

Preliminary

GENERAL DESCRIPTION

EM73P968 is an advanced single chip CMOS 4-bit one time programming (OTP) micro-controller. It contains 16K-byte ROM, 2.5K nibbles RAM, 4-bit ALU, 13-level subroutine nesting, 22-stage time base, two 12-bit timer/counters for the kernel function. EM73P968 also contains 6 interrupt sources, 1 input port, 8 bidirection ports, Max LCD display (52x5), built-in watch-dog-timer and high speed Timer/Counter.

An analog to digital (A/D) converter having 8-bit multiplier analog input and 8-bit resolution. Serial peripheral interface (SPI).

EM73P968 has plentiful operating modes (SLOW, IDLE, STOP) intended to reduce the power consumption.

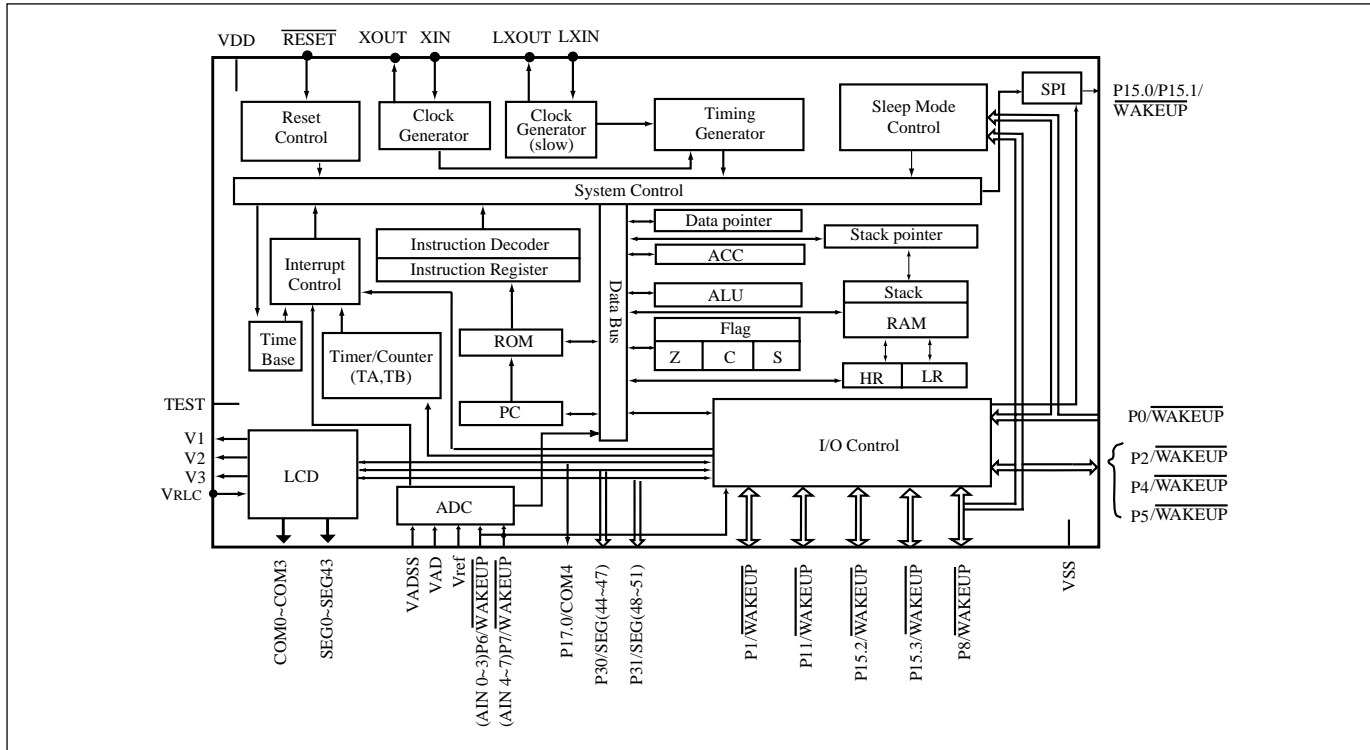
FEATURES

- Operation voltage : 2.2V ~ 6V.
- Clock source : Dual clock system. Low-frequency oscillator is Crystal or RC oscillator (32K Hz, connect an external resistor) by mask option and high-frequency oscillator is RC (Connect an external resistor) or Crystall oscillator.
- Instruction set : 107 powerful instructions.
- Instruction cycle time : Up to 2us for 4 MHz (high speed clock).
122 μ s for 32768 Hz (low speed clock with frequency Double)
- ROM capacity : 16K x 8 bits.
- RAM capacity : 2.5K x 4 bits.
- Input port : 1 port, P0(0..3), IDLE/STOP releasing function are available by mask option.
- Output port : 9 pins (P17.0, P30, P31), P17.0, P30, P31 are shared with LCD pins.
- Bidirection port : 9 ports (P1, P2, P4, P5, P6, P7, P8, P11, P15). IDLE/STOP releasing function are available by mask option for P8(0..3), P5 and P6 have high current sink.
- 12-bit timer/counter : Two 12-bit timer/counters are programmable for timer, event counter and pulse width measurement.
- A/D converter : An analog to digital (A/D) converter having 8-bit multiplier analog input and 8-bit resolution.
- SPI : Serial peripheral interface.
- Built-in watch-dog-timer : It is available by mask option.
- Built-in time base counter : 22 stages.
- Built-in high Speed Timer/Counter : Could be timer.
- Subroutine nesting : Up to 13 levels.
- Interrupt : External 2 input interrupt sources.
Internal 2 Timer overflow interrupts, 1 time base interrupt.
1 high speed counter overflow interrupt.
- LCD driver : 52 X 5 dots, 1/3 bias, 1/4 or 1/5 duty by mask option.
- Power saving function : SLOW, IDLE, STOP operation mode.
- Package type : Chip form.
QFP 128 pin.

APPLICATIONS

EM73P968 is suitable for application in family appliance, consumer products, hand held games, calculator and the toy controller.

Preliminary

FUNCTION BLOCK DIAGRAM

PIN DESCRIPTIONS

Symbol	Pin-type	Function
V_{DD}		Power supply (+)
V_{SS}		Power supply (-)
V_{ref}		ADC power (+)
V_{AD}		ADC power (+)
V_{ADSS}		ADC power (-)
RESET	RESET-A	System reset input signal, low active mask option : none pull-up
XIN/RCosc	OSC-A/OSC-H1	Crystal/RC clock source connecting pin
XOUT	OSC-A	Crystal connecting pin
LXIN	OSC-B/OSC-H2	Crystal/RC connecting pin for low speed clock source
LXOUT	OSC-B	Crystal connecting pin for low speed clock source
P0(0..3)/WAKEUP(0..3)	INPUT-K	4-bit input port with IDLE/STOP releasing function P0.0/ACLK : address counter clock for programming OTP. P0.1/PGMB : program data to OTP cells for programming OTP. P0.2/OEB : data output enable for programming OTP. P0.3/DCLK : data in/out clock signal for programming OTP. mask option 1 : wakeup disable wakeup enable mask option 2 : low current pull up normal current pull up high current pull up none

Preliminary

PIN DESCRIPTIONS

Symbol	Pin-type	Function
P8.0(INT1)/WAKEUPA /DIN P8.2(INT0)/WAKEUPC	I/O-R1	2-bit bidirection I/O port with external interrupt sources input and IDLE/STOP releasing function P8.0/DIN : data input for programming OTP mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P8.1(TRGB)/WAKEUPB, /DOUT P8.3(TRGA)/WAKEUPD	I/O-R1	2-bit bidirection I/O port with timer/counter A, B external input and IDLE/STOP releasing function P8.1/DOUT : data output for programming OTP mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P6(0..3)/WAKEUP(20..23) AIN (0..3) P7(0..3)/WAKEUP(24..27) AIN (4..7)	I/O-R1	8-bit bidirection I/O port with IDLE/STOP releasing function. Share with A/D analog input pin. mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P4(0..3)/WAKEUP(12,15)	I/O-R1	4-bit bidirection I/O port with IDLE/STOP releasing function mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none

