

## Preliminary

### GENERAL DESCRIPTION

EM73MA89B is an advanced single chip CMOS 4-bit multi-time-programmable (MTP) micro-controller. It contains 16K-byte ROM, 1012-nibble RAM, 4-bit ALU, 13-level subroutine nesting, 22-stage time base, two 12-bit timer/counters for the kernel function, and one high speed counter. EM73MA89B also equipped with 6 interrupt sources, 3~7 I/O ports (including 1 input port and 2~7 bidirection ports), LCD display (64x16 or 64x32), built-in watch-dog-timer and speech synthesizer.

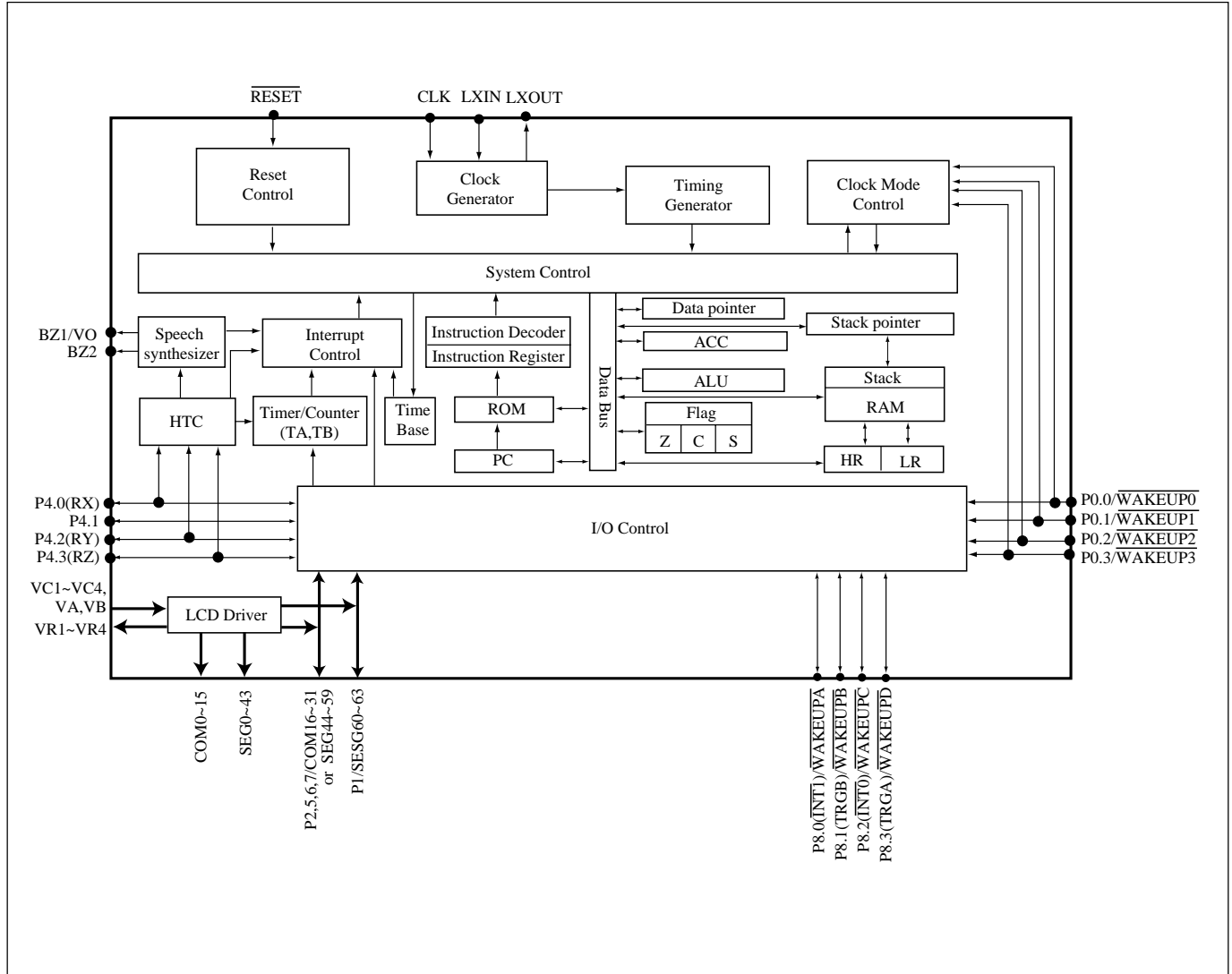
Its low power consumption and high speed feature are further strengthened with DUAL, SLOW, IDLE and STOP operation mode for optimized power saving.

### FEATURES

- Operation voltage : 2.2V to 3.6V.
- Clock source : Dual clock system. Low-frequency oscillator is 32KHz. Crystal oscillator or RC oscillator by mask option and high-frequency oscillator is a built-in internal oscillator.
- Instruction set : 107 powerful instructions.
- Instruction cycle time : 0.85 $\mu$ s for 9.2M or 1.7 $\mu$ s for 4.6M Hz selected by mask option (high speed clock). 122 $\mu$ s for 32768 Hz (low speed clock, frequency double).
- ROM capacity : 16K x 8 bits.
- RAM capacity : 1012 x 4 bits.
- Input port : 1 port (P0.0-P0.3), IDLE/STOP releasing function is available by mask option. (each input pin has a pull-up and pull-down resistor available by mask option).
- Bidirection port : 2~7 ports (P1, P2, P4, P5, P6, P7, P8). IDLE/STOP release function for P8(0..3) is available by mask option. P1, P2, P5, P6, P7 are shared with LCD pins.
- Built-in watch-dog-timer counter : It is available by mask option.
- 12-bit timer/counter : Two 12-bit timer/counters are programmable for timer, event counter and pulse width measurement mode.
- Built-in time base counter : 22 stages.
- Subroutine nesting : Up to 13 levels.
- Interrupt : External interrupt . . . . . 2 input interrupt sources.  
Internal interrupt . . . . . 2 timer overflow interrupts, 1 time base interrupt.  
1 speech/HTC interrupt.
- High speed counter : The high speed counter includes one 8-bit high speed counter and a resistor to frequency oscillator. It has resistor to frequency oscillation mode, melody mode and auto load timer mode.
- LCD driver : 64x32 or 64x16 dots, 1/32 or 1/16 duty, 1/5 bias by mask option.
- Speech synthesizer : 992K speech data ROM (use as 992K nibbles data ROM).
- PWM or current D/A : Output selection by mask option.
- Power saving function : SLOW, IDLE, STOP operation modes.
- Package type : Chip form 128 pins.

**Preliminary**

**FUNCTION BLOCK DIAGRAM**



**Preliminary**

**PIN DESCRIPTIONS**

Symbol	Pin-type	Function
VDD, VDD2		In normal mode : Power supply (+) In programming MTP mode: Power supply (+) for programming MTP
VSS		In normal mode : Power supply (-) In programming MTP mode: Power supply (-) for programming MTP
VNN		In normal mode : No connection (floating) In programming MTP mode: Power supply for programming MTP
VPP		In normal mode : No connection (floating) In programming MTP mode: High voltage (12V) power source for programming MTP
RESET	RESET-A	System reset input signal, low active mask option :           none pull-up
CLK	OSC-G	Capacitor connecting pin for internal high frequency oscillator.
LXIN	OSC-B/OSC-H	Crystal/Resistor connecting pin for low speed clock source.
LXOUT	OSC-B	Crystal connecting pin for low speed clock source.
P0(0..3)/WAKEUP0..3	INPUT-B	4-bit input port with IDLE/STOP releasing function mask option :           wakeup enable, pull-up wakeup enable, none wakeup disable, pull-up wakeup disable, pull-down wakeup disable, none  In programming MTP mode: P0.0/ACLK : address counter clock for programming MTP P0.1/PGMB : program data to MTP cells for programming MTP P0.2/OEB : data output enable for programming MTP P0.3/DCLK : data in/out clock signal for programming MTP
P4.0(RX),P4.2(RY), P4.3(RZ)	I/O-X1	3-bit bidirection I/O pins or RF oscillation input pins. mask option :           open-drain (apply to RF oscillation) high current push-pull normal current push-pull low current push-pull  In programming MTP mode: P4.0/D0 : data bus for programming MTP P4.2/D2 : data bus for programming MTP P4.3/D3 : data bus for programming MTP
P4.1	I/O-Q1	1-bit bidirection I/O pin. mask option :           open-drain high current push-pull normal current push-pull low current push-pull  In programming MTP mode: P4.1/D1 : data bus for programming MTP

## Preliminary

Symbol	Pin-type	Function
P8.0( $\overline{\text{INT1}}$ )/WAKEUPA P8.2( $\overline{\text{INT0}}$ )/WAKEUPC	I/O-X1	2-bit bidirection I/O port with external interrupt sources input and IDLE /STOP releasing function mask option :       wakeup enable, normal current push-pull wakeup enable, low current push-pull wakeup disable, high current push-pull wakeup disable, normal current push-pull wakeup disable, low current push-pull wakeup disable, open drain  In programming MTP mode: P8.0/D4 : data bus for programming MTP P8.2/D6 : data bus for programming MTP
P8.1(TRGB)/WAKEUPB P8.3(TRGA)/WAKEUPD	I/O-X1	2-bit bidirection I/O port with time/counter A,B external input and IDLE /STOP releasing function mask option :       wakeup enable, normal current push-pull wakeup enable, low current push-pull wakeup disable, high current push-pull wakeup disable, normal current push-pull wakeup disable, low current push-pull wakeup disable, open drain  In programming MTP mode: P8.1/D5 : data bus for programming MTP P8.3/D7 : data bus for programming MTP
VCA, VCB, V1~V6		LCD bias voltage pins
BZ1/VO		PWM or current D/A output pin for speech synthesizer by mask option
BZ2		PWM output pin for speech synthesizer
TEST		Tie Vss as package type, no connecting as COB type.

**\*16 COMMONS :**

COM0~COM15		LCD common output pins
SEG0~SEG59		LCD segment output pins
P1(0..3)/SEG63..60	I/O-P	4-bit bidirection I/O pins with LCD segment pins mask option :       LCD segment pin push-pull open-drain
P2(0..3),P5(0..3), P6(0..3),P7(0..3)	I/O-P	16-bit bidirection I/O pins mask option :       push-pull open-drain

**\*32 COMMONS :**

COM0~COM31		LCD common output pins
SEG0~SEG43		LCD segment output pins
P1(0..3)/SEG63..60, P2(0..3)/SEG59..56, P5(0..3)/SEG55..52, P6(0..3)/SEG51..48, P7(0..3)/SEG47..44	I/O-P	16-bit bidirection I/O pins with LCD segment pins mask option :       LCD segment pin push-pull open-drain

**Preliminary**

**FUNCTION DESCRIPTIONS**

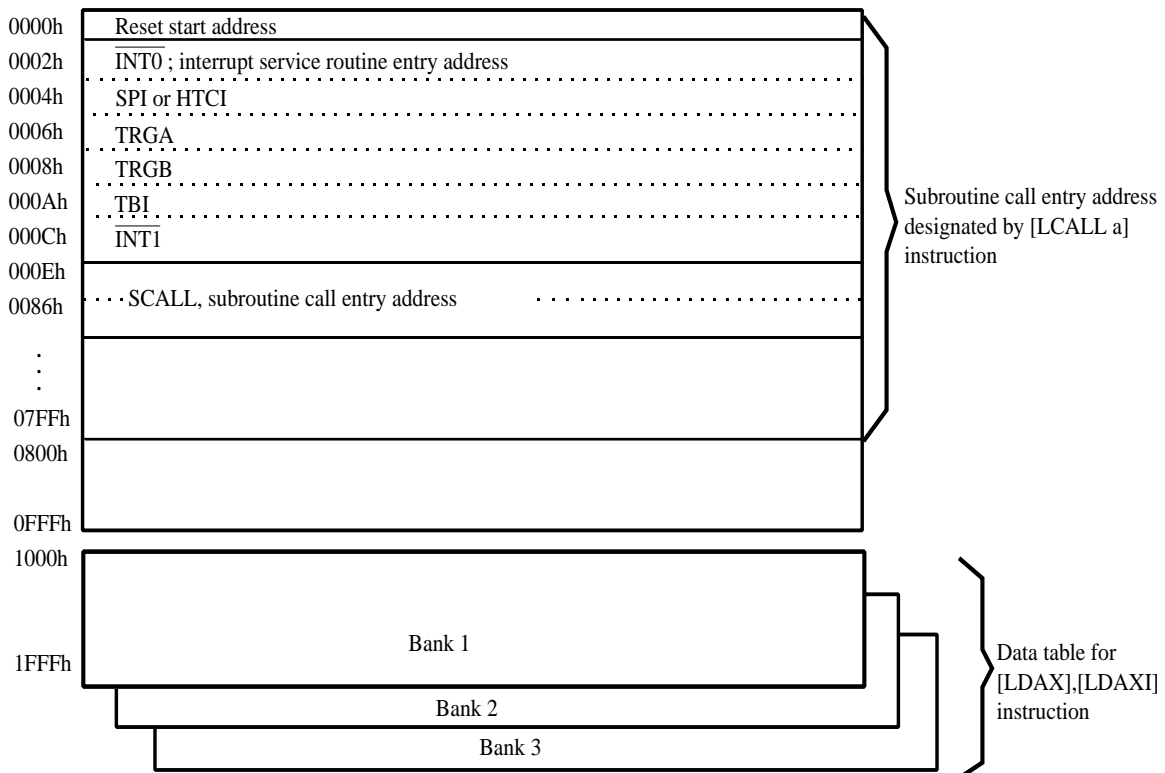
**PROGRAM ROM ( 16K X 8 bits )**

16 K x 8 bits program ROM contains user's program and some fixed data.

The basic structure of the program ROM may be categorized into 5 partitions.

1. Address 0000h : Reset start address.
2. Address 0002h - 000Ch : 6 kinds of interrupt service routine entry addresses.
3. Address 000Eh - 0086h : SCALL subroutine entry address, only available at 000Eh, 0016h, 001Eh, 0026h, 002Eh, 0036h, 003Eh, 0046h, 004Eh, 0056h, 005Eh, 0066h, 006Eh, 0076h, 007Eh, 0086h.
4. Address 0000h - 07FFh : LCALL subroutine entry address.
5. Address 0000h - 1FFFh : Except used as above function, the other region can be used as user's program and data region.

address      Bank 0 :



## Preliminary

User's program and fixed data are stored in the program ROM. User's program is executed using the PC value to fetch an instruction code.

The 16Kx8 bits program ROM can be divided into 4 banks. There are 4Kx8 bits per bank.

The program ROM bank is selected by P3(1..0). The program counter is a 13-bit binary counter. The PC and P3 are initialized to "0" during reset.

When P3(1..0)=00B or 11B, the bank0 and bank1 of program ROM will be selected. P3(1..0)=01B, the bank0 and bank2 will be selected. P3(1..0)=10B, the bank0 and bank3 will be selected.

Address	P3=xx00B P3=xx11B	P3=xx01B	P3=xx10B
0000h	Bank0	Bank0	Bank0
:			
:			
0FFFh	Bank1	Bank2	Bank3
1000h			
:			
:			
1FFFh			

**PROGRAM EXAMPLE:**

```

BANK 0
START:  :
        :
        :
        LDIA #00H           ; set program ROM to bank1
        OUTA P3
        B    XA1
        :
XA:     :
        :
        LDIA #01H           ; set program ROM to bank2
        OUTA P3
        B    XB1
        :
XB:     :
        :
        LDIA #02H           ; set program ROM to bank3
        OUTA P3
        B    XC1
        :
XC:     :
        :
        B    XD
XD:     :
        :
        :
;-----
BANK 1
XA1:   :
        :
        B    XA
        :
XA2:   :
    
```

**Preliminary**

	B	XA2			
	:				
		BANK 2			
XB1 :	:				
	:				
	B	XB			
	:				
XB2 :	:				
	B	XB2			
	:				
		BANK 3			
XC1 :	:				
	:				
	B	XC			
	:				
XC2 :	:				
	B	XC2			

Fixed data can be read out by table-look-up instruction. Table-look-up instruction is requires the Data point (DP) to indicate the ROM address in obtaining the ROM code data (Except bank 0) :

**LDAX**            **Acc ← ROM[DP]<sub>L</sub>**  
**LDAXI**          **Acc ← ROM[DP]<sub>H</sub>, DP+1**

DP is a 12-bit data register that stores the program ROM address as pointer for the ROM code data. User has to initially load ROM address into DP with instructions "STADPL", and "STADPM, STADPH", then to obtain the lower nibble of ROM code data by instruction "LDAX" and higher nibble by instruction "LDAXI".

PROGRAM EXAMPLE: Read out the ROM code of address 1777h by table-look-up instruction.

```
LDIA      #07h;
STADPL           ; [DP]L ← 07h
STADPM          ; [DP]M ← 07h
STADPH          ; [DP]H ← 07h, Load DP=777h
:
LDL          #00h;
LDH          #03h;
LDAX         ; ACC ← 6h
STAMI        ; RAM[30] ← 6h
LDAXI        ; ACC ← 5h
STAM         ; RAM[31] ← 5h
:
ORG          1777h
DATA         56h;
```

**DATA RAM ( 1012-nibble )**

A total 1012 - nibble data RAM is available from address 000 to 3FFh  
 Data RAM includes the zero page region, stacks and data areas.



