Preliminary

GENERAL DESCRIPTION

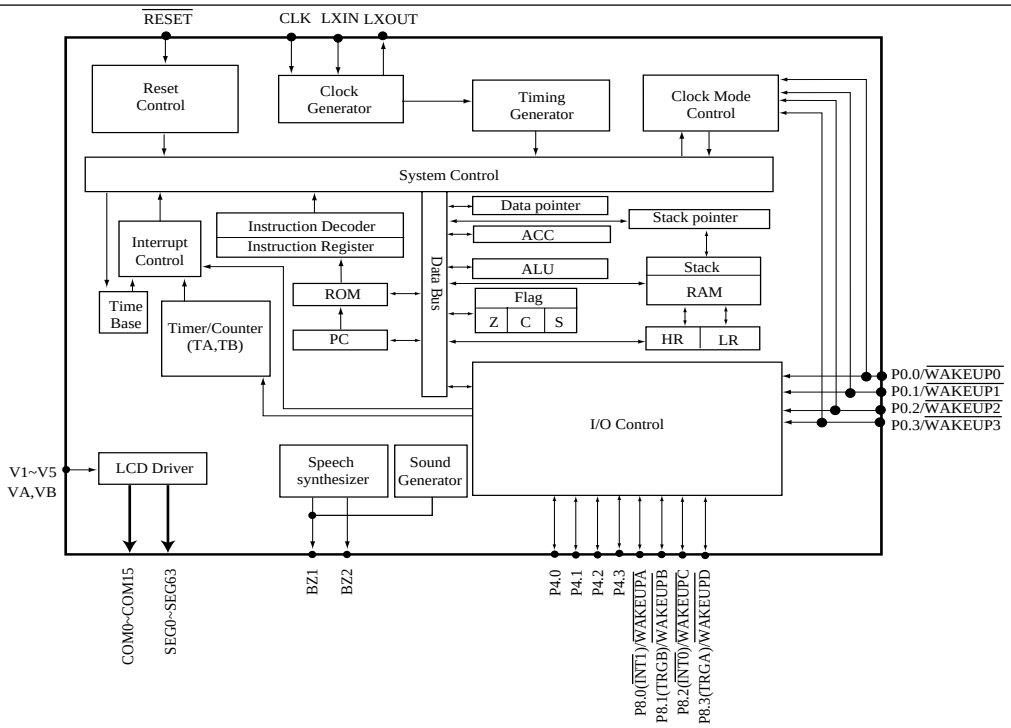
EM73A88A is an advanced single chip CMOS 4-bit micro-controller. It contains 16K-byte ROM, 500-nibble RAM, 4-bit ALU, 13-level subroutine nesting, 22-stage time base, two 12-bit timer/counters for the kernel function. EM73A88A also equipped with 6 interrupt sources, 3 I/O ports (including 1 input port and 2 bidirection ports), LCD display (64x16), built-in sound generator and speech synthesizer can direct drive speaker.
It's low power consumption and high speed feature are further strengten with DUAL, SLOW, IDLE and STOP operation mode for optimized power saving.

FEATURES

- Operation voltage : 2.2V to 4.8V.
- Clock source : Dual clock system. Low-frequency oscillator is 32 KHz Crystal or RC oscillator and high-frequency oscillator is a built-in internal oscillator (4.6 MHz).
- Instruction set : 107 powerful instructions.
- Instruction cycle time : 1.7μs for 4.6M Hz (high speed clock).
  244μs for 32768 Hz (low speed clock).
- ROM capacity : 16K x 8 bits.
- RAM capacity : 500 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 ports (P4, P8). IDLE/STOP release function for P8(0..3) is available by mask option.
- 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 interrupt.
- LCD driver : 64x16 dots, 1/16 duty, 1/5 bias with voltage multiplier.
- Sound effect : Tone generator and random generator.
- Speech synthesizer : 448K speech data ROM (use as 448K 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 109 pins.

Preliminary

FUNCTION BLOCK DIAGRAM



PIN DESCRIPTIONS

Symbol	Pin-type	Function
V_{DD}, V_{DD2}		Power supply (+)
V_{SS}		Power supply (-)
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 or RC osc connecting pin for low speed clock source.
LXOUT	OSC-B	Crystal osc 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
P4(0..3)	I/O-O	4-bit bidirection I/O port with high current source. mask option : open-drain push-pull, high current PMOS push-pull, low current PMOS
P8.0(INT1)/WAKEUPA P8.2(INT0)/WAKEUPC	I/O-L	2-bit bidirection I/O port with external interrupt sources input and IDLE /STOP releasing function mask option : wakeup enable, push-pull wakeup disable, push-pull wakeup disable, open-drain
P8.1(TRGB)/WAKEUPB P8.3(TRGA)/WAKEUPD	I/O-L	2-bit bidirection I/O port with time/counter A,B external input and IDLE /STOP releasing function

Preliminary

Symbol	Pin-type	Function
		mask option : wakeup enable, push-pull wakeup disable, push-pull wakeup disable, open-drain
BZ1		Tone / Speech PWM / D/A output pin
BZ2		Tone / Speech PWM output pin
V1, V2, V3, V4, V5, VA, VB		LCD bias pins
COM0~COM15		LCD common output pins
SEG0~SEG63		LCD segment output pins
TEST		Tie Vss as package type, no connecting as COB type

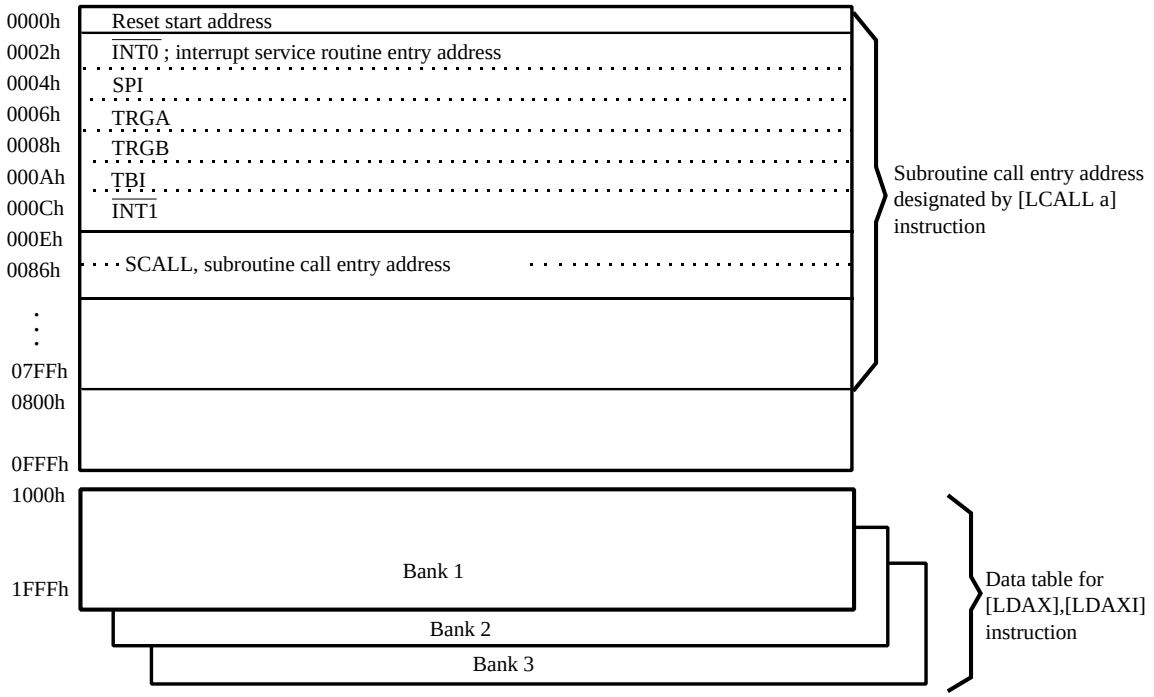
FUNCTION DESCRIPTIONS

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, the bank0 and bank1 of program ROM will be selected. P3(1..0)=01B, the bank0 and bank2 will be selected.