Preliminary

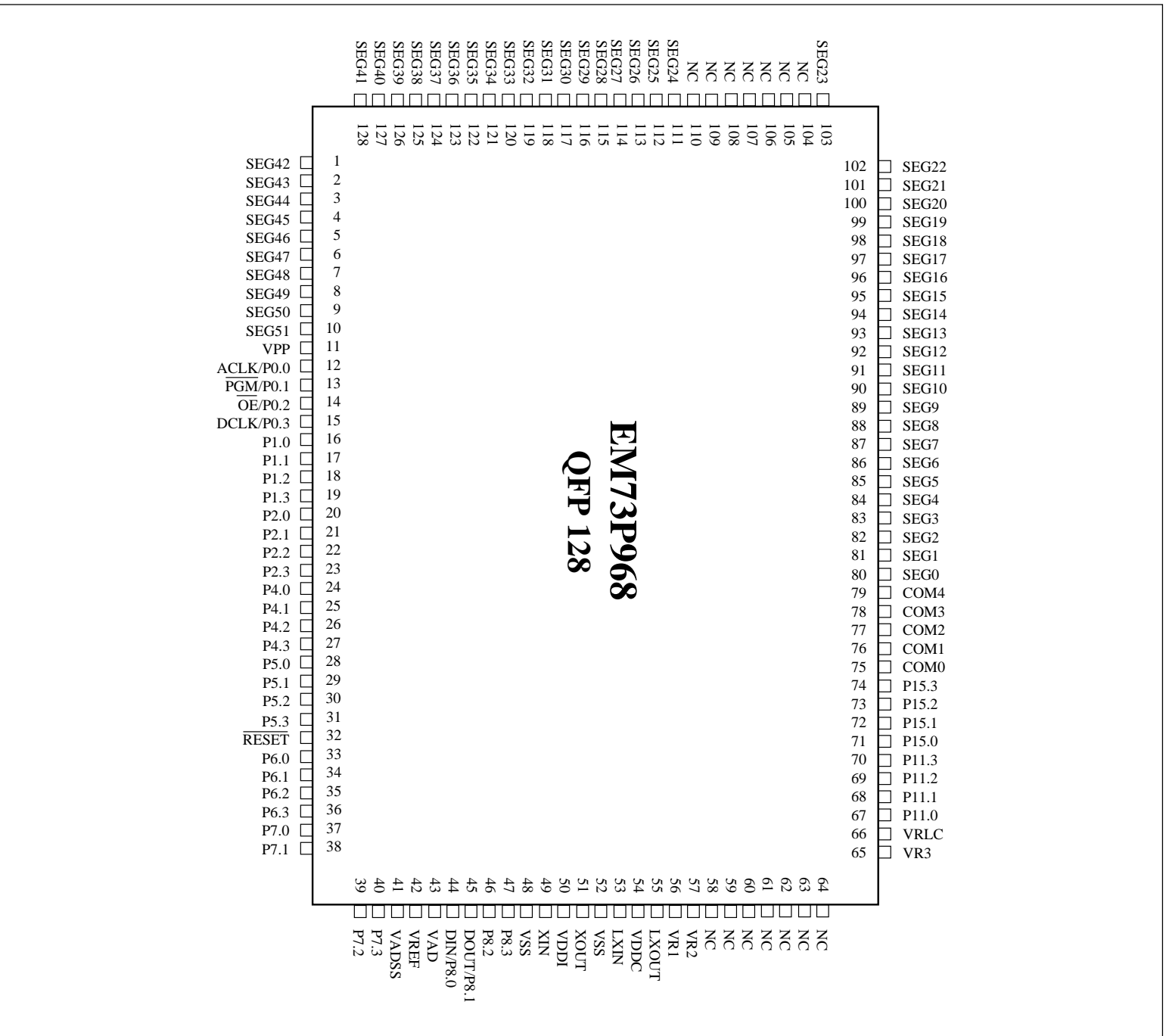
PIN DESCRIPTIONS

Symbol	Pin-type	Function
P1(0..3)/WAKEUP(4..7) P2(0..3)/WAKEUP(8..11) P5(0..3)/WAKEUP(16..19) P11(0..3)/ WAKEUP(28..31) P15.2/P15.3/ WAKEUP(34,35)	I/O-R1	18-bit bidirection I/O pins with IDLE/STOP releasing function mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P15.0/WAKEUP(32)		1-bit bidirection I/O pins with IDLE/STOP releasing function. Share with SPI data input/output pin. mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P15.1/WAKEUP(33)	I/O-R1	1-bit bidirection I/O pins with IDLE/STOP releasing function. Share with SPI clock input/output pin. mask option 1 : wakeup disable wakeup enable mask option 2 : low current push pull normal current push pull high current push pull none
P17.0/COM4	Output-L	1-bit output pin with LCD common pin mask option : LCD common pin Push pull Open-drain
P30(0..3)/SEG(51..48) P31(0..3)/SEG(47..44)	Output-M	8-bit output pins are shared with LCD segment pin mask option : LCD segment pin Low current push pull Normal current push pull High current push pull Open drain
COM0~COM3		LCD common output pins
SEG0~SEG43		LCD segment output pins
VRLC, V1, V2, V3	--	LCD bias voltage pins
TEST	--	Test pin must be connected to VSS VPP : high voltage (12V) power source for programming OTP



Preliminary

PIN ASSIGNMENT



* This specification are subject to be changed without notice.

Preliminary

FUNCTION DESCRIPTIONS

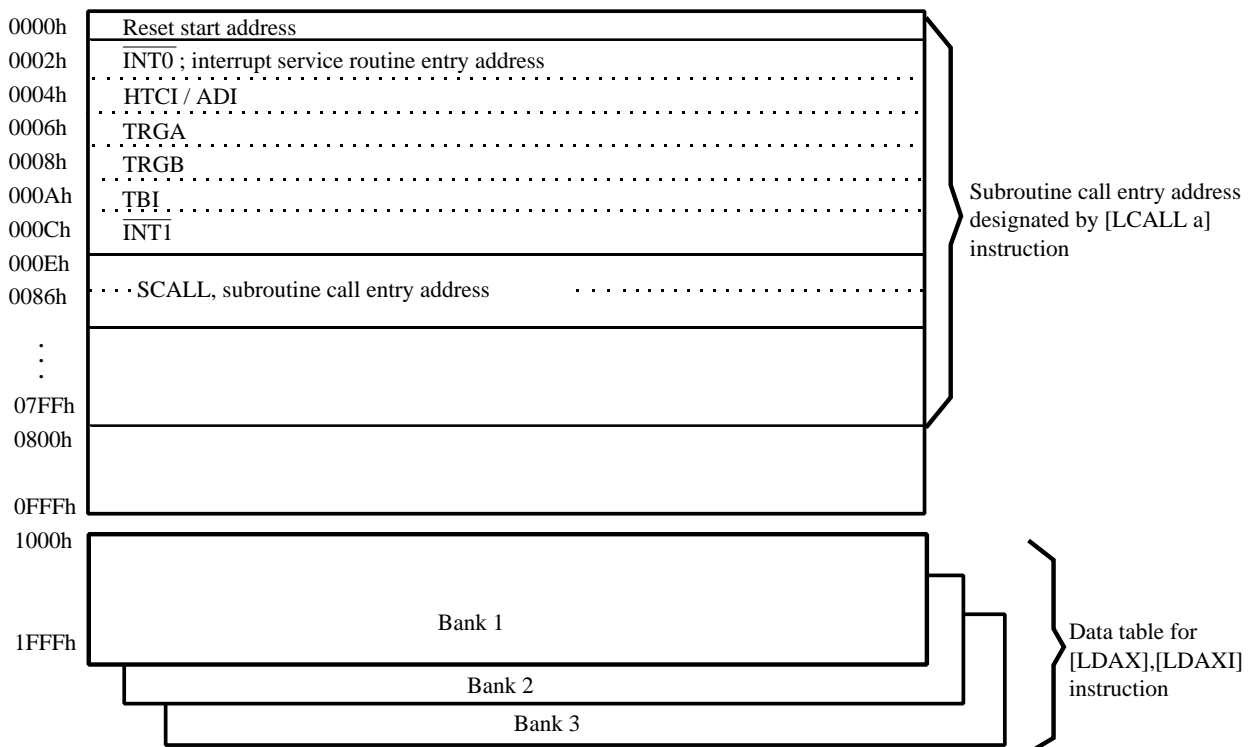
PROGRAM ROM (16K X 8 bits)

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, 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
	P3=xx11B		
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
        :
;-----
XB1 :      BANK 2
        :
        :
        B      XB
        :
XB2 :      :
        :
        B      XB2
        :
;-----
XC1 :      BANK 3
        :
        :
        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 (2548-nibble)

A total 2548 - nibble data RAM is available from address 000 to 9FFh
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=0000B	000-00Fh	ZERO PAGE															
	010-01Fh																
	020-02Fh	COM0															
	030-03Fh	COM1															
	040-04Fh	COM2															
	050-05Fh	COM3															
	060-06Fh	COM4															
	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				TCA				TCB				DP		SPW	
Bank 1																	
P9=0001B	100-10Fh																
	:																
	:																
	1F0-1FFh																
Bank 2																	
P9=0010B	200-20Fh																
	:																
	:																
	2F0-2FFh																
Bank 3																	
P9=0011B	300-30Fh																
	:																
	:																
	3F0-3FFh																
	:																
	:																
	:																
	:																
	:																
	:																
Bank 9																	
P9=1001B	900-90Fh																
	:																
	:																
	9F0-9FFh																

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 ← 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 2548 nibble data memory consists of ten banks (bank 0 ~ bank 9). There are 244x4 bits (address 000h~0F3h) in bank 0 and 2304x4 bits (address 100h ~ 9FF) in bank 1 ~ bank 9.

The bank is selected by P9.

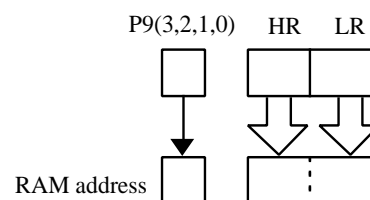
P9(3..0) Initial value : 0 0 0 0

RBK	Bank	RAM address(hex)
0 0 0 0	0	0 0 0 ~ 0 F F
0 0 0 1	1	1 0 0 ~ 1 F F
0 0 1 0	2	2 0 0 ~ 2 F F
0 0 1 1	3	3 0 0 ~ 3 F F
0 1 0 0	4	4 0 0 ~ 4 F F
0 1 0 1	5	5 0 0 ~ 5 F F
0 1 1 0	6	6 0 0 ~ 6 F F
0 1 1 1	7	7 0 0 ~ 7 F F
1 0 0 0	8	8 0 0 ~ 8 F F
1 0 0 1	9	9 0 0 ~ 9 F F
1 0 1 0~	0	0 0 0 ~ 0 F F
1 1 1 1		

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

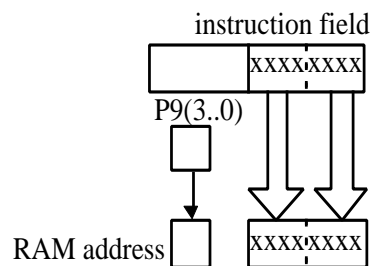
PROGRAM EXAMPLE: Load the data of RAM address "143h" to RAM address "032h".

```

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.



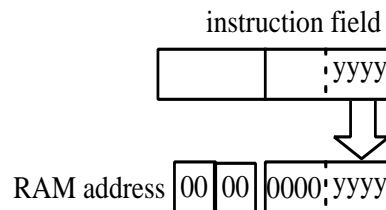
PROGRAM EXAMPLE: Load the data of RAM address "143h" to RAM address "023h".

```

OUT   #0001B,P9      ; RAM bank1
LDA   43h             ; Acc← RAM[143h]
OUT   #0001B,P9     ; RAM bank1
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_{12-6,a} (branch condition satisfied)

PC

Hold original PC value+1	a	a	a	a	a	a	a
--------------------------	---	---	---	---	---	---	---

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

PC

Original PC value + 1

LBR a

Object code: 1100 aaaa aaaa aaaa

Condition: SF=1; PC ← PC_{12,a} (branch condition satisfied)

PC

Hold +2	a	a	a	a	a	a	a	a	a	a	a	a
------------	---	---	---	---	---	---	---	---	---	---	---	---

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

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 satisfied)

PC

a	a	a	a	a	a	a	a	a	a	a	a	a
---	---	---	---	---	---	---	---	---	---	---	---	---

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

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

Preliminary

PC

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

RET

Object code: 0100 1111

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

PC

The return address stored in stack

RTI

Object code: 0100 1101

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

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

TRGH (High speed counter interrupt)

PC

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

TRGB (Time B overflow interrupt)

PC

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

TBI (Time base interrupt)

PC

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

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

PC

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

(4) Reset operation:

PC

0	0	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 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 satisfied, likewise, when SF = 0, branch condition is unsatisfied.

Preliminary

PROGRAM EXAMPLE:

Check following arithmetic operation for CF, ZF, SF

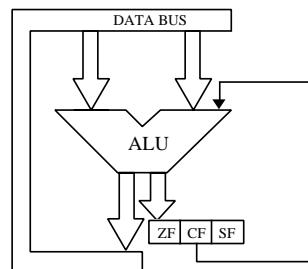
	CF	ZF	SF
LDIA #00h;	-	1	1
LDIA #03h;	-	0	1
ADDA #05h;	-	0	1
ADDA #0Dh;	-	0	0
ADDA #0Eh;	-	0	0

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".

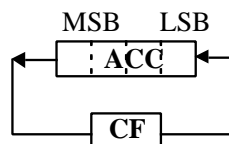
Preliminary

EXAMPLE:

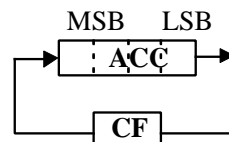
Operation	Carry	Zero
8-4=4	1	0
7-F= -8(1000)	0	0
9-9=0	1	1

(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.