*Preliminary*

Bank 0	Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
P9=xx00B	000-00Fh	ZERO PAGE																
	010-01Fh																	
	020-02Fh																	
	030-03Fh																	
	040-04Fh																	
	050-05Fh																	
	060-06Fh																	
	070-07Fh																	
	080-08Fh																	
	090-09Fh																	
	0A0-0AFh																	
	0B0-0BFh																	
	0C0-0CFh	Level 0				Level 1				Level 2				Level 3				
	0D0-0DFh	Level 4				Level 5				Level 6				Level 7				
	0E0-0EFh	Level 8				Level 8				Level 10				Level 11				
	0F0-0FFh	Level 12																
Bank 1																		
P9=xx01B	100-10Fh																	
	110-11Fh																	
	:																	
	:																	
	1E0-1EFh																	
	1F0-1FFh																	
Bank 2																		
P9=xx10B	200-20Fh																	
	210-21Fh																	
	220-22Fh																	
	230-23Fh																	
	240-24Fh																	
	250-25Fh																	
	260-26Fh																	
	270-27Fh																	
	280-28Fh																	
	290-29Fh																	
	2A0-2AFh																	
	2B0-2BFh																	
	2C0-2CFh																	
	2D0-2DFh																	
	2E0-2EFh																	
	2F0-2FFh																	
Bank 3																		
P9=xx11B	300-30Fh																	
	310-31Fh																	
	320-32Fh																	
	330-33Fh																	
	340-34Fh																	
	350-35Fh																	
	360-36Fh																	
	370-37Fh																	
	380-38Fh																	
	390-39Fh																	
	3A0-3AFh																	
	3B0-3BFh																	
	3C0-3CFh																	
	3D0-3DFh																	
	3E0-3EFh																	
	3F0-3FFh																	
		SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7	SEG8	SEG9	SEGA	SEGB	SEGC	SEG D	SEGE	SEGF	
		SEG10	SEG11	SEG12	SEG13	SEG14	SEG15	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23	SEG24	SEG25	
		SEG26	SEG27	SEG28	SEG29	SEG30	SEG31	SEG32	SEG33	SEG34	SEG35	SEG36	SEG37	SEG38	SEG39	SEG40	SEG41	
		SEG42	SEG43	SEG44	SEG45	SEG46	SEG47	SEG48	SEG49	SEG50	SEG51	SEG52	SEG53	SEG54	SEG55	SEG56	SEG57	
		SEG58	SEG59	SEG60	SEG61	SEG62	SEG63	SEG64	SEG65	SEG66	SEG67	SEG68	SEG69	SEG70	SEG71	SEG72	SEG73	

\* This specification are subject to be changed without notice.



## *Preliminary*

### ZERO- PAGE:

From 000h to 00Fh is the zero-page location. It is used as the zero-page address mode pointer for the instruction of "STD #k,y; ADD #k,y; CLR y,b; CMP k,y".

PROGRAM EXAMPLE: To write immediate data "07h" to RAM [03] and to clear bit 2 of RAM [0Eh].  
 STD #07h, 03h ; RAM[03] ← 07h  
 CLR 0Eh,2 ; RAM[0Eh]<sub>2</sub> ← 0

### STACK:

There are 13 - level (maximum) stack levels that user can use for subroutine (including interrupt and CALL). User can assign any level be the starting stack by providing the level number to stack pointer (SP). When an instruction (CALL or interrupt) is invoked, before enter the subroutine, the previous PC address is saved into the stack until returned from those subroutines, the PC value is restored by the data saved in stack.

### DATA AREA:

Except the area used by user's application, the whole RAM can be used as data area for storing and loading general data.

### ADDRESSING MODE

The 1012 nibble data memory consists of four banks (bank 0 ~ bank 3). There are 244x4 bits (address 000h~0F3h) in bank 0 and 768x4 bits (address 100h ~ 3FFh) in bank 1 ~ bank 3.

The bank is selected by P9.

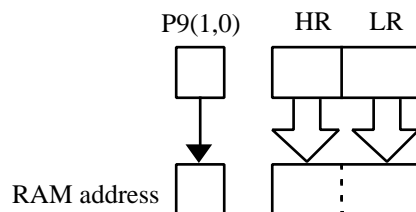
P9 Initial value : \* \* 0 0

*	*	RBK
RBK	RAM bank	
0 0	Bank0	
0 1	Bank1	
1 0	Bank2	
1 1	Bank3	

The Data Memory consists of three Address mode, namely -

#### (1) Indirect addressing mode:

The address in the bank is specified by the HL registers.



## Preliminary

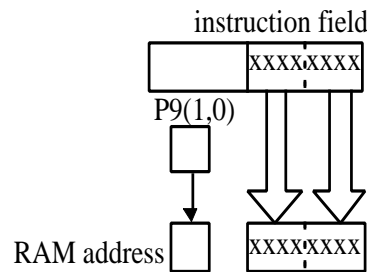
PROGRAM EXAMPLE: Load the data of RAM address "143h" to RAM address "032h".

```

OUT  #0001B,P9      ; RAM bank1
LDL  #3h            ; LR← 3
LDH  #4h            ; HR← 4
LDAM                      ; Acc← RAM[134h]
OUT  #0000B,P9     ; RAM bank0
LDL  #2h            ; LR← 2
LDH  #3h            ; HR← 3
STAM                      ; RAM[023h]← Acc
    
```

(2) Direct addressing mode:

The address in the bank is directly specified by 8 bits code of the second byte in the instruction field.



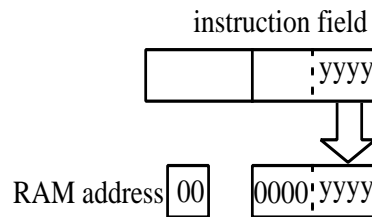
PROGRAM EXAMPLE: Load the data of RAM address "143h" to RAM address "023h".

```

OUT  #0001B,P9
LDA  43h            ; Acc← RAM[143h]
OUT  #0000B,P9
STA  23h            ; RAM[023h]← Acc
    
```

(3) Zero-page addressing mode:

The zero-page is in the bank 0 (address 000h~00Fh). The address is the lower 4 bits code of the second byte in the instruction field.



PROGRAM EXAMPLE: Write immediate "0Fh" to RAM address "005h".

```

STD  #0Fh,05h      ; RAM[05h]← 0Fh
    
```

## PROGRAM COUNTER (16K ROM) Preliminary

Program counter ( PC ) is composed by a 13-bit counter, which indicates the next executed address for the instruction of program ROM instruction.

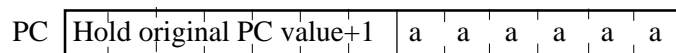
For BRANCH and CALL instructions, PC is changed by instruction indicating. PC only can indicate the address from 0000h-1FFFh. The bank number is decided by P3.

### (1) Branch instruction:

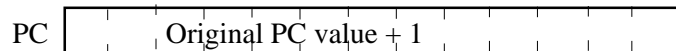
#### SBR a

Object code: 00aa aaaa

Condition: SF=1; PC ← PC<sub>12-6,a</sub> ( branch condition satisfied )



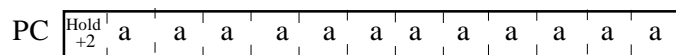
SF=0; PC← PC +1( branch condition not satisfied )



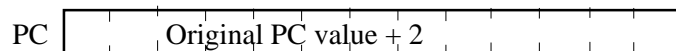
#### LBR a

Object code: 1100 aaaa aaaa aaaa

Condition: SF=1; PC ← PC<sub>12,a</sub> ( branch condition satisfied )



SF=0; PC← PC +2( branch condition not satisfied )

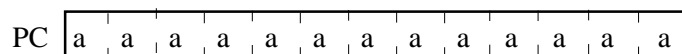


#### SLBR a

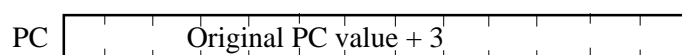
Object code: 0101 0101 1100 aaaa aaaa aaaa (a:1000h~1FFFh)

0101 0111 1100 aaaa aaaa aaaa (a:0000h~0FFFh)

Condition: SF=1; PC ← a ( branch condition satisfied )



SF=0 ; PC ← PC + 3 ( branch condition not satisfied )

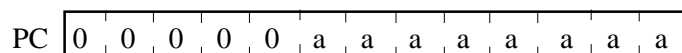


### (2) Subroutine instruction:

#### SCALL a

Object code: 1110 nnnn

Condition : PC ← a ; a=8n+6 ; n=1..Fh ; a=86h, n=0



#### LCALL a

Object code: 0100 0aaa aaaa aaaa

Condition: PC ← a

## Preliminary

PC 

0	0	a	a	a	a	a	a	a	a	a	a	a	a
---	---	---	---	---	---	---	---	---	---	---	---	---	---

**RET**

Object code: 0100 1111

Condition:  $PC \leftarrow STACK[SP]; SP + 1$

PC 

The return address stored in stack													
------------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--

**RTI**

Object code: 0100 1101

Condition :  $FLAG. PC \leftarrow STACK[SP]; EI \leftarrow 1; SP + 1$

PC 

The return address stored in stack													
------------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--

**(3) Interrupt acceptance operation:**

When an interrupt is accepted, the original PC is pushed into stack and interrupt vector will be loaded into PC. The interrupt vectors are as follows :

$\overline{INT0}$  (External interrupt from P8.2)

PC 

0	0	0	0	0	0	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---

**SPI** (speech end interrupt)

PC 

0	0	0	0	0	0	0	0	0	0	0	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---

**TRGA** (Timer A overflow interrupt)

PC 

0	0	0	0	0	0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

**TRGB** (Time B overflow interrupt)

PC 

0	0	0	0	0	0	0	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

**TBI** (Time base interrupt)

PC 

0	0	0	0	0	0	0	0	0	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---

$\overline{INT1}$  (External interrupt from P8.0)

PC 

0	0	0	0	0	0	0	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

**(4) Reset operation:**

PC 

0	0	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---

---

**Preliminary****(5) Other operations:**

- For 1-byte instruction execution: PC + 1
- For 2-byte instruction execution: PC + 2
- For 3-byte instruction execution: PC + 3

**ACCUMULATOR**

Accumulator(ACC) is a 4-bit data register for temporary data storage. For the arithmetic, logic and comparative operation..., ACC plays a role which holds the source data and result.

**FLAGS**

There are three kinds of flag, CF ( Carry flag ), ZF ( Zero flag ) and SF ( Status flag ), these three 1-bit flags are included by the arithmetic, logic and comparative .... operation.

All flags will be put into stack when an interrupt subroutine is served, and the flags will be restored after RTI instruction is executed.

**(1) Carry Flag ( CF )**

The carry flag is affected by the following operations:

- a. Addition : CF as a carry out indicator, under addition operation, when a carry-out occurs, the CF is "1", likewise, if the operation has no carry-out, CF is "0".
- b. Subtraction : CF as a borrow-in indicator, under subtraction operation, when a borrow occurs, the CF is "0", likewise, if there is no borrow-in, the CF is "1".
- c. Comparison : CF as a borrow-in indicator for Comparison operation as in the subtraction operation.
- d. Rotation : CF shifts into the empty bit of accumulator for the rotation and holds the shift out data after rotation.
- e. CF test instruction : Under TFCFC instruction, the CF content is sent into SF then clear itself as "0". Under TTSCFC instruction, the CF content is sent into SF then set itself as "1".

**(2) Zero Flag ( ZF )**

ZF is affected by the result of ALU, if the ALU operation generates a "0" result, the ZF is "1", likewise, the ZF is "0".

**(3) Status Flag ( SF )**

The SF is affected by instruction operation and system status.

- a. SF is initiated to "1" for reset condition.
- b. Branch instruction is decided by SF, when SF=1, branch condition is satisfied, likewise, when SF = 0, branch condition is unsatisfied.

## Preliminary

### PROGRAM EXAMPLE:

Check following arithmetic operation for CF, ZF, SF

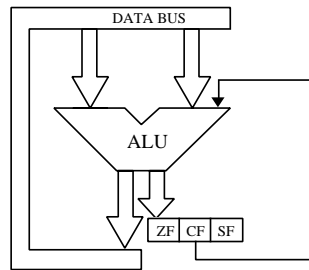
	CF	ZF	SF
LDIA #00h;	-	1	1
LDIA #03h;	-	0	1
ADDA #05h;	-	0	1
ADDA #0Dh;	-	0	0
ADDA #0Eh;	-	0	0

### ALU

The arithmetic operation of 4-bit data is performed in ALU unit. There are 2 flags that can be affected by the result of ALU operation, ZF and SF. The operation of ALU is affected by CF only.

### ALU STRUCTURE

ALU supported user arithmetic operation functions, including Addition, Subtraction and Rotation.



### ALU FUNCTION

#### (1) Addition:

ALU supports addition function with instructions ADDAM, ADCAM, ADDM #k, ADD #k,y .... . The addition operation affects CF and ZF. Under addition operation, if the result is "0", ZF will be "1", otherwise, ZF will be "0". When the addition operation has a carry-out, CF will be "1", otherwise, CF will be "0".

#### EXAMPLE:

Operation	Carry	Zero
3+4=7	0	0
7+F=6	1	0
0+0=0	0	1
8+8=0	1	1

#### (2) Subtraction:

ALU supports subtraction function with instructions SUBM #k, SUBA #k, SBCAM, DECM... . The subtraction operation affects CF and ZF. Under subtraction operation, if the result is negative, CF will be "0", and a borrow out, otherwise, if the result is positive, CF will be "1". For ZF, if the result of subtraction operation is "0", the ZF is "1", likewise, ZF is "1".

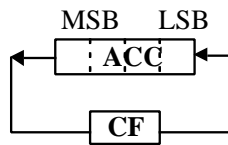
## Preliminary

EXAMPLE:

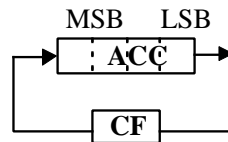
Operation	Carry	Zero
8-4=4	1	0
7-F= -8(1000)	0	0
9-9=0	1	1

(3) Rotation:

Two types of rotation operation are available, one is rotation left, the other is rotation right. RLCA instruction rotates Acc value counter-clockwise, shift the CF value into the LSB bit of Acc and hold the shift out data in CF.



RRCA instruction operation rotates Acc value clockwise, shift the CF value into the MSB bit of Acc and hold the shift out data in CF.

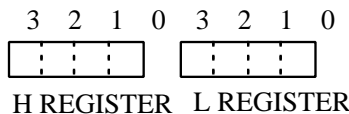


PROGRAM EXAMPLE: To rotate Acc clockwise (right) and shift a "1" into the MSB bit of Acc.  
TTCFS; CF ← 1  
RRCA; rotate Acc right and shift CF=1 into MSB.

### HL REGISTER

HL register are two 4-bit registers, they are used as a pair of pointer for the RAM memory address. They are used as also 2 independent temporary 4-bit data registers. For certain instructions, L register can be a pointer to indicate the pin number (Port4 only).

### HL REGISTER STRUCTURE



### HL REGISTER FUNCTION

(1) HL register is used as a temporary register for instructions : LDL #k, LDH #k, THA, THL, INCL, DECL, EXAL, EXAH.

PROGRAM EXAMPLE: Load immediate data "5h" into L register, "0Dh" into H register.  
LDL #05h;  
LDH #0Dh;

(2) HL register is used as a pointer for the address of RAM memory for instructions : LDAM, STAM, STAMI ..

PROGRAM EXAMPLE: Store immediate data "#0Ah" into RAM of address 35h.

**Preliminary**

```
LDL #5h;
LDH #3h;
STDMI #0Ah; RAM[35] ← Ah
```

- (3) L register is used as a pointer to indicate the bit of I/O port for instructions : SELP, CLPL, TFPL,  
(When LR = 0 indicate P4.0)

PROGRAM EXAMPLE: To set bit 0 of Port4 to "1"  
LDL #00h;  
SEPL ; P4.0 ← 1

### STACK POINTER (SP)

Stack pointer is a 4-bit register that stores the present stack level number.  
Before using stack, user must set the SP value first, CPU will not initiate the SP value after reset condition.  
When a new subroutine is received, the SP is decreased by one automatically, likewise, if returning from a subroutine, the SP is increased by one.  
The data transfer between ACC and SP is done with instructions "LDASP" and "STASP".

### DATA POINTER (DP)

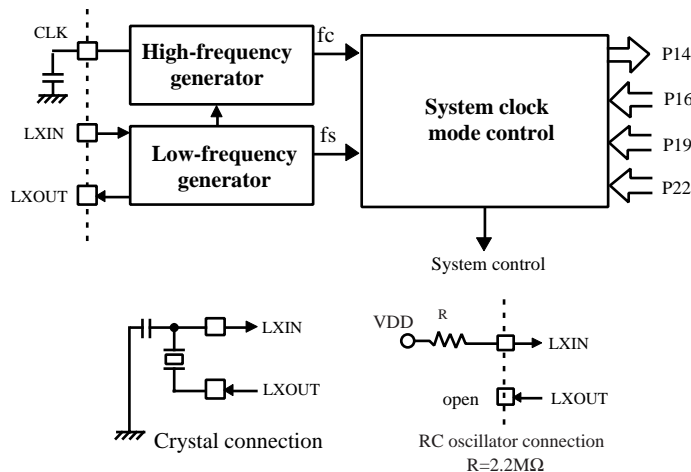
Data pointer is a 12-bit register that stores the ROM address can indicating the ROM code data specified by user (refer to data ROM).

### CLOCK AND TIMING GENERATOR

The clock generator is supported by a dual clock system. The high-frequency oscillator is internal oscillator. The low-frequency oscillator may be sourced from crystal, the working frequency is 32 KHz.

### CLOCK GENERATOR STRUCTURE

There are two clock generator for system clock control unit, P14 is the status register that hold the CPU status. P16, P19 and P22 are the command register for system clock mode control.

