. Negative voltage is need to GND for Serial Program/Read Mode |

SERIAL PROGRAM/READ CONNECTION PINS:

VPP, VBAT, GND, RESET, INT, BAK, VL4

RAM & INITIAL PAGE ASSIGNMENT

| Mode | Initial Page | Page Number | Instruction inserted for set other Page |
|------------------|---------------|-------------|---|
| Rx (0000~03FFh) | 0000~007Fh(0) | 8 (x128) | SRX X (X : 1~7h ; X : 2,3h for LCD) |
| Ry (0000~03FFh) | 0070~007Fh(7) | 60 (x16) | SRY X (X : 0~6,8~03Fh ; X : 10~1Bh for |
| Lz (000~5Fh) | 0~1Fh(0) | 3(x32) | SLZ X (X : 0~2h) |

960 x 4 bits RAM

| | |
|----------------------------|--|
| \$0000 | Rx Page 0h (Initial Page) |
| \$007F | Ry Page 0~(7)h |
| \$0080 | MHL/RHL 16 Address |
| \$00BF \$00C0 | MZR/RZR 16 Address |
| \$00FF | Rx Page 2h (LCD) |
| \$0100 | Ry Page 10~17h (LCD)Rx Page 3h,Ry Page 18~1Bh |
| \$017F | No Use |
| \$0180 \$01BF \$01C0 | |
| \$01FF | |
| \$0200 | STACK 16 Level |
| \$023E \$0240 | |
| \$027F | |
| \$0380 | Rx Page 7h Ry Page 038~03Fh |
| \$03FF | |

\$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) | Lz Page 0h (Initial Page) |
| \$1F(\$013F,013E) \$20(\$0141,0140) | Lz Page 1h |
| \$3F(\$017F,017E) \$40(\$0181,0180) | Lz Page 2h |
| \$5F(\$01BF,01BE) | |

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 |

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 | *1 | *1 | *1 | *1 | 015CH | 2E | *1 | *1 | *1 | *1 |
| 011DH | | *1 | *1 | *1 | *1 | 015DH | | *1 | *1 | *1 | *1 |
| 011EH | 0F | *1 | *1 | *1 | *1 | 015EH | 2F | *1 | *1 | *1 | *1 |
| 011FH | | *1 | *1 | *1 | *1 | 015FH | | *1 | *1 | *1 | *1 |
| 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 |

| ADDRESS | | BIT3 COM4 | BIT2 COM3 | BIT1 COM2 | BIT0 COM1 | ADDRESS | | BIT3 COM8 | BIT2 COM7 | BIT1 COM6 | BIT0 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 |
| 012FH | 18 | SEG48 | SEG48 | SEG48 | SEG48 | 016FH | 38 | SEG48 | SEG48 | SEG48 | SEG48 |
| 0130H | | SEG49 | SEG49 | SEG49 | SEG49 | 0170H | | SEG49 | SEG49 | SEG49 | SEG49 |
| 0131H | | SEG50 | SEG50 | SEG50 | SEG50 | 0171H | | SEG50 | SEG50 | SEG50 | SEG50 |
| 0132H | | 19 | SEG51 | SEG51 | SEG51 | SEG51 | | 0172H | 39 | 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 |
| 0137H | | SEG56 | SEG56 | SEG56 | SEG56 | 0177H | | SEG56 | SEG56 | SEG56 | SEG56 |
| 0138H | 1C | SEG57 | SEG57 | SEG57 | SEG57 | 0178H | 3C | 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 | *1 | *1 | *1 | *1 | 017CH | 3E | *1 | *1 | *1 | *1 |
| 013DH | | *1 | *1 | *1 | *1 | 017DH | | *1 | *1 | *1 | *1 |
| 013EH | 1F | *1 | *1 | *1 | *1 | 017EH | 3F | *1 | *1 | *1 | *1 |
| 013FH | | *1 | *1 | *1 | *1 | 017FH | | C_DC8 C_OD8 | C_DC7 C_OD7 | C_DC6 C_OD6 | C_DC5 C_OD5 |