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

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 :       :
```

```
            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]$_L$**
LDAXI **Acc ← ROM[DP]$_H$,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"

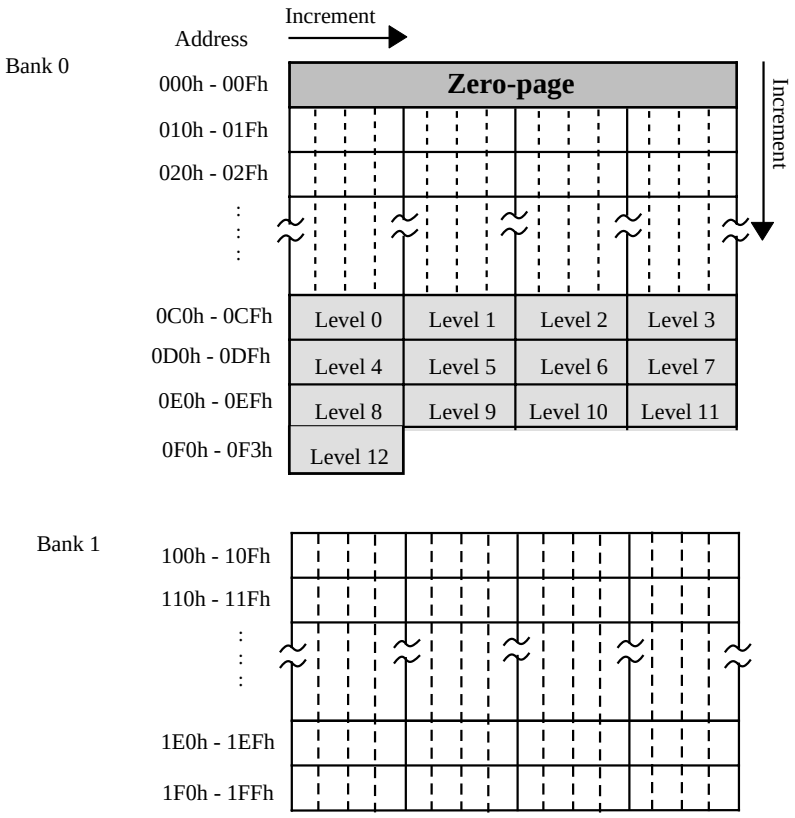
PROGRAM EXAMPLE: Read out the ROM code of address 1777h by table-look-up instruction.

```
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 (500-nibble)

A total 500 - nibble data RAM is available from address 000 to 1FFh
Data RAM includes the zero page region, stacks and data areas.

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 ; $RAM[0Eh]_2 \leftarrow 0$

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.

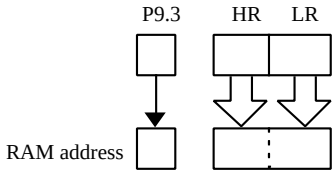
ADDRESSING MODE

The 500 nibble data memory consists of two banks (bank 0 and bank 1). There are 244x4 bits (address 000h~0F3h) in bank 0 and 256x4 bits (address 100h~1FFh) in bank 1.

Preliminary

The bank is selected by P9.3. When P9.3 is cleared to "0", the bank 0 is selected. When P9.3 is set to "1", the bank 1 is selected.
The Data Memory consists of three Address mode, namely -

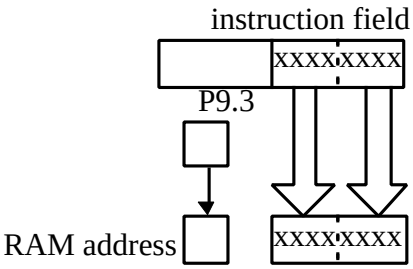
(1) Indirect addressing mode:
The address in the bank is specified by the HL registers.



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

```
SEP  P9,3   ; P9.3← 1
LDL #3h     ; LR← 3
LDH #4h     ; HR← 4
LDAM        ; Acc← RAM[134h]
CLP P9,3    ; P9.3← 0
LDL #2h     ; LR← 2
LDH #3h     ; HR← 3
STAM        ; RAM[023h]← Acc
```

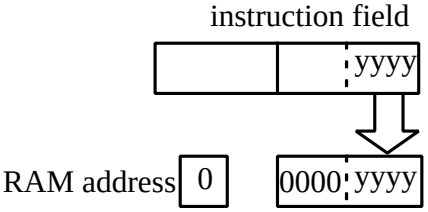
(2) Direct addressing mode:
The address in the bank is directly specified by 8 bits code of the second byte in the instruction field.



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

