GENERAL DESCRIPTION

EM73866 is an advanced single chip CMOS 4-bit micro-controller. It contains 8K-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. EM73866 also contains 6 interrupt sources, 2 input port, 7 bidirection ports, Max LCD display (32x4), built-in watch-dog-timer and high speed Timer/Counter.
EM73866 has plentiful operating modes (SLOW, IDLE, STOP) intended to reduce the power consumption.

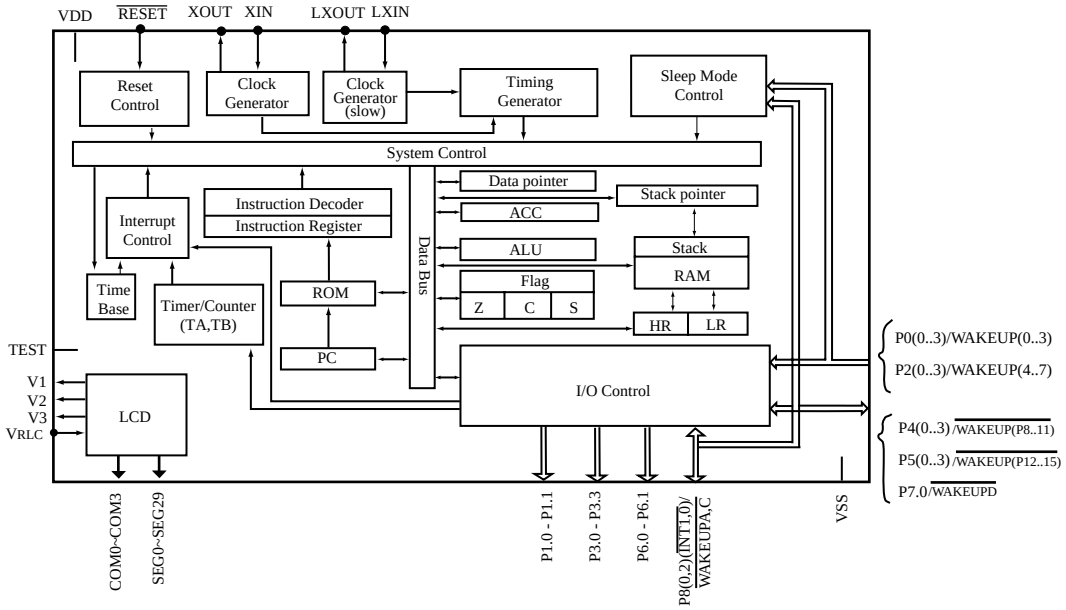
FEATURES

- 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).
  244 μs for 32768 Hz (low speed clock).
  122 μs for 32768 Hz (low speed clock with frequency Double)
- ROM capacity : 8192 X 8 bits.
- RAM capacity : 500 X 4 bits.
- Input port : 2 ports (P0, P2), P0(0..3), P2 (0..3), IDLE/STOP releasing function are available by mask option.
- Bidirection port : 7 ports (P1, P3, P4, P5, P6, P7, P8). P4.1 is shared with HTC external input. IDLE/STOP releasing function are available by mask option for P8(0..3).
- 12-bit timer/counter : Two 12-bit timer/counters are programmable for timer, event counter and pulse width measurement.
- 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, melody out or pulse width measurement.
- Subrountine 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 : Max 32 X 4 dots, 1/4, 1/3, 1/2 static four kinds of duty selectable, 1/2 or 1/3 bias choice and dynamic resistor available.
- Power saving function : SLOW, IDLE, STOP operation mode.
- Package type : Chip form 76 pins.

APPLICATIONS

EM73866 is suitable for application in family applicance, consumer products, hand held games, calculator and the toy controller.

FUNCTION BLOCK DIAGRAM



PIN DESCRIPTIONS

Symbol	Pin-type	Function
V_{DD}		Power supply (+)
V_{SS}		Power supply (-)
$\overline{RESET}$	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) P2(0..3)/WAKEUP(4..7)	INPUT-K	8-bit input pins with IDLE/STOP releasing function mask option : wakeup enable, negative edge release, pull-up wakeup enable, negative edge release, none wakeup enable, positive edge release, pull-down wakeup enable, positive edge release, none wakeup disable, pull-up wakeup disable, pull-down wakeup disable, none
P1(0..1)	I/O-Z	2-bit bidirection I/O pins with high current function source mask option 1: initial low initial high mask option 2: low current push-pull normal current push-pull high current push-pull NMOS open-drain PMOS open-drain

PIN DESCRIPTIONS

Symbol	Pin-type	Function
P3(0,1)/SEG(30,31)	I/O-O	2-bit bidirection I/O pins are shared with LCD segment pin mask option : segment pin low current push-pull normal current push-pull open-drain
P3(2,3), P6(0,1)	I/O-N	4-bit bidirection I/O pins mask option : low current push-pull normal current push-pull open-drain
P4.0/$\overline{\text{SOUND}}$/$\overline{\text{WAKEUP8}}$	I/O-R1	1-bit bidirection I/O with inverse sound output and IDLE/STOP releasing function. mask option : wakeup disable, low current push-pull wakeup disable, normal current push-pull wakeup disable, high current push-pull wakeup disable, open-drain wakeup disable, $\overline{\text{SOUND}}$ wakup enable, low current push-pull wakeup enable, normal current push-pull
P4.1(TRGH)/$\overline{\text{WAKEUP9}}$	I/O-R1	1-bit bidirection I/O with HTC output and IDLE/STOP releasing function. mask option : wakeup disble, low current push-pull wakeup disable, normal current push-pull wakeup disable, high current push-pull wakeup disable, NMOS open-drain wakeup disable, PMOS open-drain wakeup enable, low current push-pull wakeup enable, normal current push-pull
P4(2,3)/$\overline{\text{WAKEUP(10,11)}}$	I/O-R1	2-bit bidirection I/O pins with IDLE/STOP releasing function mask option : wakeup disble, low current push-pull wakeup disable, normal current push-pull wakeup disable, high current push-pull wakeup disable, NMOS open-drain wakeup disable, PMOS open-drain wakeup enable, low current push-pull wakeup enable, normal current push-pull
P5(0..3)/$\overline{\text{WAKEUP(12..15)}}$	I/O-S	4-bit bidirection I/O pins with IDLE/STOP releasing function mask option : wakeup disable, low current push-pull wakeup disable, normal current push-pull wakeup disable, open-drain wakeup enable, low current push-pull wakeup enable, normal current push-pull
P7.0/TRGA/$\overline{\text{WAKEUPD}}$ P8.1/TRGB/$\overline{\text{WAKEUPB}}$	I/O-S	2-bit bidirection I/O pins with timer/counterA, B external intput and IDLE/STOP releasing function mask option : wakeup disable, low current push-pull wakeup disable, normal current push-pull wakeup disable, open-drain wakeup enable, low current push-pull wakeup enable, normal current push-pull