| ADDRESS | | BIT3 --- | BIT2 --- | BIT1 --- | BIT0 COM9 | ADDRESS | | BIT3 --- | BIT2 --- | BIT1 --- | BIT0 --- |
|----------|----|-------------|-------------|-------------|--------------|----------|----|-------------|-------------|-------------|-------------|
| 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 | *2 | *2 | *2 | *2 |
| 0181H | | SEG2 | SEG2 | SEG2 | SEG2 | 01C1H | | *2 | *2 | *2 | *2 |
| 0182H | 41 | SEG3 | SEG3 | SEG3 | SEG3 | 01C2H | 61 | *2 | *2 | *2 | *2 |
| 0183H | | SEG4 | SEG4 | SEG4 | SEG4 | 01C3H | | *2 | *2 | *2 | *2 |
| 0184H | 42 | SEG5 | SEG5 | SEG5 | SEG5 | 01C4H | 62 | *2 | *2 | *2 | *2 |
| 0185H | | SEG6 | SEG6 | SEG6 | SEG6 | 01C5H | | *2 | *2 | *2 | *2 |
| 0186H | 43 | SEG7 | SEG7 | SEG7 | SEG7 | 01C6H | 63 | *2 | *2 | *2 | *2 |
| 0187H | | SEG8 | SEG8 | SEG8 | SEG8 | 01C7H | | *2 | *2 | *2 | *2 |
| 0188H | 44 | SEG9 | SEG9 | SEG9 | SEG9 | 01C8H | 64 | *2 | *2 | *2 | *2 |
| 0189H | | SEG10 | SEG10 | SEG10 | SEG10 | 01C9H | | *2 | *2 | *2 | *2 |
| 018AH | 45 | SEG11 | SEG11 | SEG11 | SEG11 | 01CAH | 65 | *2 | *2 | *2 | *2 |
| 018BH | | SEG12 | SEG12 | SEG12 | SEG12 | 01CBH | | *2 | *2 | *2 | *2 |
| 018CH | 46 | SEG13 | SEG13 | SEG13 | SEG13 | 01CCH | 66 | *2 | *2 | *2 | *2 |
| 018DH | | SEG14 | SEG14 | SEG14 | SEG14 | 01CDH | | *2 | *2 | *2 | *2 |
| 018EH | 47 | SEG15 | SEG15 | SEG15 | SEG15 | 01CEH | 67 | *2 | *2 | *2 | *2 |
| 018FH | | SEG16 | SEG16 | SEG16 | SEG16 | 01CFH | | *2 | *2 | *2 | *2 |
| 0190H | 48 | SEG17 | SEG17 | SEG17 | SEG17 | 01D0H | 68 | *2 | *2 | *2 | *2 |
| 0191H | | SEG18 | SEG18 | SEG18 | SEG18 | 01D1H | | *2 | *2 | *2 | *2 |
| 0192H | 49 | SEG19 | SEG19 | SEG19 | SEG19 | 01D2H | 69 | *2 | *2 | *2 | *2 |
| 0193H | | SEG20 | SEG20 | SEG20 | SEG20 | 01D3H | | *2 | *2 | *2 | *2 |
| 0194H | 4A | SEG21 | SEG21 | SEG21 | SEG21 | 01D4H | 6A | *2 | *2 | *2 | *2 |
| 0195H | | SEG22 | SEG22 | SEG22 | SEG22 | 01D5H | | *2 | *2 | *2 | *2 |
| 0196H | 4B | SEG23 | SEG23 | SEG23 | SEG23 | 01D6H | 6B | *2 | *2 | *2 | *2 |
| 0197H | | SEG24 | SEG24 | SEG24 | SEG24 | 01D7H | | *2 | *2 | *2 | *2 |
| 0198H | 4C | SEG25 | SEG25 | SEG25 | SEG25 | 01D8H | 6C | *2 | *2 | *2 | *2 |
| 0199H | | SEG26 | SEG26 | SEG26 | SEG26 | 01D9H | | *2 | *2 | *2 | *2 |

| ADDRESS | | BIT3 | BIT2 | BIT1 | BIT0 | ADDRESS | | BIT3 | BIT2 | BIT1 | BIT0 |
|----------|----|-------|-------|-------|----------------|----------|----|-------|-------|-------|-------|
| | | --- | --- | --- | COM9 | | | --- | --- | --- | --- |
| 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 |
| 019AH | 4D | SEG27 | SEG27 | SEG27 | SEG27 | 01DAH | 6D | *2 | *2 | *2 | *2 |
| 019BH | | SEG28 | SEG28 | SEG28 | SEG28 | 01DBH | | *2 | *2 | *2 | *2 |
| 019CH | 4E | SEG29 | SEG29 | SEG29 | SEG29 | 01DCH | 6E | *2 | *2 | *2 | *2 |
| 019DH | | SEG30 | SEG30 | SEG30 | SEG30 | 01DDH | | *2 | *2 | *2 | *2 |
| 019EH | 4F | *1 | *1 | *1 | *1 | 01DEH | 6F | *2 | *2 | *2 | *2 |
| 019FH | | *1 | *1 | *1 | *1 | 01DFH | | *2 | *2 | *2 | *2 |
| 01A0H | 50 | *1 | *1 | *1 | *1 | 01E0H | 70 | *2 | *2 | *2 | *2 |
| 01A1H | | *1 | *1 | *1 | *1 | 01E1H | | *2 | *2 | *2 | *2 |
| 01A2H | 51 | *1 | *1 | *1 | *1 | 01E2H | 71 | *2 | *2 | *2 | *2 |
| 01A3H | | *1 | *1 | *1 | *1 | 01E3H | | *2 | *2 | *2 | *2 |
| 01A4H | 52 | *1 | *1 | *1 | *1 | 01E4H | 72 | *2 | *2 | *2 | *2 |
| 01A5H | | *1 | *1 | *1 | *1 | 01E5H | | *2 | *2 | *2 | *2 |
| 01A6H | 53 | *1 | *1 | *1 | *1 | 01E6H | 73 | *2 | *2 | *2 | *2 |
| 01A7H | | *1 | *1 | *1 | *1 | 01E7H | | *2 | *2 | *2 | *2 |
| 01A8H | 54 | SEG41 | SEG41 | SEG41 | SEG41 | 01E8H | 74 | *2 | *2 | *2 | *2 |
| 01A9H | | SEG42 | SEG42 | SEG42 | SEG42 | 01E9H | | *2 | *2 | *2 | *2 |
| 01AAH | 55 | SEG43 | SEG43 | SEG43 | SEG43 | 01EAH | 75 | *2 | *2 | *2 | *2 |
| 01ABH | | SEG44 | SEG44 | SEG44 | SEG44 | 01EBH | | *2 | *2 | *2 | *2 |
| 01ACH | 56 | SEG45 | SEG45 | SEG45 | SEG45 | 01ECH | 76 | *2 | *2 | *2 | *2 |
| 01ADH | | SEG46 | SEG46 | SEG46 | SEG46 | 01EDH | | *2 | *2 | *2 | *2 |
| 01AEH | 57 | SEG47 | SEG47 | SEG47 | SEG47 | 01EEH | 77 | *2 | *2 | *2 | *2 |
| 01AFH | | SEG48 | SEG48 | SEG48 | SEG48 | 01EFH | | *2 | *2 | *2 | *2 |
| 01B0H | 58 | SEG49 | SEG49 | SEG49 | SEG49 | 01F0H | 78 | *2 | *2 | *2 | *2 |
| 01B1H | | SEG50 | SEG50 | SEG50 | SEG50 | 01F1H | | *2 | *2 | *2 | *2 |
| 01B2H | 59 | SEG51 | SEG51 | SEG51 | SEG51 | 01F2H | 79 | *2 | *2 | *2 | *2 |
| 01B3H | | SEG52 | SEG52 | SEG52 | SEG52 | 01F3H | | *2 | *2 | *2 | *2 |
| 01B4H | 5A | SEG53 | SEG53 | SEG53 | SEG53 | 01F4H | 7A | *2 | *2 | *2 | *2 |
| 01B5H | | SEG54 | SEG54 | SEG54 | SEG54 | 01F5H | | *2 | *2 | *2 | *2 |
| 01B6H | 5B | SEG55 | SEG55 | SEG55 | SEG55 | 01F6H | 7B | *2 | *2 | *2 | *2 |
| 01B7H | | SEG56 | SEG56 | SEG56 | SEG56 | 01F7H | | *2 | *2 | *2 | *2 |
| 01B8H | 5C | SEG57 | SEG57 | SEG57 | SEG57 | 01F8H | 7C | *2 | *2 | *2 | *2 |
| 01B9H | | SEG58 | SEG58 | SEG58 | SEG58 | 01F9H | | *2 | *2 | *2 | *2 |
| 01BAH | 5D | SEG59 | SEG59 | SEG59 | SEG59 | 01FAH | 7D | *2 | *2 | *2 | *2 |
| 01BBH | | SEG60 | SEG60 | SEG60 | SEG60 | 01FBH | | *2 | *2 | *2 | *2 |
| 01BCH | 5E | *1 | *1 | *1 | *1 | 01FCH | 7E | *2 | *2 | *2 | *2 |
| 01BDH | | *1 | *1 | *1 | *1 | 01FDH | | *2 | *2 | *2 | *2 |
| 01BEH | 5F | *1 | *1 | *1 | *1 | 01FEH | 7F | *2 | *2 | *2 | *2 |
| 01BFH | | *1 | *1 | *1 | C_DC9 C_OD9 | 01FFH | | *2 | *2 | *2 | *2 |