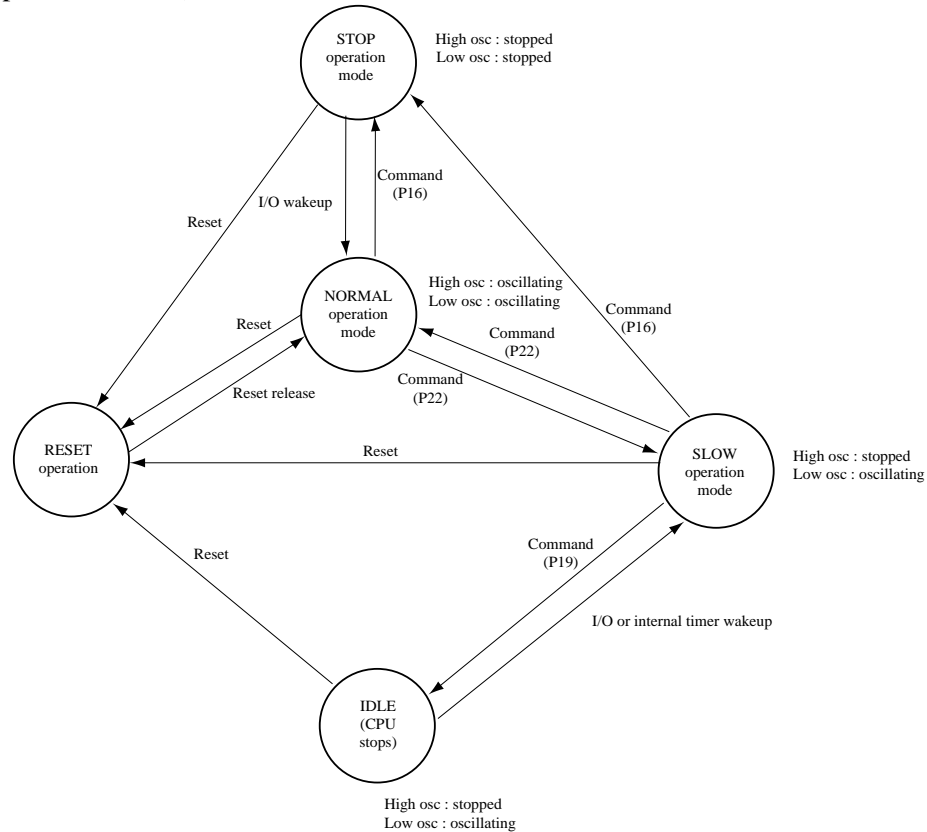




Preliminary

## SYSTEM CLOCK MODE CONTROL

The system clock mode controller can start or stop the high-frequency and low-frequency clock oscillator and switch between the basic clocks. EM73MA89B has four operation modes (DUAL, SLOW, IDLE and STOP operation modes).



Operation Mode	Oscillator	System Clock	Available function	One instruction cycle
NORMAL	High, Low frequency	High frequency clock	LCD, speech, HTC.	8 / fc
SLOW	Low frequency	Low frequency clock	LCD, HTC	4 / fs
IDLE	Low frequency	CPU stops	LCD	-
STOP	None	CPU stops	All disable	-

### DUAL OPERATION MODE

The 4-bit  $\mu$ c is in the DUAL operation mode when the CPU is reseted. This mode is dual clock system (high-frequency and low-frequency clocks oscillating). It can be changed to SLOW or STOP operation mode with the command register (P22 or P16).

LCD display, speech synthesizer and sound generator are available for the DUAL operation mode.

### SLOW OPERATION MODE

The SLOW operation mode is single clock system (low-frequency clock oscillating). It can be changed to the DUAL operation mode with the command register (P22), STOP operation mode with P16 and IDLE operation mode with P19.

LCD display is available for the SLOW operation mode. Speech synthesizer and sound generator are disabled in this mode.

**Preliminary**

P22      3      2      1      0      Initial value : \*\*\*0

3	2	1	0
*	*	*	SOM

SOM	Select operation mode
0	DUAL operation mode
1	SLOW operation mode

P14      3      2      1      0      Initial value : 0000

3	2	1	0
ACT	WKS	SINT	CPUS

CPUS	CPU status	WKS	Wakeup status
0	DUAL operation mode	0	Wakeup not by internal timer
1	SLOW operation mode	1	Wakeup by internal timer

Port14 is the status register for CPU. P14.0 (CPU status) is a read-only bit. P14.2 (wakeup status) will be set as "1" when CPU is waked by internal timer. P14.2 will be cleared as "0" when user out data to P14. P14.1 is the interrupt source selector (refer to interrupt). P14.3 is the speech acknowledge signal (refer to speech synthesizer control).

### IDLE OPERATION MODE

The IDLE operation mode suspends all CPU functions except the low-frequency clock oscillation and the LCD driver. It keeps the internal status with low power consumption without stopping the slow clock oscillator and LCD display.

LCD display is available for the IDLE operation mode. The high speed counter and speech synthesizer are disabled in this mode. The IDLE operation mode will be wakeup and return to the SLOW operation mode by the internal timing generator or I/O pins (P0(0..3)/WAKEUP 0..3 and P8(0..3)/WAKEUPA..D).

P19      3      2      1      0      Initial value : 0000

3	2	1	0
IDME		SIDR	

IDME	Enable IDLE mode	SIDR	Select IDLE releasing condition
0 1	Enable IDLE mode	0 0	P0(0..3), P8(0..3) pin input
* *	no function	0 1	P0(0..3), P8(0..3) pin input and 1 sec signal
		1 0	P0(0..3), P8(0..3) pin input and 0.5 sec signal
		1 1	P0(0..3), P8(0..3) pin input and 15.625 ms signal

### STOP OPERATION MODE

The STOP operation mode suspends system operation and holds the internal status immediately before the suspension with low power consumption. This mode will be released by reset or I/O pins (P0(0..3)/WAKEUP 0..3 or P8(0..3)/WAKEUP A..D).

LCD display, high speed counter and speech synthesizer are disabled in this mode.

**Preliminary**

P16    3    2    1    0                      Initial value : \*000

*	SWWT
---	------

SWWT	Enable STOP mode
1 0 1	Enable STOP mode
* * *	no function

**GENERAL PURPOSE REGISTER (P10)**

P10 is a 4-bit general purpose register which can be read, written and rested by all I/O instructions. (including : INA, INM, OUT, OUTA, OUTM, SEP, CLP, TTP, TFP)

**PROGRAM EXAMPLE:**

```

CHIP ROM16K
;-----RAM define area-----
DSEG
ORG 10H
HLBUF: RES 2 ; HL buffer for interrupt
P9BUF: RES 1 ; P9 (RAM bank) buffer for interrupt
:
;-----Interrupt subroutine-----
CSEG
ORG 004H
LBR SPI
:
SPI: OUTA P10 ; save Acc to general purpose register P10
INA P9
OUT #0000B,P9
STA P9BUF ; save RAM bank to P9BUF
EXHL HLBUF ; save HL to HLBUF
:
:
EXHL HLBUF ; restore HLBUF to HL
LDA P9BUF ; restore P9BUF to RAM bank
OUTA P9
INA P10 ; restore register P10 to Acc
RTI

```

} 10 instruction bytes

} 10 instruction bytes

***Preliminary***

**TIME BASE INTERRUPT (TBI)**

The time base can be used to generate a single fixed frequency interrupt. Eight types of frequencies can be selected with the "P25" setting.

P25    3    2    1    0  
    initial value : 0000

P25	NORMAL operation mode	SLOW operation mode
0 0 x x	Interrupt disable	Interrupt disable
0 1 0 0	Interrupt frequency LXIN / 2 <sup>3</sup> Hz	Reserved
0 1 0 1	Interrupt frequency LXIN / 2 <sup>15</sup> Hz	Interrupt frequency LXIN / 2 <sup>15</sup> Hz
0 1 1 0	Interrupt frequency LXIN / 2 <sup>5</sup> Hz	Reserved
0 1 1 1	Interrupt frequency LXIN / 2 <sup>14</sup> Hz	Interrupt frequency LXIN / 2 <sup>14</sup> Hz
1 1 0 0	Interrupt frequency LXIN / 2 <sup>1</sup> Hz	Reserved
1 1 0 1	Interrupt frequency LXIN / 2 <sup>6</sup> Hz	Interrupt frequency LXIN / 2 <sup>6</sup> Hz
1 1 1 0	Interrupt frequency LXIN / 2 <sup>8</sup> Hz	Interrupt frequency LXIN / 2 <sup>8</sup> Hz
1 1 1 1	Interrupt frequency LXIN / 2 <sup>10</sup> Hz	Interrupt frequency LXIN / 2 <sup>10</sup> Hz
1 0 x x	Reserved	Reserved

**TIMER / COUNTER (TIMER A, TIMER B)**

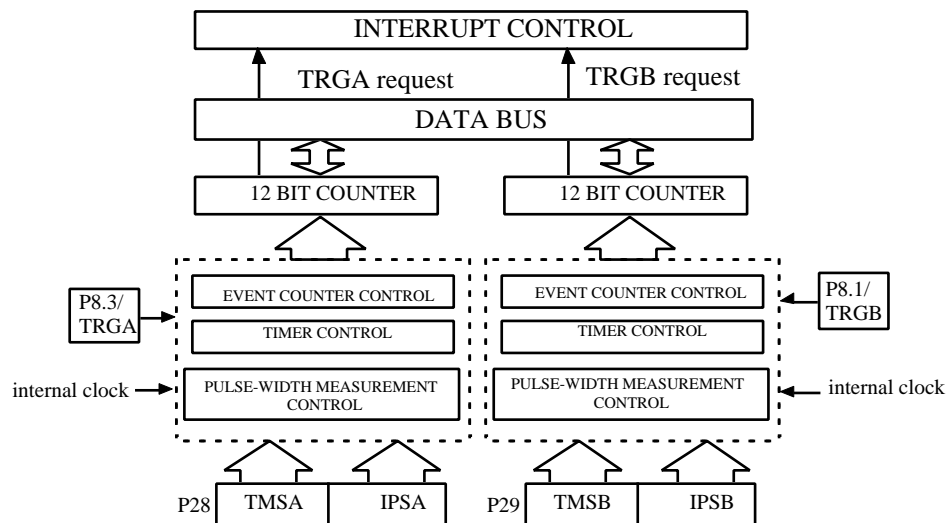
Timer/counters support three special functions:

1. Even counter
2. Timer.
3. Pulse-width measurement.

These three functions can be executed by 2 timer/counter independently.

With timerA, the counter data is saved in timer register TAH, TAM, TAL. User can set counter initial value and read the counter value by instruction "LDATAH(M,L)" and "STATAH(M,L)". With timer B register is TBH, TBM, TBL and the W/R instruction are "LDATBH (M,L)" and "STATBH (M,L)".

The basic structure of timer/counter is composed by two identical counter module, these two modules can be set initial timer or counter value to the timer registers, P28 and P29 are the command registers for timerA and timer B, user can choose different operation modes and internal clock rates by setting these two registers. When timer/counter overflows, it will generate a TRGA(B) interrupt request to interrupt control unit.



## Preliminary

### TIMER/COUNTER CONTROL

P8.1/TRGB, P8.3/TRGA are the external timer inputs for timerB and timerA, they are used in event counter and pulse-width measurement mode.

Timer/counter command port: P28 is the command port for timer/counterA and P29 is for the timer/counterB.

P28, P29    3   2   1   0                    Initial value : 0000

TMSA(B)	IPSA(B)
---------	---------

TMSA(B)	Mode selection
0 0	Stop
0 1	Event counter mode
1 0	Timer mode
1 1	Pulse width measurement mode

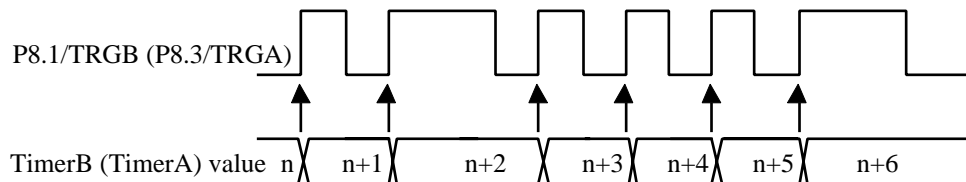
IPSA	Clock rate selection		IPSB	Clock rate selection	
	NORMAL mode	SLOW mode		NORMAL mode	SLOW mode
0 0	LXIN/2 <sup>3</sup> HZ	Reserved	0 0	Depend on high speed timer/counter	
0 1	LXIN/2 <sup>7</sup> HZ	LXIN/2 <sup>7</sup> HZ	0 1	LXIN/2 <sup>5</sup> HZ	LXIN/2 <sup>5</sup> HZ
1 0	LXIN/2 <sup>11</sup> HZ	LXIN/2 <sup>11</sup> HZ	1 0	LXIN/2 <sup>9</sup> HZ	LXIN/2 <sup>9</sup> HZ
1 1	LXIN/2 <sup>15</sup> HZ	LXIN/2 <sup>15</sup> HZ	1 1	LXIN/2 <sup>13</sup> HZ	LXIN/2 <sup>13</sup> HZ

### TIMER/COUNTER FUNCTION

Timer/counterA,B are programmable for timer, event counter and pulse width measurement mode. Each timer/counter can execute any of these functions independently.

#### EVENT COUNTER MODE

Under event counter mode, the timer/counter is increased by one at any rising edge of P8.1/TRGB for timerB (P8.3/TRGA for timer A). When timerB (timerA) counts overflow, it will provide an interrupt request TRGB (TRGA) to interrupt control unit.



PROGRAM EXAMPLE: Enable timerA with P28

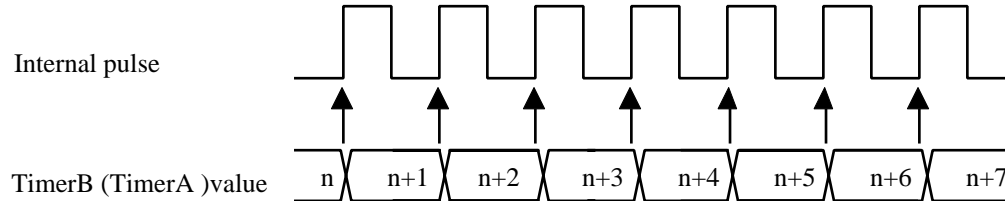
```
LDIA    #0100b;
OUTA    P28    ; Enable timerA with event counter mode
```

## Preliminary

### TIMER MODE

Under timer mode, the timer/counter is increased by one at any rising edge of internal pulse. User can choose up to 4 types of internal pulse rate by setting IPSB for timerB (IPSA for timerA).

When timer/counter counts overflow, an interrupt request will be sent to interrupt control unit.



**PROGRAM EXAMPLE:** To generate TRGA interrupt request after 60 ms with system clock LXIN=32KHz

```
LDIA    #0100B    ;
EXAE    ; enable mask 2
EICIL   110111b  ; interrupt latch ←0, enable EI
LDIA    #0Ah;
STATAL;
LDIA    #00h;
STATAM;
LDIA    #0Fh;
STATAH;
LDIA    #1000B;
OUTA    P28      ; enable timerA with internal pulse rate: LXIN/23 Hz
```

**NOTE:** The preset value of timer/counter register is calculated as following procedure.

Internal pulse rate:  $LXIN/2^3$  ;  $LXIN = 32KHz$

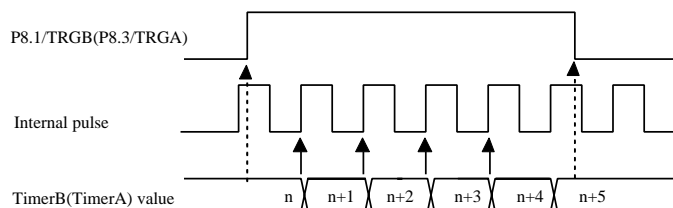
The time of timer counter count one =  $2^3 / LXIN = 8/32768=0.244ms$

The number of internal pulse to get timer overflow =  $60 ms/0.244ms = 245.901= 0F6h$

The preset value of timer/counter register =  $1000h - 0F6h = F0Ah$

### PULSE WIDTH MEASUREMENT MODE

Under the pulse width measurement mode, the counter is increased at the rising edge of internal pulse during external timer/counter input (P8.1/TRGB, P8.3/TRGA) in high level, interrupt request is generated as soon as timer/counter count overflow.



**PROGRAM EXAMPLE:** Enable timerA by pulse width measurement mode.

```
LDIA    #1100b  ;
OUTA    P28     ; Enable timerA with pulse width measurement mode.
```

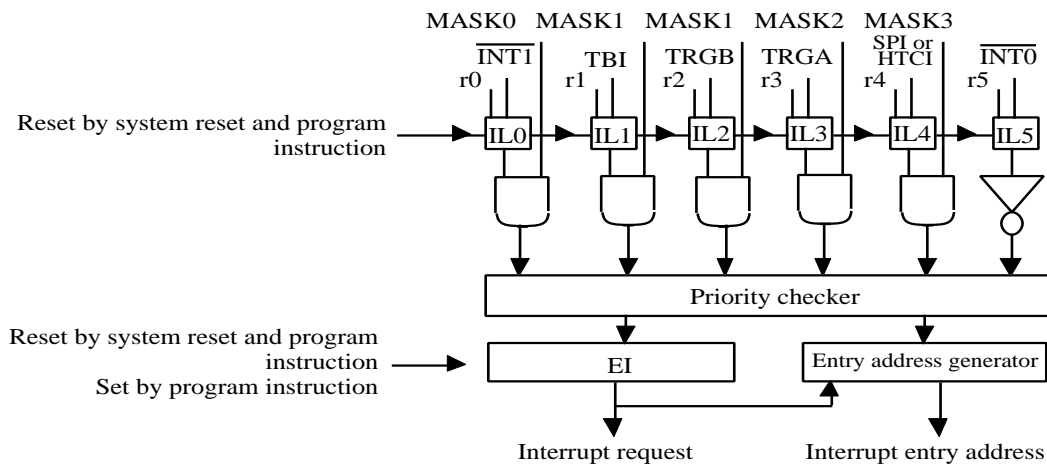
## Preliminary

### INTERRUPT FUNCTION

Six interrupt sources are available, 2 from external interrupt sources and 4 from internal interrupt sources. Multiple interrupts are admitted according to their priority.

Type	Interruptsource	Priority	Interrupt Latch	Interrupt Enablecondition	ProgramROM entry address
External	External interrupt ( $\overline{INT0}$ )	1	IL5	EI=1	002h
Internal	Speech or HTC interrupt (SPI or HTCI)	2	IL4	EI=1, MASK3=1	004h
Internal	TimerA overflow interrupt (TRGA)	3	IL3	EI=1, MASK2=1	006h
Internal	TimerB overflow interrupt (TRGB)	4	IL2	EI=1, MASK1=1	008h
Internal	Time base interrupt (TBI)	5	IL1		00Ah
External	External interrupt ( $\overline{INT1}$ )	6	IL0	EI=1, MASK0=1	00Ch

### INTERRUPT STRUCTURE



Interrupt controller:

**IL0-IL5** : Interrupt latch. Hold all interrupt requests from all interrupt sources. IL's can not be set by program, but can be reset by program or system reset, so IL can only decide which interrupt source can be accepted.