PIN DESCRIPTIONS

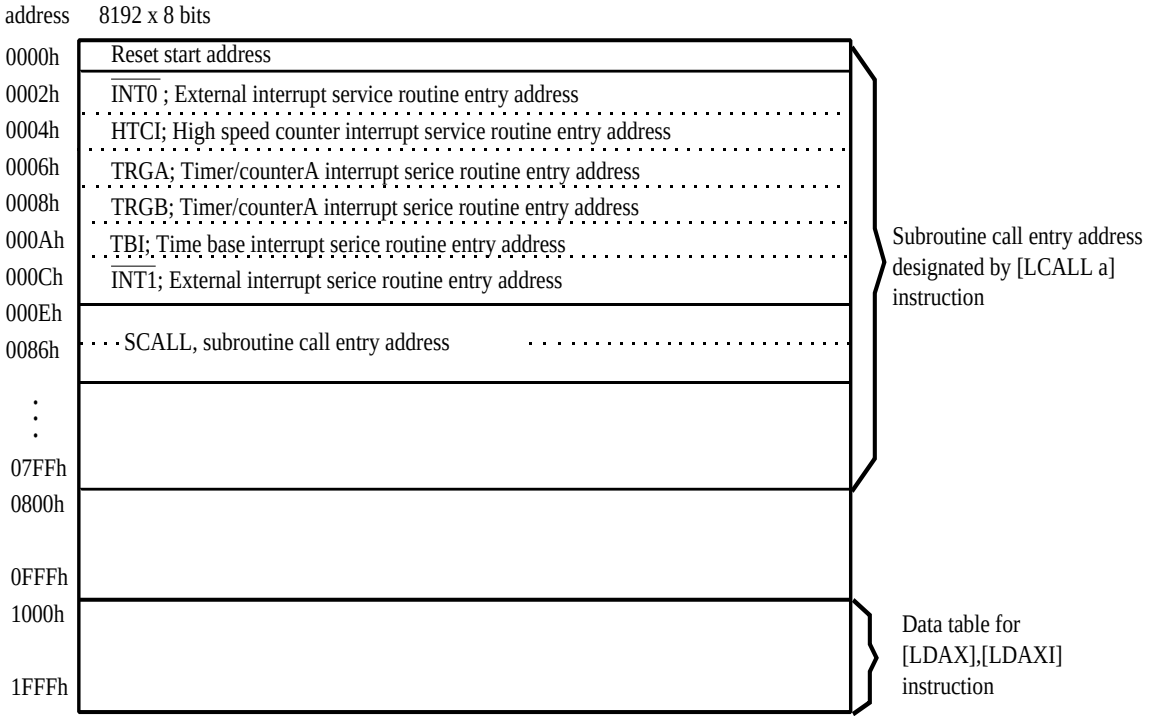
Symbol	Pin-type	Function
P8.0($\overline{INT1}$)/$\overline{WAKEUPA}$ P8.2($\overline{INT0}$)/$\overline{WAKEUPC}$	I/O-S	2-bit bidirection I/O pins with interrupt 0, 1 external intput and IDLE/STOP releasing function mask option : wakeup disable, low current push-pull wakeup disable, normal current push-pull wakeup disable, open-drain wakeup enable, low current push-pull wakeup enable, normal current push-pull
COM(0..3)	--	LCD common pins
SEG(0..29)	--	LCD segment pins
V1, V2, V3, V_{RLC}	--	LCD bias pins

FUNCTION DESCRIPTIONS

PROGRAM ROM (8K X 8 bits)

8 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.



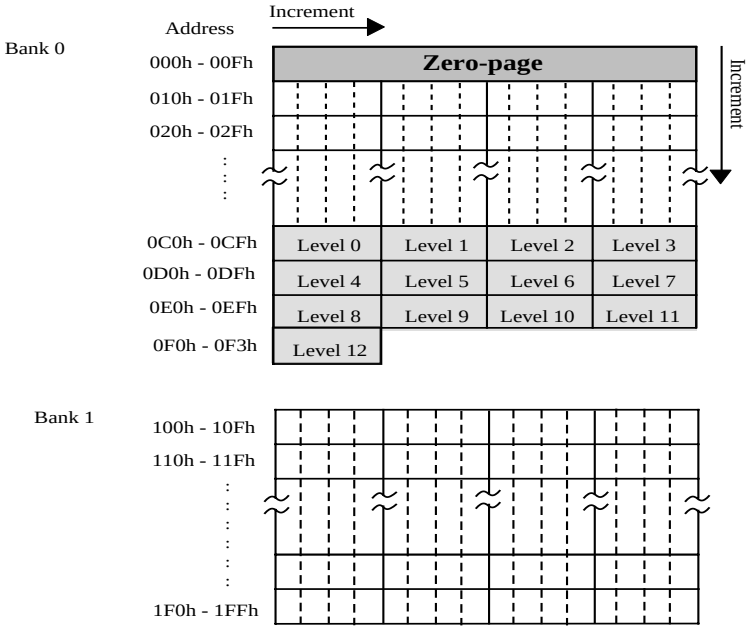
DP is a 12-bit data register that stores the program ROM address as pointer for the ROM code data. User has to initially load ROM address into DP with instructions "STADPL", and "STADPM, STADPH", then then to obtain the lower nibble of ROM code data by instruction "LDAX" and higher nibble by instruction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



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 (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.

SPECIAL PURPOSE REGISTER:

The instruction concerning with "Timer/counter", "Data Pointer" and "Stack Pointer" at instruction table 14 be sure the RAM bank must be set in Bank0.

DATA AREA:

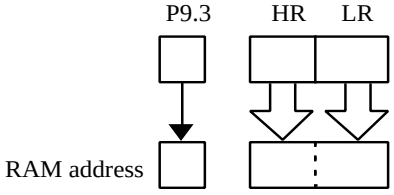
Except the area used by user's application, the whole RAM can be used as data area for storing and loading general data.

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.

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.
There are 3 addressing mode to access the data memory, namely -

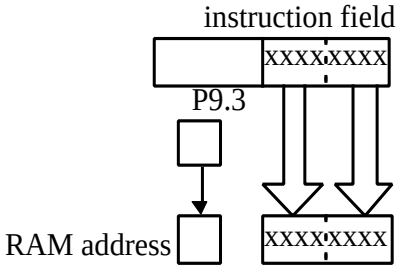
(1) Indirect addressing mode:
The address in the certain bank is specified by the HL registers.