*1: Data RAM Only.

*2: RAM No Use.

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 |
| 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 | |
| | 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 |
| 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 Condition: BCF=0, 1/3Bias, 1/16Duty, LCD Alternating Frequency = PH5, Charge Pump Cycle = PH4, Only 32.768 KHz Crystal oscillator operating, without loading.

| Name | Sym. | Power Mode | Min. | Typ. | Max. | Unit |
|-----------|--------|-------------------------------------|------|------|------|------|
| | IHALT1 | 1.5V Mode, VBAT=1.5V | | 5 | | uA |
| HALT mode | 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#1: 1.5V Power Mode

At#2: 3V Power Mode

At Ta=-40°C to 80°C, GND= 0V; X'tal No Matched Capacitance

| 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 | VDDB | Crystal Mode BCF=1, #1 | 1.4 | | V |
| | | Crystal Mode BCF=1, #2 | 1.8 | | V |
| Oscillator Sustain Voltage | VDDB | 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 BAK=VBAT for BCF=0 | 2.2 | 3.6 | V |
| | | 3V Power Mode BAK=VL1 for BCF=0 | 2.4 | 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/2 | 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 | 500 KHz |
| BAK=2.2V | 4 MHz |

INTERNAL RC FREQUENCY RANGE

At Ta=-40°C to 80°C, GND= 0V

| Option Mode | BAK | Min. | Typ. | Max. |
|-------------|------|---------|---------|---------|
| 250 KHz | 1.5V | 100 KHz | 180 KHz | 300 KHz |
| | 3.0V | 200 KHz | 250 KHz | 300 KHz |
| 500 KHz | 1.5V | 200 KHz | 360 KHz | 600 KHz |
| | 3.0V | 400 KHz | 500 KHz | 600 KHz |

ELECTRICAL CHARACTERISTICS

At#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

Input Resistance

| 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 | COM5~9 IOB~D ELC, ELP BZB, BZ | 1.0 | | | V |
| | Voh2a | Ioh=-1mA, #2 | | 2.0 | | | V |
| Output "L" Voltage | Vol1a | Iol=200uA, #1 | | | | 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 |