**MASK0-MASK3** : Except  $\overline{INT0}$ , MASK register may permit or inhibit all interrupt sources.

**EI** : Enable interrupt Flip-Flop may permit or inhibit all interrupt sources, when interrupt occurs, EI is auto cleared to "0", after RTI instruction is executed, EI is auto set to "1" again.

**Priority checker** : Check interrupt priority when multiple interrupts occur.

*Preliminary*

**INTERRUPT OPERATION**

The procedure of interrupt operation:

1. Push PC and all flags to stack.
2. Set interrupt entry address into PC.
3. Set SF = 1.
4. Clear EI to inhibit other interrupts occur.
5. Clear the IL with which interrupt source has already been accepted.
6. Excute interrupt subroutine from the interrupt entry address.
7. CPU accept RTI, restore PC and flags from stack. Set EI to accept other interrupt requests.

PROGRAM EXAMPLE: To enable interrupt of "INT0, TRGA"

```
LDIA    #0100B ;
EXAE           ; set mask register "0100b"
EICL    010111B ; enable interrupt F.F. and clear IL3 and IL5
```

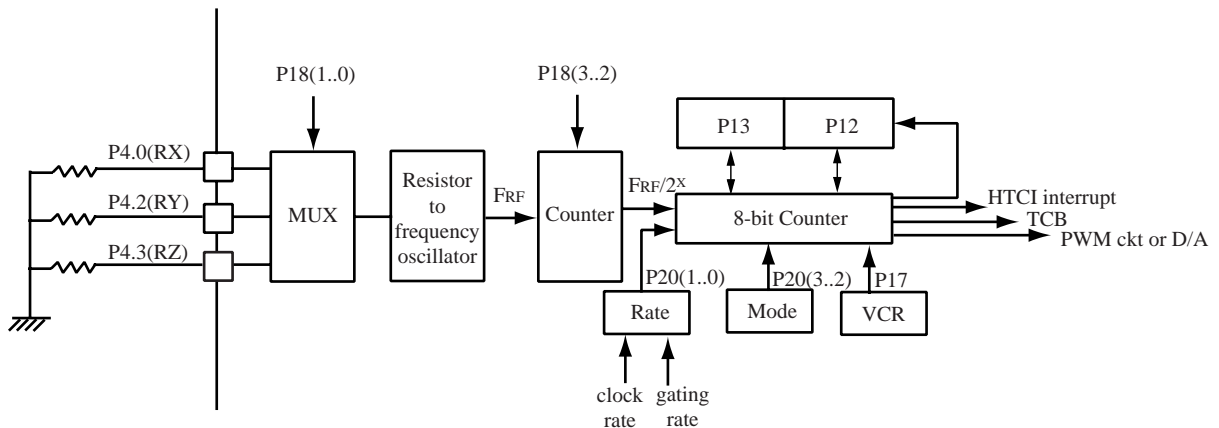
**INTERRUPT SOURCE SELECTION REGISTER**



P14.1 is the interrupt source selection register for speech ending interrupt (SPI) and high speed counter overflow interrupt (HTCI) selection. When SINT=0, the program address "0004H" is the interrupt entry address of SPI. When SINT=1, the program address "0004H" is the interrupt entry address of HTCI. P14.0 and P14.2 are the CPU flages (refer to system operation mode). P14.3 is the speech acknowledge signal (refer to speech synthesizer control).

**HIGH SPEED COUNTER**

EM73MA89B has one high speed counter for resistor to frequency oscillation mode, melody mode and auto timer mode. This function is available for the DUAL and SLOW operation mode. The resistor to frequency oscillation (RFO) circuit as show below :





## Preliminary

### CONTROL OF HIGH SPEED COUNTER

The high speed counter is controlled by the command registers (P20, P18) :

P20    3    2    1    0            Initial value : 0000

MODE	RATE
------	------

MODE	Selection of HTC mode
0 0	Disable HTC
0 1	Auto load timer mode
1 0	Melody mode
1 1	Resistor to frequency oscillation mode

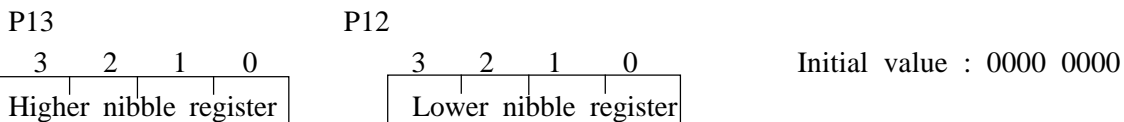
RATE ( Hz )	Internal pulse rate / Counter start request frequency	
	Resistor to frequency oscillation mode	Auto load timer mode / Melody mode internal pulse rate
0 0	$LXIN / 2^6$	$CLK / 2^4$
0 1	$LXIN / 2^{10}$	$CLK / 2^5$
1 0	$LXIN / 2^{14}$	$CLK / 2^6$
1 1	$LXIN / 2^{15}$	$CLK / 2^7$

P18    3    2    1    0            Initial value : 0000

RFIP	RFIN
------	------

RFIP	Input frequency of RFO	RFIN	Selection of RFO Pin
0 0	$F_{RF}$	0 0	Normal I/O
0 1	$F_{RF} / 4$	0 1	P4.0 (RX) for RFO
1 0	$F_{RF} / 16$	1 0	P4.2 (RY) for RFO
1 1	$F_{RF} / 64$	1 1	P4.3 (RZ) for RFO

P12 and P13 are the 8-bit binary counter registers of the HTC. P12 is lower nibble register and P13 is higher nibble register.



The HTC can be set initial value and send counter value to counter registers (P13 and P12), P20 and P18 are the command ports for HTC, user can choose different operation mode and different internal clockrate by setting the port. The timer/counter increase one at the rising edge of internal pulse. The HTC can generate an overflow interrupt (HTCI) when it overflows. The HTCI can't be generated when the HTC is in the melody mode or disabled.

**Preliminary**

### 8-BIT BINARY COUNTER

Write the preset value to the registers

The value of 8-bit binary counter can be presetted by P13 and P12. The value of registers can be loaded into the 8-bit binary counter when the counter starts counting or occurs overflow. When the 8-bit binary counter overflows, the HTCI interrupt will be generated. If you write values to the registers before the next overflow occurs, the preset value can be changed.

Read the count value from the registers

The count value of 8-bit binary counter can be read out from P13 and P12. The value is unstable when you read out the value during counting. Thus, you must disable the counter before reading out the value.

### 20-BIT COUNTER FUNCTION

The 8-bit binary counter is connected to TCB which is one 12-bit general counter and becomes to the 20-bit counter. The TCB increases one when the 8-bit binary counter overflows and generates an overflow interrupt (TRGB) when the TCB overflows. The TRGB cannot be generated when the HTC is in the melody or disable.

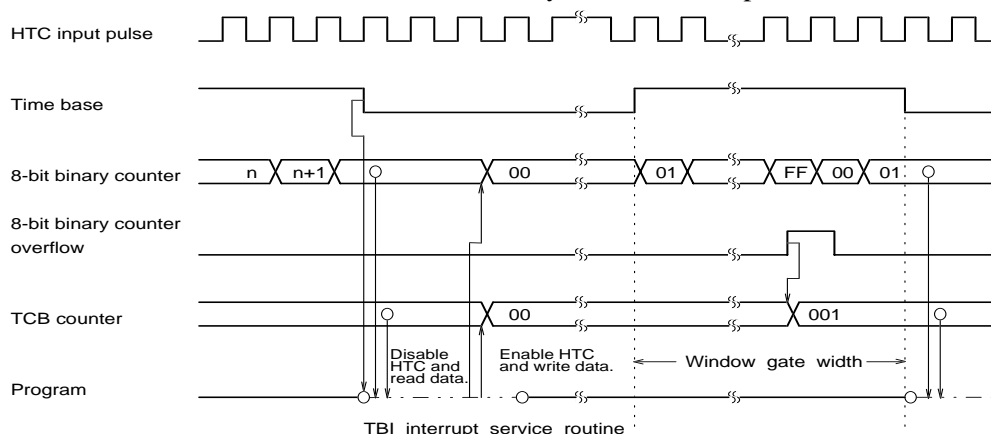
### FUNCTION OF HIGH SPEED COUNTER

The HTC has three modes which are RFO mode, melody mode and auto load timer mode.

The HTC is disabled when the CPU is reseted or in the STOP/IDLE operation mode. Users must enable it by yourself when the CPU is waked up.

#### Resistor to frequency oscillation mode

In this mode, the HTC is counted by the rising edges of input pulses from P4.1 (CS) and the value of window gate width is specified by P20. In this case, the window gate width interval is from the time base output fall to rise and the value of window gate width setting is the same as the time base interrupt frequency. The time base can be generated a fixed frequency interrupt when the time base interrupt (TBI) is enabled. The content of the HTC can be read and initialized by the TBI interrupt service routine.



ex. TBI interrupt frequency is  $LXIN/2^{15}$  Hz (P25=0101B). The pulse rate of RFO is  $LXIN/2^{15}$  Hz (P20=1111B). The window gate width of RFO is  $2^{14}/LXIN$  sec. (LXIN=32KHz)

---

**Preliminary**

## PROGRAM EXAMPLE

```

        DSEG
        ORG      00H

HLBUF:  RES     2

P9BUF:  RES     1

RFCON:  RES     1
        :
        CSEG
        ORG      00H
        LBR      MAIN          ; initial jump
        ORG      0AH
        LBR      TBI           ; timebase interrupt vector address
        :
                                ; timebase interrupt service routine
TBI:    OUTA     P10
        INA     P9
        OUT     #0,P9
        STA     P9BUF
        EXHL    HLBUF
        CMP     #00H,RFCON
        B       TBI1
        STD     #01H,RFCON
        LDIA    #00H          ; initial TCB & HTC register
        OUTA    P13
        OUTA    P12
        STATBL
        STATBM
        STATBH
        B       TBIEND
```

## Preliminary

```

TBI1:  OUTA    P10
       INA     P9
       OUT     #0,P9
       STA     P9BUF
       EXHL    HLBUF
       LDIA    #00H           ; disable RFO before reading the counter value
       OUTA    P20
       INA     P12           ; store the counter value to RAM[00] - RAM[04]
       STA     00H
       INA     P13
       STA     01H
       LDATBL
       STA     02H
       LDATBM
       STA     03H
       LDATBH
       STA     04H

TBIEND: EXHL    HLBUF
        LDA     P9BUF
        OUTA    P9
        INA     P10
        RTI

MAIN:   STD     #00H,RFCON    ; main program
        LDIA    #0001B       ; P4.0 (RX) output
        OUTA    P18
        LDIA    #0010B       ; enable timebase interrupt
        EXAE
        EICIL    0
        LDIA    #1111B       ; the pulse rate of RFO=215/LXIN sec.
        OUTA    P20
        LDIA    #0101B       ; enable timebase, interrupt frequency : LXIN / 215 Hz
        OUTA    P25
        :
    
```

## Preliminary

### Auto load timer mode

In this mode, there are four different internal pulse rates can be selected by P20. The HTC loads the initial values by the counter registers (P12, P13) and increases at the rising edges of internal pulse generated by the time base. The value of TCB increases one when the high speed counter overflows and generates an overflow interrupt (TRGB) when the TCB overflows. This mode is only available for DUAL operation mode.

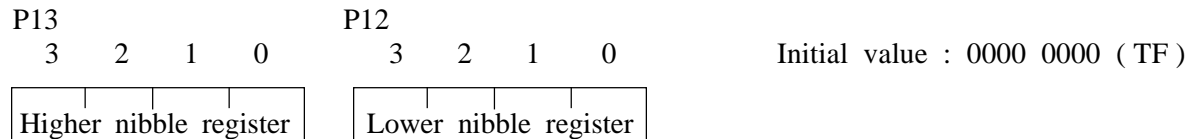
### PROGRAM EXAMPLE :

```

LDIA          #00H          ; initial TCB & HTC register
STATBL
STATBM
STATBH
OUTA          P13
OUTA          P12
OUTA          P18
LDIA          #0111B        ; enable timer mode, internal pulse rate : CLK/27
OUTA          P20
:
LDIA          #00H          ; disable timer mode
OUTA          P20
INA           P12          ; store the counter value to RAM[00] - RAM[04]
STA           00H
INA           P13
STA           01H
LDATBL
STA           02H
LDATBM
STA           03H
LDATBH
STA           04H
    
```

### Melody mode

In the melody mode, HTC will output the square wave to the PWM circuit or D/A converter. The 8-bit tone frequency register is P13 and P12. The tone frequency will be changed when users output the different data to P12. Thus, the data must be output to P13 before P12 when users want to change the 8-bit tone frequency (TF). This mode is only available for DUAL operation mode.



\*\*  $F_{TONE} = [ (CLK / 2^X) / (100H - TF) ] / 2$ , TF = 0 ~ 255

\*\* Example : CLK = 4.6MHz, RATE = 01, TF = 11001110 B= 0CEH.  
 $F_{TONE} = [ (4.6MHz / 2^5) / (100H - 0CEH) ] / 2 = 1430 \text{ Hz.}$

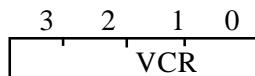
## Preliminary

Volume control register (P17)

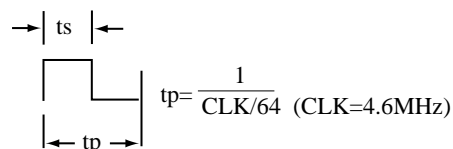
The are 16 levels of volume for sound generator. P17 is the volume control register.

Port17

Initial value : 0000



VCR				ts/tp
1	1	1	1	15/16
1	1	1	0	14/16
:	:	:	:	:
0	0	0	1	1/16
0	0	0	0	0/16



**PROGRAM EXAMPLE:**

```

LDIA    #0CH
OUTA    P13
LDIA    #0EH
OUTA    P12
LDIA    #0111B           ; volume control
OUTA    P17
LDIA    #1010B           ; 1430 Hz tone output
OUTA    P20
    
```

**LCD DRIVER**

It can directly drive the liquid crystal display (LCD) and has 64 segment pins, 16 or 32 common pins by mask option. There are total 64x16 or 64x32 dots can be display. The V1~V5, VA and VB pins have to connect the capacitors for LCD voltage multiplier.

	<b>16 common pins</b>	<b>32 common pins</b>
Display dots	16x64 dots	32x64 dots
Bias	1/5 bias	1/5 bias
Duty	1/16 duty	1/32 duty
LCD display RAM	Bank2 (P9=xx10B)	Bank2(P9=xx10B), Bank3(P9=xx11B)
I/O or LCD pin by mask option	COM0..15, SEG0..59, P1[0..3]/SEG63..60	COM0..31, SEG0..43, P7[0..3]/SEG47..44, P6[0..3]/SEG51..48, P5[0..3]/SEG55..52, P2[0..3]/SEG59..56, P1[0..3]/SEG63..60

Preliminary

**LCD driver control command register (P27) :**

Port27    3   2   1   0    Initial value : 0000



LDC	LCD display control
0	LCD display disable
1	LCD display enable

VREF	Reference voltage	V5(1/5bias)*1
0 0 0	0.85V	4.25V
0 0 1	0.90V	4.50V
0 1 0	0.95V	4.75V
0 1 1	1.00V	5.00V
1 0 0	1.05V	5.25V
1 0 1	Reserved	Reserved
1 1 *	Reserved	Reserved

\* : Don't care.

\*1: V5 is LCD working voltage  
(suggestion only).

Example :

```
LDIA #1001B ; enable LCD.
OUTA P27
:
LDIA #0000B ; disable LCD.
OUTA P27
```

**LCD display data area:**

The LCD display data is stored in the display data area of the data memory (RAM).