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[143h]
CLP     P9,3            ; P9.3← 0
LDL     #2h             ; LR← 2
LDH     #3h             ; HR← 3
STAM                    ; RAM[032h]← Acc
```

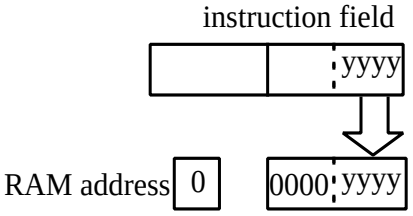
(2) Direct addressing mode:
The address in the bank is directly specified by 8 bits code of the second byte in the instruction field.



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 address is the lower 4 bits code of the second byte in the instruction field.This kind of instructions are only available for the zero page. Area in bank0, even the P9.3 is set.



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

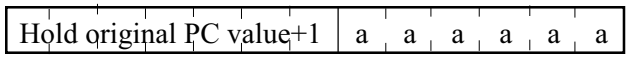
```
STD     #0Fh, 05h       ; RAM[05h]← 0Fh
```

PROGRAM COUNTER (8K ROM)

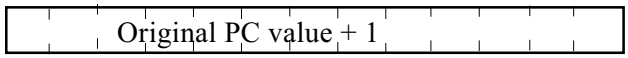
Program counter (PC) is composed by a 13-bit counter, which indicates the next executed address for the instruction of program ROM instruction.
For BRANCH and CALL instructions, PC is changed by instruction indicated. PC can only indicate the address from 0000h-1FFFh.

(1) Branch instruction:

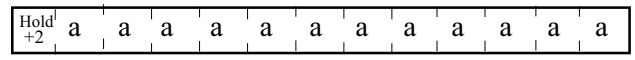
SBR a
Object code: 00aa aaaa
Condition: SF=1; PC ← PC $_{12\text{-}6.a}$ (branch condition satisified)



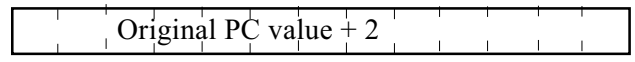
SF=0; PC← PC +1(branch condition not satisified)



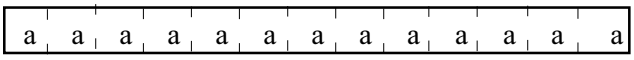
LBR a
Object code: 1100 aaaa aaaa aaaa
Condition: SF=1; PC ← PC $_{12.a}$ (branch condition satisified)



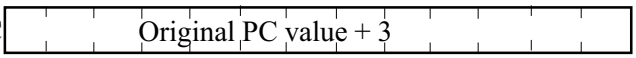
SF=0; PC← PC +2(branch condition not satisified)



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)

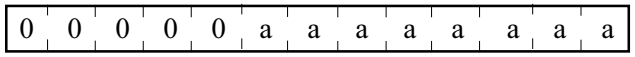


SF=0 ; PC ← PC + 3 (branch condition not satisified)



(2) Subrountine instruction:

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



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

PC	0	0	a	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 following:

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

PC	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	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 is a 4-bit data register for temporary data. For the arithematic, logic and comparative opertion .., ACC plays a role which holds the source data and result.

FLAGS

There are 3 kinds of flag, CF (Carry flag), ZF (Zero flag), SF (Status flag), these 3 1-bit flags are affected by the arithematic, logic and comparative .... operation.
All flags will be put into stack when an interrupt subrountine is served, and the flags will be restored after RTI instruction executed.

(1) Carry Flag (CF)

The carry flag is affected by following operation :

a. Addition : CF as a carry out indicator, when the addition operation has a carry-out, CF will be "1", in another word, if the operation has no carry-out, CF will be "0".

b. Subtraction : CF as a borrow-in indicator, when the subtraction operation must has a borrow-in, the CF will be "0", in another word, if no borrow-in, CF will be "1".

c. Comparision : CF is as a borrow-in indicator for Comparision operation as the same as 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 : For TFCFC instruction, the content of CF sends into SF then clear itself "0". For TTSFC instruction, the content of CF sends into SF then set itself "1".

(2) Zero Flag (ZF)

ZF is affected by the result of ALU, if the ALU operation generate a "0" result, the ZF will be "1", otherwise, the ZF will be "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 will be satisified, otherwise, branch condition will not be satisified by SF = 0.

PROGRAM EXAMPLE:
Check following arithematic operation for CF, ZF, SF

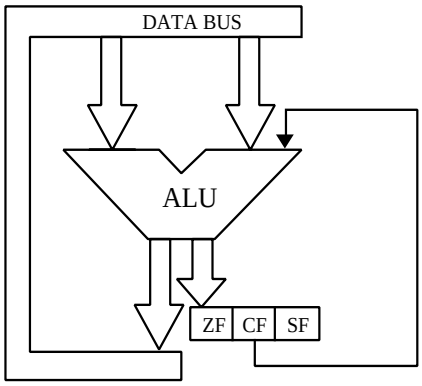
	CF	ZF	SF
LDIA #00h;	-	1	1
LDIA #03h;	-	0	1
ADDA #05h;	-	0	1
ADDA #0Dh;	-	0	0
ADDA #0Eh;	-	0	0

ALU

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

ALU STRUCTURE

ALU supported user arithematic operation function, including : addition, subtraction and rotaion.



ALU FUNCTION

(1) Addition:

For instruction ADDAM, ADCAM, ADDM #k, ADD #k,y .... ALU supports additional function. The additional operation can affect CF and ZF. For additional operation, if the result is "0", ZF will be "1", otherwise, not equal "0", 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:

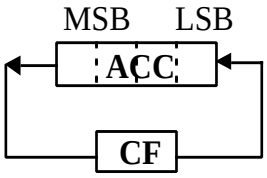