```
SEP  P9,3   ; P9.3← 1
LDA 43h     ; Acc← RAM[143h]
CLP P9,3    ; P9.3← 0
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
```

Preliminary

PROGRAM COUNTER (16K ROM)

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 $_{12\text{-}6.a}$ (branch condition satisified)

PC | Hold original PC value+1 | a a a a a a |

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

PC | Original PC value + 1 |

LBR a
Object code: 1100 aaaa aaaa aaaa
Condition: SF=1; PC ← PC $_{12.a}$ (branch condition satisified)

PC | Hold +2 | a a a a a a a a a a a a a |

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

PC | Original PC value + 2 |

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 satisified)

PC | a a a a a a a a a a a a a |

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

PC | Original PC value + 3 |

(2) Subroutine instruction:

SCALL a
Object code: 1110 nnnn
Condition : PC ← a ; a=8n+6 ; n=1..Fh ; a=86h, n=0

PC | 0 0 0 0 0 a a a a a a a a |

LCALL a
Object code: 0100 0aaa aaaa aaaa
Condition: PC ← a

PC	0	0	a	a	a	a	a	a	a	a	a	a	a

RET
Object code: 0100 1111
Condition: PC ← STACK[SP]; SP + 1

PC	The return address stored in stack

RT I
Object code: 0100 1101
Condition : FLAG. PC ← STACK[SP]; EI ← 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 :

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

PC	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	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

$\overline{\text{INT1}}$ (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

(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 arithematic, logic and comparative opertion.., 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 arithematic, 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 occures, 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 occures, the CF is "0", likewise, if there is no borrow-in, the CF is "1".

c. Comparision: CF as a borrow-in indicator for Comparision 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 TTSFC 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 satisified, likewise, when SF = 0, branch condition is unsatisified.

Preliminary

PROGRAM EXAMPLE:

Check following arithematic operation for CF, ZF, SF

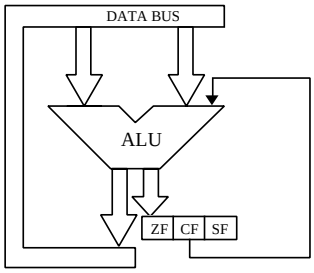
	CF	ZF	SF
LDIA #00h;	-	1	1
LDIA #03h;	-	0	1
ADDA #05h;	-	0	1
ADDA #0Dh;	-	0	0
ADDA #0Eh;	-	0	0

ALU

The arithematic 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 arithematic operation functions, including Addition, Subtraction and Rotaion.



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".

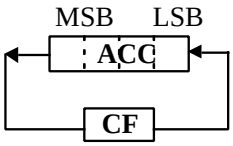
Preliminary

EXAMPLE:

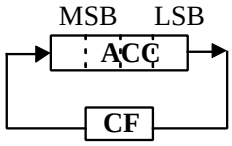
Operation	Carry	Zero
8-4=4	1	0
7-F= -8(1000)	0	0
9-9=0	1	1

(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.

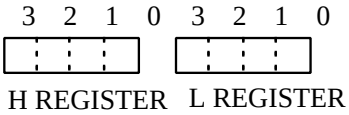


PROGRAM EXAMPLE: To rotate Acc clockwise (right) and shift a "1" into the MSB bit of Acc.
TTCFS; CF ← 1
RRCA; rotate Acc right and shift CF=1 into MSB.

HL REGISTER

HL register are two 4-bit registers, they are used as a pair of pointer for the RAM memoryaddress. 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)

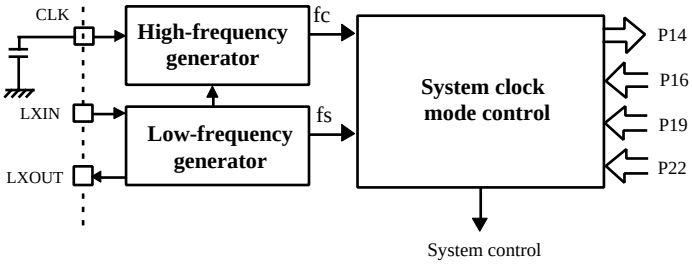
Data pointer is a 12-bit register that stores the ROM address can indicating the ROM code data specified by user (refer to data ROM).

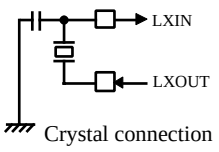
CLOCK AND TIMING GENERATOR

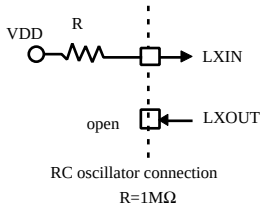
The clock generator is supported by a dual clock system. The high-frequency oscillator is internal oscillator, the working frequency is 4.6 MHz. The low-frequency oscillator may be sourced from crystal or RC osc, the working frequency is 32 KHz.

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.



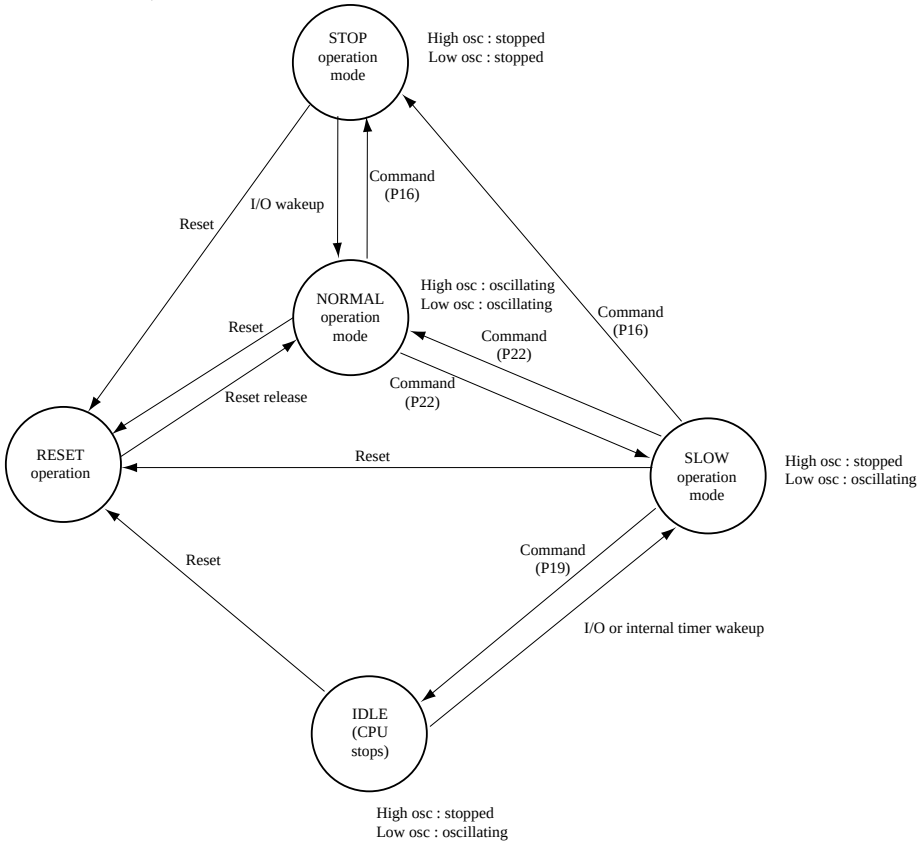




Preliminary

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. EM73A88A 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, sound gen.	8 / fc
SLOW	Low frequency	Low frequency clock	LCD	8 / fs
IDLE	Low frequency	CPU stops	LCD	-
STOP	None	CPU stops	All disable	-

DUAL OPERATION MODE

The 4-bit µc 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 IDEL 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 : 0000

*	SOM

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

P14 3 2 1 0 Initial value : *000

*	WKS	LFS	CPUS

LFS	Low-frequency status
0	LXIN source is not stable
1	LXIN source is stable

CPUS	CPU status
0	DUAL operation mode
1	SLOW operation mode

WKS	Wakeup status
0	Wakeup not by internal timer
1	Wakeup by internal timer

Port14 is the status register for CPU. P14.0 (CPU status) and P14.1 (Low-frequency status) are read-only bits. 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.

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. Sound generator is 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

*	IDME	SIDR

IDME	Enable IDLE mode
1	Enable IDLE mode
0	no function

SIDR	Select IDLE releasing condition
0 0	P0(0..3), P8(0..3) pin input
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 and P8(0..3)/WAKEUP A..D).

LCD display and sound generator are disabled in the STOP operation mode.

Preliminary

P16 3 2 1 0 Initial value : 0000

*	SPME	SWWT

SPME	Enable STOP mode
1	Enable STOP mode
0	no function

SWWT	Set wake-up warm-up time
0 0	2^{14}/LXIN
0 1	2^{10}/LXIN
1 0	2^{12}/LXIN
1 1	no function

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	DUAL operation mode	SLOW operation mode
0 0 x x	Interrupt disable	Interrupt disable
0 1 0 0	Interrupt frequency LXIN / 2^3 Hz	Reserved
0 1 0 1	Interrupt frequency LXIN / 2^4 Hz	Reserved
0 1 1 0	Interrupt frequency LXIN / 2^5 Hz	Reserved
0 1 1 1	Interrupt frequency LXIN / 2^{14} Hz	Interrupt frequency LXIN / 2^{14} Hz
1 1 0 0	Interrupt frequency LXIN / 2^1 Hz	Reserved
1 1 0 1	Interrupt frequency LXIN / 2^6 Hz	Interrupt frequency LXIN / 2^6 Hz
1 1 1 0	Interrupt frequency LXIN / 2^8 Hz	Interrupt frequency LXIN / 2^8 Hz
1 1 1 1	Interrupt frequency LXIN / 2^{10} Hz	Interrupt frequency LXIN / 2^{10} Hz
1 0 x x	Reserved	Reserved

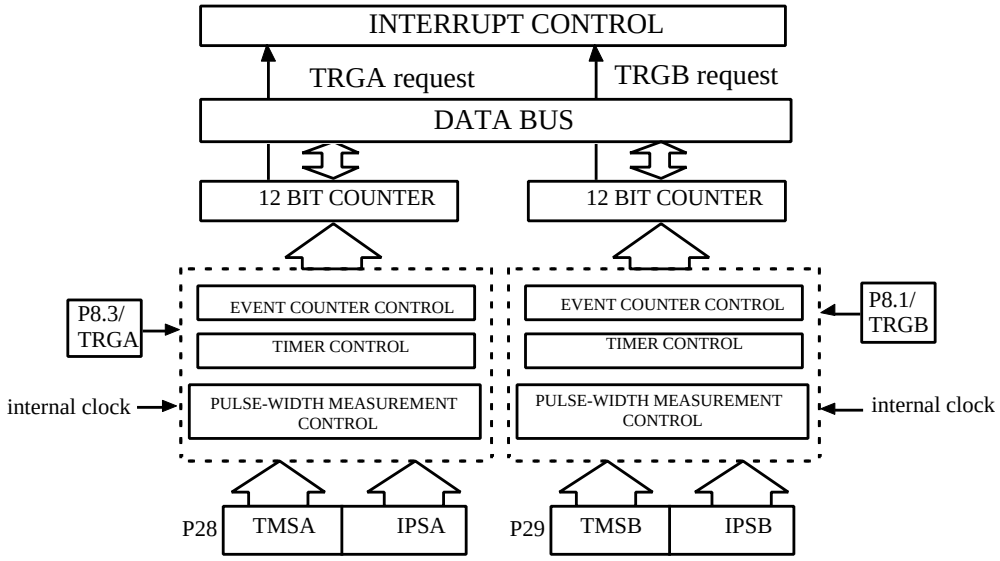
TIMER / COUNTER (TIMERA, TIMERB)

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.

Port 28

3	2	1	0
TMSA			IPSA

Initial state: 0000

Port 29

3	2	1	0
TMSB			IPSB

Initial state: 0000

TIMER/COUNTER MODE SELECTION	
TMSA (B)	Function description
0 0	Stop
0 1	Event counter mode
1 0	Timer mode
1 1	Pulse width measurement mode

INTERNAL PULSE-RATE SELECTION		
IPSA(B)	DUAL mode	SLOW mode
0 0	LXIN/2^3 Hz	Reserved
0 1	LXIN/2^7 Hz	LXIN/2^7 Hz
1 0	LXIN/2^{11}Hz	LXIN/2^{11}Hz
1 1	LXIN/2^{15}Hz	LXIN/2^{15}Hz

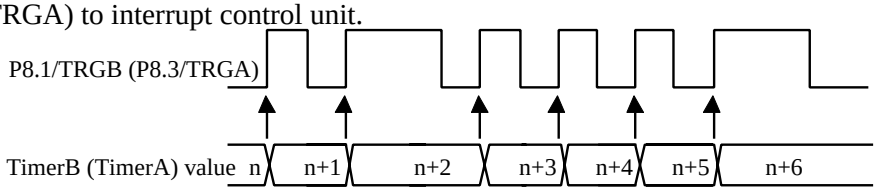
Preliminary

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.

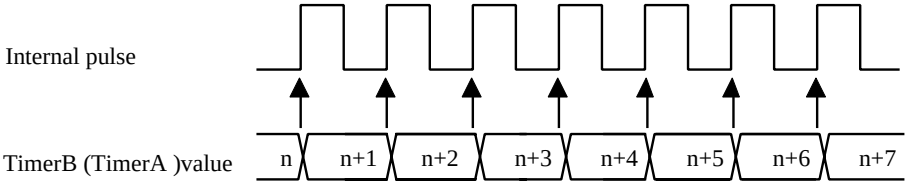


PROGRAM EXAMPLE: Enable timerA with P28

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

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 LXlN=32KHz

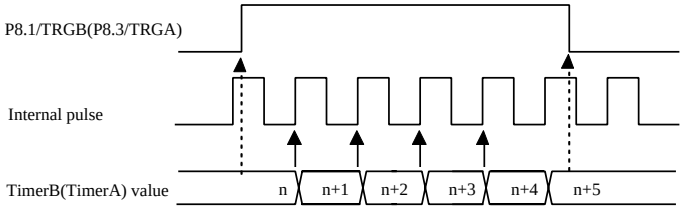
```
LDIA #0100B;
EXAE;     enable mask 2
EICIL 110111b; interrupt latch ←0, enable EI
LDIA #0Ah;
STATAL;
LDIA #00h;
STATAM;
LDIA #0Fh;
STATAH;
LDIA #1000B;
OUTA P28; enable timerA with internal pulse rate: LXIN/2³ Hz
```

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

Preliminary

Under the pulse width measurement mode, the counter is incresed 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.

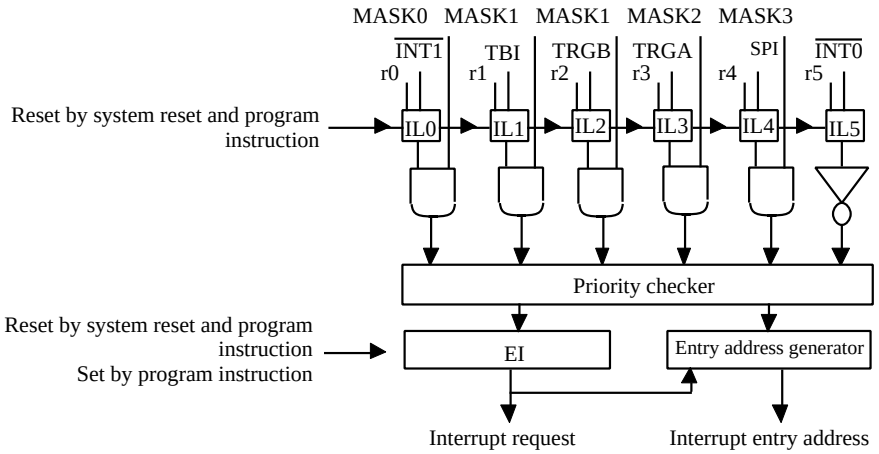
```
LDIA #1100b;
OUTA P28; Enable timerA with pulse width measurement mode.
```

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 Enable condition	ProgramROM entry address
External	External interrupt($\overline{INT0}$)	1	IL5	EI=1	002h
Internal	speech end interrupt (SPI)	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.

Preliminary

EI : Enable interrupt Flip-Flop may promit 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.

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 "1100b"
EICIL 010111B ; enable interrupt F.F. and clear IL3 and IL5
```