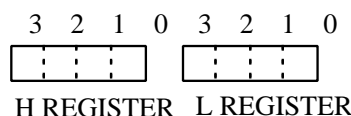


PROGRAM EXAMPLE: To rotate Acc clockwise (right) and shift a "1" into the MSB bit of Acc.
 TTCFS; CF ← 1
 RRCA; rotate Acc right and shift CF=1 into MSB.

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)

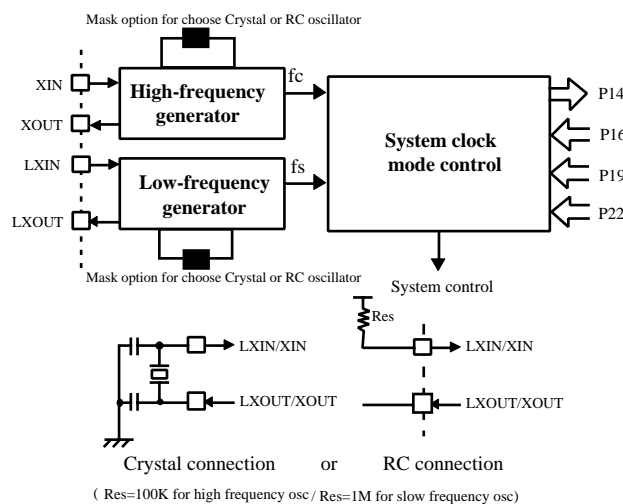
Data pointer is a 12-bit register that stores the ROM address can indicating the ROM code data specified by user (refer to data ROM).

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

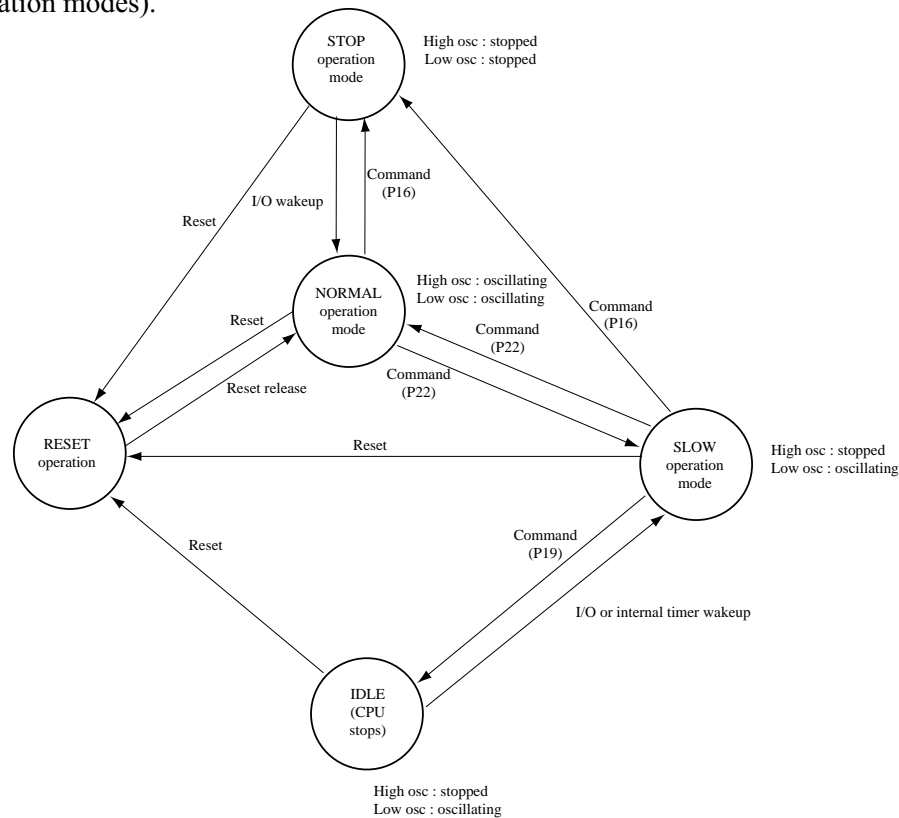
There are two clock generator for system clock control unit, P14 is the status register that hold the CPU status. P16, P19 and P22 are the command register for system clock mode control.



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. EM73P968 has four operation modes (NORMAL, SLOW, IDLE and STOP operation modes).



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

NORMAL OPERATION MODE

The 4-bit μ c is in the NORMAL 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 and high speed timer/counter are available for the NORMAL operation mode.

SLOW OPERATION MODE

The SLOW operation mode is single clock system (low-frequency clock oscillating). It can be changed to the NORMAL 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.

Preliminary

P22 3 2 1 0 Initial value : 0000

*	SOM		
---	-----	--	--

SOM	Low-frequency
000	2 ³ /LXIN RC solw to normal
001	2 ⁴ /LXIN RC solw to normal
010	2 ¹¹ /LXIN X'tal slow to normal
011	2 ¹² /LXIN X'tal slow to normal
1**	normal to slow

P14 3 2 1 0 Initial value : *000

INT2_S	WKS	SPI_F	CPUS
--------	-----	-------	------

SPI_F	SPI Flag	CPUS	CPU status
0	SPI register is empty	0	NORMAL operation mode
1	SPI register is full	1	SLOW operation

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. P14.2 (wakeup status) will be set to "1" when CPU is wake-up by internal timer. P14.2 will be cleared to "0" when user out data to P14. INT2_S is low, the program address "0004H" is the interrupt entry address of HTCI. INT2_S is high, the program address "0004H" is the interrupt entry address of ADI.

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 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, P1(0..3)/WAKEUP 4..7, P2(0..3)/WAKEUP 8..11, P4(0..3)/WAKEUP 12..15, P5(0..3)/WAKEUP 16..19, P6(0..3)/WAKEUP 20..23, P7(0..3)/WAKEUP 24..27, P8(0..3)/WAKEUPA..D, P11(0..3)/WAKEUP 28..31, and P15(0..3)/WAKEUP 32..35).

P19 3 2 1 0 Initial value : 0000

IDME		SIDR	
------	--	------	--

IDME	Enable IDLE mode
0 1	Enable IDLE mode
* *	no function

SIDR	Select IDLE releasing condition
0 0	P0,P1,P2,P4,P5,P6,P7,P8,P11,P15 pin input
0 1	P0,P1,P2,P4,P5,P6,P7,P8,P11,P15 pin input and 1 sec signal
1 0	P0,P1,P2,P4,P5,P6,P7,P8,P11,P15 pin input and 0.5 sec signal
1 1	P0,P1,P2,P4,P5,P6,P7,P8,P11,P15 pin input and 15.625 ms signal

Preliminary

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)/ $\overline{\text{WAKEUP}}\ 0..3$, P1(0..3)/ $\overline{\text{WAKEUP}}\ 4..7$, P2(0..3)/ $\overline{\text{WAKEUP}}\ 8..11$, P4(0..3)/ $\overline{\text{WAKEUP}}\ 12..15$, P5(0..3)/ $\overline{\text{WAKEUP}}\ 16..19$, P6(0..3)/ $\overline{\text{WAKEUP}}\ 20..23$, P7(0..3)/ $\overline{\text{WAKEUP}}\ 24..27$, P8(0..3)/ $\overline{\text{WAKEUPA}}..D$, P11(0..3)/ $\overline{\text{WAKEUP}}\ 28..31$, and P15(0..3)/ $\overline{\text{WAKEUP}}\ 32..35$).

LCD display and high speed timer/counter with melody output are disabled in STOP mode.

Initial value : 0000

P16 3 2 1 0

*	SWWT	Set wake up Stop wake up time (go to NORMAL)
*	1 0 0	$2^9/\text{XIN}$ for RC osc.
*	1 0 1	$2^{10}/\text{XIN}$ for RC osc.
*	1 1 0	$2^{18}/\text{XIN}$ for Crystal osc.
*	1 1 1	$2^{19}/\text{XIN}$ for Crystal osc.

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 HTCI
:
HTCI: 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 ³ Hz	Reserved
0 1 0 1	Interrupt frequency LXIN / 2 ¹⁵ Hz	Interrupt frequency LXIN / 2 ¹⁵ Hz
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

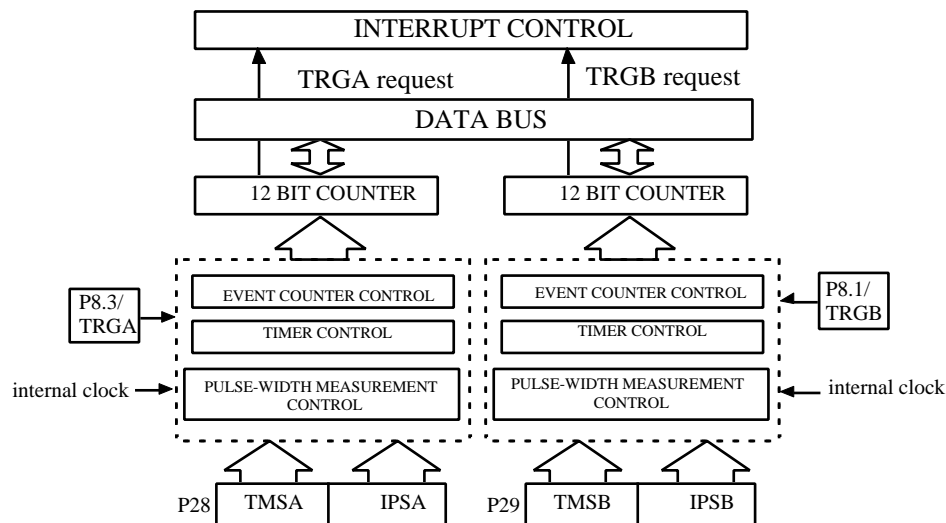
TIMER / COUNTER (TIMER A, TIMER B)

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

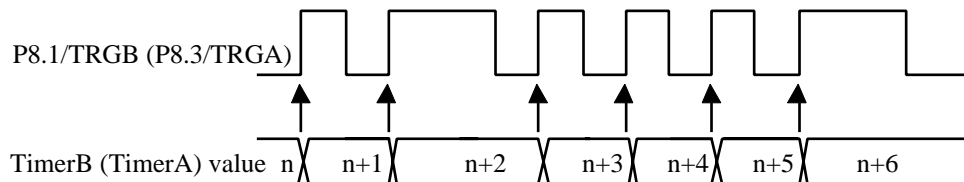
IPSA	Clock rate selection		IPSB	Clock rate selection	
	NORMAL mode	SLOW mode		NORMAL mode	SLOW mode
0 0	LXIN/2 ³ HZ	Reserved	0 0	Depend on high speed timer/counter	
0 1	LXIN/2 ⁷ HZ	LXIN/2 ⁷ HZ	0 1	LXIN/2 ⁵ HZ	LXIN/2 ⁵ HZ
1 0	LXIN/2 ¹¹ HZ	LXIN/2 ¹¹ HZ	1 0	LXIN/2 ⁹ HZ	LXIN/2 ⁹ HZ
1 1	LXIN/2 ¹⁵ HZ	LXIN/2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

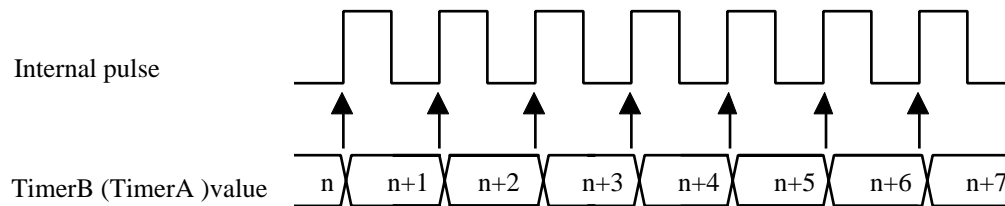
```
LDIA    #0100b;
OUTA    P28      ; Enable timerA with event counter mode
```

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: $LXIN/2^3$; $LXIN = 32KHz$

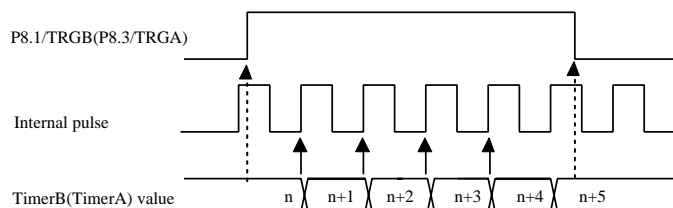
The time of timer counter count one = $2^3 / LXIN = 8/32768=0.244ms$

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

HIGH SPEED TIMER/COUNTER

EM73P968 has one 8-bit high speed timer/counter (HTC). It supports two special functions : auto load timer and melody output. The HTC is available for the NORMAL and SLOW operation mode.