For instruction SUBM #k, SUBA #k, SBCAM, DECM... ALU supports user subtraction function. The subtraction operation can affect CF and ZF, For subtraction operation, if the result is negative, CF will be "0", it means a borrow out, otherwise, if the result is positive, CF will be "1". For ZF, if the result of subtraction operation is "0", the ZF will be "1", otherwise, ZF will be "1".

EXAMPLE:

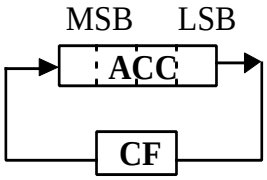
Operation	Carry	Zero
8-4=4	1	0
7-F= -8(1000)	0	0
9-9=0	1	1

(3) Rotation:

There are two kinds of rotation operation, one is rotation left, the other is rotation right.
RLCA instruction rotates Acc value to left, shift the CF value into the LSB bit of Acc and the shift out data will be hold in CF.



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



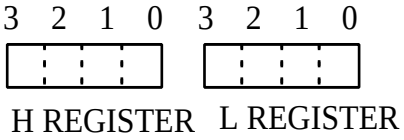
PROGRAM EXAMPLE: To rotate Acc right and shift a "1" into the MSB bit of Acc.

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 address of RAM memory and also 2 independent temporary 4-bit data registers. For some instruction, L register can be a pointer to indicate the pin number (Port4).

HL REGISTER STRUCTURE



HL REGISTER FUNCTION

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

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

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

PROGRAM EXAMPLE: Store immediate data #Ah into RAM of address 35h.
LDL #5h;
LDH #3h;
STDMI #0Ah; RAM[35] ← Ah

(3) For instruction : SELP, CLPL, TFPL, L regieter be a pointer to indicate the bit of I/O port.

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 which 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 accepted, the SP will be decreased one automatically, in another word, if returning from a subroutine, the SP will be increased one.
The data transfer between ACC and SP is by instruction of "LDASP" and "STASP". When "LDASP" and "STASP" are used, Port 9.3 must be set to 0.

DATA POINTER (DP)

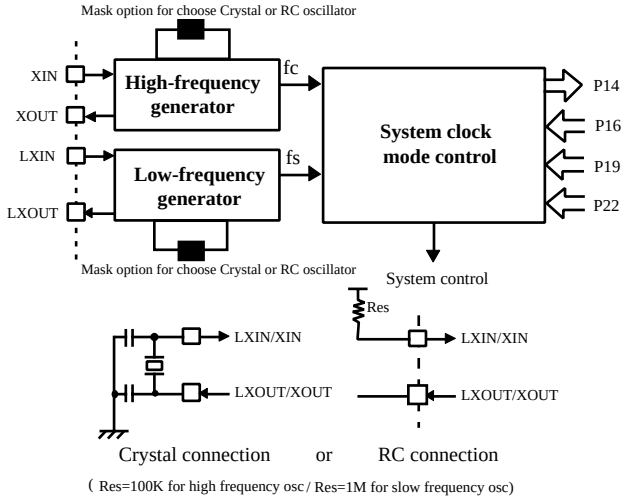
Data pointer is a 12-bit register which stores the address of ROM can indicate the ROM code data specified by user (refer to data ROM).
When "LDADPL", "LDADPM" and "LDADPH" are used, Port 9.3 must be set to 0.

CLOCK AND TIMING GENERATOR

The clock generator is supported by a dual clock system, the clock source comes from crystal (resonator) or RC oscillation for both high or low frequency osc. are decided by mask option.

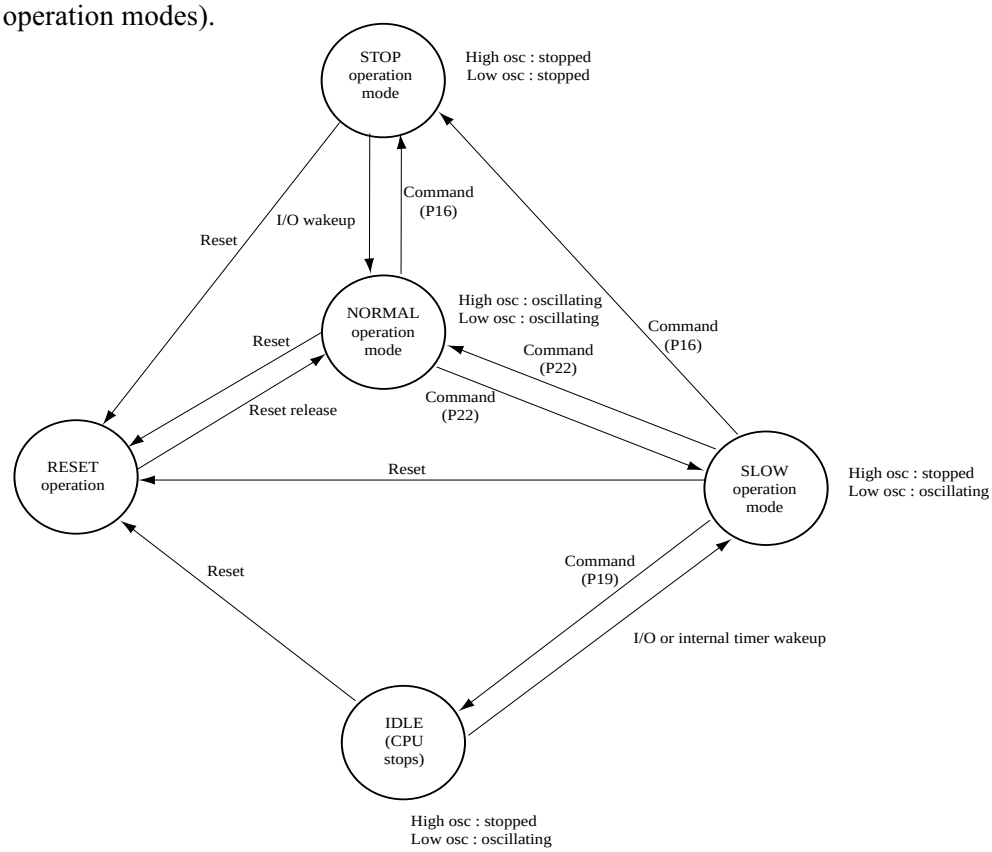
CLOCK GENERATOR STRUCTURE

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



SYSTEM CLOCK MODE CONTROL

The system clock mode controller can start or stop the high-frequency and low-frequency clock oscillator and switch between these two basic clocks. EM73866 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, High speed timer	8 / fc
SLOW	Low frequency	Low frequency clock	LCD, High speed timer	8 / fs
				8 / fs x 2 (freq.Double)
IDLE	Low frequency	CPU stops	LCD	-
STOP	None	CPU stops	All disable	-

NORMAL OPERATION MODE

The 4-bit µc is in the NORMAL operation mode when the CPU is reseted. This mode is a dual clock system (high-frequency and low-frequency clocks oscillating). It can be changed to SLOW or STOP operation mode by the command register (P22 or P16).
LCD display and high speed timer/counter with melody output are available for the NORMAL operation mode.

SLOW OPERATION MODE

The SLOW operation mode is a single clock system (low-frequency clock oscillating). It can be changed to the DUAL operation mode by the command register (P22), STOP operation mode by P16 and IDLE operation mode by P19.
LCD display and high speed timer/counter with melody output are available for the SLOW operation mode.

P22 Initial value : 0000

3	2	1	0
*	SOM		

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

P14 Initial value : *000

3	2	1	0
*	WKS	LFS	CPUS

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

CPUS	CPU status
0	NORMAL 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 to "1" when CPU is wake-up by internal timer. P14.2 will be cleared to "0" when user out data to P14.

IDLE OPERATION MODE

The IDLE operation mode suspends all SLOW operations except for the low-frequency clock and LCD driver. It retains the internal status with low power consumption without stopping the clock function 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)/$\overline{\text{WAKEUP}}$(0..3), P2(0..3)/$\overline{\text{WAKEUP}}$(4..7), P4(0..3)/$\overline{\text{WAKEUP}}$(8..11), P5(0..3)/$\overline{\text{WAKEUP}}$(12..15), P7.0/$\overline{\text{WAKEUPD}}$ or P8(0..2)/$\overline{\text{WAKEUP}}$(A..C)).

P19 3 2 1 0 Initial value : 0000