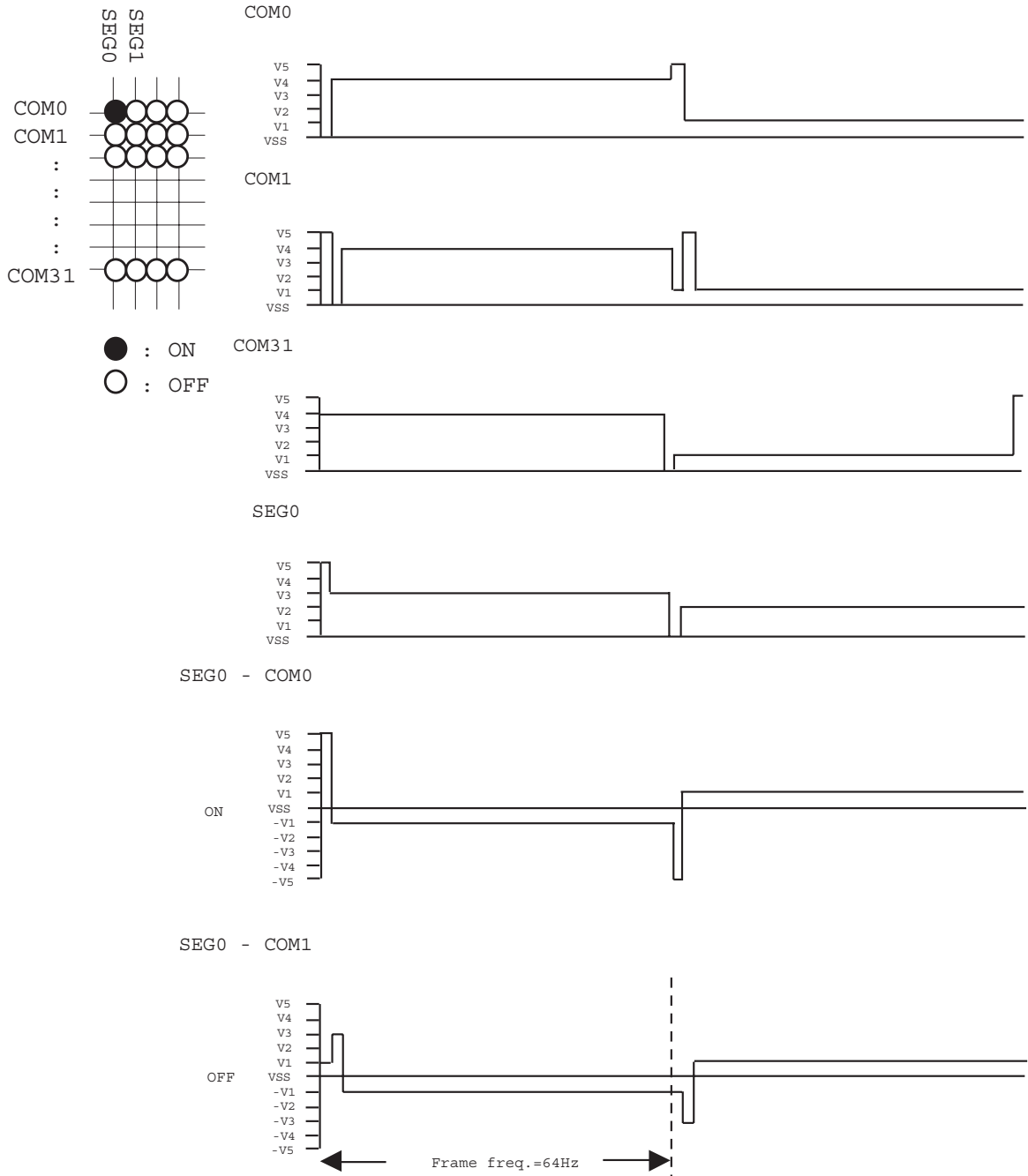
Bank 2	Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
P9=xx10B	200-20Fh								COM0									
	210-21Fh								COM1									
	220-22Fh								COM2									
	230-23Fh								COM3									
	240-24Fh								COM4									
	250-25Fh								COM5									
	260-26Fh								COM6									
	270-27Fh								COM7									
	280-28Fh								COM8									
	290-29Fh								COM9									
	2A0-2AFh								COM10									
	2B0-2BFh								COM11									
	2C0-2CFh								COM12									
	2D0-2DFh								COM13									
	2E0-2EFh								COM14									
2F0-2FFh								COM15										
Bank 3 P9=xx11B	300-30Fh								COM16									
	310-31Fh								COM17									
	320-32Fh								COM18									
	330-33Fh								COM19									
	340-34Fh								COM20									
	350-35Fh								COM21									
	360-36Fh								COM22									
	370-37Fh								COM23									
	380-38Fh								COM24									
	390-39Fh								COM25									
	3A0-3AFh								COM26									
	3B0-3BFh								COM27									
	3C0-3CFh								COM28									
	3D0-3DFh								COM29									
	3E0-3EFh								COM30									
3F0-3FFh								COM31										

\* This specification are subject to be changed without notice.

**Preliminary**

**LCD waveform :**

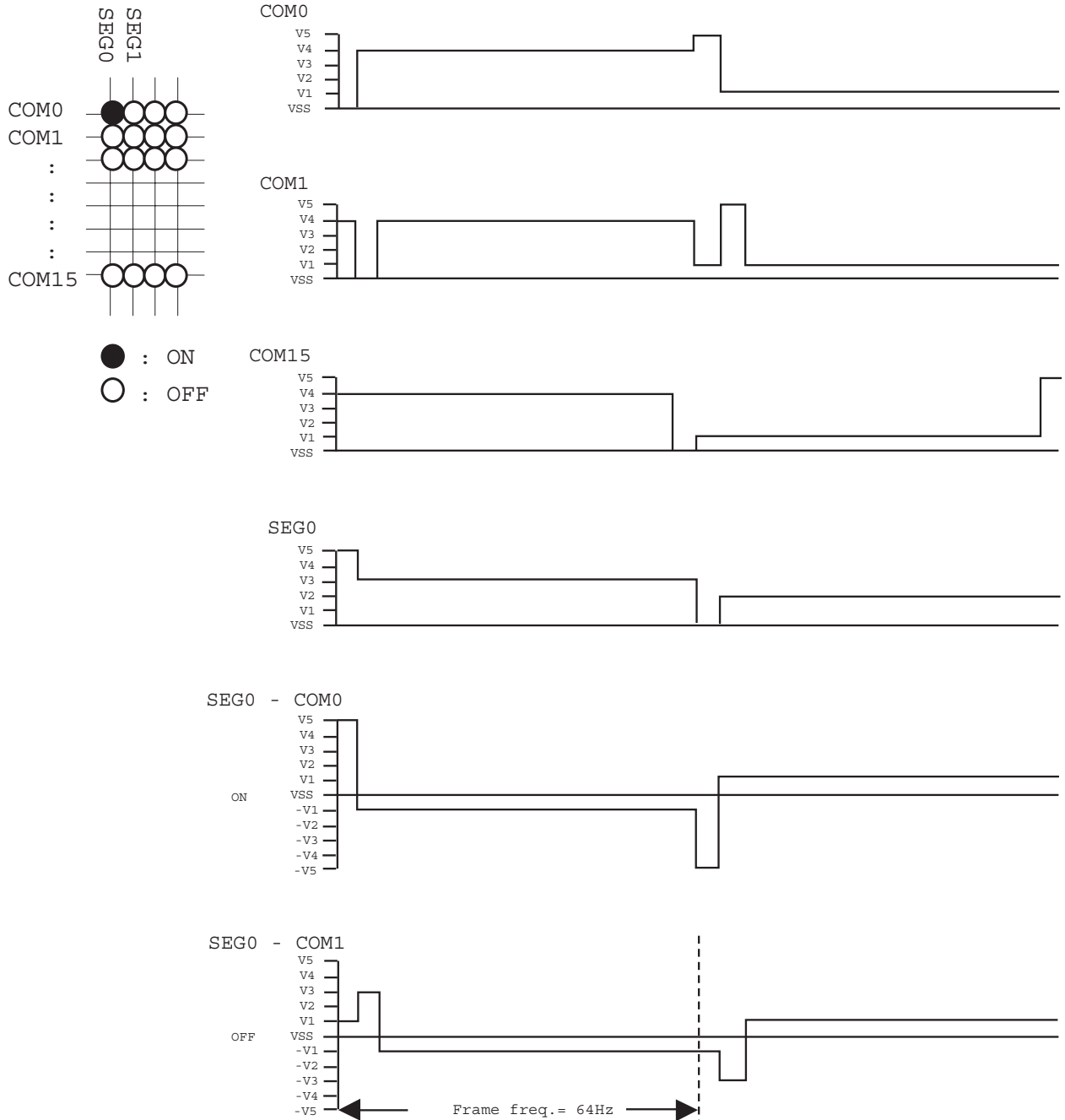
(1)1/32 duty, 1/5 bias





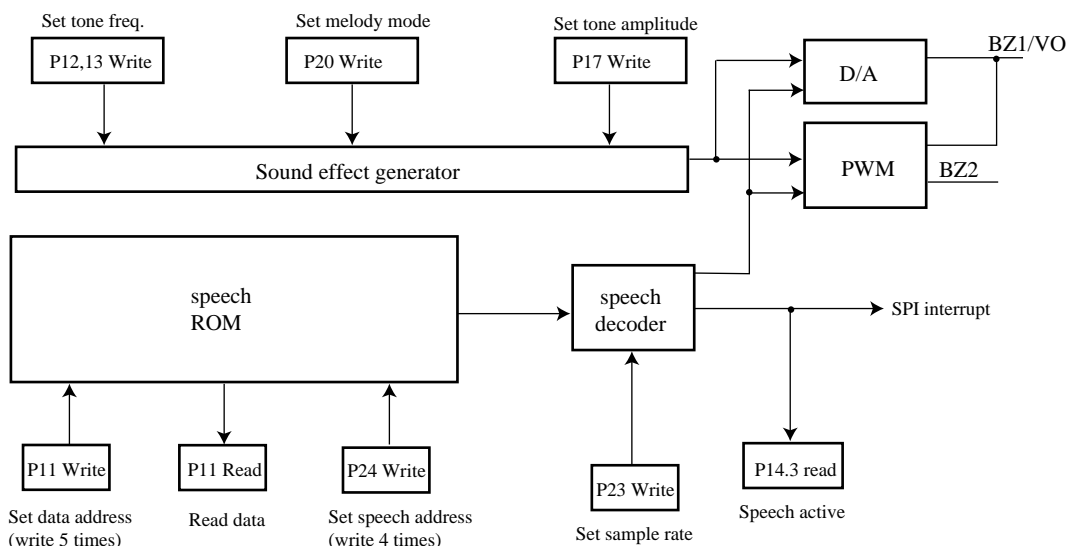
**Preliminary**

(2)1/16 duty, 1/5 bias



*Preliminary*

**SPEECH SYNTHESIZER**



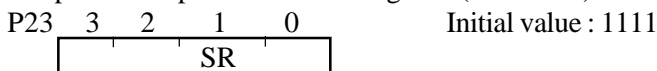
Block diagram of speech and sound effect

EM73MA89B speech synthesizer operates as following :

1. Send the speech start address to the address latch by writing P24 four times.
2. Choose the sampling rate, enable the speech synthesizer by writing P23.
3. The ROM address counters send the ROM address A6 .. A19 to the speech ROM.
4. ACT is the speech acknowledge signal. When the speech synthesizer has voice output. ACT is high. When ACT is changed from high to low, the speech synthesizer can generate the speech ending interrupt SPI. The ACT signal can be read from P14.3.

**SPEECH SYNTHESIZER CONTROL**

Speech sample rate control register (P23 write) :



SR	Sample rate selection
0000	CLK/64/2/3
0001	CLK/64/2/4
0010	CLK/64/3/3
0011	CLK/64/3/4
0100	CLK/64/2/7
0101	CLK/64/4/4
0110	CLK/64/6/3
0111	CLK/64/6/4
1***	Disable speech

port 23 -- initialization is "1111".  
port 24 -- initialization is pointed to the low-nibble of start address latch.

The frequency of CLK is decided by mask option.

## Preliminary

Speech active flag (P14.3 read) :

P14    3    2    1    0                      Initial value : 0000

ACT	WKS	SINT	CPUS
-----	-----	------	------

ACT is the speech acknowledge signal. When the speech synthesizer has voice output, ACT is high. When ACT is high → low, the speech synthesizer can generate the speech ending interrupt SPI.

P14(0,2) are CPU status flags (refer to CPU status). P14.1 is the interrupt source selector (refer to interrupt).

Speech start address register (P24 write) :

P24    3    2    1    0                      Initial value : 1111

Port 24
---------

P24L1	P24L2	P24L3	P24L4
A9   A8   A7   A6	A13   A12   A11   A10	A17   A16   A15   A14	-   -   A19   A18

Send the speech start address to the speech synthesizer by writing P24 four times. There is a pointer counter to point the address latch (P24L1, P24L2, P24L3, P24L4). It will increase one when write P24. So, the first time writing P24 to P24L1, the second time is P24L2, the third time is P24L3, the fourth time is P24L4 and the fifth time is P24L1 latch again, ... etc. The pointer counter point to P24L1 when CPU is reset or P23 is written. In the DUAL operation mode, the speech synthesizer is available. In the other operation modes, it is disable.

**PROGRAM EXAMPLE:**

```

CHIP ROM16K
;-----RAMdefinearea-----
DSEG
ORG 10H
HLBUF: RES 2 ; HL buffer for interrupt
P9BUF: RES 1 ; P9 (RAM bank) buffer for interrupt
:
;-----Constant-----
ACT EQU 143
SPEECH EQU 43200H
:
;-----Interrupt subroutine-----
CSEG
ORG 004H
LBR SPI
:
SPI: OUTA P10 ; save Acc to general purpose register P10
INA P9
OUT #0000B,P9
STA P9BUF ; save RAM bank to P9BUF
EXHL HLBUF ; save HL to HLBUF
:
:
EXHL HLBUF ; restore HLBUF to HL
LDA P9BUF ; resotre P9BUF to RAM bank
OUTA P9
INA P10 ; restore register P10 to Acc
RTI

```

} 10 instruction bytes

} 10 instruction bytes

**Preliminary**

```

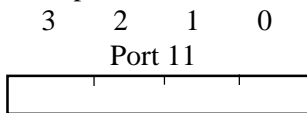
;-----Mainprogram-----
MAIN:
:
LDIA #0000B
OUTA P14 ; select SPI interrupt
LDIA #SPEECH/40H ; set speech start address
OUTA P24
LDIA #SPEECH/400H
OUTA P24
LDIA #SPEECH/4000H
OUTA P24
LDIA #SPEECH/40000H
OUTA P24
LDIA #0011B ; set sampling rate and start playing
OUTA P23
:
WAIT:
TTP P14,3 ; wait speed end
B WAIT
:

```

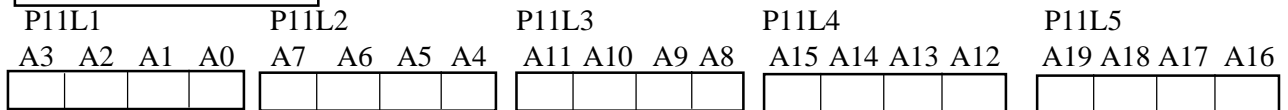
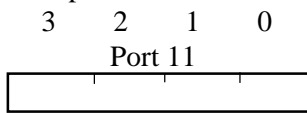
### USING SPEECH ROM AS DATA ROM

The speech ROM can be used for speech synthesizer and for data ROM simultaneously. First, write initial address to P11 five times, then you can read P11 to get data, and the address counter increases one automatically. The read operation should be all done before you leave normal mode and change to slow mode.

Get speech ROM data (P11 read) :



Set speech ROM address (P11 write) :



### PROGRAM EXAMPLE:

```

DATA_ADR EQU 12345H ; the start address of the speech ROM
:
LDIA #DATA_ADR
OUTA P11
LDIA #DATA_ADR/10H
OUTA P11
LDIA #DATA_ADR/100H
OUTA P11
LDIA #DATA_ADR/1000H
OUTA P11
LDIA #DATA_ADR/10000H
OUTA P11

```

## Preliminary

```

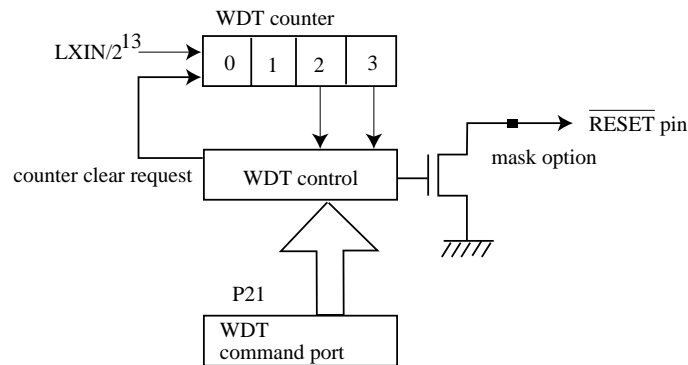
; READ DATA
INA          P11          ; read DATA_ADR
STA          TEMP
INA          P11
STA          TEMP+1
:

```

### WATCH-DOG-TIMER (WDT)

Watch-dog-timer can help user to detect the malfunction (runaway) of CPU and give system a timeup signal every certain time. User can use the time up signal to give system a reset signal when system is fail. This function is available by mask option. If the mask option of WDT is enabled, it will stop counting when CPU is reseted or in the STOP operation mode.

The basic structure of Watch-Dog-Timer control is composed by a 4-stage binary counter and a control unit. The WDT counter counts for a certain time to check the CPU status, if there is no malfunction happened, the counter will be cleared and continue counting. Otherwise, if there is a malfunction happened, the WDT control will send a WDT signal (low active) to reset CPU. The WDT checking period is assign by P21 (WDT command port).



P21 is the control port of watch-dog-timer, and the WDT time up signal is connected to  $\overline{\text{RESET}}$ .

```

Port 21   3   2   1   0   Initial value :0000
          CWC * 1 * WDT

```

CWC	Clear watchdog timer counter
0	Clear counter then return to 1
1	Nothing

WDT	Set watch-dog-timer detect time
0	$3 \times 2^{13}/\text{LXIN} = 3 \times 2^{13}/32\text{K Hz} = 0.75 \text{ sec}$
1	$7 \times 2^{13}/\text{LXIN} = 7 \times 2^{13}/32\text{K Hz} = 1.75 \text{ sec}$

### PROGRAM EXAMPLE

To enable WDT with  $7 \times 2^{13}/\text{LXIN}$  detection time.

```

LDIA #0001B
OUTA P21 ; set WDT detection time and clear WDT counter
:
:

```

## Preliminary

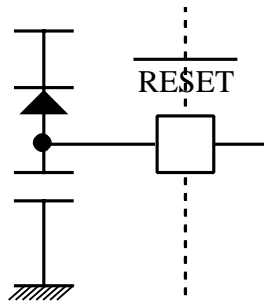
### RESETTING FUNCTION

When CPU in normal working condition and  $\overline{\text{RESET}}$  pin is held in low level for three instruction cycles at least, then CPU begins to initialize the whole internal states, when  $\overline{\text{RESET}}$  pin changes to high level, CPU begins to work in normal condition.

The CPU internal state during reset condition is as following table :

Hardware condition in RESET state	Initial value
Program counter	0000h
Status flag	01h
Interrupt enable flip-flop ( EI )	00h
MASK0 ,1, 2, 3	00h
Interrupt latch ( IL )	00h
P3, 9, 10, 11, 12, 13, 14, 16, 18, 19, 20, 21, 22, 25, 27, 28, 29	00h
P0, 1, 2, 4, 5, 6, 7, 8, 17, 23, 24	0Fh
CLK, LXIN	Start oscillation

The  $\overline{\text{RESET}}$  pin is a hysteresis input pin and it has a pull-up resistor available by mask option. The simplest RESET circuit is connect  $\overline{\text{RESET}}$  pin with a capacitor to  $V_{SS}$  and a diode to  $V_{DD}$ .