At#1: VL1=1.2V

At#2: VL2=2.4V

At#3: VL1=1.05V

At#4: VL2=2.10V

SEGMENT DRIVER OUTPUT CHARACTERISTICS

| Name | Symb. | Condition | For | Min. | Typ. | Max. | Unit. | |
|------------------------------|---------|---------------------|-------|-------|------|------|-------|---|
| 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 | | | V | |
| Output "M1" Voltage | Vom112h | Iol/h=+/-1uA,#1,#2 | | 1.0 | | 1.4 | V | |
| | Vom134h | Iol/h=+/-1uA,#3,#4 | | 0.85 | | 1.25 | V | |
| Output "M2" Voltage | Vom212h | Iol/h=+/-1uA,#1,#2 | | 2.2 | | 2.6 | V | |
| | Vom234h | Iol/h=+/-1uA,#3,#4 | | 1.95 | | 2.30 | V | |
| Output "L" Voltage | Vol12h | Iol=1uA,#1,#2 | | | | 0.2 | V | |
| | Vol34h | Iol=1uA,#3,#4 | | | | 0.2 | V | |
| Output "H" Voltage | Voh12i | Ioh=-10uA,#1,#2 | | COM-n | 3.4 | | | V |
| | Voh34i | Ioh=-10uA,#3,#4 | | | 2.95 | | | V |
| Output "M1" Voltage | Vom112i | Iol/h=+/-10uA,#1,#2 | | | 1.0 | | 1.4 | V |
| | Vom134i | Iol/h=+/-10uA,#3,#4 | | | 0.85 | | 1.25 | V |
| Output "M2" Voltage | Vom212i | Iol/h=+/-10uA,#1,#2 | 2.2 | | | 2.6 | V | |
| | Vom234i | Iol/h=+/-10uA,#3,#4 | 1.90 | | | 2.30 | V | |
| Output "L" Voltage | Vol12i | Iol=10uA,#1,#2 | | | | 0.2 | V | |
| | Vol34i | Iol=10uA,#3,#4 | | | | 0.2 | V | |
| 1/4 Bias display Mode | | | | | | | | |
| Output "H" Voltage | Voh12j | Ioh=-1uA,#1,#2 | SEG-n | | 4.6 | | | V |
| | Voh34j | Ioh=-1uA,#3,#4 | | | 4.00 | | | |
| Output "M2" Voltage | Vom212j | Iol/h=+/-1uA,#1,#2 | | | 2.2 | | 2.6 | V |
| | Vom234j | Iol/h=+/-1uA,#3,#4 | | 1.90 | | 2.30 | V | |
| Output "L" Voltage | Vol12j | Iol=1uA,#1,#2 | | | | 0.2 | V | |
| | Vol12j | Iol=1uA,#3,#4 | | | | 0.2 | V | |
| Output "H" Voltage | Voh12k | Ioh=-10uA,#1,#2 | | COM-n | 4.6 | | | V |
| | Voh34k | Ioh=-10uA,#3,#4 | | | 4.00 | | | V |
| Output "M1" Voltage | Vom112k | Iol/h=+/-10uA,#1,#2 | | | 1.0 | | 1.4 | V |
| | Vom134k | Iol/h=+/-10uA,#3,#4 | | | 0.85 | | 1.25 | V |
| Output "M3" Voltage | Vom312k | Iol/h=+/-10uA,#1,#2 | | | 3.4 | | 3.8 | V |
| | Vom334k | Iol/h=+/-10uA,#3,#4 | | | 2.95 | | 3.35 | V |
| Output "L" Voltage | Vol12k | Iol=10uA,#1,#2 | | | | 0.2 | V | |
| | Vol34k | Iol=10uA,#3,#4 | | | | 0.2 | V | |

ORDERING INFORMATION

The ordering information:

| Ordering number | Package |
|-----------------|-------------------------|
| TM89P51M-000 | Wafer / Dice blank chip |
| TM89P51M-COD | Wafer / Dice with code |