LCD DRIVER

It can directly drive the liquid crystal display (LCD) and has 64 segments, 16 commons output pins. There are total 64x16 dots can be display. The V1~V5 are the LCD bias voltage input pins.

(1) LCD driver control command register:

Port27 3 2 1 0 Initial value: 0000

LDC	*	*

LCD DISPLAY CONTROL	
LDC	Function description
0 0	LCD display disable
0 1	Blanking
1 0	no function
1 1	LCD display enable

* : Don't care.

P27 is the LDC driver control command register. The initial value is 0000.
When LDC (bit2 and bit3 of P27) is set to "00", the LCD display is disabled.
When LDC is set to "01", the LCD is blanking, the COM pins are inactive and the SEG pins output the display data continuously.
When LDC is set to "11", the LCD display is enabled.

(2) LCD display data area:

The LCD display data is stored in the display data area of the data memory (RAM). The LCD display data area is as illustrated below :

Preliminary

The display data from the display data area are automatically read out and send to the LCD driver directly by the hardware. Therefore, the display patterns can be changed only by overwritting the contents of the display data area through software.

The dispaly memory area that is not used to store the LCD display data could be used as the ordinary data memory.

LCD display data area :

Bank1

P9.3=1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
100-10Fh	COM0															
110-11Fh	COM1															
120-12Fh	COM2															
130-13Fh	COM3															
140-14Fh	COM4															
150-15Fh	COM5															
160-16Fh	COM6															
170-17Fh	COM7															
180-18Fh	COM8															
190-19Fh	COM9															
1A0-1AFh	COM10															
1B0-1BFh	COM11															
1C0-1CFh	COM12															
1D0-1DFh	COM13															
1E0-1EFh	COM14															
1F0-1FFh	COM15															

SEG0 SEG1 SEG2 SEG3 SEG4 SEG5 SEG6 SEG7 SEG8 SEG9 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

P26 is the start address register of LCD common pin.

Port26 3 2 1 0 Initial value: 0000

3	2	1	0
CSA			

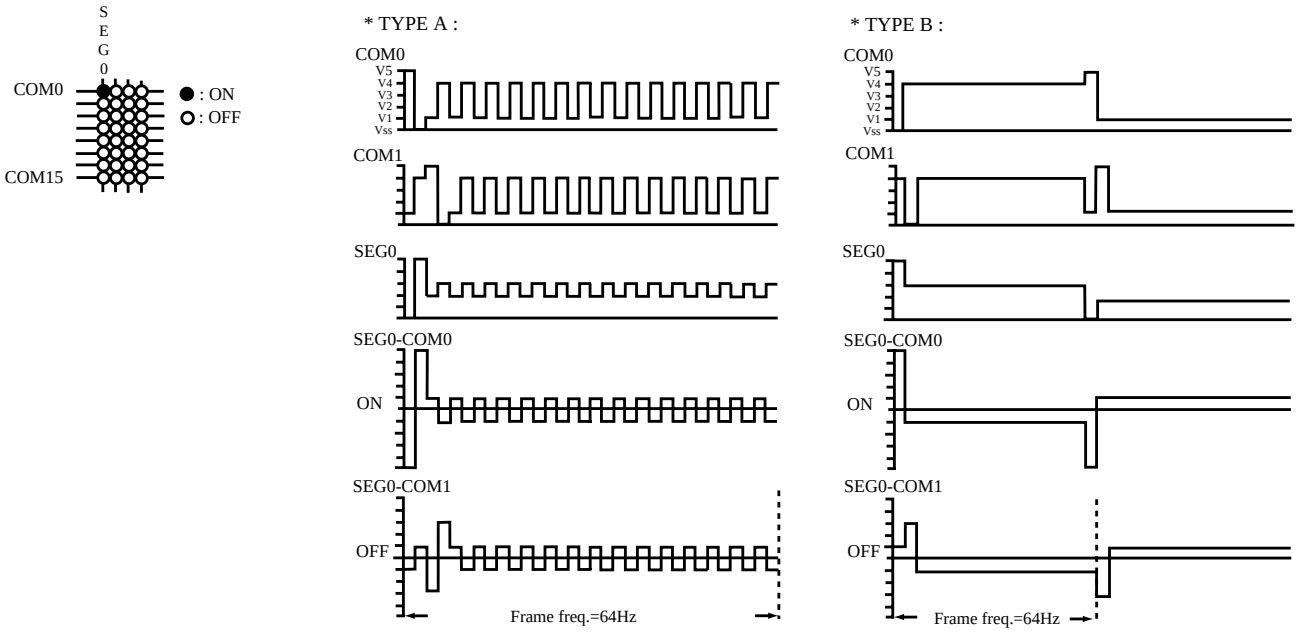
CSA	Common start address register															
	RAM															
	100-109h	110-119h	120-129h	130-139h	140-149h	150-159h	160-169h	170-179h	180-189h	190-199h	1A0-1A9h	1B0-1B9h	1C0-1C9h	1D0-1D9h	1E0-1EF9h	1F0-1F9h
0000	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15
0001	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14
0010	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13
0011	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12
0100	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11
0101	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10
0110	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9
0111	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8
1000	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6	COM7
1001	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5	COM6
1010	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4	COM5
1011	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3	COM4
1100	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2	COM3
1101	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1	COM2
1110	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0	COM1
1111	COM1	COM2	COM3	COM4	COM5	COM6	COM7	COM8	COM9	COM10	COM11	COM12	COM13	COM14	COM15	COM0

PROGRAM EXAMPLE:

```
LDIA    #0000B
OUTA    P26
LDIA    #1100B  ; LCD display enable
OUTA    P27
LDIA    #1010B  ; store 1010B to RAM[101h]
SEP     P9,3
STA     01H
```

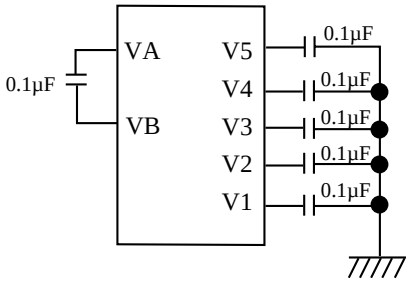
Preliminary

(3) LCD waveform : (1/5 bias)

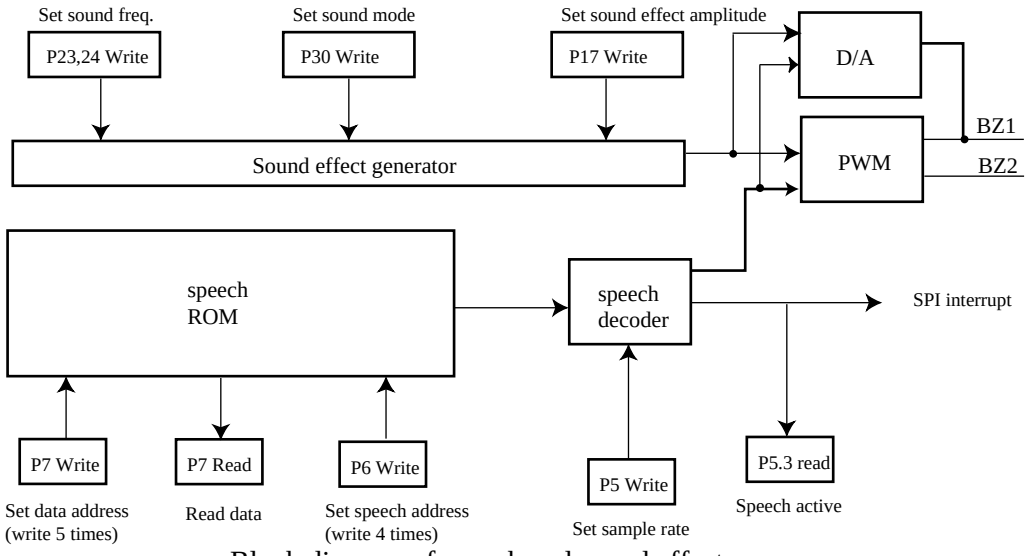


(4) LCD drive voltage :

• The LCD bias voltage is supplied by voltage multiplier. The application circuit is illustated as below :



SPEECH SYNTHESIZER



Block diagram of speech and sound effect

Preliminary

EM73A88A speech synthesizer operates as following :

1. Send the speech start address to the address latch by writing P6 four times.
2. Choose the sampling rate, enable the speech synthesizer by writing P5.
3. The ROM address counters send the ROM address A6 .. A17 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 P5.3.