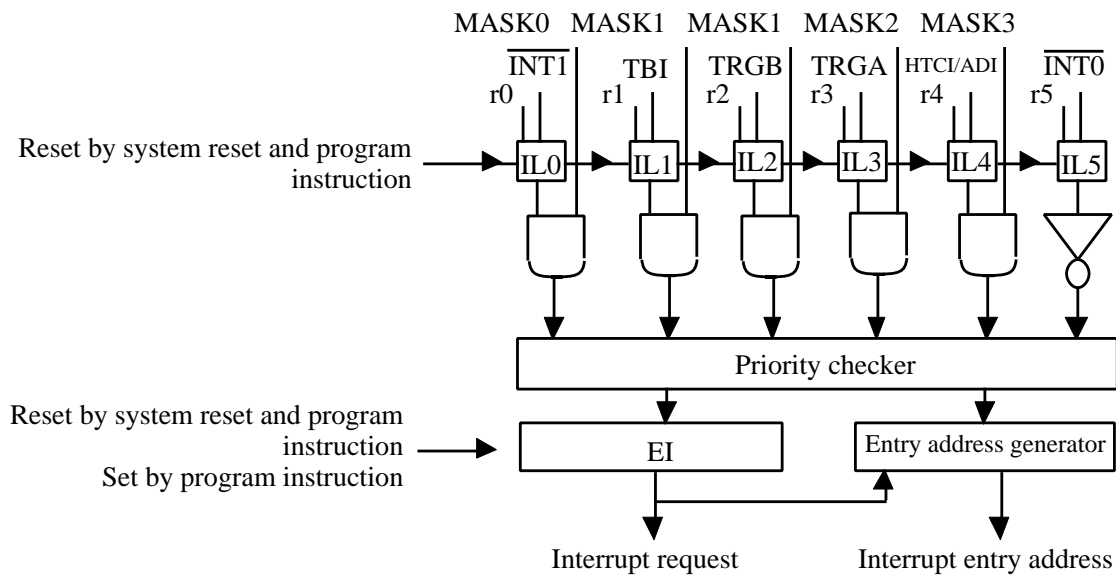
The HTC can be set initial value and send counter value to counter registers (P12 and P13), P20 is the command port for HTC, user can choose different operation mode and different internal clockrate by setting the port. The timer/counter increase one at the rising edge of internal pulse. The HTC can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

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	HTC interrupt (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



Preliminary

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 $\overline{\text{INT0}}$, 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.

INTERRUPT OPERATION

The procedure of interrupt operation:

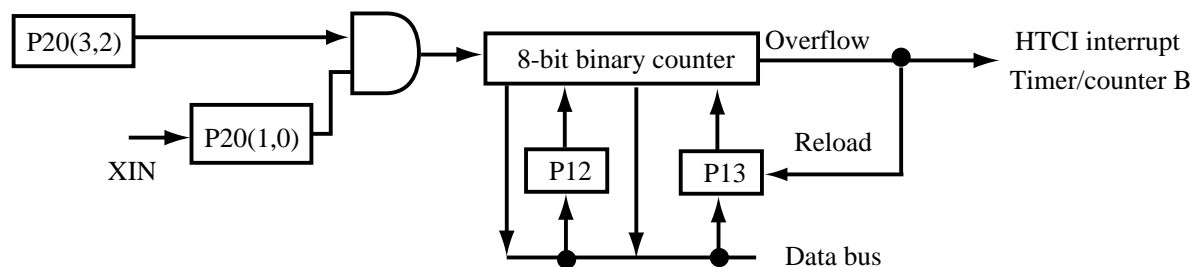
1. Push PC and all flags to stack.
2. Set interrupt entry address into PC.
3. Set SF = 1.
4. Clear EI to inhibit other interrupts occur.
5. Clear the IL with which interrupt source has already been accepted.
6. Excute interrupt subroutine from the interrupt entry address.
7. CPU accept RTI, restore PC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

HIGH SPEED COUNTER

EM73P968 has one high speed counter for auto load timer mode. This function is available for the NORMAL operation mode.



Preliminary

FUNCTION OF HIGH SPEED COUNTER

The HTC has auto load timer mode.

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

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 NORMAL operation mode.

PROGRAM EXAMPLE :

```

LDIA          #00H          ; initial TCB & HTC register
STATBL
STATBM
STATBH
OUTA          P13
OUTA          P12
LDIA          #1011B        ; 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

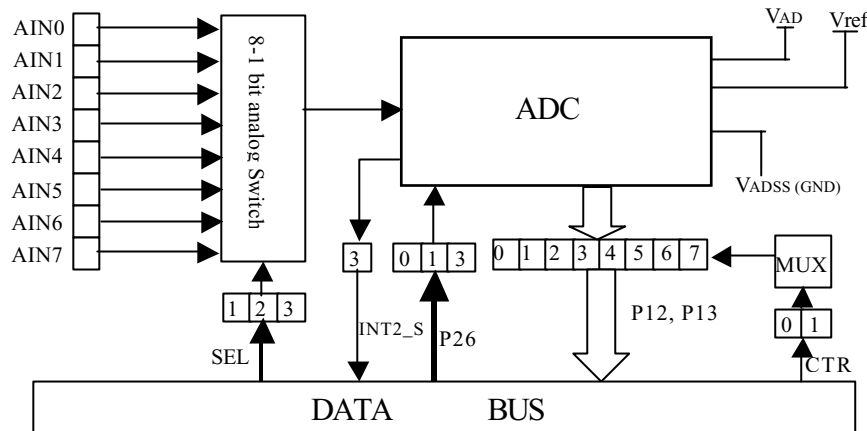
```

Preliminary

ANALOG – TO-DIGITAL CONVERTER (ADC)

The analog – to - digital consists of an 8-bit analog multiplexer (P6, P7), one control register (P26), two data register (P12,P13), and ADC with 8-bit resolution.

The ADC module utilizes successive approximation to convert the unknown analog signal to a digital value. The result is fed to the P12,P13, Input channel are select by the analog input multiplexer the P17 register bits SEL0, SEL1 and SEL2. The A/D converter is disable when the CPU is reset or in the STOP/IDLE/SLOW operation mode. User must enable it by self when the CPU is NORMAL operation mode.



A/D channel control register

P17(BIT)	3	2	1	0	Initial value :0000
SYMBOL	SEL2	SEL1	SEL0	COM4	

SEL0~ SEL2: Analog Input select

A/D input share with P6 & P7.

Analog Input Select			Input channel	Share with pin
SEL2	SEL1	SEL0		
0	0	0	AIN0	P6.0
0	0	1	AIN1	P6.1
0	1	0	AIN2	P6.2
0	1	1	AIN3	P6.3
1	0	0	AIN4	P7.0
1	0	1	AIN5	P7.1
1	1	0	AIN6	P7.2
1	1	1	AIN7	P7.3

Preliminary

ADC control register

P26(BIT)	3	2	1	0
SYMBOL	ADEN	*	F_RUN	START

Initial value : 0000

Port 26 is A/D control register , when P26.3 (ADEN) is high A/D converter enable , P26.3 is low A/D converter disable , P26.1(F_RUN) is high, select A/D conversion is free run , P26.1(F_RUN) is low , A/D could not convert P26.0(START) is high , A/D converter is only one time.

A/D clock rate control register

P23(BIT)	3	2	1	0
SYMBOL	*	*	A/D rate select	

Initial value : 0000

A/D rate	A/D clock rate
0 0	$CLK / 2^5$
0 1	$CLK / 2^6$
1 0	$CLK / 2^7$
1 1	$CLK / 2^7$

CLK=system clock (4M)

ADC Data Register (P12,P13)

When we use ADC , first ADC must get P12,P13 ,because P12,P13 share with SPI , ADC and HTC when the A/D conversion is complete ,the result is load to the P12,P13, and the ADC can generate an interrupt (ADI), the INT2_S (P14.3) is set high.

Preliminary

PROGRAM EXAMPLE : input P6.0 an analog message to coverter

```

CHIP      16K
;----- RAM define area -----
DSEG     10H
ADCBUF:   RES      2
;----- interrupt subroutine -----
CSEG
LBR      START
ORG      004H
LBR      ADI
;-----
START:
LDIA     #0001B   ; A/D clock rate=60K
OUTA     P23
LDIA     #0001B
OUTA     P18      ; P12,P13 → ADC
LDIA     #1001B
OUTA     P26      ; ADC enable & ADC run one time
LDIA     #0000B
OUTA     P17      ; P6.0 input an analog

LOOP:
B        LOOP    ; wait the ADC interrupt to occur & interrupt Flag to be Set
                (INT2 _S)
B        LOOP
:
:

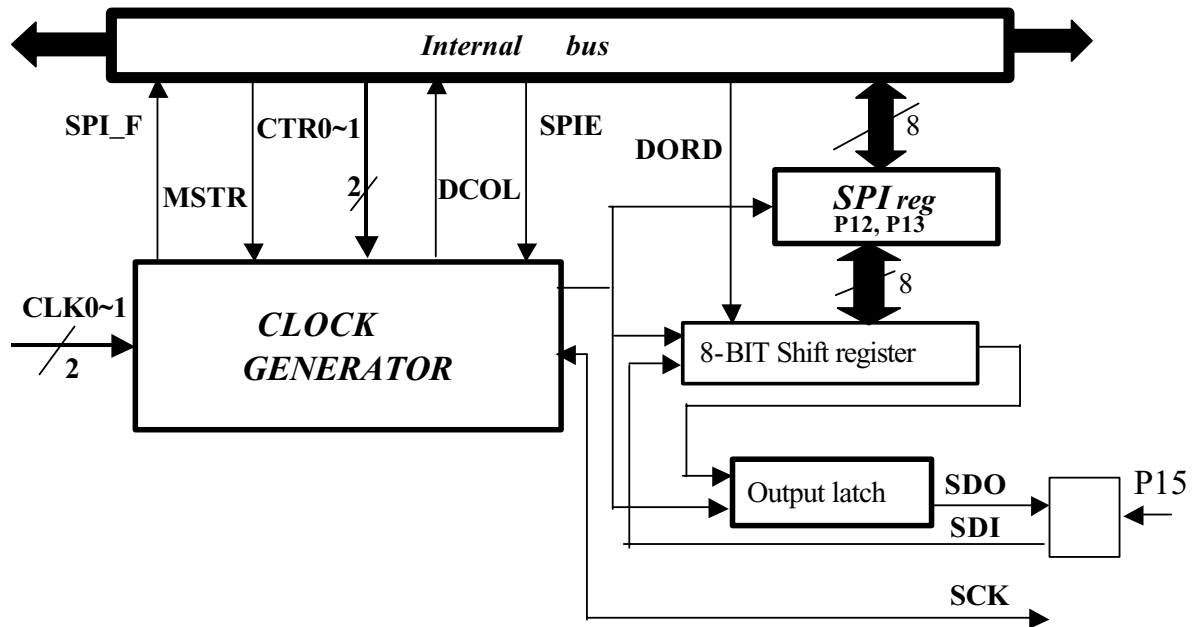
ADI:
INA     P12
STA     ADBUF
INA     P13
STA     ADBUF
RET