IDME	SIDR

IDME	Enable IDLE mode
0 1	Enable IDLE mode
* *	Reserved

SIDR	Select IDLE releasing condition
0 0	P0(0..3), P2(0..3), P4(0..3), P5(0..3), P7.0, P8(0..2) pin input
0 1	P0(0..3), P2(0..3), P4(0..3), P5(0..3), P7.0, P8(0..2) pin input and 1 sec signal (0.5 sec with frequency Double)
1 0	P0(0..3), P2(0..3), P4(0..3), P5(0..3), P7.0, P8(0..2) pin input and 0.5 sec signal (0.25 sec with frequency Double)
1 1	P0(0..3), P2(0..3), P4(0..3), P5(0..3), P7.0, P8(0..2) pin input and 15. 625 ms signal (7.8 ms with frequency Double)

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)/, P2 (0.. 3), P4(0..3), P5(0..3), P7.0 or P8(0..2)).

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/XIN for RC osc.
*	1 0 1	2^10/XIN for RC osc.
*	1 1 0	2^18/XIN for Crystal osc.
*	1 1 1	2^19/XIN for Crystal osc.

TIME BASE INTERRUPT (TBI)

The time base can be used to generate a fixed frequency interrupt. There are 8 kinds of frequencies can be selected by setting P25.

P25 3 2 1 0

initial value : 0000

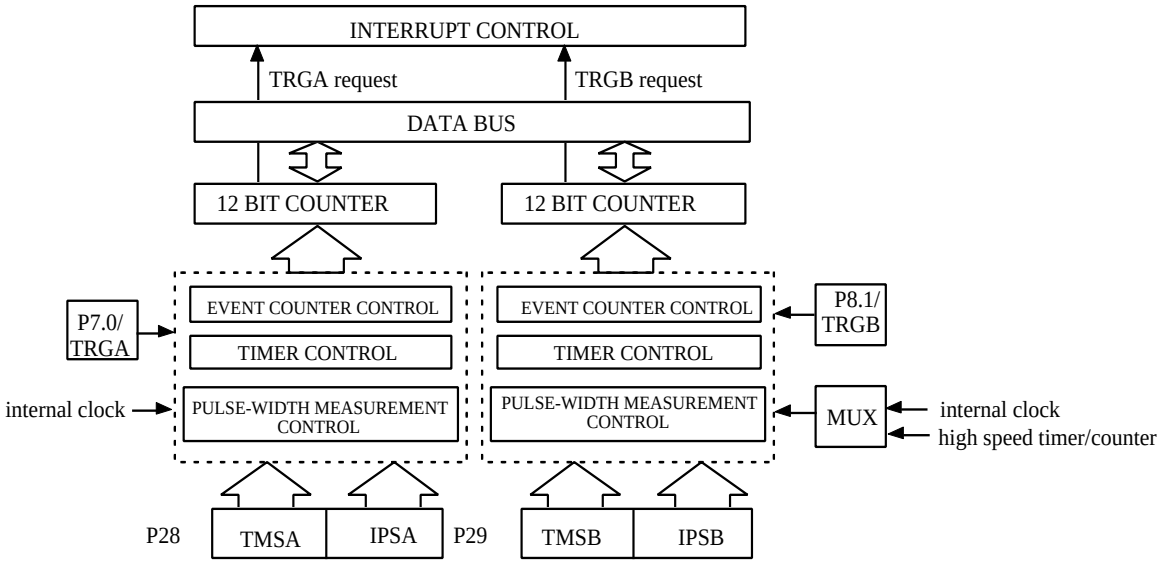
P25	Interrupt Rate			
	Low-frequency		With low-frequency double	
	NORMAL mode	SLOW mode	NORMAL mode	SLOW mode
0 0 x x	disable	disable	disable	disable
0 1 0 0	LXIN / 2^3 Hz	Reserved	LXIN / 2^2 Hz	Reserved
0 1 0 1	LXIN / 2^4 Hz	Reserved	LXIN / 2^3 Hz	Reserved
0 1 1 0	LXIN / 2^5 Hz	Reserved	LXIN / 2^4 Hz	Reserved
0 1 1 1	LXIN / 2^{14} Hz	LXIN / 2^{14} Hz	LXIN / 2^{13} Hz	LXIN / 2^{13} Hz
1 1 0 0	LXIN / 2^1 Hz	Reserved	LXIN / 2 Hz	Reserved
1 1 0 1	LXIN / 2^6 Hz	LXIN / 2^6 Hz	LXIN / 2^5 Hz	LXIN / 2^5 Hz
1 1 1 0	LXIN / 2^8 Hz	LXIN / 2^8 Hz	LXIN / 2^7 Hz	LXIN / 2^7 Hz
1 1 1 1	LXIN / 2^{10} Hz	LXIN / 2^{10} Hz	LXIN / 2^9 Hz	LXIN / 2^9 Hz
1 0 x x	Reserved	Reserved	Reserved	Reserved

TIMER / COUNTER (TIMERA, TIMERB)

Timer/counters can support user three special functions:
1. Even counter
2. Timer.
3. Pulse-width measurement.

These three functions can be executed by 2 timer/counter independently.
For timerA, the counter data is saved in timer register TAH, TAM, TAL, which user can set counter initial value and read the counter value by instruction "LDATAH(M,L), STATAH(M,L)" and timer register is TBH, TBM, TBL and W/R instruction "LDATBH (M,L), STATBH (M,L)".

The basic structure of timer/counter is composed by two same structure counter, these two counters can be set initial value and send counter value to timer register, P28 and P29 are the command ports for timerA and timer B, user can choose different operation mode and different internal clock rate by setting these two ports. When timer/counter overflow, it will generate a TRGA(B) interrupt request to interrupt control unit.



TIMER/COUNTER CONTROL

P8.1/TRGB, P7.0/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

IPSA	INTERNAL PULSE-RATE SELECTION			
	Low-frequency		With low-frequency double	
	NORMAL mode	SLOW mode	NORMAL mode	SLOW mode
0 0	LXIN/2^3 Hz	Reserved	LXIN/2^2 Hz	Reserved
0 1	LXIN/2^7 Hz	LXIN/2^7 Hz	LXIN/2^6 Hz	LXIN/2^6 Hz
1 0	LXIN/2^{11} Hz	LXIN/2^{11} Hz	LXIN/2^{10} Hz	LXIN/2^{10} Hz
1 1	LXIN/2^{15} Hz	LXIN/2^{15} Hz	LXIN/2^{14} Hz	LXIN/2^{14} Hz

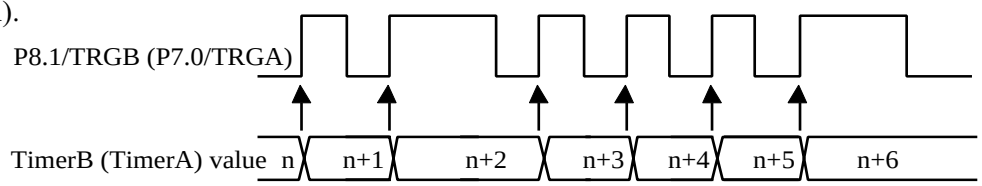
IPSB	INTERNAL PULSE-RATE SELECTION			
	Low-frequency		With low-frequency double	
	NORMAL mode	SLOW mode	NORMAL mode	SLOW mode
0 0	Depend on high speed timer/counter		Depend on high speed timer/counter	
0 1	LXIN/2^5 Hz	LXIN/2^5 Hz	LXIN/2^4 Hz	LXIN/2^4 Hz
1 0	LXIN/2^9 Hz	LXIN/2^9 Hz	LXIN/2^8 Hz	LXIN/2^8 Hz
1 1	LXIN/2^{13} Hz	LXIN/2^{13} Hz	LXIN/2^{12} Hz	LXIN/2^{12} Hz

TIMER/COUNTER FUNCTION

Timer/counter A can be programmable for timer, event counter and pulse width measurement. Each timer/ counter can execute any one of these functions independly.

EVENT COUNTER MODE

For event counter mode, timer/counter increases one at any rising edge of P8.1/TRGB for timerB (P7.0/ TRGA for timer A). When timerB (timerA) counts overflow, it will give interrupt control an interrupt request TRGB (TRGA).

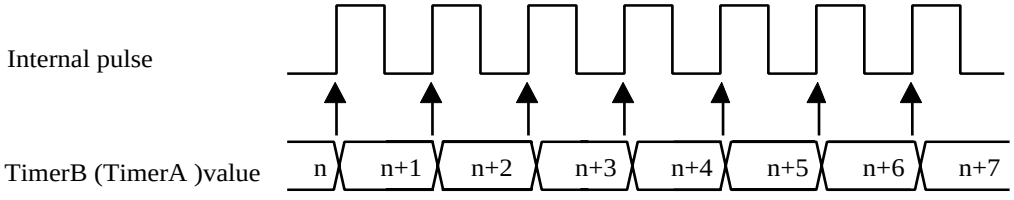


PROGRAM EXAMPLE: Enable timerA with P28

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

TIMER MODE

For timer mode, timer/counter increase one at any rising edge of internal pulse. User can choose 4 kinds of internal pulse rate by setting IPSB for timerB (IPSA for timerA).
When timer/counter counts overflow, TRGB (TRGA) will be generated to interrupt control unit.



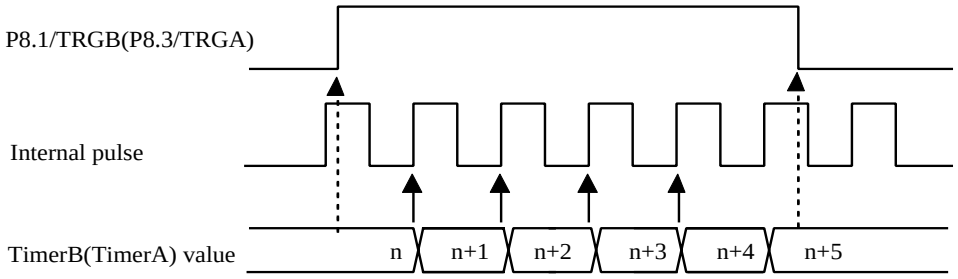
PROGRAM EXAMPLE: To generate TRGA interrupt request after 60 ms with system clock LXlN=32KHz