APPENDIX A TM89P51M INSTRUCTION TABLE

| Instruction | | Machine Code | Function | | Flag/Remark |
|----------------------------------|---------|---------------------|--|---|---|
| NOP | | 0000 0000 0000 0000 | No Operation | | |
| IDC& IDC% IDC\$ | | 0000 0001 010B B010 | '&,\$': HL '%,\$': ZR | ←HL+1 ←ZR+1 | '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11 |
| IDC8& IDC8% IDC8\$ | | 0000 0001 010B B100 | '&,\$': HL '%,\$': ZR | ←HL+2 ←ZR+2 | '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11 |
| IDCH& IDCH% IDCH\$ | | 0000 0001 010B B110 | '&,\$': HL '%,\$': ZR | ←HL+4 ←ZR+4 | '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11 |
| LID LID& LID% LID\$ | @ZR,@HL | 0000 0001 011B B010 | (@ZR) '&,\$': HL '%,\$': ZR | ← (@HL) ←HL+1 ←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 '%,\$': ZR | ← (@ZR) ←HL+1 ←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 '%,\$': ZR | ← (@HL)8 bits ←HL+2 ←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 '%,\$': ZR | ← (@ZR)8 bits ←HL+2 ←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 '%,\$': ZR | ← (@HL)16 bits ←HL+4 ←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 '%,\$': ZR | ← (@ZR)16 bits ←HL+4 ←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 | ← (7SEG ← (Ry)) ← ZR+2 | ' ': B4=0 '#': B4=1 |
| LCB LCB# | @ZR,Ry | 0000 0001 101B YYYY | (@ZR)8 bits '#': ZR | ← (7SEG ← (Ry)) ← ZR+2 Blank Zero | ' ': B4=0 '#': B4=1 |
| LCP LCP# | @ZR,Ry | 0000 0001 110B YYYY | (@ZR)8 bits '#': ZR | ← (Ry) & AC ← ZR+2 | ' ': B4=0 '#': B4=1 |
| LCD LCD& LCD% LCD\$ | @ZR,@HL | 0000 0001 111B B000 | (@ZR)8 bits '&,\$': HL '%,\$': ZR | ← T(@HL) ←HL+1 ←ZR+2 | ' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 '\$': B4,3=11 |
| LCDH LCDH& LCDH% | @ZR,@HL | 0000 0001 111B B100 | (@ZR)16 bits '&,\$': HL '%,\$': ZR | ← T(@HL)16 bits ←HL+2 ←ZR+4 | ' ': B4,3=00 '&': B4,3=01 '%': B4,3=10 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|------------------------------|--------------------------|---|---|--|--|
| LCDH\$ | | | | | 'S': B4,3=11 |
| LCT LCT& LCT% LCT\$ | @ZR,@HL | 0000 0001 111B B001 | (@ZR)8 bits '&,\$': HL '%,\$': ZR | ← (7SEG ← (@HL)) ←HL+1 ←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 '%,\$': ZR | ← (7SEG ← (@HL)) ←HL+1 ←ZR+2 Blank Zero | '?' : B4,3=00 '&?' : B4,3=01 '%?' : B4,3=10 '\$?' : B4,3=11 |
| 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 |
| LCE LCE# | Lz,@HL | 0000 100Z ZZZZ 110B | (Lz) '#': HL | ← (@HL)8bits ← HL+2 | '?' : B0=0 '#?' : B0=1 |
| LCE LCE# | Lz,@ZR | 0000 100Z ZZZZ 111B | (Lz) '#': ZR | ← (@ZR)8bits ← ZR+2 | '?' : B0=0 '#?' : B0=1 |
| 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 SMUI SMUI# | Rx @HL @ZR | 0000 1111 0XXX XXXX 0000 1111 1B00 0000 0000 1111 1B10 0000 | MUI MUI '#': HL MUI '#': ZR | ← (Rx) ← (@HL) ← HL+1 ← (@ZR) ← ZR+1 | '?' : B6=0 '#?' : B6=1 |
| FRQ | D,Rx | 0001 00DD 0XXX XXXX | FREQ D=00 D=01 D=10 | ← (Rx) & AC : 1/4 Duty : 1/3 Duty : 1/2 Duty | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|--|--------|---------------------|-----------------------------|-------------------------------------|---|
| | | | D=11 | : 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 | ZRBF8~9 | ← (Rx)0~1 | |
| MVV | Rx | 0001 1111 0XXX XXXX | IDBF12 | ← (Rx)0 | |
| 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 |
| 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’: ? = ACB |