```

Preliminary

SERIAL PERIPHERAL INTERFACE (SPI)

The Serial Peripheral Interface (SPI) circuitry consists of two control register P18, P24, one data register (P12, P13), one shift register. The MSTR select the source of the serial clock from the internal or the external clock. at the same time, only transfer can occur or receive can occur. The SPI is available for the NORMAL operation mode.



SPI Control Register :

P24(Bit)	3	2	1	0
SYMBOL	MSTR	DORO	CLKS1	CLKS0

CLKS0~CLKS1: SPI transmission clock rate select

This is the clock rate selection bits, on master mode, we have four Kinds of rate can select.

Clock Rate	P24(1,0 BIT)	
	CLKS1	CLKS0
$F_c/2^5$	0	0
$F_c/2^6$	0	1
$F_c/2^7$	1	0
$F_c/2^8$	1	1

DORD: Data transmission order

0: LSB first

the data in the 8-bit shift register is shifted in/out LSB first

1: MSB first

the data in the 8-bit shift register is shifted in/out LSB first

Preliminary

MSTR: master or slave mode select

0: Master mode

SPI is in master mode and SCK is configured as an output pin.

SPI clock source is internal clock.

1: slave mode

SPI is in slave mode and SCK is configured as an input pin.

SCK receives the serial clock externally.

P18(Bit)	3	2	1	0
SYMBOL	SPIE	*	CTR1	CTR0

SPIE: Serial Peripheral Interface Enable

1: Serial Peripheral Interface Enable

0: Serial Peripheral Interface disable

P12, P13 control table

CTR1	CTR0	Select resume
0	0	HTC counter
0	1	A/D converter
1	0	SPI shift data
1	1	Unused

SPI control bit:

SPI_F (P14.1): SPI control flag

when SPI register (P12, P13) is empty SPI_F clear 0

when SPI register (P12, P13) is full, SPI_F set 1

P3(Bit)	3	2	1	0
SYMBOL	DCOL	*	ROM bank select	

DCOL (P3.3): SPI control flag

When SPI shift register is empty DCOL clear 0.

When SPI shift register is full DCOL set 1.

SDO: Serial data out (share with P15.0)

When MSTR set to 0 , SDO is an output pin, share with P15.0,

When the SPI is enable , data are shift out form SDO (P15.0)

SDI: Serial data out (share with P15.0)

When MSTR set to 1 , SDI is an input pin, share with P15.0,

When the SPI is enable , data are shift in form SDI (P15.0)

SCK: Serial Clock (share with P15.1)

The SCK pin for synchronization of both input and output data stream through SDI and SDO pins. When the MSTR is set, SCK become an output and the Serial clock is supplied to the internal system. When the MSTR is clear, SCK become an input and the Serial clock is supplied to the external system. The clock speed in slave mode is dependent upon the speed of the external system and has a maximum speed up till the internal system clock.

Preliminary

SCK: Serial Clock (share with P15.1)

The SCK pin for synchronization of both input and output data stream through SDI and SDO pins. When the MSTR is set, SCK become an output and the Serial clock is supplied to the internal system. When the MSTR is clear, SCK become an input and the Serial clock is supplied to the external system. The clock speed in slave mode is dependent upon the speed of the external system and has a maximum speed up till the internal system clock.

PROGRAM EXAMPLE :

transmission 16 bit (ABAB H) serial data LBS first, clock rate $F_c/2^8$ ($F_c=4\text{MHz}$)

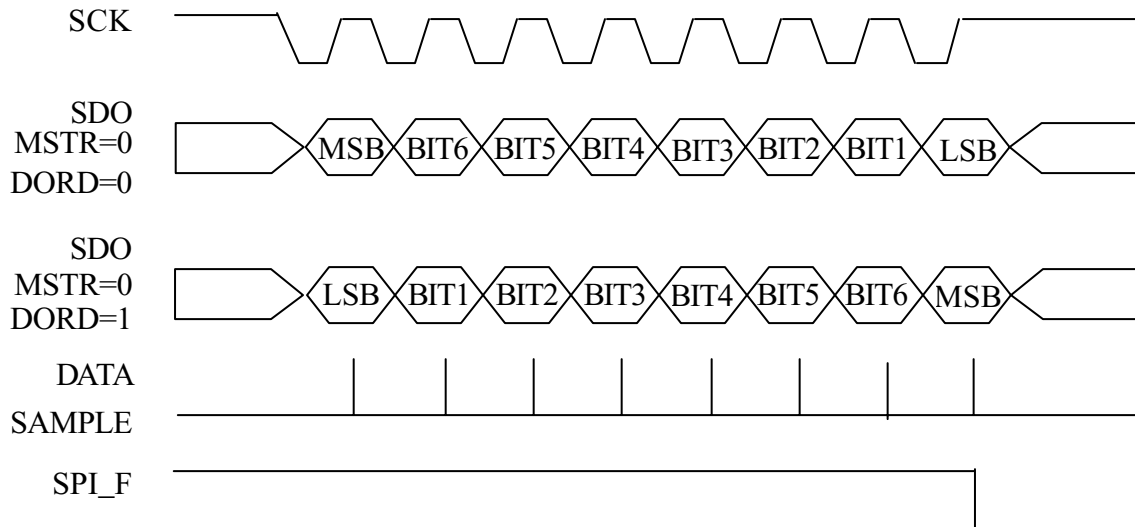
```

        LDIA        #1010B
        OUTA        P18          ; enable SPI & P12,P13a SPI
        LDIA        #0011B
        OUTA        P24          ; transmission LBS first &  $F_c/2^8$  clock rate
        LDIA        #0AH
        OUTA        P13          ; 0AH → P13,
        LDIA        #0BH
        OUTA        P12          ; 0BH → P12
        SEP         P14,1       ; SPI register (P12, P13) is full
NEXT:
        TTP         P14,1
        B           NEXT        ; wait SPI register is empty and input next data (8 bits)
        LDIA        #0AH
        OUTA        P13          ; 0AH → P13
        LDIA        #0BH
        OUTA        P12          ; 0BH → P12
        SEP         P14,1
NEXT1:
        TTP         P14,1       ; wait SPI register is empty and input next data (8 bits)
        B           NEXT1
NEXT2:
        TTP         P3.3
        B           NEXT2       ; wait all data transfer over
        LDIA        #0
        OUTA        P18          ; SPI disable
    
```

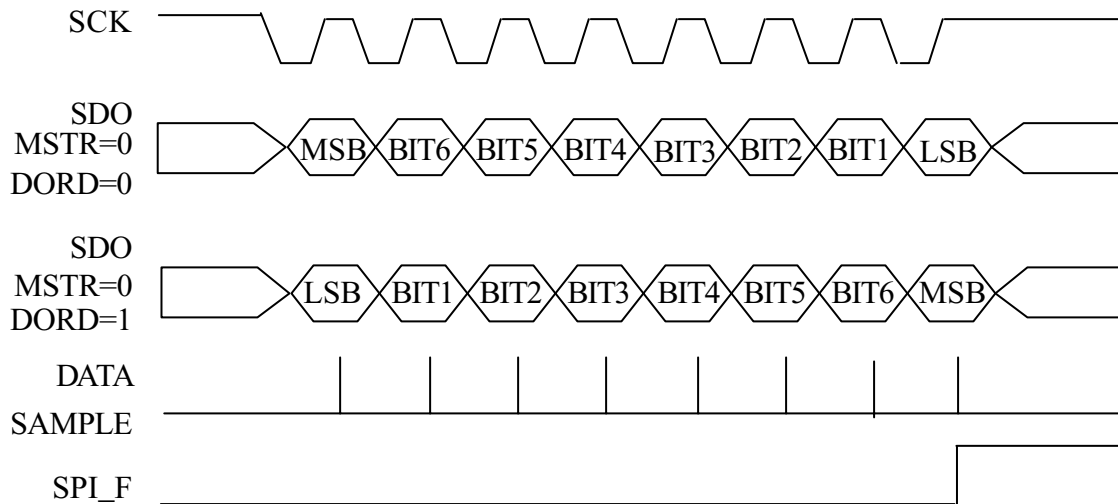
Preliminary

SPI TIMING DIAGRAM

DATA OUTPUT TIMING



DATA INPUT TIMING



Preliminary

LCD DRIVER

EM73P968 can directly drive the liquid crystal display (LCD) and has 52 segment and 4 or 5 common output pins by mask option. There are total 52x4 or 52x5 dots can be display. The VRLC pin is the LCD driver power input, there is the voltage of (VCC-VRLC) to LCD.

P17.0 share with com 4. When the mask option select 1/4 duty, the P17.0 is an output pin and LCD have 4 common. When the mask option select 1/5 duty, the P17.0 is a LCD pin and LCD have 5 common.

LCD driver control command register (P27) :

Port27 3 2 1 0 Initial value : 0000

LDC	*	*	*
------------	---	---	---

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

* : Don't care.

Example :

LDIA #1000B ; enable LCD, reference voltage of LCD is 1.5V.