SPEECH SYNTHESIZER CONTROL

Speech sample rate control register (P5 write) :

3	2	1	0
	SR		

Initial value : *111

SR	Sample rate selection		Sample rate
000	PWM on	CLK/64/1/3	24K
001		CLK/64/1/4	18K
010		CLK/64/2/3	12K
011		CLK/64/2/4	9K
100		CLK/64/3/3	8K
101		CLK/64/3/4	6K
111	PWM off		

port 5 -- initialization is "*111".
port 6 -- initialization is pointed to the low-nibble of start address latch.

CLK=4.6 MHz

Speech active flag (P5 read) :

3	2	1	0
ACT	*	*	*

Initial value : 0***

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.

Speech start address register (P6 write) :

3	2	1	0
Port 6			

Initial value : 1111

P6L1				P6L2				P6L3				P6L4			
A9	A8	A7	A6	A13	A12	A11	A10	A17	A16	A15	A14	-	-	-	A18

Send the speech start address to the speech synthesizer by writing P6 four times. There is a pointer counter to point the address latch (P6L1, P6L2, P6L3, P6L4). It will increase one when write P6. So, the first time writing P6 to P6L1, the second time is P6L2, the third time is P6L3, the fourth time is P6L4 and the fifth time is P6L1 latch again, ... etc. The pointer counter point to P6L1 when CPU is reset or P5 is writen.
In the NORMAL operation mode, the speech synthesizer is available. In the other operation modes, it is disable.

Preliminary

PROGRAM EXAMPLE:

```
SP_ADR1   EQU     1234H       ; the start address of the speech section
          :
          LDIA    #SP_ADR1
          OUTA    P6
          LDIA    #SP_ADR1/10H
          OUTA    P6
          LDIA    #SP_ADR1/100H
          OUTA    P6
          LDIA    #SP_ADR1/1000H
          OUTA    P6
; set sample rate & start speech
          LDIA    #0010B
          OUTA    P5
; wait speech end
WAIT      TTP     P5,3        ; get speech active flag
          B       WAIT
```

USING SPEECH ROM AS DATA ROM

The speech ROM can be used for speech synthesizer and for data ROM simutaneously.
First, write initial address to P7 (five times), and after four cycles, you can read P7 to get data, and address counter increases one automatically.The following read operations must be at an internval of instruction cycles which are more than 3.
The read operation should be all done before you leave normal mode and change to slow mode.

Get speech ROM data (P7 read) :

3	2	1	0
Port 7			

Set speech ROM address (P7 write) :

3	2	1	0
Port 7			

P7L1			
A3	A2	A1	A0

P7L2			
A7	A6	A5	A4

P7L3			
A11	A10	A9	A8

P7L4			
A15	A14	A13	A12

P7L5			
-	A18	A17	A16

PROGRAM EXAMPLE:

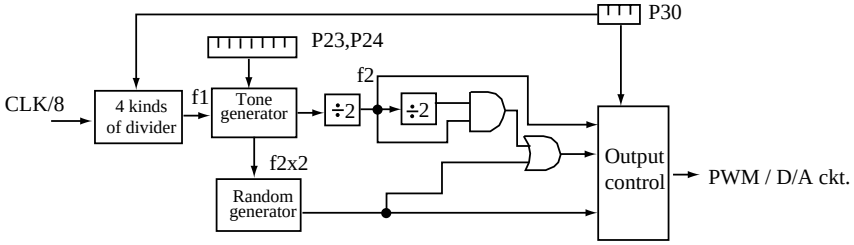
```
D_ADR1    EQU     12345H      ; the start address of the speech ROM
          :
          LDIA    #D_ADR1
          OUTA    P7
          LDIA    #D_ADR1/10H
          OUTA    P7
          LDIA    #D_ADR1/100H
          OUTA    P7
          LDIA    #D_ADR1/1000H
          OUTA    P7
          LDIA    #D_ADR1/10000H
          OUTA    P7
```

```
        NOP
        NOP                                  } 4 cycles
        NOP
        NOP
; READ DATA
        INA         P7          ; read D_ADR1
        STA         TEMP
        NOP                      } 3 cycles
        INA         P7          ; read D_ADR1+1
```

MELODY (SOUND EFFECT) CONTROL

One channel melody/sound effect output, controlled by port 23, 24, 17, and 30.

There is a built-in sound effect. It includes the tone generator and random generator. The tone generator is a binary down counter and the random generator is a 9-bit liner feedback shift register.



Sound effect command register (P30)

There are 4 kinds of basic frequency for sound generator which can be selected by P30. The output of sound effect is tone and random combination.

Port30

3	2	1	0
BFREQ		SMODE	

Initial value : 0000

BFREQ	Basic frequency (f1) select
0 0	CLK/16
0 1	CLK/32
1 0	CLK/64
1 1	Reserved

(CLK=4.6MKz)

SMODE	Sound generator mode
0 0	Disable
0 1	Tone output
1 0	Random output
1 1	Tone+random output

Tone frequency register (P23, P24)

The 8-bit tone frequency register is P24 and P23. The tone frequency will be changed when user output the different data to P23. Thus, the data must be output to P24 before P23 when users want to change the 8-bit tone frequency (TF).

Port24

3	2	1	0
Higher nibble register			

Port23

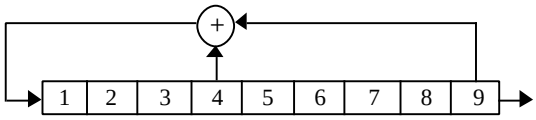
3	2	1	0
Lower nibble register			

Initial value : 1111 1111

** f1=CLK/2^X, f2=f1/(TF+1)/2, TF=1~255, TF-0
** Example : CLK=4.6 MHz, BFREQ=10, TF=00110001B.
⇒ f1=143.75K Hz, f2=143.75K Hz/50/2=1430 Hz

Random generator

$f(x)=x^9+x^4+1$



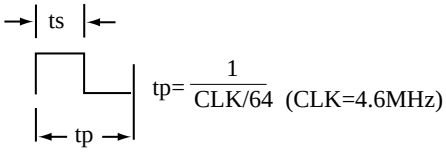
Volume control register (P17)

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

Port17 Initial value : 1111

3	2	1	0
VCR			

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:

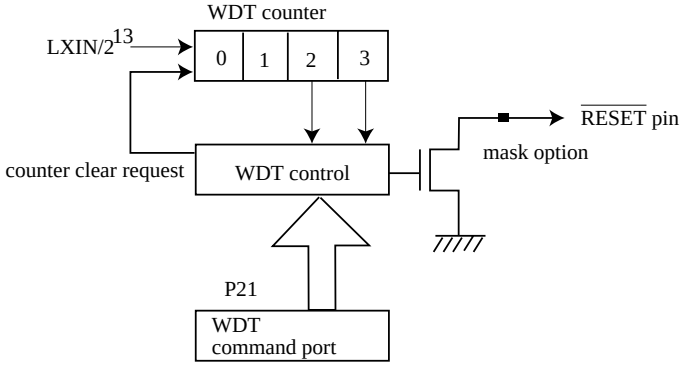
```
LDIA    #1001B      ; basic frequency : CLK/32, tone output
OUTA    P30
LDIA    #0111B      ; volume control
OUTA    P17
LDIA    #0011B      ; 1430 Hz tone output
OUTA    P24
LDIA    #0001B
OUTA    P23
```

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).



Preliminary

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	*	*	WDT

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

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

PROGRAM EXAMPLE

To enable WDT with 7 x 2^{13}/LXIN detection time.

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

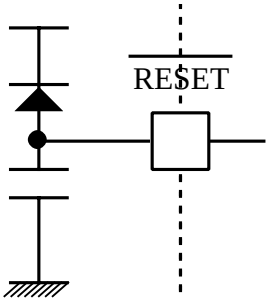
RESETTING FUNCTION

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, 14, 16, 19, 21, 22, 25, 26, 27, 28, 29, 30	00h
P5	07h
P0, 4, 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}.