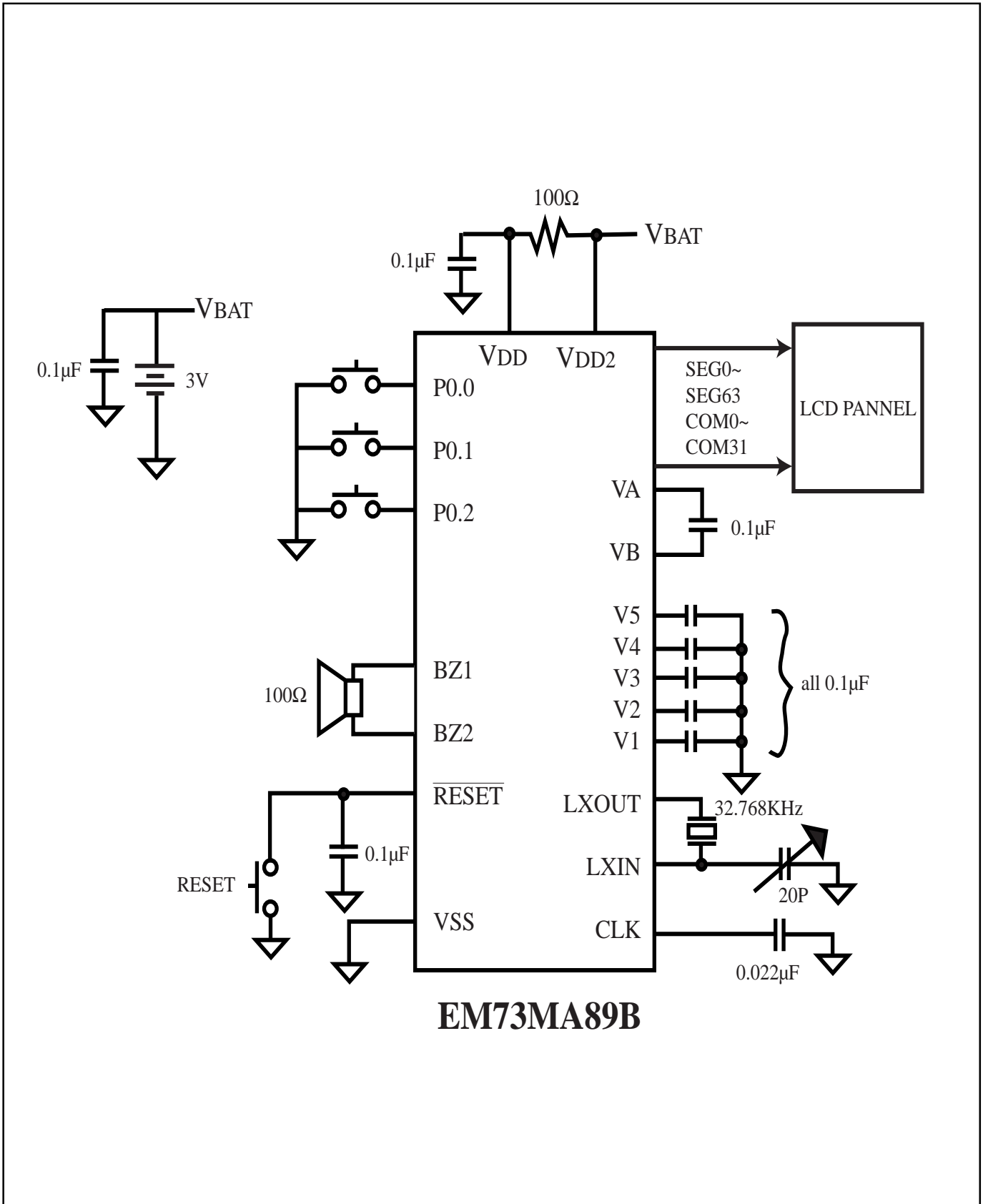
**Preliminary**

**EM73MA89B I/O PORT DESCRIPTION :**

Port	Input function	Output function	Note
0	E Input port , wakeup function		
1	E Input port	E Output port / LCD segment pins	
2	E Input port	E Output port / LCD pins	
3	I ROM bank selection	I ROM bank selection	
4	E Input port	E Output port / RFO pins	
5	E Input port	E Output port / LCD pins	
6	E Input port	E Output port / LCD pins	
7	E Input port	E Output port / LCD pins	
8	E Input port, wakeup function, external interrupt input	E Output port	
9	I RAM bank selection	I RAM bank selection	
10	I General purpose register	I General purpose register	
11	I Read data register	I Data ROM address register	
12	--	I High speed counter register	Low nibble
13	--	I High speed counter register	High nibble
14	I CPU status, ACT flag	I CPU status, interrupt souce selector	
15	--	--	
16		I STOP mode control register	
17		I TONE volume control register	
18		I HTC control register	
19		I IDLE mode control register	
20		I HTC control register	
21		I WDT control register	
22		I DUAL/SLOW mode control register	
23		I Speech sampling rate register	
24		I Speech start address register	
25		I Timebase control register	
26		--	
27		I LCD control register	
28		I Timer/counter A control register	
29		I Timer/counter B control register	
30		--	
31		--	

**Preliminary**

**APPLICATION CIRCUIT**

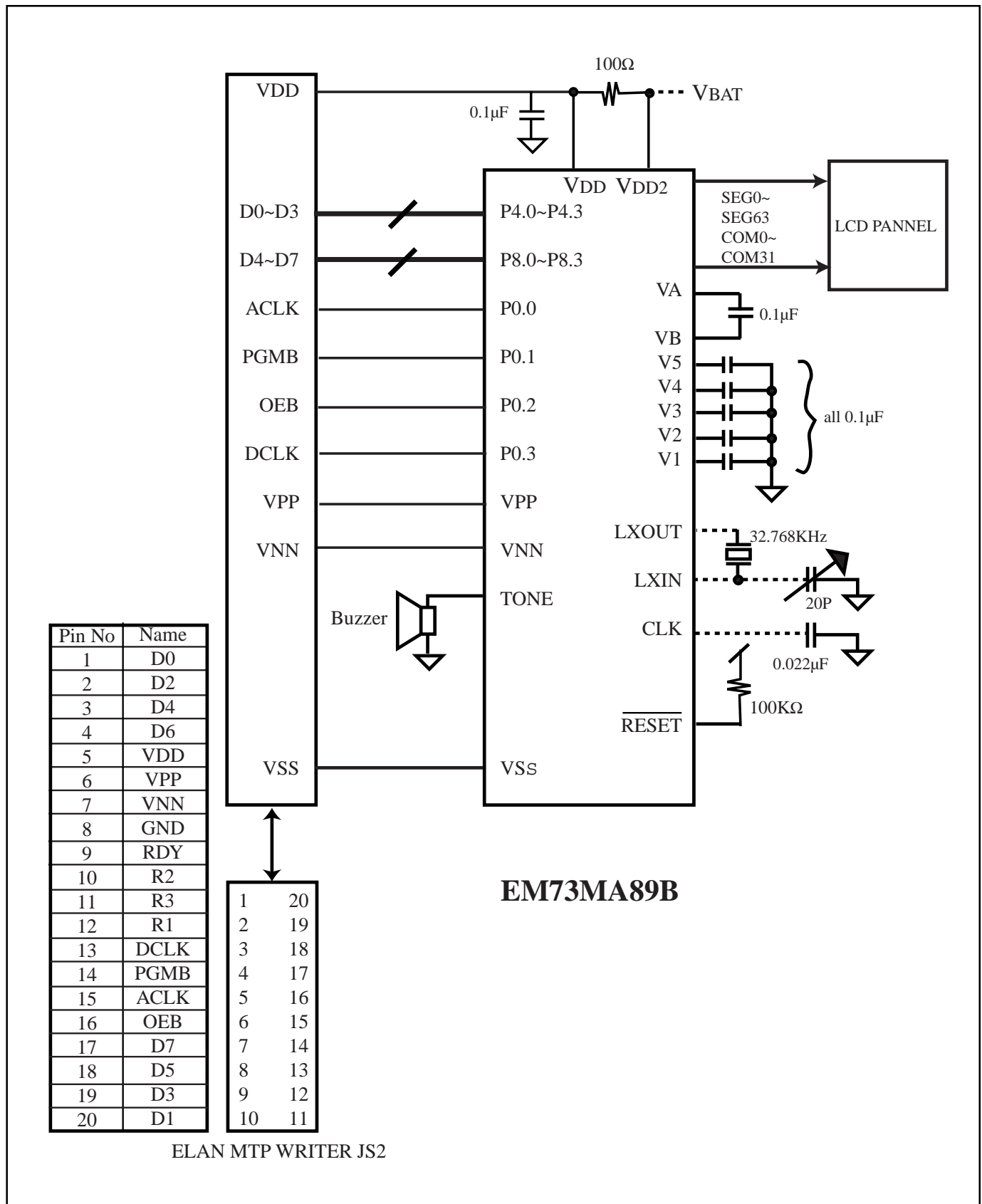


\* This specification are subject to be changed without notice.



**Preliminary**

**APPLICATION CIRCUIT (MTP PROGRAMMING MODE)**

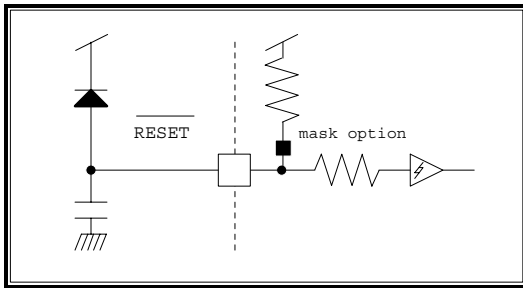


\* This specification are subject to be changed without notice.

**Preliminary**

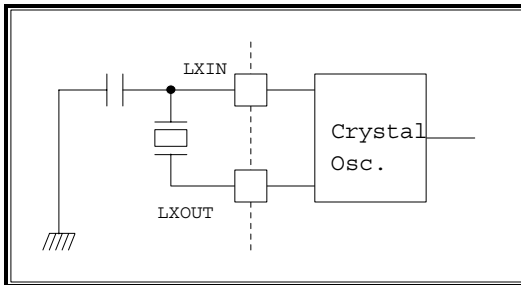
**RESET PIN TYPE**

TYPE RESET-A

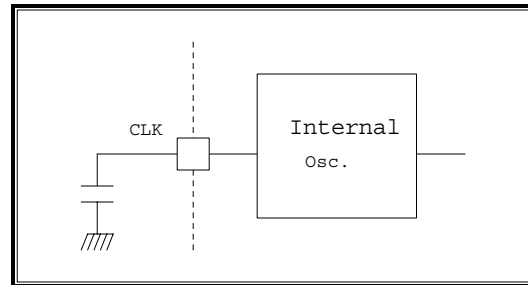


**OSCILLATION PIN TYPE**

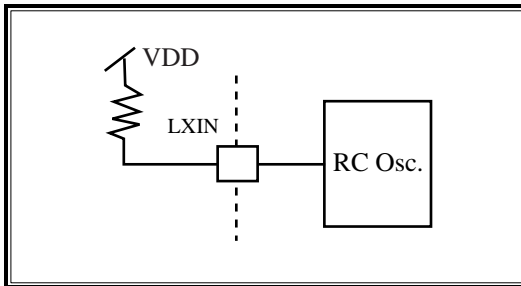
TYPE OSC-B



TYPE OSC\_G

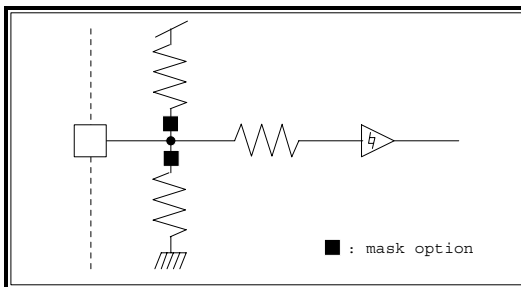


TYPE OSC-H

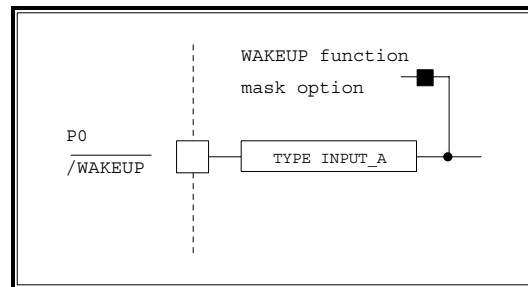


**INPUT PIN TYPE**

TYPE INPUT-A



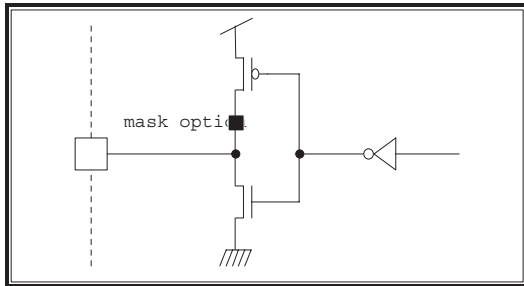
TYPE INPUT-B



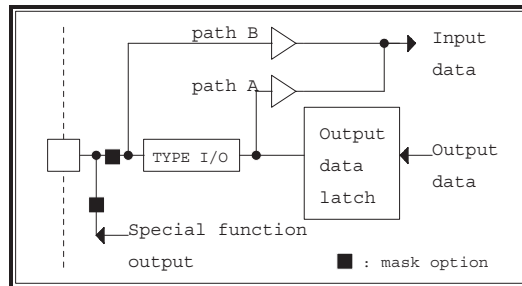
**Preliminary**

**I/O PIN TYPE**

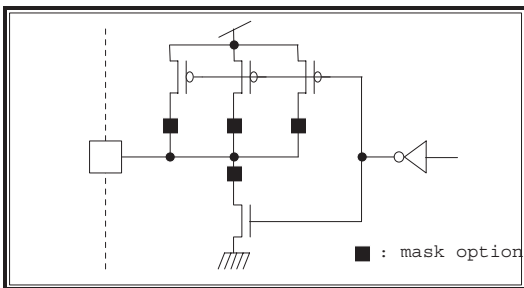
TYPE I/O



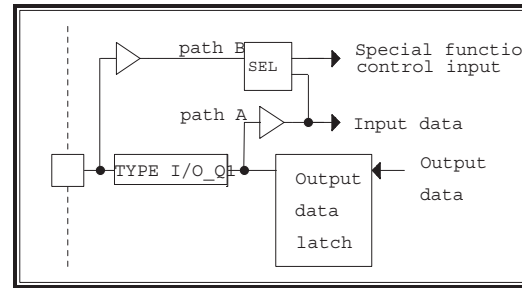
TYPE I/O-P



TYPE I/O-Q1



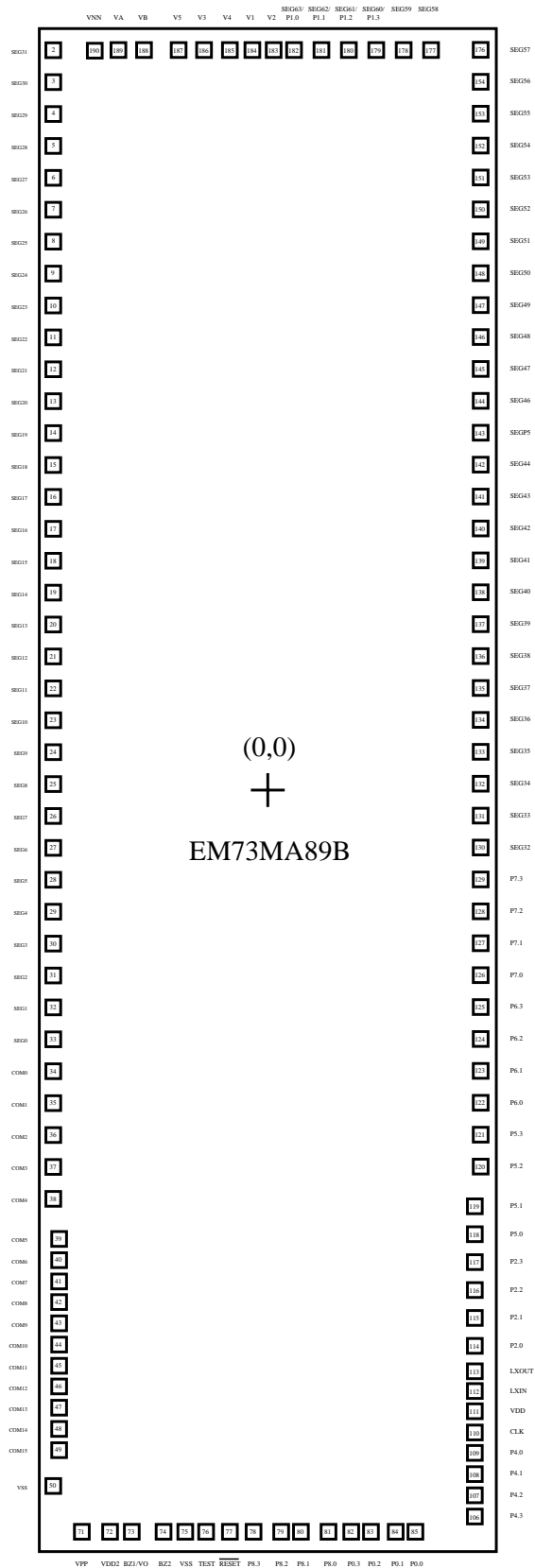
TYPE I/O-X1



- Path A : For set and clear bit of port instructions, data goes through path A from output data latch to CPU.
- Path B : For input and test instructions, data from output pin go through path B to CPU and the output data latch will be set to high.

Preliminary

PAD DIAGRAM (16 COMMONS)



\* This specification are subject to be changed without notice.