```
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 = 0F0AH

PULSE WIDTH MEASUREMENT MODE

For the pulse width measurement mode, the counter only incresed by the rising edge of internal pulse rate as external timer/counter input (P8.1/TRGB, P7.0/TRGA), interrupt request will be generated as soon as timer/counter count overflow.



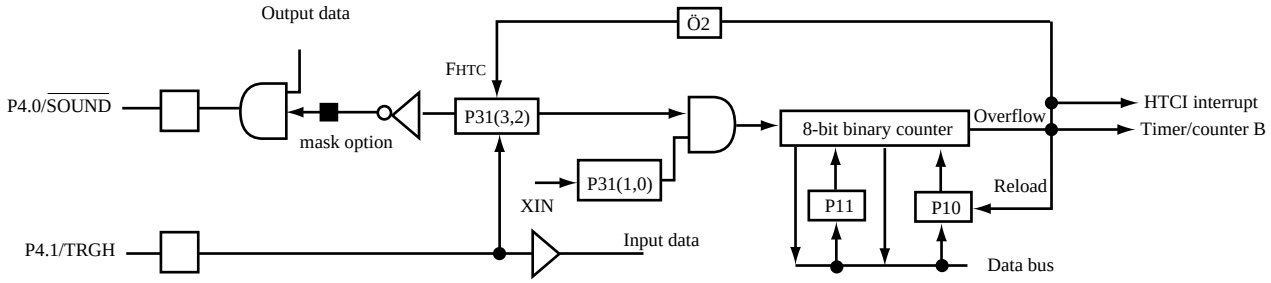
PROGRAM EXAMPLE: Enable timerA by pulse width measurement mode.

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

HIGH SPEED TIMER/COUNTER

EM73866 has one 8-bit high speed timer/counter (HTC). It supports three special functions : auto load timer, melody output and pulse width measurement modes. 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 (P11 and P10), P31 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.



P31 is the command register of the 8-bit high speed timer/counter.

P31 | 3 | 2 | 1 | 0 | Initial value : 0000

HTMS HIPS

HTMS	Mode selection
0 0	Stop
0 1	Auto load timer mode
1 0	Melody mode
1 1	Pulse width measurement mode

HIPS	CLOCK RATE SELECTION			
	Low-frequency		With low-frequency double	
	NORMAL mode	SLOW mode	NORMAL mode	SLOW mode
0 0	LXIN/2^0 Hz	LXIN/2^0 Hz	2xLXIN Hz	2xLXIN Hz
0 1	LXIN/2^2 Hz	LXIN/2^2 Hz	LXIN/2^1Hz	LXIN/2^1 Hz
1 0	RCIN/2^4 Hz	Reserved	RCIN/2^3 Hz	Reserved
1 1	RCIN/2^6 Hz	Reserved	RCIN/2^5Hz	Reserved

P11 and P10 are the counter registers of the 8-bit high speed timer/counter. P10 is the lower nibble register and P11 is the higher nibble register. (HT is the value of counter registers.)

P11 | 3 | 2 | 1 | 0 | Higher nibble register

P10 | 3 | 2 | 1 | 0 | Lower nibble register

Initial value : 0000 0000 (HT)

** F_{HTC}=[(XIN/2^X)/(100H-HT)]/2, HT=0~255
** Example : LXIN=32K Hz, HIPS=01, HT=11110000B=0F0H.
⇒F_{HTC}=[(32K Hz/2^2)/(100H-0f0H)]/2=256 Hz.

```
LDIA    #1111B
OUTA    P11
LDIA    #0000B
OUTA    P10
LDIA    #1001B
OUTA    P31
```

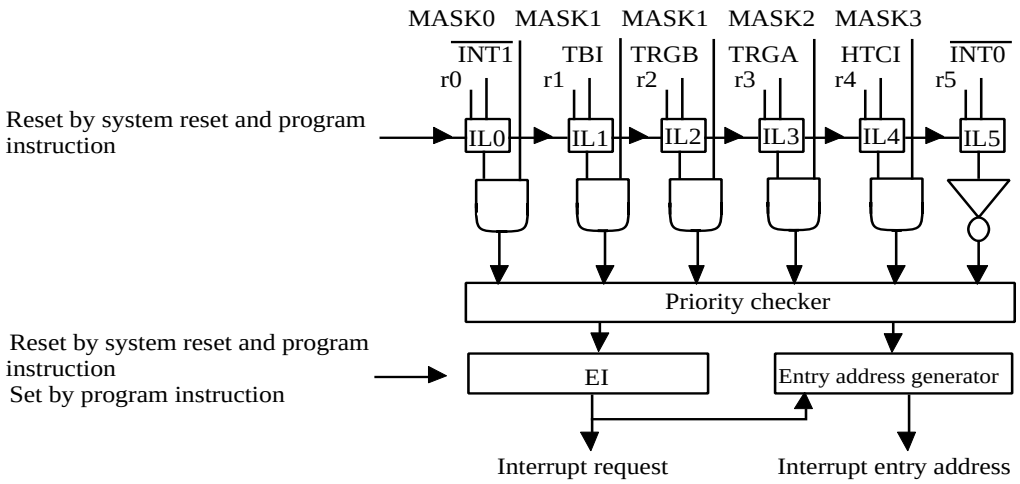
The value of 8-bit binary up counter can be presetted by P10 and P11. The value of registers can loaded into the HTC when the counter starts counting or occurs overflow. If user write value to the registers before the next overflow occurs, the preset value can be changed.
The preset value will be changed when users output the different data to P10 and P11.
The count value of HTC can be read from P10 and P11. The value is unstable when user read the value during counting. Thus, user must disable the counter before reading the value.
The P4.0/SOUND and SOUND pins will output the squre wave in the melody mode. When the CPU is not in the melody mode, the P4.0/SOUND is high and SOUND is low.
The P4.1/RGH pin will be the input pin in the pulse width measurement mode. User must output high to P4.1/TRGH and then it can be the HTC external input pin. When the HTC is disabled, the P4.1 pin is a normal I/O pin.

INTERRUPTFUNCTION

There are 6 interrupt sources, 2 external interrupt sources, 4 internal interrupt sources. Multiple interrupts are admitted according the priority.

Type	Interrupt source	Priority	Interrupt Latch	Interrupt Enable condition	Program ROM entry address
External	External interrupt ($\overline{INT0}$)	1	IL5	EI=1	002H
Internal	High speed timer overflow 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



Interrupt controller:

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

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

EI : Enable interrupt Flip-Flop can promit or inhibit all interrupt sources, when interrupt happened, EI is cleared to "0" automatically, after RTI instruction happened, EI will be set to "1" again.

Priority checker: Check interrupt priority when multiple interrupts happened.