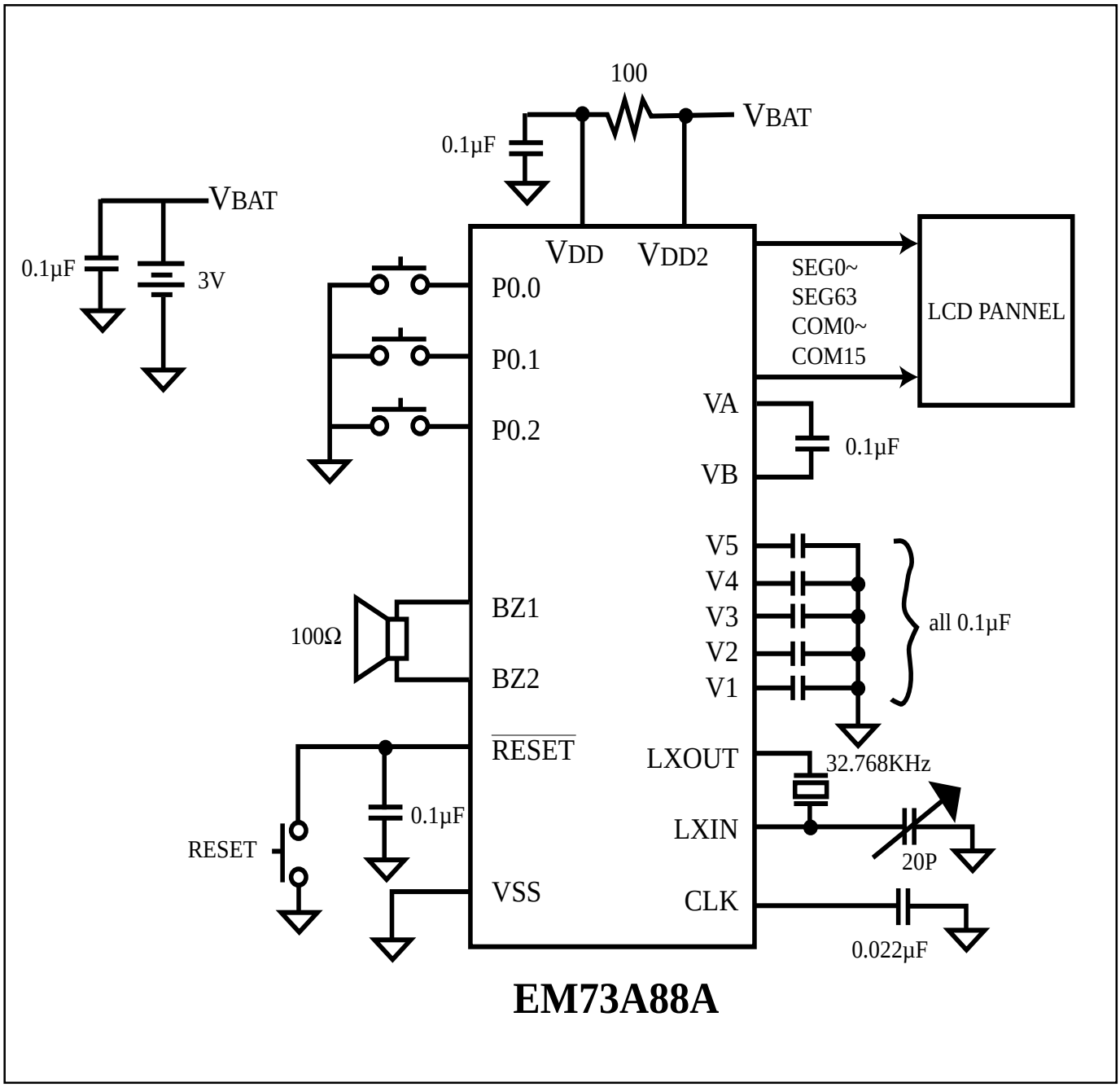
Preliminary

EM73A88A I/O PORT DESCRIPTION :

Port	Input function		Output function		Note
0	E	Input port , wakeup function			
1		--		--	
2		--		--	
3		--	I	P3(1..0) : ROM bank selection	
4	E	Input port	E	Output port	
5	I	P5.3 : Speech active signal (ACT)	I	Speech sample rate register	
6		--	I	Speech start address register	
7	I	DATA ROM data	I	Data start address register	
8	E	Input port, wakeup function, external interrupt input	E	Output port	
9		--	I	P9.3 : RAM bank selection	
10		--		--	
11		--		--	
12		--		--	
13		--		--	
14	I	CPU status register		--	
15		--		--	
16			I	STOP mode control register	
17			I	Sound effect volume control register	
18				--	
19			I	IDLE mode control register	
20				--	
21			I	WDT control register	
22			I	DUAL/SLOW mode control register	
23			I	Sound effect frequency register	low nibble
24			I	Sound effect frequency register	high nibble
25			I	Timebase control register	
26			I	LCD common start address register	
27			I	LCD control register	
28			I	Timer/counter A control register	
29			I	Timer/counter B control register	
30			I	Sound effect command register	
31				--	

Preliminary

APPLICATION CIRCUIT



Preliminary

ABSOLUTE MAXIMUM RATINGS

Items	Sym.	Ratings	Conditions
Supply Voltage	V_{DD}	-0.5V to 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°C
Operating Temperature	T_{OPR}	0°C to 50°C	
Storage Temperature	T_{STG}	-55°C to 125°C	

RECOMMANDED OPERATING CONDITIONS

Items	Sym.	Ratings	Condition
Supply Voltage	V_{DD}	2.2V to 4.8V	
Input Voltage	V_{IH}	0.90xV_{DD} to V_{DD}	
	V_{IL}	0V to 0.10xV_{DD}	
Operating Frequency	F_C	4.6MHz	CLK
	Fs	32KHz	LXIN,LXOUT

DC ELECTRICAL CHARACTERISTICS (V_{DD}=3±0.3V, V_{SS}=0V, T_{OPR}=25°C)

Parameters	Sym.	Min.	Typ.	Max.	Unit	Conditions
Supply current	I_{DD}	-	0.5	1.2	mA	V_{DD}=3.3V,no load,DUAL mode,Fs=32KHz, Fc=4.6MHz
		-	25	38	µA	V_{DD}=3.3V,SLOW mode,LCD on
		-	20	33	µA	V_{DD}=3.3V,IDLE mode,LCD on
		-	7	12	µA	V_{DD}=3.3V,IDLE mode,LCD off
		-	0.1	1	µA	V_{DD}=3.3V, STOP mode
Hysteresis voltage	V_{HYS+}	0.50V_{DD}	-	0.75V_{DD}	V	$\overline{RESET}$, P0, P8
	V_{HYS-}	0.20V_{DD}	-	0.40V_{DD}	V	
Input current	I_{IH}	-	-	±1	µA	P0, $\overline{RESET}$, V_{DD}=3.3V,V_{IH}=3.3/0V
		-	-	±1	µA	Open-drain, V_{DD}=3.3V,V_{IH}=3.3/0V
	I_{IL}	-	-250	-500	µA	Normal current push-pull,V_{DD}=3.3V,P4(low), P8
Output voltage	V_{OH}	2.4	-	-	V	Push-pull, P4(high current PMOS), SOUND, V_{DD}=2.7V, I_{OH}=-0.9mA
		2.0	2.4	-	V	Push-pull, P4(low current PMOS), P8, V_{DD}=2.7V, I_{OH}=-40µA
	V_{OL}	-	0.15	0.3	V	V_{DD}=2.7V,I_{OL}=0.9mA, P4, P8
Leakage current	I_{LO}	-	-	1	µA	Open-drain, V_{DD}=3.3V, V_O=3.3V
Input resistor	R_{IN}	100	200	300	KΩ	P0
		300	600	900	KΩ	$\overline{RESET}$

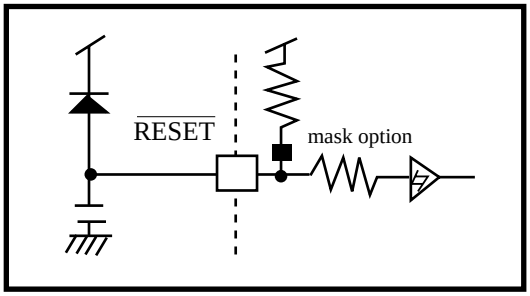
Preliminary

DC ELECTRICAL CHARACTERISTICS (V_{DD}=3±0.3V, V_{SS}=0V, T_{OPR}=25°C)

Parameters	Sym.	Min.	Typ.	Max.	Unit	Conditions
Output current of BZ1, BZ2	I_{OH}	25	-	60	mA	V_{DD}=3V,V_{BZ}=1.5V,
	I_{OL}	25	-	60	mA	
Output current of VO	-	2	3	4	mA	V_{DD}=3V,V_O=0.7V
LCD bias voltage	V_1	-	0.9	-	V	VDD=3V, LCD on, no load
	V_2	-	1.8	-	V	
	V_3	-	2.7	-	V	
	V_4	-	3.6	-	V	
	V_5	-	4.5	-	V	