| Instruction | | Machine Code | Function | | Flag/Remark |
|--|--------|---------------------|-----------------------------|--|--|
| SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*# | @HL | 0010 001B 1B00 B000 | AC '*': (@HL) '#': HL | $\leftarrow (@HL) + ? + CF$ $\leftarrow AC$ $\leftarrow HL+1$ | 'SBCM': ? = MUB ' ': B8=0 '*': B8=1 ' ': B6=0 '#': B6=1 'SBC' : B3=0 'SBCM' : B3=1 |
| SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*# | @HL,DA | 0010 001B 1B01 B000 | AC '*': (@HL) '#': HL | $\leftarrow BCD((@HL)+?+CF)$ $\leftarrow AC$ $\leftarrow HL+1$ | |
| SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*# | @ZR | 0010 001B 1B10 B000 | AC '*': (@ZR) '#': ZR | $\leftarrow (@ZR) + ? + CF$ $\leftarrow AC$ $\leftarrow ZR+1$ | |
| SBC SBC# SBC* SBC*# SBCM SBCM# SBCM* SBCM*# | @ZR,DA | 0010 001B 1B11 B000 | AC '*': (@ZR) '#': ZR | $\leftarrow BCD((@ZR)+?+CF)$ $\leftarrow AC$ $\leftarrow ZR+1$ | |
| ADD ADD* | Rx | 0010 010B 0XXX XXXX | AC '*': (Rx) | $\leftarrow (Rx) + AC$ $\leftarrow AC$ | CF 'ADD' : ? = AC |
| ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# | @HL | 0010 010B 1B00 B000 | AC '*': (@HL) '#': HL | $\leftarrow (@HL) + ?$ $\leftarrow AC$ $\leftarrow HL+1$ | 'ADDM' : ? = MU ' ': 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 | $\leftarrow BCD((@HL)+?)$ $\leftarrow AC$ $\leftarrow HL+1$ | 'ADD' : B3=0 'ADDM' : B3=1 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|--|--------|---------------------|-------------------------------|---|---|
| ADDM*# | | | | | |
| ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# | @ZR | 0010 010B 1B10 B000 | AC ‘*’ : (@ZR) ‘#’ : ZR | $\leftarrow (@ZR) + ?$ $\leftarrow AC$ $\leftarrow ZR+1$ | |
| ADD ADD# ADD* ADD*# ADDM ADDM# ADDM* ADDM*# | @ZR,DA | 0010 010B 1B11 B000 | AC ‘*’ : (@ZR) ‘#’ : ZR | $\leftarrow BCD((@ZR)+?)$ $\leftarrow AC$ $\leftarrow ZR+1$ | |
| SUB SUB* | Rx | 0010 011B 0XXX XXXX | AC ‘*’ : (Rx) | $\leftarrow (Rx) + ACB + 1$ $\leftarrow AC$ | CF |
| SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*# | @HL | 0010 011B 1B00 B000 | AC ‘*’ : (@HL) ‘#’ : HL | $\leftarrow (@HL) + ? + 1$ $\leftarrow AC$ $\leftarrow HL+1$ | ‘SUB’ : ? = ACB ‘SUBM’ : ? = MUB ‘ ’ : B8=0 ‘*’ : B8=1 |
| SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*# | @HL,DA | 0010 011B 1B01 B000 | AC ‘*’ : (@HL) ‘#’ : HL | $\leftarrow BCD((@HL)+?+1)$ $\leftarrow AC$ $\leftarrow HL+1$ | ‘SUB’ : B3=0 ‘SUBM’ : B3=1 |
| SUB SUB# SUB* SUB*# SUBM SUBM# SUBM* SUBM*# | @ZR | 0010 011B 1B10 B000 | AC ‘*’ : (@ZR) ‘#’ : ZR | $\leftarrow (@ZR) + ? + 1$ $\leftarrow AC$ $\leftarrow ZR+1$ | |
| SUB SUB# SUB* SUB*# SUBM | @ZR,DA | 0010 011B 1B11 B000 | AC ‘*’ : (@ZR) ‘#’ : ZR | $\leftarrow BCD((@ZR)+?+1)$ $\leftarrow AC$ $\leftarrow ZR+1$ | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|------------------------------|------|---------------------|-----------------------------|---------------------------------|------------------------|
| SUBM# SUBM* SUBM*# | | | | | |
| 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 | |
| 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 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|------|---------------------|----------|--|------------------------|
| SBCI | Ry,D | 0011 0010 DDDD YYYY | AC | $\leftarrow (Ry) + DB + CF$ | CF |
| SBCI* | Ry,D | 0011 0011 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) + DB + CF$ | CF |
| ADDI | Ry,D | 0011 0100 DDDD YYYY | AC | $\leftarrow (Ry) + D$ | CF |
| ADDI* | Ry,D | 0011 0101 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) + D$ | CF |
| SUBI | Ry,D | 0011 0110 DDDD YYYY | AC | $\leftarrow (Ry) + DB + 1$ | CF |
| SUBI* | Ry,D | 0011 0111 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) + DB + 1$ | CF |
| ADNI | Ry,D | 0011 1000 DDDD YYYY | AC | $\leftarrow (Ry) + D$ | |
| ADNI* | Ry,D | 0011 1001 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) + D$ | |
| ANDI | Ry,D | 0011 1010 DDDD YYYY | AC | $\leftarrow (Ry) \text{ AND } D$ | |
| ANDI* | Ry,D | 0011 1011 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) \text{ AND } D$ | |
| EORI | Ry,D | 0011 1100 DDDD YYYY | AC | $\leftarrow (Ry) \text{ EOR } D$ | |
| EORI* | Ry,D | 0011 1101 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) \text{ EOR } D$ | |
| ORI | Ry,D | 0011 1110 DDDD YYYY | AC | $\leftarrow (Ry) \text{ OR } D$ | |
| ORI* | Ry,D | 0011 1111 DDDD YYYY | AC,(Ry) | $\leftarrow (Ry) \text{ OR } D$ | |
| INC* | Rx | 0100 0000 0XXX XXXX | AC,(Rx) | $\leftarrow (Rx) + 1$ | ' ': B6=0 '#': B6=1 |
| INC* | @HL | 0100 0000 1B00 0000 | AC,(@HL) | $\leftarrow (@HL) + 1$ | CF |
| INC*# | | | '#': HL | $\leftarrow HL+1$ | |
| INC* | @ZR | 0100 0000 1B10 0000 | AC,(@ZR) | $\leftarrow (@ZR) + 1$ | |
| INC*# | | | '#': ZR | $\leftarrow ZR+1$ | |
| DEC* | Rx | 0100 0001 0XXX XXXX | AC,(Rx) | $\leftarrow (Rx) - 1$ | ' ': B6=0 '#': B6=1 |
| DEC* | @HL | 0100 0001 1B00 0000 | AC,(@HL) | $\leftarrow (@HL) - 1$ | CF |
| DEC*# | | | '#': HL | $\leftarrow HL+1$ | |
| DEC* | @ZR | 0100 0001 1B10 0000 | AC,(@ZR) | $\leftarrow (@ZR) - 1$ | |
| DEC*# | | | '#': ZR | $\leftarrow ZR+1$ | |
| MULH | Rx | 0100 0010 0XXX XXXX | MU AC | $\leftarrow (\text{Hex. : H}) \text{ MUI } x (Rx)$ $\leftarrow (\text{Hex. : L}) \text{ MUI } x (Rx)$ | ' ': B6=0 '#': B6=1 |
| MULH | @HL | 0100 0010 1B00 0000 | MU AC | $\leftarrow (\text{Hex. : H}) \text{ MUI } x (@HL)$ $\leftarrow (\text{Hex. : L}) \text{ MUI } x (@HL)$ | |
| MULH# | | | '#': HL | $\leftarrow HL+1$ | |
| MULH | @ZR | 0100 0010 1B10 0000 | MU AC | $\leftarrow (\text{Hex. : H}) \text{ MUI } x (@ZR)$ $\leftarrow (\text{Hex. : L}) \text{ MUI } x (@ZR)$ | |
| MULH# | | | '#': ZR | $\leftarrow ZR+1$ | |
| MULD | Rx | 0100 0011 0XXX XXXX | MU AC | $\leftarrow (\text{Dec. : H}) \text{ MUI } x (Rx)$ $\leftarrow (\text{Dec. : L}) \text{ MUI } x (Rx)$ | ' ': B6=0 '#': B6=1 |
| MULD | @HL | 0100 0011 1B00 0000 | MU AC | $\leftarrow (\text{Dec. : H}) \text{ MUI } x (@HL)$ $\leftarrow (\text{Dec. : L}) \text{ MUI } x (@HL)$ | |