OUTA P27

:

LDIA #0000B ; disable LCD

OUTA P27

LCD RAM

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																		
20H-2CH	COM0																																																	
30H-3CH	COM1																																																	
40H-4CH	COM2																																																	
50H-5CH	COM3																																																	
60H-6CH	COM4																																																	
SEGO1	SEGO2	SEGO3	SEGO4	SEGO5	SEGO6	SEGO7	SEGO8	SEGO9	SEGO10	SEGO11	SEGO12	SEGO13	SEGO14	SEGO15	SEGO16	SEGO17	SEGO18	SEGO19	SEGO20	SEGO21	SEGO22	SEGO23	SEGO24	SEGO25	SEGO26	SEGO27	SEGO28	SEGO29	SEGO30	SEGO31	SEGO32	SEGO33	SEGO34	SEGO35	SEGO36	SEGO37	SEGO38	SEGO39	SEGO40	SEGO41	SEGO42	SEGO43	SEGO44	SEGO45	SEGO46	SEGO47	SEGO48	SEGO49	SEGO50	SEGO51

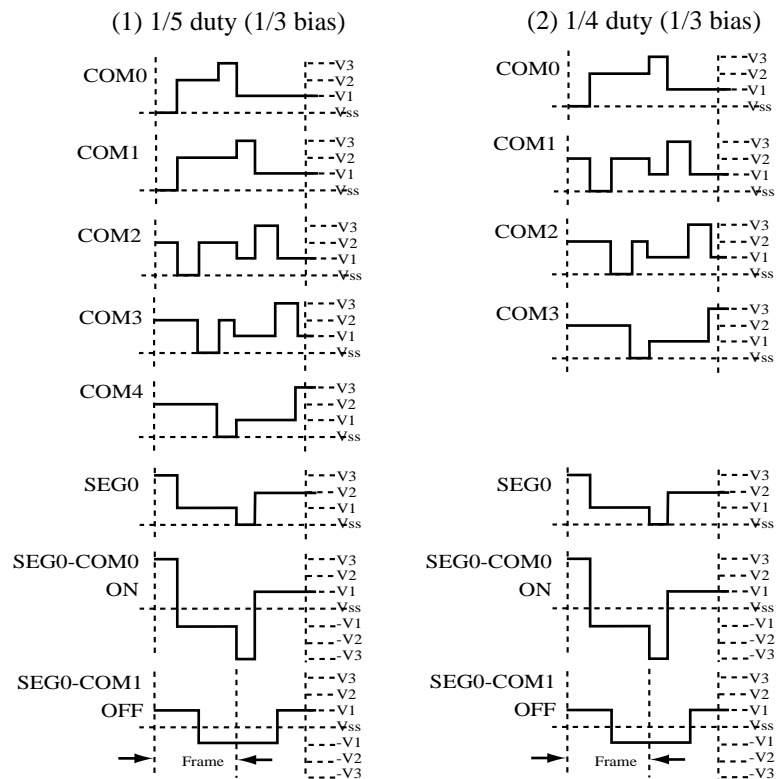
Preliminary

Driving Method	RAM address	SEG0	SEG1	SEG2	SEG3
		bit0	bit1	bit2	bit3
1/5 duty	1/4 duty	20H			
		30H			
		40H			
		50H			
	-	60H			

Driving Method	RAM address	SEG4	SEG5	SEG6	SEG7
		bit0	bit1	bit2	bit3
1/5 duty	1/4 duty	21H			
		31H			
		41H			
		51H			
	-	61H			

⋮

Driving Method	RAM address	SEG48	SEG49	SEG50	SEG51
		bit0	bit1	bit2	bit3
1/5 duty	1/4 duty	2CH			
		3CH			
		4CH			
		5CH			
	-	6CH			



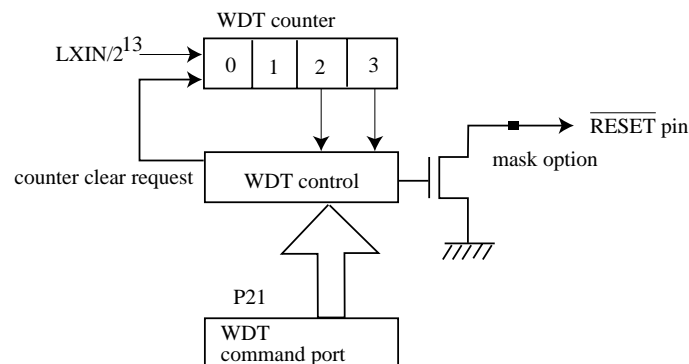
Preliminary

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	*	*	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

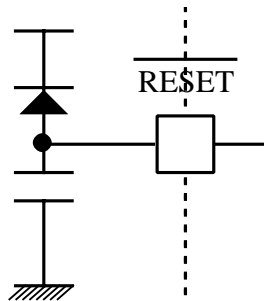
RESETTING FUNCTION

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, 12, 13, 14, 16, 19, 20, 21, 22, 25, 27, 28, 29	00h
P0, 1, 2, 4, 5, 6, 7, 8, 11, 15, 17, 30, 31	0Fh
LXIN, XIN	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

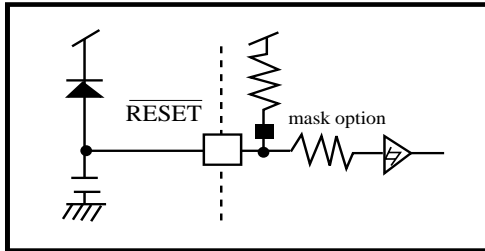
EM73P968 I/O PORT DESCRIPTION :

Port	Input function	Output function	Note
0	E Input port, wakeup function		
1	E Input port, wakeup function	E Output port	
2	E Input port, wakeup function	E Output port	
3	I ROM bank selection	I ROM bank selection, P3.3 SPI use	
4	E Input port, wakeup function	E Output port	
5	E Input port, wakeup function	E Output port	
6	E Input port, wakeup function share with A/D input	E Output port	
7	E Input port, wakeup function share with A/D input	E Output port	
8	E Input port, wakeup function, external interrupt input	E Output port, P8.0($\overline{\text{INT1}}$), P8.1(TRGB), P8.2($\overline{\text{INT0}}$), P8.3(TRGA)	
9	I RAM bank selection	I RAM bank selection	
10	I General purpose register	I General purpose register	
11	E Input port, wakeup function	E Output port	
12	SPI input data register	I High speed counter register share with SPI output data, A/D resolution data	Low nibble
13	SPI input data register	I High speed counter register share with SPI output data, A/D resolution data	High nibble
14	I CPU status	I CPU status, interrupt source selector	
15	E Input port, wakeup function P15.0 input data with SPI, P15.1 input clock with SPI	E Output port, P15.0 output data with SPI, P15.1 output clock with SPI	
16		I STOP mode control register	
17		I Output port P17.0/COM4 P17.1-P17.3 A/D control register	
18		I Interrupt status register P12, P13 control register	
19		I IDLE mode control register	
20		I HTC control register	
21		I WDT control register	
22		I NORMAL/SLOW mode control register	
23		I ADC control register	
24		I SPI control register	
25		I Timebase control register	
26		I A/D control register	
27		I LCD control register	
28		I Timer / counter A control register	
29		I Timer / counter B control register	
30		I Output port / SEG(51..48)	
31		I Output port / SEG(47..44)	

Preliminary

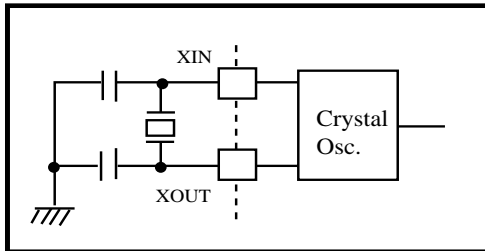
RESET PIN TYPE

TYPE RESET-A

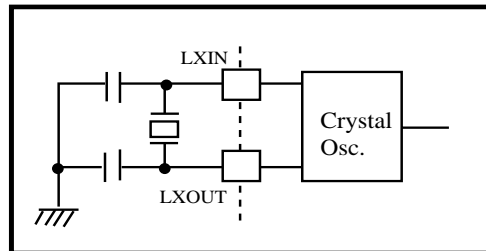


OSCILLATION PIN TYPE

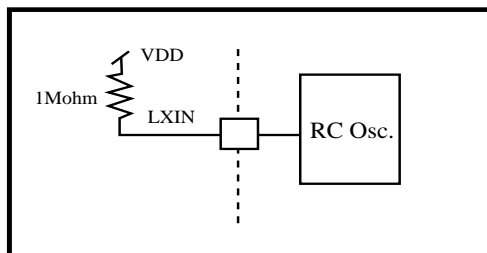
TYPE OSC-A



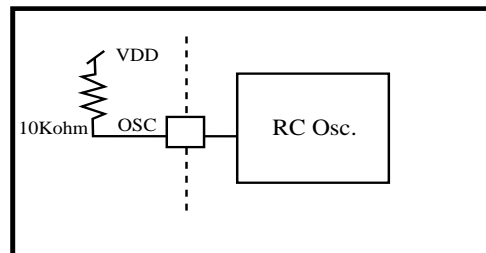
TYPE OSC-B



TYPE OSC-H1 (Low frequency)

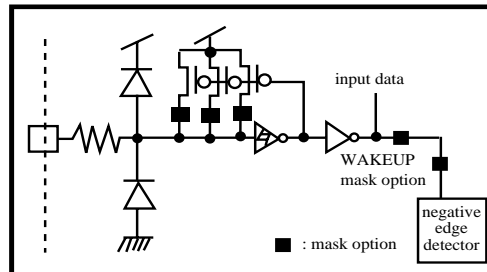


TYPE OSC-H2 (High frequency)



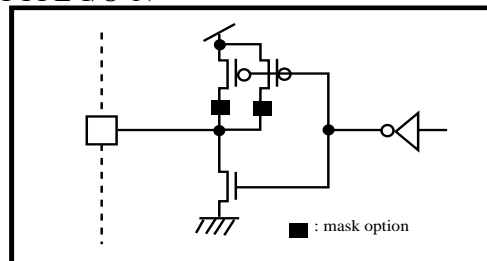
INPUT PIN TYPE

TYPE INPUT-K

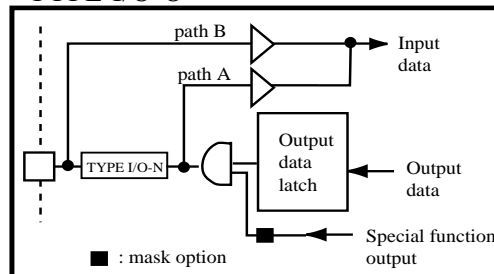


I/O PIN TYPE

TYPE I/O-N



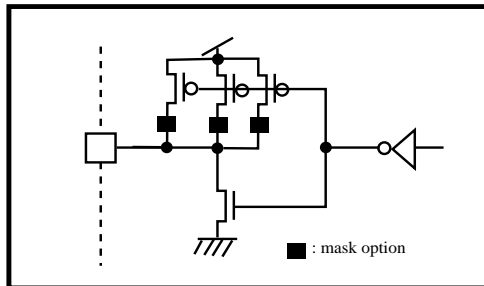
TYPE I/O-O



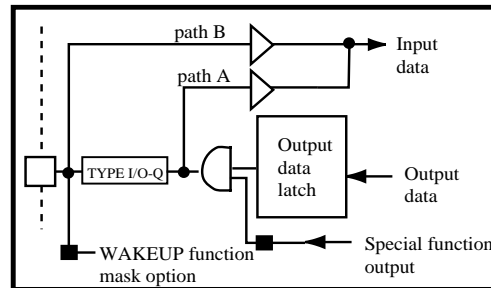
* This specification are subject to be changed without notice.