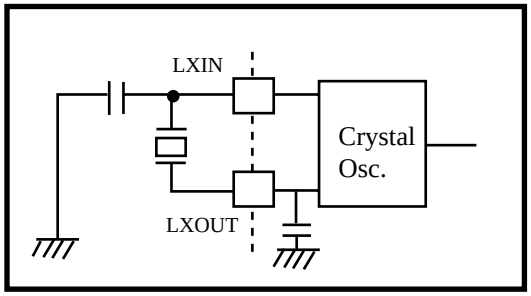
RESET PIN TYPE

TYPE RESET-A

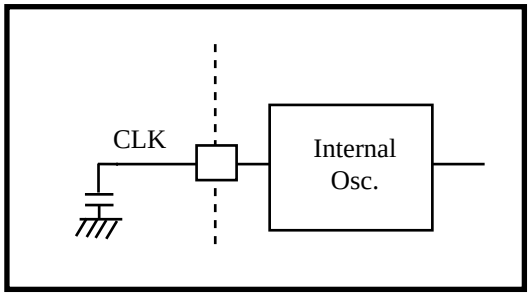


OSCILLATION PIN TYPE

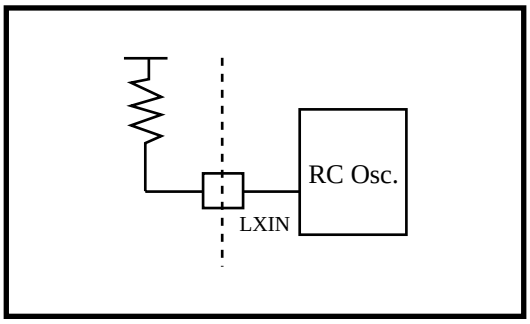
TYPE OSC-B



TYPE OSC-G

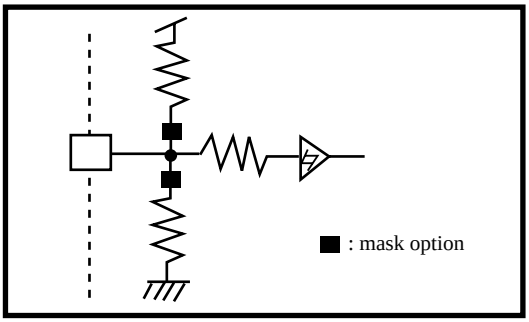


TYPE OSC-H

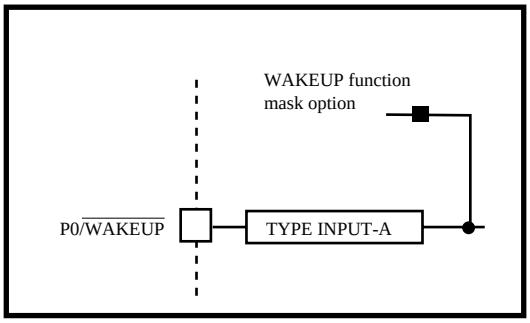


INPUT PIN TYPE

TYPE INPUT-A



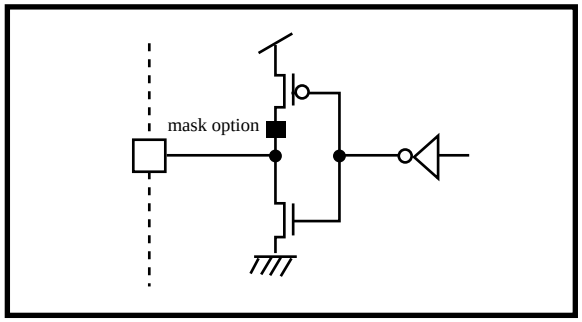
TYPE INPUT-B



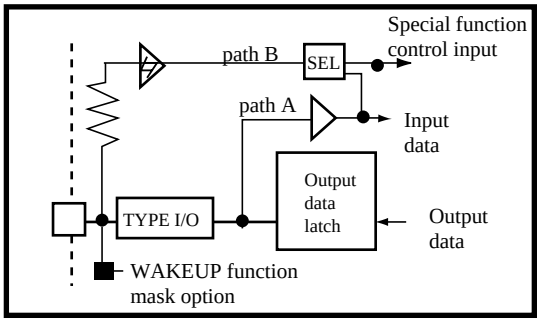
Preliminary

I/O PIN TYPE

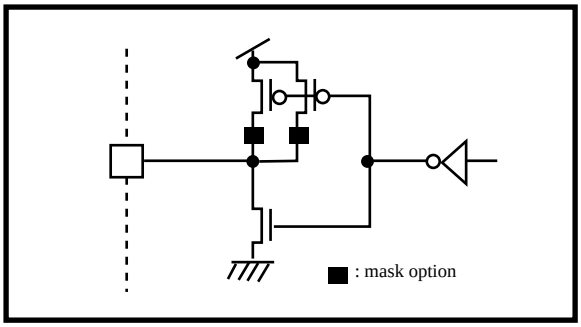
TYPE I/O



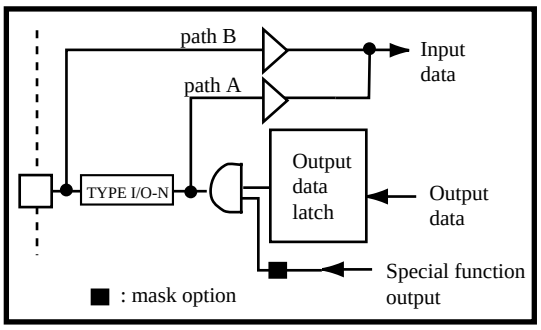
TYPE I/O-L



TYPE I/O-N



TYPE I/O-O

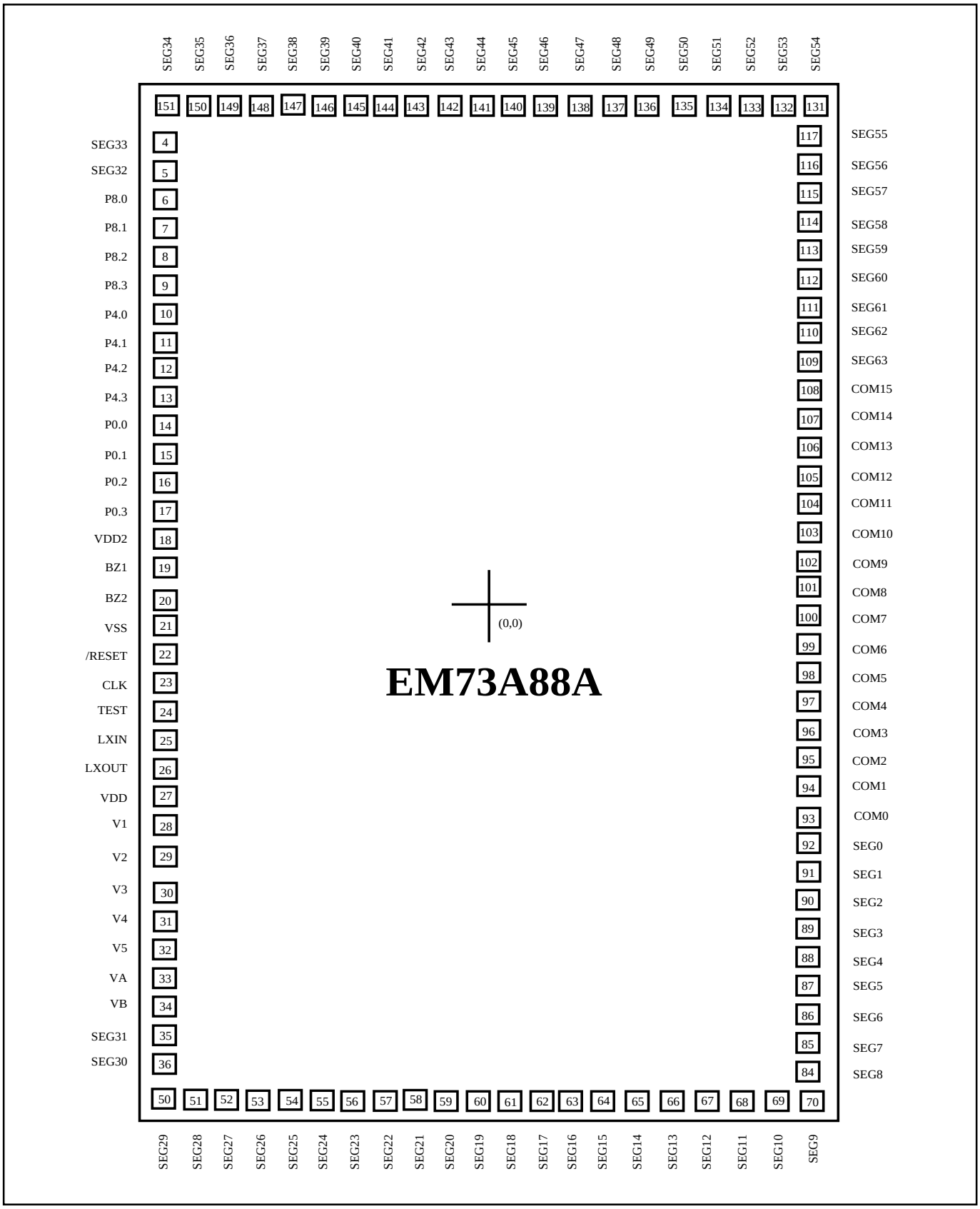


Path A : For set and clear bit of port instructions, data goes through path A from output data latch to CPU.