| MULD# | | | '#': HL | $\leftarrow HL+1$ | |
| MULD | @ZR | 0100 0011 1B10 0000 | MU AC | $\leftarrow (\text{Dec. : H}) \text{ MUI } x (@ZR)$ $\leftarrow (\text{Dec. : L}) \text{ MUI } x (@ZR)$ | |
| MULD# | | | '#': ZR | $\leftarrow ZR+1$ | |
| IPA | Rx | 0100 0110 0XXX XXXX | AC,(Rx) | $\leftarrow \text{Port}(A)$ | |
| IPB | Rx | 0100 0110 1XXX XXXX | AC,(Rx) | $\leftarrow \text{Port}(B)$ | |
| IPC | Rx | 0100 0111 0XXX XXXX | AC,(Rx) | $\leftarrow \text{Port}(C)$ | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|-------|---------------------|--------------------------------|--------------------------------|---|
| 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 | #0 |
| 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 |
| RTM2I | 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) | ← TM3(0~3) | |
| MSD | Rx | 0100 1110 0XXX XXXX | AC,(Rx) | ← STS4 | B3 : No Use B2 : RFOVF B1 : WDF B0 : CSF |
| RTM3I | 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 | ← (@HL)(n+1) ← CF | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|------------------------------|------------------|--|--|--|------------------------------|
| RRC RRC# | @ZR | 0101 0010 1B10 0000 | CF '#': HL ACn, (@ZR)n AC3, (@ZR)3 CF '#': ZR | ← (@HL)0 ← HL+1 ← (@HL)(n+1) ← CF ← (@ZR)0 ← ZR+1 | |
| RLC 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 DAA* DAA* DAA*# | Rx @HL @ZR | 0101 0100 0000 0000 0101 0101 0XXX XXXX 0101 0101 1B00 0000 0101 0101 1B10 0000 | AC AC,(Rx) AC,(@HL) '#': HL AC,(@ZR) '#': ZR | ← BCD(CF,AC) for add. ← BCD(CF,AC) for add. ← BCD(CF,AC) for add. ← HL+1 ← BCD(CF,AC) for add. ← ZR+1 | CF ' ': B6=0 '#': B6=1 |
| DAS DAS* DAS* DAS*# | Rx @HL @ZR | 0101 0110 0000 0000 0101 0111 0XXX XXXX 0101 0111 1B00 0000 0101 0111 1B10 0000 | AC AC,(Rx) AC,(@HL) '#': HL AC,(@ZR) '#': ZR | ← BCD(CF,AC) for sub. ← BCD(CF,AC) for sub. ← BCD(CF,AC) for sub. ← HL+1 ← BCD(CF,AC) for sub. ← ZR+1 | CF ' ': 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 | ← 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(8 bits) ← HL+2 | #0 |
| LDS8 | @ZR,D | 0110 011B DDDD DDDD | (@ZR) | ← D(8 bits) | ' ': B8=0 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|--------|---------------------|----------------------------|-------------------|---|
| LDS8# | | | '#': ZR | ← ZR+2 | '#': 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 | |
| MHL | X | 0110 1101 10XX XX00 | IDBF0~12 | ← (Rx)16 bits | Rx : 0 0000 10XX XX00b~ 0 0000 10XX XX11b |
| RHL | X | 0110 1101 10XX XX01 | (Rx)16 bits (Rx)15~13=0 | ← IDBF0~12 | Rx : 0 0000 10XX XX00b~ 0 0000 10XX XX11b |
| MZR | X | 0110 1101 10XX XX10 | ZRBF0~9 | ← (Rx)16 bits | Rx : 0 0000 11XX XX00b~ 0 0000 11XX XX11b |
| RZR | X | 0110 1101 10XX XX11 | (Rx)16 bits (Rx)15~10=0 | ← ZRBF0~9 | 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 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|---|----------------------------|-------------|--|---|
| 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 00XX XXXX | X5~0 | : Set Ry Page (0~6, 8~1B,20~3Fh) for Rx 0~1BF,200~3FFh | Inserted by Compiler |
| ERY | X | 1101 0010 00XX XXXX | X5~0 | : Enable Ry Page (0~6,8~ 1B, 20~3Fh) for Rx 0~1BF,200~3FFh Set & Lock | <ol style="list-style-type: none"> Before execute "CLPG (X1=1)", Page still can be changed by executing "ERY" again. Between jump location, can't have any "ERY/ERX/ELZ/CLPG" instructions. Interrupt will be masked by hardware till all page set be cleared by "CLPG". |
| SRX | X | 1101 0100 0000 0XXX | X2~0 | :Set Rx Page (1~7h) | Inserted by Compiler |
| ERX | X | 1101 0110 0000 0XXX | X2~0 | :Enable Rx Page (1~7h) Set & Lock | <ol style="list-style-type: none"> Before execute "CLPG (X0=1)", Page still can be changed by executing "ERX" again. Between jump location, can't have any "ERY/ERX/ELZ/CLPG" instructions. Interrupt will be masked by hardware till all page set be cleared by "CLPG". |
| SLZ | X | 1101 0100 0100 00XX | X1~0 | :Set Lz Page(1~2h) | Inserted by Compiler |
| ELZ | X | 1101 0110 0100 00XX | X1~0 | :Enable Lz Page(1~2h) Set & Lock | <ol style="list-style-type: none"> Before execute "CLPG (X2=1)", Page still can be changed by executing "ELZ" again. Between jump location, can't have any "ERY/ERX/ELZ/CLPG" instructions. |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|----------|----------------------------|---|--|---|
| | | | | | 3. Interrupt will be masked by hardware till all page set be cleared by "CLPG". |
| SPBK | X | 1101 0110 1000 00XX | X1~0 | :Set POM BANK X=00~03H | 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= 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 merges to TM2 Add "SETDAT SD" to the next address If TM1 merges to 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 SD12~0 (Rx)15~12= 1010/1000 0111~0100 0011~0000 (Rx)11~6 | ← (Rx)16 bits :Rx Address (SD1,0=00b) for 16 bits data mode :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 SD:0000~03FFH |
| 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 merges to TM2 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|---|---------------------|--|--|---|
| T2RH | | 1101 0111 0100 0110 | Timer2 SD12~0 (Rx)15~12= 1010/1000 0111~0100 0011~0000 (Rx)11~6 (Rx)5~0 | ← (Rx)16 bits :Rx Address (SD1,0=00b) for 16 bits 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:0000~03FFH If TM1 is merged to 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 to TM3 |
| T3RH | | 1101 0111 0100 1010 | Timer3 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 16 bits 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 (Timer3) Value | Add "SETDAT SD" to the next address SD:0000~03FFH If TM1 is merged to TM3 |
| SPKXH | D | 1101 0111 0100 110D | KO16~1 D=1 D=0 SD15~0 | ← SETDAT SD (16 bits) : KEY_S released by scanning cycle : KEY_S released by normal key scanning : Enable KO16~1=1 | Add "SETDAT SD" to the next address |
| SPKRH | D | 1101 0111 0100 111D | KO16~1 | ← (Rx)16 bits | Add "SETDAT SD" to the next address |