Preliminary

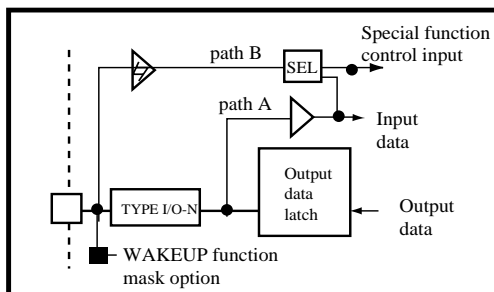
TYPE I/O-Q



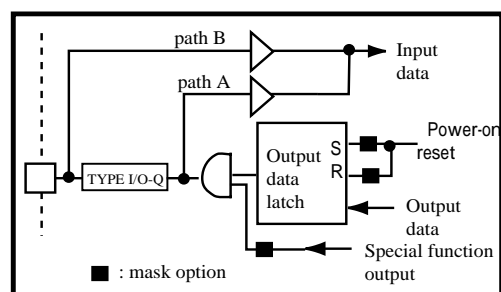
TYPE I/O-R1



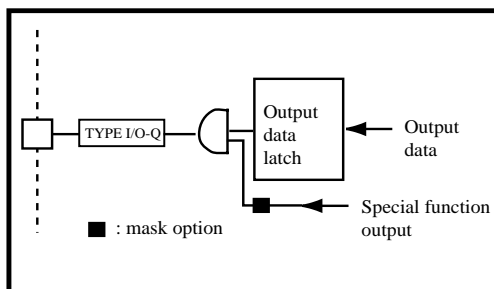
TYPE I/O-S



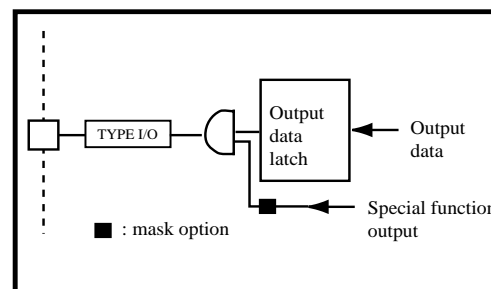
TYPE I/O-Z



OUTPUT-L



OUTPUT-M



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

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	$T_{OPR}=50^{\circ}C$
Operating Temperature	T_{OPR}	-30°C to 70°C	
Storage Temperature	T_{STG}	-55°C to 125°C	

RECOMMENDED OPERATING CONDITIONS

Items	Sym.	Ratings			Condition
			Min.	Max.	
Supply Voltage	V_{DD}	Normal	2.2V	6.0V	4MHz by RC osc
		Slow	2.2V		
		Idle	2.2V		
		Stop	2.0V		
Input Voltage schmitt circuit	V_{IH}	0.80x V_{DD} to V_{DD}			$V_{DD} : 2.0\sim 5.5V$
	V_{IL}	0V to 0.20 to V_{DD}			
Operating Frequency	F_C	4MHz			Osc
	F_s	32KHz			LXIN, LXOUT (crystal osc)

AD CONVERTER CHARACTERISTICS ($V_{AD}=5.0V$, $V_{REF}=5.0V$, $V_{SS}=0V$)

Characteristic	Sym.	Min.	Max.	Unit	Condition
Resolution	-	8	8	bit	
Conversion range		V_{SS}	V_{AD}	V	$V_{AD}=5V$
Quantization error			± 1	LSB	
Sampling rate			10	CLK	$V_{DD}=5V$
A/D supply current	AIDD1	-	1.0	mA	ADEN=0
	AIDD2		5	μA	ADEN=1
Analog input impedance	RAN		3	M Ω	
Vref current	AIfref	-	0.2	mA	

Preliminary

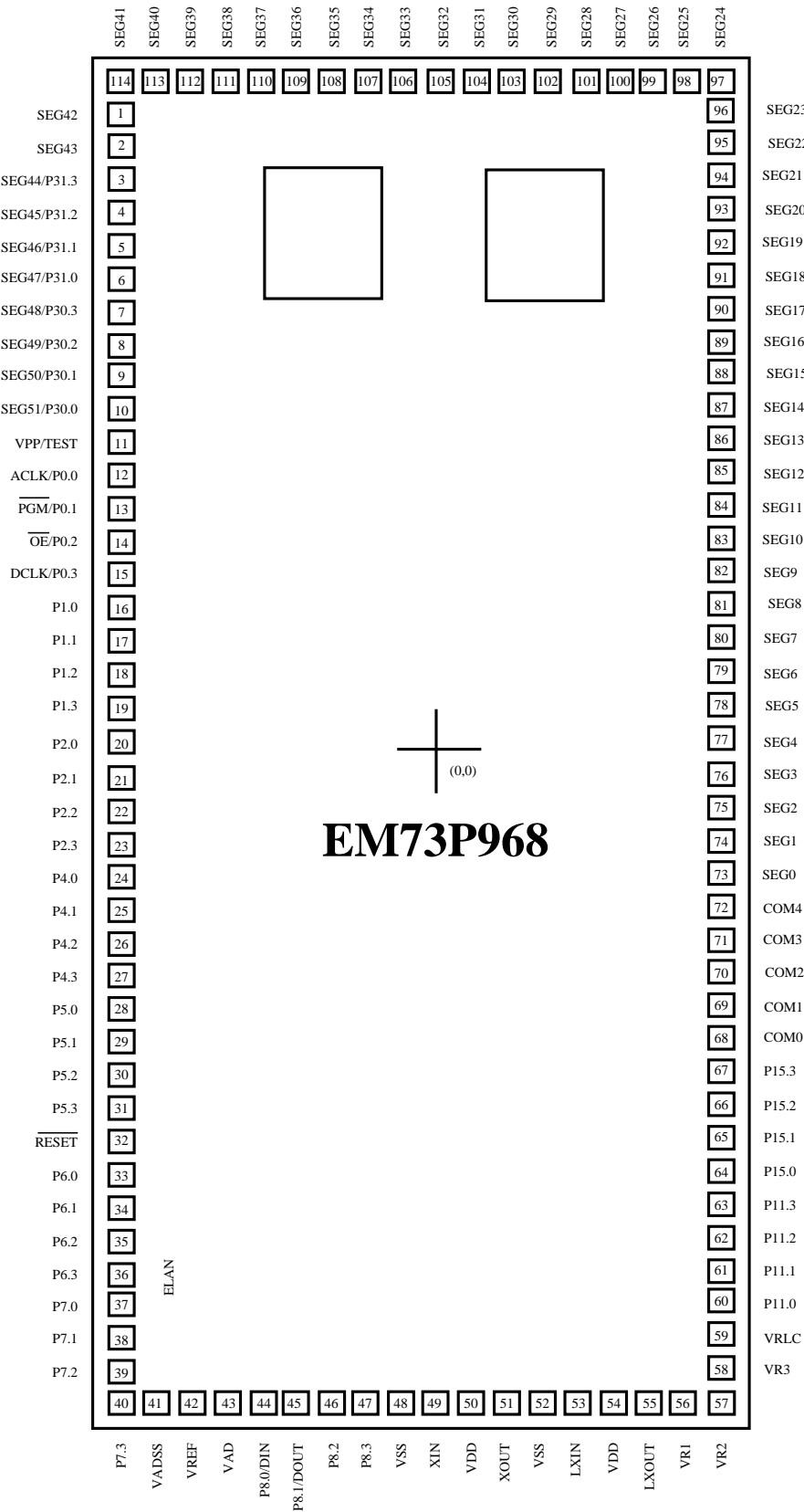
DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5\pm 0.5V$, $V_{SS}=0V$, $T_{OPR}=25^{\circ}C$)

Parameters		Sym.	Min.	Typ.	Max.	Unit	Conditions	
Supply current	I_{DD_Xtal}	-	1.5	2	mA	$V_{DD}=5.5V$, no load, NORMAL mode, $F_c=4MHz$, $F_s=32KHz$ (crystal)		
		-	100	150	μA	$V_{DD}=5.5V$, no load, SLOW mode, $F_s=32KHz$ (crystal)		
		-	80	100	μA	$V_{DD}=5.5V$, no load, $R_{V_{RLC}}=68K$, IDLE mode, $F_s=32KHz$ (crystal)		
		-	0.1	1	μA	$V_{DD}=5.5V$, STOP mode (crystal)		
	I_{DD_RC}	-	650	1000	μA	$V_{DD}=5.5V$, no load, NORMAL mode, $F_c=4MHz$, $F_s=32KHz$ (RC, OSC)		
		-	80	120	μA	$V_{DD}=5.5V$, no load, SLOW mode, $F_s=32KHz$ (RC, OSC)		
		-	45	70	μA	$V_{DD}=5.5V$, no load, $R_{V_{RLC}}=68K$, IDLE mode, $F_s=32KHz$ (RC, OSC)		
Hysteresis voltage	V_{HYS+}	$0.50V_{DD}$	-	$0.75V_{DD}$	V	RESET, all I/O ports		
	V_{HYS-}	$0.20V_{DD}$	-	$0.40V_{DD}$	V			
Input current	I_{IH}	-	-	± 1	μA	RESET, P0, $V_{DD}=5.5V$, $V_{IH}=5.5/0V$		
		-	-	± 1	μA	Open-drain, $V_{DD}=5.5V$, $V_{IH}=5.5/0V$		
	High current	I_{IL1}	11	14	18	mA	P1, P2	I/O port acts as input(push-pull), optional, $V_{DD}=4.5V$, $V_{IL}=0.2V$
	High current	I_{IL}	450	550	650	μA	P0,	
	Normal current		50	60	80	μA	P4~P8,	
Low current	20		24	28	μA	P11, P15		
Output Voltage	V_{OH}	2.2	-	-	V	$V_{DD}=4.5V$, see I_{OH} =typical. for P4~P8, P11, P15, P30, P31		
		-	-	0.2	V	$V_{DD}=4.5V$, $I_{OL}=0.5mA$, P1, P2, P4, P7, P8, P11, P15, P17.0, P30, P31		
	-	-	1.0	V	$V_{DD}=4.5V$, $I_{OL}=16mA$, P5, P6			
	High current	I_{OH1}	9	11	14	mA	P1, P2	$V_{DD}=4.5V$, $V_{OH}=2.2V$
	Normal current		45	55	65	μA		
	Low current		18	22	27	μA		
	High current	I_{OH}	400	450	500	μA	P4~P8,	
	Normal current		45	55	65	μA	P11, P15, P30, P31,	
	Low current		16	20	25	μA	optional	
	Normal current	45	55	65	μA	P17.0		
Leakage current	I_{LO}	-	-	1	μA	Open-drain, $V_{DD}=5.5V$, $V_o=5.5V$		
Input resistor	R_{IN}	-	-	-	K Ω	RESET		
High Frequency Variation			20	30	%	$V_{DD}=2.2\sim 5.5V\pm 10\%$ RC OSC $R=100K\pm 2\%$, $f_c=4MHz$		
Low Frequency Variation			20	30	%	$V_{DD}=2.2\sim 5.5V\pm 10\%$ RC OSC $R=1M\Omega\pm 2\%$, $f_s=32KHz$		

Note : RESET pin must add to a pull-up resistor.