**Preliminary**

Pad No.	Symbol	X	Y
1	NC		
2	SEG31	-1300.0	4508.2
3	SEG30	-1300.0	4314.0
4	SEG29	-1300.0	4119.8
5	SEG28	-1300.0	3925.6
6	SEG27	-1300.0	3731.4
7	SEG26	-1300.0	3537.2
8	SEG25	-1300.0	3343.0
9	SEG24	-1300.0	3148.8
10	SEG23	-1300.0	2954.6
11	SEG22	-1300.0	2760.4
12	SEG21	-1300.0	2566.2
13	SEG20	-1300.0	2372.0
14	SEG19	-1300.0	2177.8
15	SEG18	-1300.0	1983.6
16	SEG17	-1300.0	1789.4
17	SEG16	-1300.0	1595.2
18	SEG15	-1300.0	1401.0
19	SEG14	-1300.0	1206.8
20	SEG13	-1300.0	1012.6
21	SEG12	-1300.0	818.4
22	SEG11	-1300.0	624.2
23	SEG10	-1300.0	430.0
24	SEG9	-1300.0	235.8
25	SEG8	-1300.0	41.6
26	SEG7	-1300.0	-152.6
27	SEG6	-1300.0	-346.8
28	SEG5	-1300.0	-541.0
29	SEG4	-1300.0	-735.2
30	SEG3	-1300.0	-929.4
31	SEG2	-1300.0	-1123.6
32	SEG1	-1300.0	-1317.8
33	SEG0	-1300.0	-1512.0
34	COM0	-1300.0	-1706.2
35	COM1	-1300.0	-1900.4
36	COM2	-1300.0	-2094.6
37	COM3	-1300.0	-2288.8
38	COM4	-1300.0	-2483.0
39	COM5	-1264.4	-2727.0
40	COM6	-1264.4	-2856.3



**Preliminary**

Pad No.	Symbol	X	Y
41	COM7	-1264.4	-2985.6
42	COM8	-1264.4	-3111.2
43	COM9	-1264.4	-3240.5
44	COM10	-1264.4	-3369.8
45	COM11	-1264.4	-3499.1
46	COM12	-1264.4	-3624.7
47	COM13	-1264.4	-3754.0
48	COM14	-1264.4	-3883.3
49	COM15	-1264.4	-4012.6
50	VSS	-1300.0	-4230.7
51	NC		
52	NC		
53	NC		
54	NC		
55	NC		
56	NC		
57	NC		
58	NC		
59	NC		
60	NC		
61	NC		
62	NC		
63	NC		
64	NC		
65	NC		
66	NC		
67	NC		
68	NC		
69	NC		
70	NC		
71	VPP	-1126.2	-4509.4
72	VDD2	-954.5	-4509.4
73	BZ1/VO	-822.4	-4509.4
74	BZ2	-629.1	-4509.4
75	VSS	-496.9	-4509.4
76	TEST	-370.2	-4509.4
77	RESET	-225.6	-4509.4
78	P8.3	-81.6	-4509.4
79	P8.2	86.4	-4509.4
80	P8.1	206.4	-4509.4



**Preliminary**

Pad No.	Symbol	X	Y
81	P8.0	374.4	-4509.4
82	P0.3	512.5	-4509.4
83	P0.2	632.5	-4509.4
84	P0.1	783.0	-4509.4
85	P0.0	903.0	-4509.4
86	NC		
87	NC		
88	NC		
89	NC		
90	NC		
91	NC		
92	NC		
93	NC		
94	NC		
95	NC		
96	NC		
97	NC		
98	NC		
99	NC		
100	NC		
101	NC		
102	NC		
103	NC		
104	NC		
105	NC		
106	P4.3	1264.4	-4416.5
107	P4.2	1264.4	-4287.4
108	P4.1	1264.4	-4158.3
109	P4.0	1264.4	-4029.2
110	CLK	1264.4	-3905.1
111	VDD	1264.4	-3775.8
112	LXIN	1264.4	-3652.6
113	LXOUT	1264.4	-3532.3
114	P2.0	1264.4	-3377.8
115	P2.1	1264.4	-3207.9
116	P2.2	1264.4	-3038.0
117	P2.3	1264.4	-2868.1
118	P5.0	1264.4	-2698.2
119	P5.1	1264.4	-2528.3
120	P5.2	1300.0	-2286.7



**Preliminary**

Pad No.	Symbol	X	Y
121	P5.3	1300.0	-2092.5
122	P6.0	1300.0	-1898.3
123	P6.1	1300.0	-1704.1
124	P6.2	1300.0	-1509.9
125	P6.3	1300.0	-1315.7
126	P7.0	1300.0	-1121.5
127	P7.1	1300.0	-927.3
128	P7.2	1300.0	-733.1
129	P7.3	1300.0	-538.9
130	SEG32	1300.0	-344.7
131	SEG33	1300.0	-150.5
132	SEG34	1300.0	43.7
133	SEG35	1300.0	237.9
134	SEG36	1300.0	432.1
135	SEG37	1300.0	626.3
136	SEG38	1300.0	820.5
137	SEG39	1300.0	1014.7
138	SEG40	1300.0	1208.9
139	SEG41	1300.0	1403.1
140	SEG42	1300.0	1597.3
141	SEG43	1300.0	1791.5
142	SEG44	1300.0	1985.7
143	SEG45	1300.0	2179.9
144	SEG46	1300.0	2374.1
145	SEG47	1300.0	2568.3
146	SEG48	1300.0	2762.5
147	SEG49	1300.0	2956.7
148	SEG50	1300.0	3150.9
149	SEG51	1300.0	3345.1
150	SEG52	1300.0	3539.3
151	SEG53	1300.0	3733.5
152	SEG54	1300.0	3927.7
153	SEG55	1300.0	4121.9
154	SEG56	1300.0	4316.1
155	NC		
156	NC		
157	NC		
158	NC		
159	NC		
160	NC		





**Preliminary**

Pad No.	Symbol	X	Y
161	NC		
162	NC		
163	NC		
164	NC		
165	NC		
166	NC		
167	NC		
168	NC		
169	NC		
170	NC		
171	NC		
172	NC		
173	NC		
174	NC		
175	NC		
176	SEG57	1300.0	4510.3
177	SEG58	998.1	4509.4
178	SEG59	831.4	4509.4
179	SEG60/P1.3	664.8	4509.4
180	SEG61/P1.2	498.1	4509.4
181	SEG62/P1.1	331.4	4509.4
182	SEG63/P1.0	164.7	4509.4
183	V2	40.7	4509.4
184	V1	-88.8	4509.4
185	V4	-225.7	4509.4
186	V3	-382.9	4509.4
187	V5	-537.4	4509.4
188	VB	-748.0	4509.4
189	VA	-903.7	4509.4
190	VNN	-1047.6	4505.5
191	NC		
192	NC		
193	NC		
194	NC		
195	NC		
196	NC		
197	NC		
198	NC		
199	NC		
200	NC		



**Preliminary**

Pad No.	Symbol	X	Y
201	NC		
202	NC		
203	NC		
204	NC		
205	NC		
206	NC		
207	NC		
208	NC		

Unit :  $\mu\text{m}$

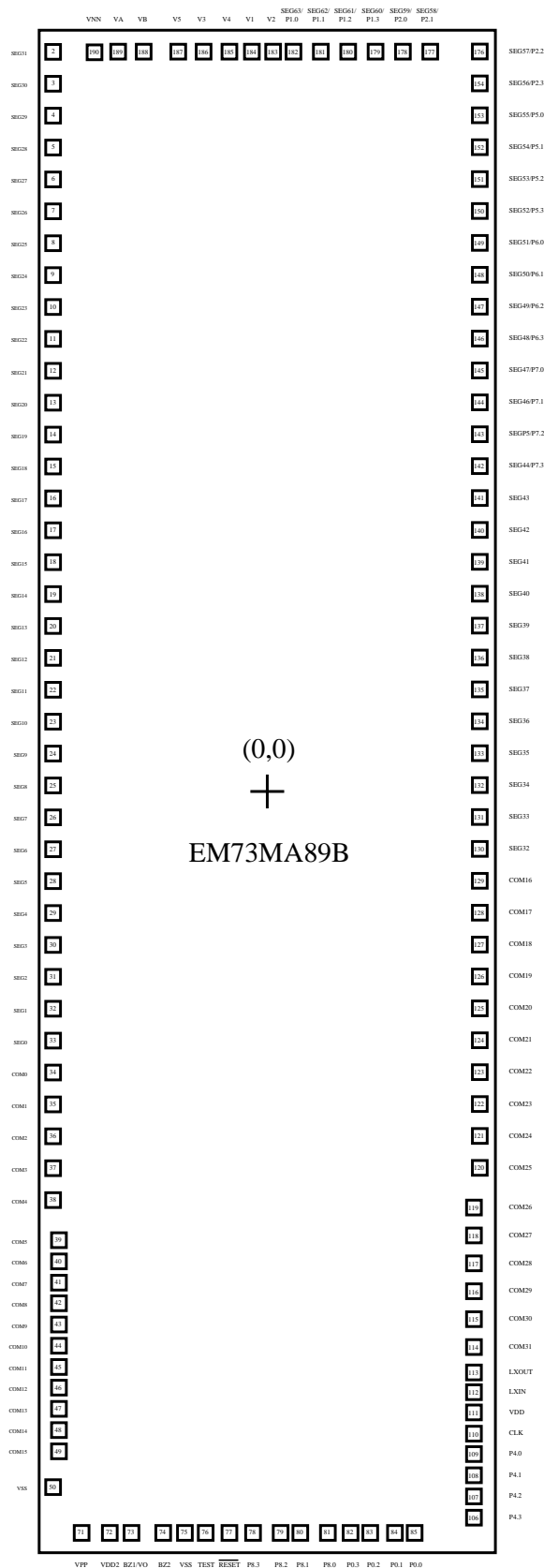
Chip size : 2890 x 9380 $\mu\text{m}$

Note : For PCB layout, IC substrate must be floated or connected to Vss.



Preliminary

PAD DIAGRAM (32 COMMONS)



\* This specification are subject to be changed without notice.



**Preliminary**

Pad No.	Symbol	X	Y
1	NC		
2	SEG31	-1300.0	4508.2
3	SEG30	-1300.0	4314.0
4	SEG29	-1300.0	4119.8
5	SEG28	-1300.0	3925.6
6	SEG27	-1300.0	3731.4
7	SEG26	-1300.0	3537.2
8	SEG25	-1300.0	3343.0
9	SEG24	-1300.0	3148.8
10	SEG23	-1300.0	2954.6
11	SEG22	-1300.0	2760.4
12	SEG21	-1300.0	2566.2
13	SEG20	-1300.0	2372.0
14	SEG19	-1300.0	2177.8
15	SEG18	-1300.0	1983.6
16	SEG17	-1300.0	1789.4
17	SEG16	-1300.0	1595.2
18	SEG15	-1300.0	1401.0
19	SEG14	-1300.0	1206.8
20	SEG13	-1300.0	1012.6
21	SEG12	-1300.0	818.4
22	SEG11	-1300.0	624.2
23	SEG10	-1300.0	430.0
24	SEG9	-1300.0	235.8
25	SEG8	-1300.0	41.6
26	SEG7	-1300.0	-152.6
27	SEG6	-1300.0	-346.8
28	SEG5	-1300.0	-541.0
29	SEG4	-1300.0	-735.2
30	SEG3	-1300.0	-929.4
31	SEG2	-1300.0	-1123.6
32	SEG1	-1300.0	-1317.8
33	SEG0	-1300.0	-1512.0
34	COM0	-1300.0	-1706.2
35	COM1	-1300.0	-1900.4
36	COM2	-1300.0	-2094.6
37	COM3	-1300.0	-2288.8
38	COM4	-1300.0	-2483.0
39	COM5	-1264.4	-2727.0
40	COM6	-1264.4	-2856.3



**Preliminary**

Pad No.	Symbol	X	Y
41	COM7	-1264.4	-2985.6
42	COM8	-1264.4	-3111.2
43	COM9	-1264.4	-3240.5
44	COM10	-1264.4	-3369.8
45	COM11	-1264.4	-3499.1
46	COM12	-1264.4	-3624.7
47	COM13	-1264.4	-3754.0
48	COM14	-1264.4	-3883.3
49	COM15	-1264.4	-4012.6
50	VSS	-1300.0	-4230.7
51	NC		
52	NC		
53	NC		
54	NC		
55	NC		
56	NC		
57	NC		
58	NC		
59	NC		
60	NC		
61	NC		
62	NC		
63	NC		
64	NC		
65	NC		
66	NC		
67	NC		
68	NC		
69	NC		
70	NC		
71	VPP	-1126.2	-4509.4
72	VDD2	-954.5	-4509.4
73	BZ1/VO	-822.4	-4509.4
74	BZ2	-629.1	-4509.4
75	VSS	-496.9	-4509.4
76	TEST	-370.2	-4509.4
77	RESET	-225.6	-4509.4
78	P8.3	-81.6	-4509.4
79	P8.2	86.4	-4509.4
80	P8.1	206.4	-4509.4



**Preliminary**

Pad No.	Symbol	X	Y
81	P8.0	374.4	-4509.4
82	P0.3	512.5	-4509.4
83	P0.2	632.5	-4509.4
84	P0.1	783.0	-4509.4
85	P0.0	903.0	-4509.4
86	NC		
87	NC		
88	NC		
89	NC		
90	NC		
91	NC		
92	NC		
93	NC		
94	NC		
95	NC		
96	NC		
97	NC		
98	NC		
99	NC		
100	NC		
101	NC		
102	NC		
103	NC		
104	NC		
105	NC		
106	P4.3	1264.4	-4416.5
107	P4.2	1264.4	-4287.4
108	P4.1	1264.4	-4158.3
109	P4.0	1264.4	-4029.2
110	CLK	1264.4	-3905.1
111	VDD	1264.4	-3775.8
112	LXIN	1264.4	-3652.6
113	LXOUT	1264.4	-3532.3
114	COM31	1264.4	-3377.8
115	COM30	1264.4	-3207.9
116	COM29	1264.4	-3038.0
117	COM28	1264.4	-2868.1
118	COM27	1264.4	-2698.2
119	COM26	1264.4	-2528.3
120	COM25	1300.0	-2286.7



**Preliminary**

Pad No.	Symbol	X	Y
121	COM24	1300.0	-2092.5
122	COM23	1300.0	-1898.3
123	COM22	1300.0	-1704.1
124	COM21	1300.0	-1509.9
125	COM20	1300.0	-1315.7
126	COM19	1300.0	-1121.5
127	COM18	1300.0	-927.3
128	COM17	1300.0	-733.1
129	COM16	1300.0	-538.9
130	SEG32	1300.0	-344.7
131	SEG33	1300.0	-150.5
132	SEG34	1300.0	43.7
133	SEG35	1300.0	237.9
134	SEG36	1300.0	432.1
135	SEG37	1300.0	626.3
136	SEG38	1300.0	820.5
137	SEG39	1300.0	1014.7
138	SEG40	1300.0	1208.9
139	SEG41	1300.0	1403.1
140	SEG42	1300.0	1597.3
141	SEG43	1300.0	1791.5
142	SEG44/P7.3	1300.0	1985.7
143	SEG45/P7.2	1300.0	2179.9
144	SEG46/P7.1	1300.0	2374.1
145	SEG47/P7.0	1300.0	2568.3
146	SEG48/P6.3	1300.0	2762.5
147	SEG49/P6.2	1300.0	2956.7
148	SEG50/P6.1	1300.0	3150.9
149	SEG51/P6.0	1300.0	3345.1
150	SEG52/P5.3	1300.0	3539.3
151	SEG53/P5.2	1300.0	3733.5
152	SEG54/P5.1	1300.0	3927.7
153	SEG55/P5.0	1300.0	4121.9
154	SEG56/P2.3	1300.0	4316.1
155	NC		
156	NC		
157	NC		
158	NC		
159	NC		
160	NC		

**Preliminary**

Pad No.	Symbol	X	Y
161	NC		
162	NC		
163	NC		
164	NC		
165	NC		
166	NC		
167	NC		
168	NC		
169	NC		
170	NC		
171	NC		
172	NC		
173	NC		
174	NC		
175	NC		
176	SEG57/P2.2	1300.0	4510.3
177	SEG58/P2.1	998.1	4509.4
178	SEG59/P2.0	831.4	4509.4
179	SEG60/P1.3	664.8	4509.4
180	SEG61/P1.2	498.1	4509.4
181	SEG62/P1.1	331.4	4509.4
182	SEG63/P1.0	164.7	4509.4
183	V2	40.7	4509.4
184	V1	-88.8	4509.4
185	V4	-225.7	4509.4
186	V3	-382.9	4509.4
187	V5	-537.4	4509.4
188	VB	-748.0	4509.4
189	VA	-903.7	4509.4
190	VNN	-1047.6	4505.5
191	NC		
192	NC		
193	NC		
194	NC		
195	NC		
196	NC		
197	NC		
198	NC		
199	NC		
200	NC		





**Preliminary**

Pad No.	Symbol	X	Y
201	NC		
202	NC		
203	NC		
204	NC		
205	NC		
206	NC		
207	NC		
208	NC		

Unit : um

Chip size :2890 x 9380um

Note : For PCB layout, IC substrate must be floated or connected to Vss.