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



Preliminary

Pad No.	Symbol	X	Y
1			
2			
3			
4	SEG33	-1104.7	1829.1
5	SEG32	-1104.7	1708.6
6	P8.0	-1099.8	1593.1
7	P8.1	-1099.8	1482.6
8	P8.2	-1099.8	1372.2
9	P8.3	-1099.8	1261.7
10	P4.0	-1099.8	1151.2
11	P4.1	-1099.8	1040.8
12	P4.2	-1099.8	930.3
13	P4.3	-1099.8	819.9
14	P0.0	-1099.8	709.4
15	P0.1	-1099.8	598.9
16	P0.2	-1099.8	488.5
17	P0.3	-1099.8	378.0
18	VDD2	-1099.8	257.5
19	BZ1	-1099.8	133.7
20	BZ2	-1099.8	23.3
21	VSS	-1099.8	-92.8
22	/RESET	-1099.8	-223.6
23	CLK	-1099.8	-334.0
24	TEST	-1099.8	-444.5
25	LXIN	-1099.8	-555.0
26	LXOUT	-1099.8	-665.4
27	VDD	-1099.8	-785.9
28	V1	-1104.7	-915.2
29	V2	-1104.7	-1025.7
30	V3	-1104.7	-1136.1
31	V4	-1104.7	-1246.6
32	V5	-1104.7	-1359.6
33	VA	-1104.7	-1475.1
34	VB	-1104.7	-1590.6
35	SEG31	-1104.7	-1708.6
36	SEG30	-1104.7	-1829.1
37			
38			
39			
40			

Preliminary

Pad No.	Symbol	X	Y
41			
42			
43			
44			
45			
46			
47			
48			
49			
50	SEG29	-1113.7	-1949.7
51	SEG28	-993.1	-1949.7
52	SEG27	-875.1	-1949.7
53	SEG26	-759.6	-1949.7
54	SEG25	-646.6	-1949.7
55	SEG24	-536.2	-1949.7
56	SEG23	-425.7	-1949.7
57	SEG22	-317.8	-1949.7
58	SEG21	-212.3	-1949.7
59	SEG20	-106.9	-1949.7
60	SEG19	-1.5	-1949.7
61	SEG18	103.9	-1949.7
62	SEG17	209.3	-1949.7
63	SEG16	314.8	-1949.7
64	SEG15	422.7	-1949.7
65	SEG14	533.2	-1949.7
66	SEG13	643.6	-1949.7
67	SEG12	756.6	-1949.7
68	SEG11	872.1	-1949.7
69	SEG10	993.1	-1949.7
70	SEG9	1113.7	-1949.7
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			

Preliminary

Pad No.	Symbol	X	Y
81			
82			
83			
84	SEG8	1104.7	-1829.1
85	SEG7	1104.7	-1708.6
86	SEG6	1104.7	-1588.0
87	SEG5	1104.7	-1470.0
88	SEG4	1104.7	-1354.5
89	SEG3	1104.7	-1241.5
90	SEG2	1104.7	-1131.1
91	SEG1	1104.7	-1020.6
92	SEG0	1104.7	-910.2
93	COM0	1104.7	-799.7
94	COM1	1104.7	-689.2
95	COM2	1104.7	-581.3
96	COM3	1104.7	-474.4
97	COM4	1104.7	-369.0
98	COM5	1104.7	-263.6
99	COM6	1104.7	-158.2
100	COM7	1104.7	-52.7
101	COM8	1104.7	52.7
102	COM9	1104.7	158.1
103	COM10	1104.7	263.5
104	COM11	1104.7	368.9
105	COM12	1104.7	474.4
106	COM13	1104.7	581.3
107	COM14	1104.7	689.2
108	COM15	1104.7	799.7
109	SEG63	1104.7	910.2
110	SEG62	1104.7	1020.6
111	SEG61	1104.7	1131.1
112	SEG60	1104.7	1241.5
113	SEG59	1104.7	1354.5
114	SEG58	1104.7	1470.0
115	SEG57	1104.7	1588.0
116	SEG56	1104.7	1708.6
117	SEG55	1104.7	1829.1
118			
119			
120			

Preliminary

Pad No.	Symbol	X	Y
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131	SEG54	1113.7	1949.7
132	SEG53	993.1	1949.7
133	SEG52	872.1	1949.7
134	SEG51	756.6	1949.7
135	SEG50	643.6	1949.7
136	SEG49	533.2	1949.7
137	SEG48	422.7	1949.7
138	SEG47	314.8	1949.7
139	SEG46	209.3	1949.7
140	SEG45	103.9	1949.7
141	SEG44	-1.5	1949.7
142	SEG43	-106.9	1949.7
143	SEG42	-212.3	1949.7
144	SEG41	-317.8	1949.7
145	SEG40	-425.7	1949.7
146	SEG39	-536.2	1949.7
147	SEG38	-646.6	1949.7
148	SEG37	-759.6	1949.7
149	SEG36	-875.1	1949.7
150	SEG35	-993.1	1949.7
151	SEG34	-1113.7	1949.7

Unit : µm
Chip Size : 2520 x 4210 µm
Note : For PCB layout, IC substrate must be floated or connected to Vss.

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]$_L$	1	2	-	Z	1
LDAXI	0110 0111	Acc←ROM[DP]$_H$,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'

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[HLl - 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_{12-6}.a_{5-0} 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)	If SF=1 then PC←a else null	3	3	-	-	1
	0101 0111 1100 aaaa aaaa aaaa (a:0000~0FFFh)						

(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] - 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	$RAM[HL]_b \leftarrow 0$	1	1	-	-	1
CLP p,b	0110 1101 11bb pppp	$PORT[p]_b \leftarrow 0$	2	2	-	-	1
CLPL	0110 0000	$PORT[LR_{3-2}+4]LR_{1-0} \leftarrow 0$	1	2	-	-	1
CLR y,b	0110 1100 11bb yyyy	$RAM[y]_b \leftarrow 0$	2	2	-	-	1
SEM b	1111 01bb	$RAM[HL]_b \leftarrow 1$	1	1	-	-	1
SEP p,b	0110 1101 01bb pppp	$PORT[p]_b \leftarrow 1$	2	2	-	-	1
SEPL	0110 0010	$PORT[LR_{3-2}+4]LR_{1-0} \leftarrow 1$	1	2	-	-	1
SET y,b	0110 1100 01bb yyyy	$RAM[y]_b \leftarrow 1$	2	2	-	-	1
TF y,b	0110 1100 00bb yyyy	$SF \leftarrow RAM[y]_b'$	2	2	-	-	*
TFA b	1111 10bb	$SF \leftarrow Acc_b'$	1	1	-	-	*
TFM b	1111 11bb	$SF \leftarrow RAM[HL]_b'$	1	1	-	-	*
TFP p,b	0110 1101 00bb pppp	$SF \leftarrow PORT[p]_b'$	2	2	-	-	*
TFPL	0110 0001	$SF \leftarrow PORT[LR_{3-2}+4]LR_{1-0}'$	1	2	-	-	*
TT y,b	0110 1100 10bb yyyy	$SF \leftarrow RAM[y]_b$	2	2	-	-	*
TTP p,b	0110 1101 10bb pppp	$SF \leftarrow PORT[p]_b$	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 nnnn	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←$[DP]_L$	2	2	-	Z	1
LDADPM	0110 1010 1111 1101	Acc←$[DP]_M$	2	2	-	Z	1
LDADPH	0110 1010 1111 1110	Acc←$[DP]_H$	2	2	-	Z	1
LDASP	0110 1010 1111 1111	Acc←SP	2	2	-	Z	1
LDATAL	0110 1010 1111 0100	Acc←$[TA]_L$	2	2	-	Z	1
LDATAM	0110 1010 1111 0101	Acc←$[TA]_M$	2	2	-	Z	1
LDATAH	0110 1010 1111 0110	Acc←$[TA]_H$	2	2	-	Z	1
LDATBL	0110 1010 1111 1000	Acc←$[TB]_L$	2	2	-	Z	1
LDATBM	0110 1010 1111 1001	Acc←$[TB]_M$	2	2	-	Z	1
LDATBH	0110 1010 1111 1010	Acc←$[TB]_H$	2	2	-	Z	1
STADPL	0110 1001 1111 1100	$[DP]_L$←Acc	2	2	-	-	1
STADPM	0110 1001 1111 1101	$[DP]_M$←Acc	2	2	-	-	1
STADPH	0110 1001 1111 1110	$[DP]_H$←Acc	2	2	-	-	1
STASP	0110 1001 1111 1111	SP←Acc	2	2	-	-	1
STATAL	0110 1001 1111 0100	$[TA]_L$←Acc	2	2	-	-	1
STATAM	0110 1001 1111 0101	$[TA]_M$←Acc	2	2	-	-	1
STATAH	0110 1001 1111 0110	$[TA]_H$←Acc	2	2	-	-	1
STATBL	0110 1001 1111 1000	[$TB]_L$←Acc	2	2	-	-	1
STATBM	0110 1001 1111 1001	$[TB]_M$←Acc	2	2	-	-	1
STATBH	0110 1001 1111 1010	$[TB]_H$←Acc	2	2	-	-	1

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_{CC}	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)	$ROM[DP]_L$	Low 4-bit of program memory
$ROM[DP]_H$	High 4-bit of program memory	$[DP]_L$	Low 4-bit of data pointer register
$[DP]_M$	Middle 4-bit of data pointer register	$[DP]_H$	High 4-bit of data pointer register
$[TA]_L([TB]_L)$	Low 4-bit of timer/counter A (timer/counter B) register	$[TA]_M([TB]_M)$	Middle 4-bit of timer/counter A (timer/counter B) register
$[TA]_H([TB]_H)$	High 4-bit of timer/counter A (timer/counter B) register	LR_{1-0}	Contents of bit assigned by bit 1 to 0 of LR
LR_{3-2}	Bit 3 to 2 of LR	a_{5-0}	Bit 5 to 0 of destination address for branch instruction
PC_{12-6}	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