Preliminary

PAD DIAGRAM



* This specification are subject to be changed without notice.

Preliminary

Pad No.	Symbol	X	Y
1	SEG42	-935.0	2080.0
2	SEG43	-935.0	1960.0
3	SEG44/P31.3	-935.0	1845.0
4	SEG45/P31.2	-935.0	1730.0
5	SEG46/P31.1	-935.0	1615.0
6	SEG47/P31.0	-935.0	1505.0
7	SEG48/P30.3	-935.0	1395.0
8	SEG49/P30.2	-935.0	1285.0
9	SEG50/P30.1	-935.0	1175.0
10	SEG51/P30.0	-935.0	1065.0
11	VPP/TEST	-935.0	955.0
12	P0.0/ACLK	-935.0	847.5
13	P0.1/PGM	-935.0	740.0
14	P0.2/ \overline{OE}	-935.0	632.5
15	P0.3/DCLK	-935.0	525.0
16	P1.0	-935.0	420.0
17	P1.1	-935.0	315.0
18	P1.2	-935.0	210.0
19	P1.3	-935.0	105.0
20	P2.0	-935.0	0.0
21	P2.1	-935.0	-105.0
22	P2.2	-935.0	-210.0
23	P2.3	-935.0	-315.0
24	P4.0	-935.0	-420.0
25	P4.1	-935.0	-525.0
26	P4.2	-935.0	-632.5
27	P4.3	-935.0	-740.0
28	P5.0	-935.0	-847.5
29	P5.1	-935.0	-955.0
30	P5.2	-935.0	-1065.0
31	P5.3	-935.0	-1175.0
32	\overline{RESET}	-935.0	-1285.0
33	P6.0	-935.0	-1395.0
34	P6.1	-935.0	-1505.0
35	P6.2	-935.0	-1615.0
36	P6.3	-935.0	-1730.0
37	P7.0	-935.0	-1845.0
38	P7.1	-935.0	-1960.0
39	P7.2	-935.0	-2080.0
40	P7.3	-935.0	-2200.0

Preliminary

Pad No.	Symbol	X	Y
41	VADSS	-815.0	-2200.0
42	VREF	-700.0	-2200.0
43	VAD	-590.0	-2200.0
44	P8.0/DIN	-480.0	-2200.0
45	P8.1/DOUT	-372.5	-2200.0
46	P8.2	-265.0	-2200.0
47	P8.3	-157.5	-2200.0
48	VSS	-52.5	-2200.0
49	XIN	52.5	-2200.0
50	VDD	157.5	-2200.0
51	XOUT	265.0	-2200.0
52	VSS	372.5	-2200.0
53	LXIN	480.0	-2200.0
54	VDD	590.0	-2200.0
55	LXOUT	700.0	-2200.0
56	VR1	815.0	-2200.0
57	VR2	935.0	-2200.0
58	VR3	935.0	-2080.0
59	VRLC	935.0	-1960.0
60	P11.0	935.0	-1845.0
61	P11.1	935.0	-1730.0
62	P11.2	935.0	-1615.0
63	P11.3	935.0	-1505.0
64	P15.0	935.0	-1395.0
65	P15.1	935.0	-1285.0
66	P15.2	935.0	-1175.0
67	P15.3	935.0	-1065.0
68	COM0	935.0	-955.0
69	COM1	935.0	-847.5
70	COM2	935.0	-740.0
71	COM3	935.0	-632.5
72	COM4	935.0	-525.0
73	SEG0	935.0	-420.0
74	SEG1	935.0	-315.0
75	SEG2	935.0	-210.0
76	SEG3	935.0	-105.0
77	SEG4	935.0	0.0
78	SEG5	935.0	105.0
79	SEG6	935.0	210.0
80	SEG7	935.0	315.0

Preliminary

Pad No.	Symbol	X	Y
81	SEG8	935.0	420.0
82	SEG9	935.0	525.0
83	SEG10	935.0	632.5
84	SEG11	935.0	740.0
85	SEG12	935.0	847.5
86	SEG13	935.0	955.0
87	SEG14	935.0	1065.0
88	SEG15	935.0	1175.0
89	SEG16	935.0	1285.0
90	SEG17	935.0	1395.0
91	SEG18	935.0	1505.0
92	SEG19	935.0	1615.0
93	SEG20	935.0	1730.0
94	SEG21	935.0	1845.0
95	SEG22	935.0	1960.0
96	SEG23	935.0	2080.0
97	SEG24	935.0	2200.0
98	SEG25	815.0	2200.0
99	SEG26	700.0	2200.0
100	SEG27	590.0	2200.0
101	SEG28	480.0	2200.0
102	SEG29	372.5	2200.0
103	SEG30	265.0	2200.0
104	SEG31	157.5	2200.0
105	SEG32	52.5	2200.0
106	SEG33	-52.5	2200.0
107	SEG34	-157.5	2200.0
108	SEG35	-265.0	2200.0
109	SEG36	-372.5	2200.0
110	SEG37	-480.0	2200.0
111	SEG38	-590.0	2200.0
112	SEG39	-700.0	2200.0
113	SEG40	-815.0	2200.0
114	SEG41	-935.0	2200.0

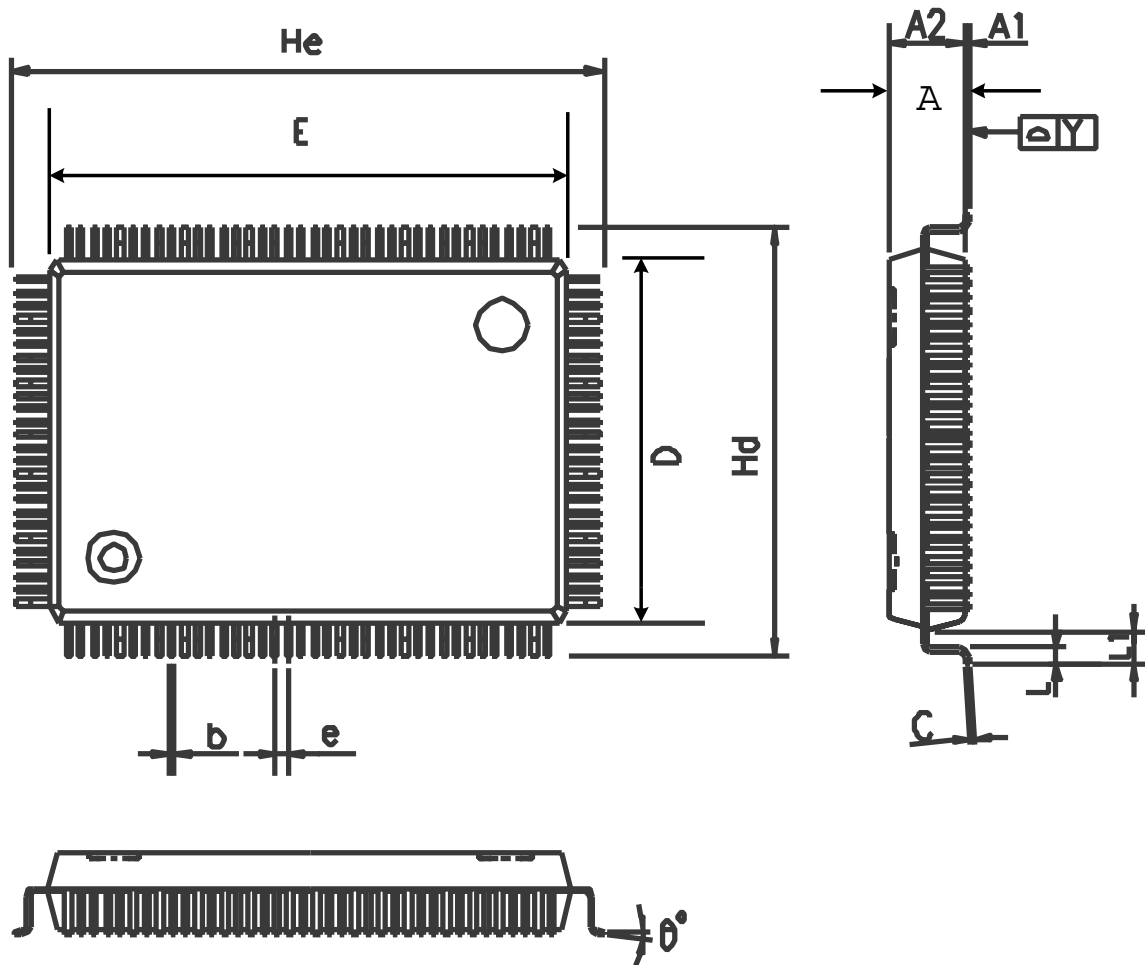
Unit : um

Chip size : 2130 x 4660um

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

Preliminary

PACKAGE DIMENSION



Symbol	Min	Normal	Max
A			3.400
A2	2.540	2.720	2.900
A1	0.250	0.350	0.450
b		0.2(TYP)	
c		0.15(TYP)	
D	13.900	14.000	14.100
Hd	17.000	17.200	17.400
E	19.900	20.000	20.100
He	23.000	23.200	23.400
L	0.650	0.800	0.950
L1	1.400	1.600	1.800
e		0.5(bsc)	
θ	0		7

All dimensions are in millimeters.

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'

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 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) 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] _b - 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 ← 0	1	1	-	-	1
CLP p,b	0110 1101 11bb pppp	PORT[p] _b ← 0	2	2	-	-	1
CLPL	0110 0000	PORT[LR _{3,2} +4]LR _{1,0} ← 0	1	2	-	-	1
CLR y,b	0110 1100 11bb yyyy	RAM[y] _b ← 0	2	2	-	-	1
SEM b	1111 01bb	RAM[HL] _b ← 1	1	1	-	-	1
SEP p,b	0110 1101 01bb pppp	PORT[p] _b ← 1	2	2	-	-	1
SEPL	0110 0010	PORT[LR _{3,2} +4]LR _{1,0} ← 1	1	2	-	-	1
SET y,b	0110 1100 01bb yyyy	RAM[y] _b ← 1	2	2	-	-	1
TF y,b	0110 1100 00bb yyyy	SF ← RAM[y] _b '	2	2	-	-	*
TFA b	1111 10bb	SF ← Acc _b '	1	1	-	-	*
TFM b	1111 11bb	SF ← RAM[HL] _b '	1	1	-	-	*
TFP p,b	0110 1101 00bb pppp	SF ← PORT[p] _b '	2	2	-	-	*
TFPL	0110 0001	SF ← PORT[LR _{3,2} +4]LR _{1,0} '	1	2	-	-	*
TT y,b	0110 1100 10bb yyyy	SF ← RAM[y] _b	2	2	-	-	*
TTP p,b	0110 1101 10bb pppp	SF ← PORT[p] _b	2	2	-	-	*

(9) Subroutine

Mnemonic	Object code (binary)	Operation description	Byte	Cycle	Flag		
					C	Z	S
LDCALL a	0100 0aaa aaaa aaaa	STACK[SP] ← PC, SP ← SP - 1, PC ← a	2	2	-	-	-
SCALL a	1110 mnm	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

* 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 _{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