| Instruction | | Machine Code | Function | | Flag/Remark |
|---------------|-----|---------------------|--|---|---|
| | | | D=1 D=0 (Rx)15~0 | : KEY_S released by scanning cycle : KEY_S released by normal key scanning : Enable KO16~1=1 | SD:0000~03FFH |
| 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 | IDBF12~0 | ← SETDAT SD (16 bits) SD : 0000~1FFFH | Add “SETDAT SD” to the next address |
| SZRX | | 1101 0111 0101 1101 | ZRBF9~0 | ← SETDAT SD (16 bits) SD : 0000~03FFH | Add “SETDAT SD” to the next address |
| CPHLH | | 1101 0111 0101 1110 | (PC+2) | ← force “NOP” if SETDAT SD (16 bits) = IDBF12~0 | 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) = ZRBF9~0 | 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) ← 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 (16 bits) ← PC+X+2 | Add “SETDAT SD” to the next X(<16) Addresses range |
| 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 | ← T(@HL)16 bits ← HL+2 | ‘ ’: B3=0 ‘#’: B3=1 |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-----------------|-------|---------------------|--|---|----------------------------|
| | | | TD15~12= 1010~1000 0111~0100 0011~0000 | :Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ | |
| | | | TD11~6 | : Set Counter 11~6 (Timer1) Value | If TM1 is merged to TM2 |
| | | | TD5~0 | : Set Counter 5~0 (Timer2) Value | |
| T3TH T3TH# | @HL | 1101 0111 1010 B000 | Timer3(3->1) ‘#’ : HL | ← T(@HL)16bits ← HL+2 | ‘ ’ : B3=0 ‘#’ : B3=1 |
| | | | TD15~12= 1010~1000 0111~0100 0011~0000 | :Ctm=INT/CX :Ctm=PH13/11/7/5 :Ctm=FREQ/PH15/3/9/ | |
| | | | TD11~6 | : Set Counter 11~6 (Timer1) Value | If TM1 is merged to TM3 |
| | | | TD5~0 | : Set Counter 5~0 (Timer3) Value | |
| SPKTH SPKTH# | D,@HL | 1101 0111 1011 B00X | KO16~1 ‘#’ : HL D=1 D=0 | ← T(@HL)16bits ← HL+2 : KEY_S released by scanning cycle : KEY_S released by normal key scanning | ‘ ’ : B3=0 ‘#’ : B3=1 |
| | | | TD15~0 | : Enable KO16~1=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)0~1 AC1~3/(Rx)2~3 =0 | ← ZRBF8~9 | |
| RVV | Rx | 1101 1011 0XXX XXXX | AC,(Rx)0 AC1~3/(Rx)1~3 =0 | ← IDBF12 | |
| 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 | : Ctm = PH13 : Ctm = PH11 : Ctm = PH7 : Ctm = PH5 : Ctm = FREQ | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|-----|---------------------|--|--|---|
| | | | X8,7,6=010 X8,7,6=001 X8,7,6=000 X5~0 | : 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 |
| 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) | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|-----|---------------------|--|--|------------------------------------|
| 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 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 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 | |
| SCNT | X | 1111 0100 010X XX00 | X4=0 X4=1 | Counter clock source for not CX control Mode : CX : FREQ | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|---|---------------------|--|---|-------------|
| | | | X3,2=00 X3,2=01 X3,2=10 X3,2=11 | 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 X8,7,6=011 | BCLK/8 BCLK/4 BCLK/2 BCLK FREQB | |

| Instruction | | Machine Code | Function | | Flag/Remark |
|-------------|---|---------------------|---|---|-------------|
| | | | 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 | 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 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) : 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