Preliminary

### ABSOLUTE MAXIMUM RATINGS

Items	Sym.	Ratings	Conditions
Supply Voltage	$V_{DD}$	-0.5V to 3.6V	
Input Voltage	$V_{IN}$	-0.5V to $V_{DD}+0.5V$	
Output Voltage	$V_O$	-0.5V to $V_{DD}+0.5V$	
Power Dissipation	$P_D$	300mW	$T_{OPR}=50^{\circ}C$
Operating Temperature	$T_{OPR}$	-30°C to 70°C	
Storage Temperature	$T_{STG}$	-55°C to 125°C	

### RECOMMENDED OPERATING CONDITIONS

Items	Sym.	Ratings	Conditions
Supply Voltage	$V_{DD}$	2.2V to 3.6V	
Input Voltage	$V_{IH}$	$0.90 \times V_{DD}$ to $V_{DD}$	
	$V_{IL}$	0V to $0.10 \times V_{DD}$	
Operating Frequency	$F_C$	4.6MHz ~ 9.2MHz	CLK
	$F_S$	32KHz	LXIN,LXOUT

### DC ELECTRICAL CHARACTERISTICS ( $V_{DD}=3\pm 0.3V$ , $V_{SS}=0V$ , $T_{OPR}=25^{\circ}C$ )

Parameters	Sym.	Min.	Typ.	Max.	Unit	Conditions
Supply current	$I_{DD}$	-	0.5	1.2	mA	$V_{DD}=3.3V$ , DUAL mode, no load, $F_C=4.6MHz$ , $F_S=32KHz$
		-	35	45	$\mu A$	$V_{DD}=3.3V$ , SLOW mode, $F_S=32KHz$ , LCD on
		-	30	40	$\mu A$	$V_{DD}=3.3V$ , IDLE mode, LCD on
		-	7	12	$\mu A$	$V_{DD}=3.3V$ , IDLE mode, LCD off
		-	0.1	1	$\mu A$	$V_{DD}=3.3V$ , STOP mode
Hysteresis voltage	$V_{HYS+}$	$0.50V_{DD}$	-	$0.75V_{DD}$	V	RESET, P0, P8
	$V_{HYS-}$	$0.20V_{DD}$	-	$0.40V_{DD}$	V	
Input current	$I_{IH}$	-	-	$\pm 1$	$\mu A$	P0, RESET, $V_{DD}=3.3V$ , $V_{IH}=3.3/0V$
		-	-	$\pm 1$	$\mu A$	Open-drain, $V_{DD}=3.3V$ , $V_{IH}=3.3/0V$
	$I_{IL}$	-	-250	-500	$\mu A$	Push-pull (normal current push-pull) $V_{DD}=3.3V$ , $V_{IL}=0.4V$
		-	-20	-25	$\mu A$	Push-pull (low current push-pull) $V_{DD}=3.3V$ , $V_{IL}=0.4V$
Output voltage	$V_{OH}$	2.4	-	-	V	Push-pull, (high current push-pull) $V_{DD}=2.7V$ , $I_{OH}=-0.9mA$
		2.0	2.4	-	V	Push-pull, (normal current push-pull) $V_{DD}=2.7V$ , $I_{OH}=-40\mu A$
	$V_{OL}$	-	0.15	0.3	V	$V_{DD}=2.7V$ , $I_{OL}=0.9mA$
Leakage current	$I_{LO}$	-	-	1	$\mu A$	Open-drain, $V_{DD}=3.3V$ , $V_O=3.3V$
Input resistor	$R_{IN}$	100	200	300	K $\Omega$	P0, $V_{DD}=3.3V$
		300	600	900	K $\Omega$	RESET, $V_{DD}=3.3V$

**Preliminary**

Output current of BZ1, BZ2	$I_{OH}$	30	-	-	mA	$V_{DD}=3.0V, V_{BZ}=1.5V,$ mask option : small size
	$I_{OL}$	30	-	-	mA	
	$I_{OH}$	75	-	-	mA	$V_{DD}=3.0V, V_{BZ}=1.5V,$ mask option : large size
	$I_{OL}$	75	-	-	mA	
Output current of VO		2	3	4	mA	$V_{DD}=3.0V, v_o=0.7V$
LCD reference voltage	$V_{REF}$	0.765	0.85	0.935	V	$V_{DD}=3.0V, \text{no load}, VREF=000$
		0.81	0.90	0.99	V	$V_{DD}=3.0V, \text{no load}, VREF=001$
		0.855	0.95	1.045	V	$V_{DD}=3.0V, \text{no load}, VREF=010$
		0.9	1.00	1.1	V	$V_{DD}=3.0V, \text{no load}, VREF=011$
		0.945	1.05	1.155	V	$V_{DD}=3.0V, \text{no load}, VREF=100$

Preliminary

## INSTRUCTION TABLE

### (1) Data Transfer

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
LDA x	0110 1010 xxxx xxxx	Acc←RAM[x]	2	2	-	Z	1
LDAM	0101 1010	Acc ←RAM[HL]	1	1	-	Z	1
LDAX	0110 0101	Acc←ROM[DP] <sub>L</sub>	1	2	-	Z	1
LDAXI	0110 0111	Acc←ROM[DP] <sub>H</sub> ,DP+1	1	2	-	Z	1
LDH #k	1001 kkkk	HR←k	1	1	-	-	1
LDHL x	0100 1110 xxxx xx00	LR←RAM[x],HR←RAM[x+1]	2	2	-	-	1
LDIA #k	1101 kkkk	Acc←k	1	1	-	Z	1
LDL #k	1000 kkkk	LR←k	1	1	-	-	1
STA x	0110 1001 xxxx xxxx	RAM[x]←Acc	2	2	-	-	1
STAM	0101 1001	RAM[HL]←Acc	1	1	-	-	1
STAMD	0111 1101	RAM[HL]←Acc, LR-1	1	1	-	Z	C
STAMI	0111 1111	RAM[HL]←Acc, LR+1	1	1	-	Z	C'
STD #k,y	0100 1000 kkkk yyyy	RAM[y]←k	2	2	-	-	1
STDMI #k	1010 kkkk	RAM[HL]←k, LR+1	1	1	-	Z	C'
THA	0111 0110	Acc←HR	1	1	-	Z	1
TLA	0111 0100	Acc←LR	1	1	-	Z	1

### (2) Rotate

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
RLCA	0101 0000	←CF←Acc←	1	1	C	Z	C'
RRCA	0101 0001	↪CF↪Acc↪	1	1	C	Z	C'

### (3) Arithmetic operation

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
ADCAM	0111 0000	Acc←Acc + RAM[HL] + CF	1	1	C	Z	C'
ADD #k,y	0100 1001 kkkk yyyy	RAM[y]←RAM[y] + k	2	2	-	Z	C'
ADDA #k	0110 1110 0101 kkkk	Acc←Acc+k	2	2	-	Z	C'
ADDAM	0111 0001	Acc←Acc + RAM[HL]	1	1	-	Z	C'
ADDH #k	0110 1110 1001 kkkk	HR←HR+k	2	2	-	Z	C'
ADDL #k	0110 1110 0001 kkkk	LR←LR+k	2	2	-	Z	C'
ADDM #k	0110 1110 1101 kkkk	RAM[HL]←RAM[HL] + k	2	2	-	Z	C'
DECA	0101 1100	Acc←Acc-1	1	1	-	Z	C
DECL	0111 1100	LR←LR-1	1	1	-	Z	C
DECM	0101 1101	RAM[HL]←RAM[HL] -1	1	1	-	Z	C
INCA	0101 1110	Acc←Acc + 1	1	1	-	Z	C'

## Preliminary

INCL	0111 1110	LR←LR + 1	1	1	-	Z	C'
INCM	0101 1111	RAM[HL]←RAM[HL]+1	1	1	-	Z	C'
SUBA #k	0110 1110 0111 kkkk	Acc←k-Acc	2	2	-	Z	C
SBCAM	0111 0010	Acc←RAM[HL] - Acc - CF'	1	1	C	Z	C
SUBM #k	0110 1110 1111 kkkk	RAM[HL]←k - RAM[HL]	2	2	-	Z	C

#### (4) Logical operation

Mnemonic	Object code (binary)	Operation description	Byte	Cycle	Flag		
					C	Z	S
ANDA #k	0110 1110 0110 kkkk	Acc←Acc&k	2	2	-	Z	Z'
ANDAM	0111 1011	Acc←Acc & RAM[HL]	1	1	-	Z	Z'
ANDM #k	0110 1110 1110 kkkk	RAM[HL]←RAM[HL]&k	2	2	-	Z	Z'
ORA #k	0110 1110 0100 kkkk	Acc←Acc   k	2	2	-	Z	Z'
ORAM	0111 1000	Acc ← Acc   RAM[HL]	1	1	-	Z	Z'
ORM #k	0110 1110 1100 kkkk	RAM[HL]←RAM[HL]   k	2	2	-	Z	Z'
XORAM	0111 1001	Acc←Acc^RAM[HL]	1	1	-	Z	Z'

#### (5) Exchange

Mnemonic	Object code (binary)	Operation description	Byte	Cycle	Flag		
					C	Z	S
EXA x	0110 1000 xxxx xxxx	Acc↔RAM[x]	2	2	-	Z	1
EXAH	0110 0110	Acc↔HR	1	2	-	Z	1
EXAL	0110 0100	Acc↔LR	1	2	-	Z	1
EXAM	0101 1000	Acc↔RAM[HL]	1	1	-	Z	1
EXHL x	0100 1100 xxxx xx00	LR↔RAM[x], HR↔RAM[x+1]	2	2	-	-	1

#### (6) Branch

Mnemonic	Object code (binary)	Operation description	Byte	Cycle	Flag		
					C	Z	S
SBR a	00aa aaaa	If SF=1 then PC←PC <sub>12-6</sub> ·a <sub>5-0</sub> else null	1	1	-	-	1
LBR a	1100 aaaa aaaa aaaa	If SF= 1 then PC←a else null	2	2	-	-	1
SLBR a	0101 0101 1100 aaaa aaaa aaaa (a:1000~1FFFh) 0101 0111 1100 aaaa aaaa aaaa (a:0000~0FFFh)	If SF=1 then PC←a else null	3	3	-	-	1

#### (7) Compare

Mnemonic	Object code (binary)	Operation description	Byte	Cycle	Flag		
					C	Z	S
CMP #k,y	0100 1011 kkkk yyyy	k-RAM[y]	2	2	C	Z	Z'
CMPA x	0110 1011 xxxx xxxx	RAM[x]-Acc	2	2	C	Z	Z'

## Preliminary

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
CMPAM	0111 0011	RAM[HL] <sub>b</sub> - Acc	1	1	C	Z	Z'
CMPH #k	0110 1110 1011 kkkk	k - HR	2	2	-	Z	C
CMPIA #k	1011 kkkk	k - Acc	1	1	C	Z	Z'
CMPL #k	0110 1110 0011 kkkk	k-LR	2	2	-	Z	C

### (8) Bit manipulation

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
CLM b	1111 00bb	RAM[HL] <sub>b</sub> ← 0	1	1	-	-	1
CLP p,b	0110 1101 11bb pppp	PORT[p] <sub>b</sub> ← 0	2	2	-	-	1
CLPL	0110 0000	PORT[LR <sub>3-2</sub> +4]LR <sub>1-0</sub> ← 0	1	2	-	-	1
CLR y,b	0110 1100 11bb yyyy	RAM[y] <sub>b</sub> ← 0	2	2	-	-	1
SEM b	1111 01bb	RAM[HL] <sub>b</sub> ← 1	1	1	-	-	1
SEP p,b	0110 1101 01bb pppp	PORT[p] <sub>b</sub> ← 1	2	2	-	-	1
SEPL	0110 0010	PORT[LR <sub>3-2</sub> +4]LR <sub>1-0</sub> ← 1	1	2	-	-	1
SET y,b	0110 1100 01bb yyyy	RAM[y] <sub>b</sub> ← 1	2	2	-	-	1
TF y,b	0110 1100 00bb yyyy	SF ← RAM[y] <sub>b</sub> '	2	2	-	-	*
TFA b	1111 10bb	SF ← Acc <sub>b</sub> '	1	1	-	-	*
TFM b	1111 11bb	SF ← RAM[HL] <sub>b</sub> '	1	1	-	-	*
TFP p,b	0110 1101 00bb pppp	SF ← PORT[p] <sub>b</sub> '	2	2	-	-	*
TFPL	0110 0001	SF ← PORT[LR <sub>3-2</sub> +4]LR <sub>1-0</sub> '	1	2	-	-	*
TT y,b	0110 1100 10bb yyyy	SF ← RAM[y] <sub>b</sub>	2	2	-	-	*
TTP p,b	0110 1101 10bb pppp	SF ← PORT[p] <sub>b</sub>	2	2	-	-	*

### (9) Subroutine

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
LCALL a	0100 0aaa aaaa aaaa	STACK[SP] ← PC, SP ← SP - 1, PC ← a	2	2	-	-	-
SCALL a	1110 nmn	STACK[SP] ← PC, SP ← SP - 1, PC ← a, a = 8n + 6 (n = 1~15), 0086h (n = 0)	1	2	-	-	-
RET	0100 1111	SP ← SP + 1, PC ← STACK[SP]	1	2	-	-	-

### (10) Input/output

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
INA p	0110 1111 0100 pppp	Acc ← PORT[p]	2	2	-	Z	Z'
INM p	0110 1111 1100 pppp	RAM[HL] ← PORT[p]	2	2	-	-	Z'
OUT #k,p	0100 1010 kkkk pppp	PORT[p] ← k	2	2	-	-	1
OUTA p	0110 1111 000p pppp	PORT[p] ← Acc	2	2	-	-	1
OUTM p	0110 1111 100p pppp	PORT[p] ← RAM[HL]	2	2	-	-	1

Preliminary

**(11) Flag manipulation**

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
TFCFC	0101 0011	SF←CF', CF←0	1	1	0	-	*
TTCFS	0101 0010	SF←CF, CF←1	1	1	1	-	*
TZS	0101 1011	SF←ZF	1	1	-	-	*

**(12) Interrupt control**

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
CIL r	0110 0011 11rr rrrr	IL←IL & r	2	2	-	-	1
DICIL r	0110 0011 10rr rrrr	EIF←0, IL←IL&r	2	2	-	-	1
EICIL r	0110 0011 01rr rrrr	EIF←1, IL←IL&r	2	2	-	-	1
EXAE	0111 0101	MASK↔Acc	1	1	-	-	1
RTI	0100 1101	SP←SP+1, FLAG.PC ←STACK[SP], EIF ←1	1	2	*	*	*

**(13) CPU control**

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
NOP	0101 0110	no operation	1	1	-	-	-

**(14) Timer/Counter & Data pointer & Stack pointer control**

Mnemonic	Object code ( binary )	Operation description	Byte	Cycle	Flag		
					C	Z	S
LDADPL	0110 1010 1111 1100	Acc←[DP] <sub>L</sub>	2	2	-	Z	1
LDADPM	0110 1010 1111 1101	Acc←[DP] <sub>M</sub>	2	2	-	Z	1
LDADPH	0110 1010 1111 1110	Acc←[DP] <sub>H</sub>	2	2	-	Z	1
LDASP	0110 1010 1111 1111	Acc←SP	2	2	-	Z	1
LDATAL	0110 1010 1111 0100	Acc←[TA] <sub>L</sub>	2	2	-	Z	1
LDATAM	0110 1010 1111 0101	Acc←[TA] <sub>M</sub>	2	2	-	Z	1
LDATAH	0110 1010 1111 0110	Acc←[TA] <sub>H</sub>	2	2	-	Z	1
LDATBL	0110 1010 1111 1000	Acc←[TB] <sub>L</sub>	2	2	-	Z	1
LDATBM	0110 1010 1111 1001	Acc←[TB] <sub>M</sub>	2	2	-	Z	1
LDATBH	0110 1010 1111 1010	Acc←[TB] <sub>H</sub>	2	2	-	Z	1
STADPL	0110 1001 1111 1100	[DP] <sub>L</sub> ←Acc	2	2	-	-	1
STADPM	0110 1001 1111 1101	[DP] <sub>M</sub> ←Acc	2	2	-	-	1
STADPH	0110 1001 1111 1110	[DP] <sub>H</sub> ←Acc	2	2	-	-	1
STASP	0110 1001 1111 1111	SP←Acc	2	2	-	-	1
STATAL	0110 1001 1111 0100	[TA] <sub>L</sub> ←Acc	2	2	-	-	1
STATAM	0110 1001 1111 0101	[TA] <sub>M</sub> ←Acc	2	2	-	-	1
STATAH	0110 1001 1111 0110	[TA] <sub>H</sub> ←Acc	2	2	-	-	1
STATBL	0110 1001 1111 1000	[ TB] <sub>L</sub> ←Acc	2	2	-	-	1
STATBM	0110 1001 1111 1001	[TB] <sub>M</sub> ←Acc	2	2	-	-	1
STATBH	0110 1001 1111 1010	[TB] <sub>H</sub> ←Acc	2	2	-	-	1

\* This specification are subject to be changed without notice.

**Preliminary**

\*\*\*\*\* SYMBOL DESCRIPTION

Symbol	Description	Symbol	Description
HR	H register	LR	L register
PC	Program counter	DP	Data pointer
SP	Stack pointer	STACK[SP]	Stack specified by SP
A <sub>CC</sub>	Accumulator	FLAG	All flags
CF	Carry flag	ZF	Zero flag
SF	Status flag	EI	Enable interrupt register
IL	Interrupt latch	MASK	Interrupt mask
PORT[p]	Port ( address : p )	TA	Timer/counter A
TB	Timer/counter B	RAM[HL]	Data memory (address : HL )
RAM[x]	Data memory (address : x )	ROM[DP] <sub>L</sub>	Low 4-bit of program memory
ROM[DP] <sub>H</sub>	High 4-bit of program memory	[DP] <sub>L</sub>	Low 4-bit of data pointer register
[DP] <sub>M</sub>	Middle 4-bit of data pointer register	[DP] <sub>H</sub>	High 4-bit of data pointer register
[TA] <sub>L</sub> ([TB] <sub>L</sub> )	Low 4-bit of timer/counter A (timer/counter B) register	[TA] <sub>M</sub> ([TB] <sub>M</sub> )	Middle 4-bit of timer/counter A (timer/counter B) register
[TA] <sub>H</sub> ([TB] <sub>H</sub> )	High 4-bit of timer/counter A (timer/counter B) register	LR <sub>1-0</sub>	Contents of bit assigned by bit 1 to 0 of LR
LR <sub>3-2</sub>	Bit 3 to 2 of LR	a <sub>5-0</sub>	Bit 5 to 0 of destination address for branch instruction
PC <sub>12-6</sub>	Bit 12 to 6 of program counter	←	Transfer
↔	Exchange	+	Addition
-	Substraction	&	Logic AND
	Logic OR	^	Logic XOR
'	Inverse operation	.	Concatenation
#k	4-bit immediate data	x	8-bit RAM address
y	4-bit zero-page address	p	4-bit or 5-bit port address
b	Bit address	r	6-bit interrupt latch