INTERRUPT FUNCTION

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 happened.
5. Clear the IL for which interrupt source has already be accepted.
6. To excute interrupt subrountine 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    #1100B    ;
EXAE              ; set mask register "1100B"
EICIL   111111B   ; enable interrupt F.F.
```

LCD DRIVER

EM73866 can directly drive the liquid crystal display (LCD) and has max. 32 segment and 4 common output pins. There are total 32 x 4 dots can be display. The VRLC pin is the LCD driver power input, there is the voltage of (V_{CC}-VRLC) to LCD.

CONTROL OF LCD DRIVER

The LCD driver control command register is P27. When LDC is 00, the LCD is disabled and changes the duty only. When LDC is 01, the LCD is blanking, the COM pins are inactive and the SEG pins continuously output the display data. When LDC is 11, the LCD driver enables, the power switch is turned on and it cannot off forever except the CPU is reseted or sleeping. User must enable the LCD driver by self when the CPU is waked up.

Port27 3 2 1 0 Initial value : 0000

LDC	DUTY

LDC	LCD display control
0 0	LCD display disable & change duty
0 1	Blanking
1 0	Reserved
1 1	LCD display enable

DUTY	Driving method select	Frame frequency
0 0	1/4 duty (1/3 bias)	85 Hz
0 1	1/3 duty (1/3 bias)	114 Hz
1 0	1/2 duty (1/2 bias)	171 Hz
1 1	Static	85Hz

The storing region of data RAM for LCD display data.

	RAM address	COM3 bit3	COM2 bit2	COM1 bit1	COM0 bit0
SEG0	20H				
SEG1	21H				
SEG2	22H				
:	:				
SEG32	3FH				

The relation between LCD display data and driving method

Driving method	bit3	bit2	bit1	bit0
1/4 duty	COM3	COM2	COM1	COM0
1/3 duty	-	COM2	COM1	COM0
1/2 duty	-	-	COM1	COM0
Static	-	-	-	COM0

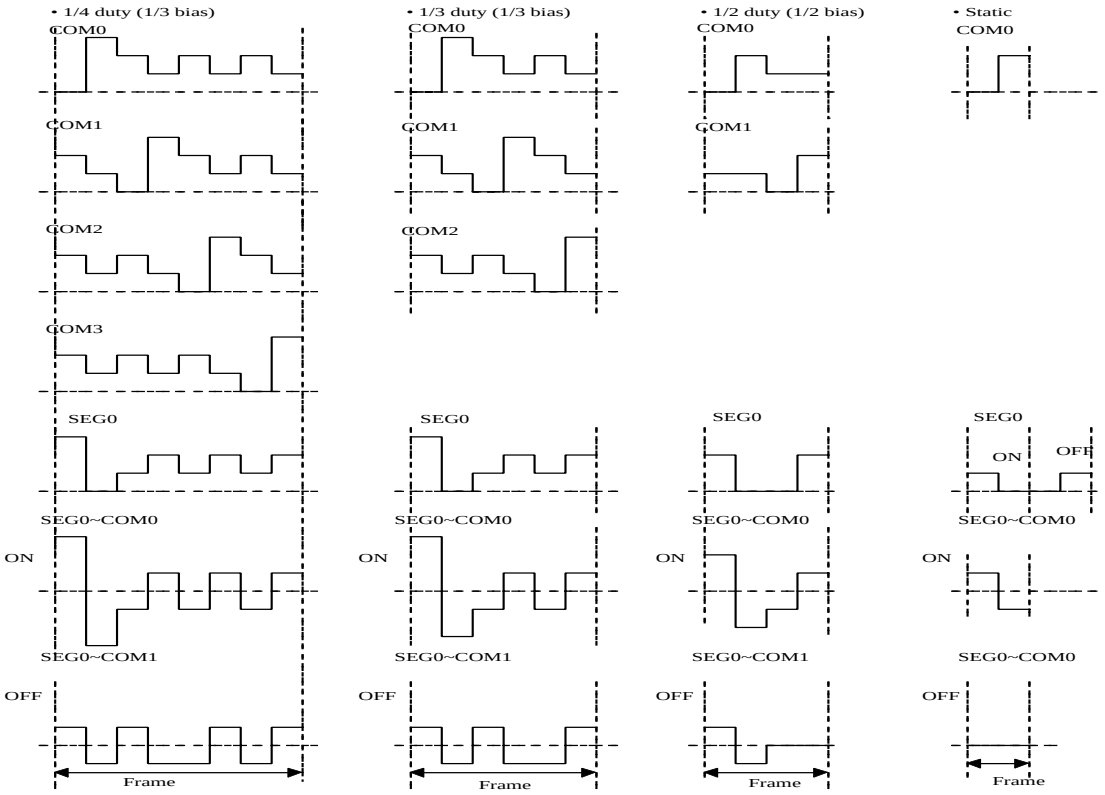
PROGRAM EXAMPLE:

```
LDIA    #0000B
OUTA    P27     ; Set LCD duty
LDIA    #1100B
OUTA    P27     ; Enable LCD
LDIA    #1010B
STA     24H
```

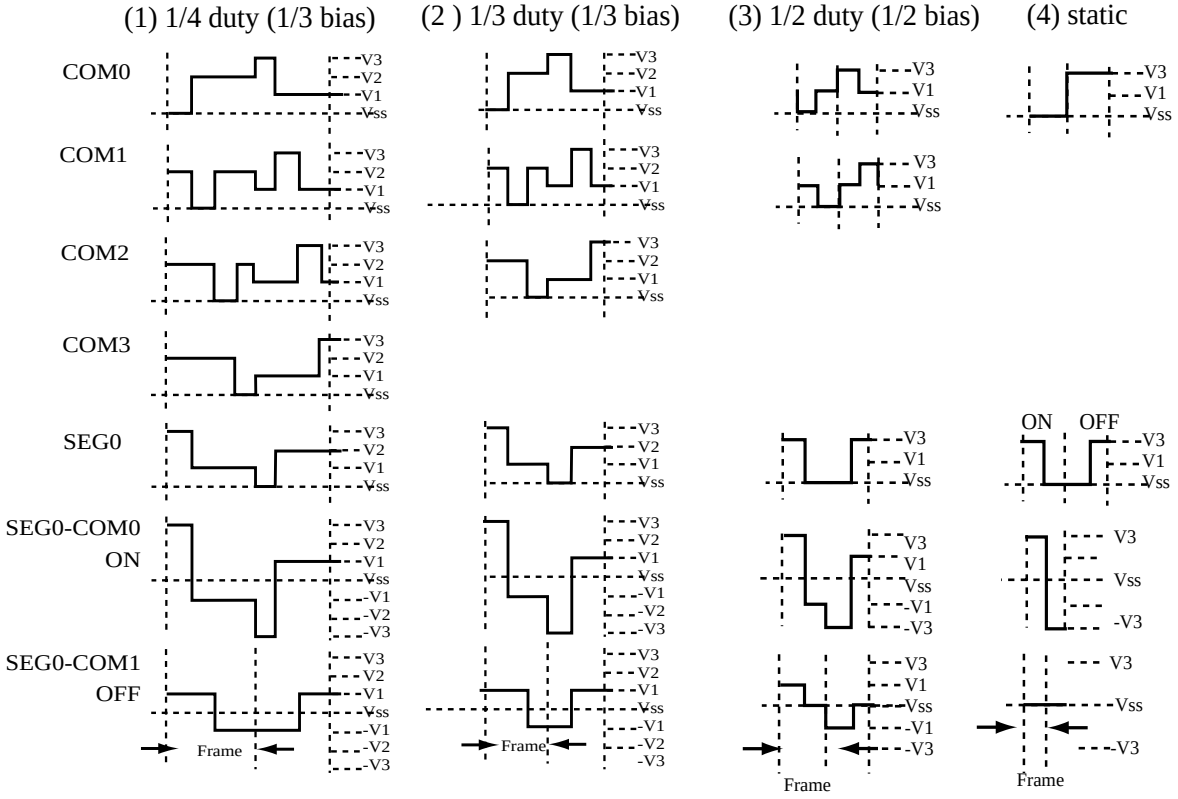
LCD driving methods

There are four kinds of driving methods can be selected by DUTY (P27.0~P27.1). Ther are two waveform types, but for the reason of the number of voltage transition point in type A is greater than type B, so type B gets a better display performance. The driving waveforms type A and B of LCD driver are as below :

TYPE A :

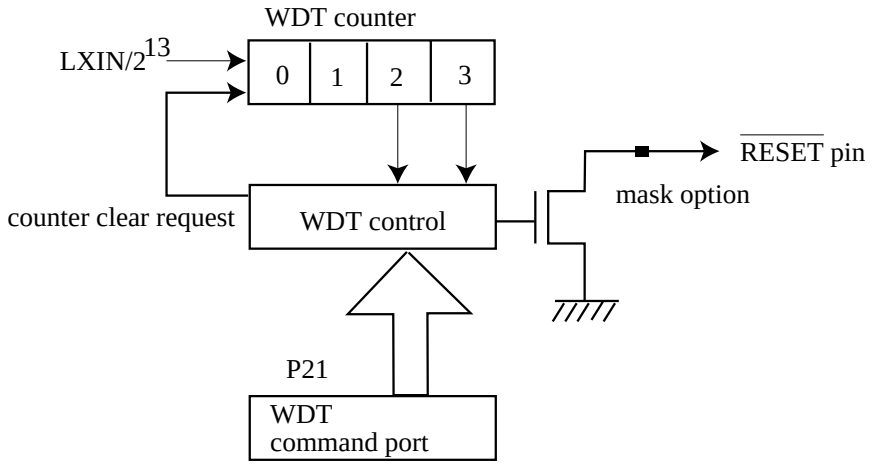

• 1/4 duty (1/3 bias)
• 1/3 duty (1/3 bias)
• 1/2 duty (1/2 bias)
• Static
COM0
COM1
COM2
COM3
SEG0
SEG0~COM0
SEG0~COM1
ON
OFF
Frame

TYPE B :


(1) 1/4 duty (1/3 bias)
(2) 1/3 duty (1/3 bias)
(3) 1/2 duty (1/2 bias)
(4) static
COM0
COM1
COM2
COM3
SEG0
SEG0-COM0
ON
SEG0-COM1
OFF
V3
V2
V1
Vss
-V1
-V2
-V3
Frame

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 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	
	Low-frequency	With low-frequency double
0	3 x 2^{13}/LXIN = 3 x 2^{13}/32K Hz = 0.75 sec	3 x 2^{12}/LXIN = 3 x 2^{12}/32K Hz = 0.375 sec
1	7 x 2^{13}/LXIN = 7 x 2^{13}/32K Hz = 1.75 sec	7 x 2^{12}/LXIN = 7 x 2^{12}/32K Hz = 0.875 sec

PROGRAM EXAMPLE

To enable WDT with $7x2^{13}$/LXIN detection time.

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

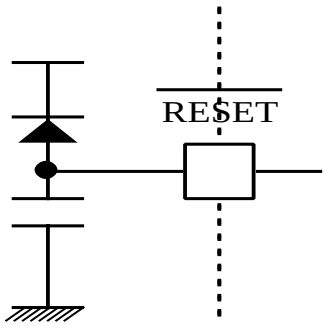
RESETTING FUNCTION

When CPU in normal working condition and $\overline{\text{RESET}}$ pin holds in low level for three instruction cycles at least, then CPU begins to initialize the whole internal states, and 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
P10, 11,14, 16, 19, 21, 22, 25, 27, 28, 29, 31	00h
P1, 3, 4, 5, 6, 7, 8	0Fh
Both oscillator	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}.



EM73866 I/O PORT DESCRIPTION :

Port	Input function		Output function		Note
0	E	Input port , wakeup function			
1	E	Input port	E	Output port	
2	E	Input port , wakeup function		--	
3	E	Input port	E	Output port, shared with segment	
4	E	Input port , wakeup function	E	Output port	
5	E	Input port , wakeup function	E	Output port, P4.0/$\overline{\text{SOUND}}$, P4.1(TRGH)	
6	E	Input port	E	Output port	
7	E	Input port, wakeup function	E	Output port, P7.0(TRGA)	
8	E	Input port, wakeup function	E	Output port,P8.0($\overline{\text{INT1}}$),P8.1(TRGB),P8.2($\overline{\text{INT1}}$)	
9		--		--	
10		--	I	High speed Timer/Counter Register	Low nibble
11		--	I	High speed Timer/Counter Register	High nibble
12		--		--	
13		--		--	
14	I	CPU status	I	Clear P14.0 to 0	
15		--		--	
16			I	STOP mode control register	
17				--	
18				--	
19			I	IDLE mode control register	
20				--	
21			I	WDT control register	
22			I	Slow mode control register	
23				--	
24				--	
25			I	Timebase control register	
26				--	
27			I	LCD control register	
28			I	Timer/counter A control register	
29			I	Timer/counter B control register	
30				--	
31			I	HTC control register	

E : External port
I : Internal port

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

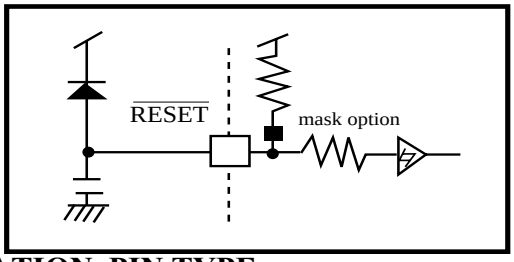
Items	Sym.	Ratings			Condition
			Min.	Max.	
Supply Voltage	V_{DD}	Normal	2.2V	6.0V	4MHz by RC osc
		Slow	2.2V		
		Idle	2.2V		
		Stop	2.0V		
Input Voltage schmitt circuit	V_{IH}	0.80xV_{DD} to V_{DD}			V_{DD} : 2.0~5.5V
	V_{IL}	0V to 0.20 to V_{DD}			
Operating Frequency	F_C	4MHz			Osc
	Fs	32KHz			LXIN,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

DC ELECTRICAL CHARACTERISTICS (V_{DD}=5±0.5V, V_{SS}=0V, T_{OPR}=25°C)

Parameters	Sym.	Min.	Typ.	Max.	Unit	Conditions
Supply current	I_{DD_Xtal}	-	1	2	mA	V_{DD}=5.5V,no load,NORMAL mode, Fc=4MHz, Fs=32KHz (crystal)
		-	100	150	μA	V_{DD}=5.5V,no load,SLOW mode,Fs=32KHz (crystal)
		-	80	100	μA	V_{DD}=5.5V,no load,RV_{RLC}=68K,IDLE mode, Fs=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, Fc=4MHz,Fs=32KHz (RC, OSC)
		-	80	120	μA	V_{DD}=5.5V,no load,SLOW mode,Fs=32KHz (RC, OSC)
		-	45	70	μA	V_{DD}=5.5V,no load,RV_{RLC}=68K,IDLE mode, Fs=32KHz (RC, OSC)
		-	0.1	1	μA	V_{DD}=5.5V,STOP mode (RC, OSC)
Hysteresis voltage	V_{HYS+}	$0.50V_{DD}$	-	$0.75V_{DD}$	V	$\overline{RESET}$,all I/O ports except P3
	V_{HYS-}	$0.20V_{DD}$	-	$0.40V_{DD}$	V	
Input current	I_{IH}	-	-	±1	μA	$\overline{RESET}$, P0,P2, V_{DD}=5.5V,V_{IH}=5.5/0V
		-	-	±1	μA	Open-drain,V_{DD}=5.5V,V_{IH}=5.5/0V
High current		12	16	20	mA	P1,P4 — I/O port acts as input(push-pull), optional,V_{DD}=4.5V,V_{IL}=0.2V
Normal current	I_{IL}	450	550	650	μA	P1,P3~8
Low current		20	24	28	μA	P1,P3~8
Output Voltage	V_{OH1}	3	-	-	V	V_{DD}=4.5V,I_{OH}=3mA for P1
	V_{OH}	2.2	-	-	V	V_{DD}=4.5V,see I_{OH}=typical. for P3~8
	V_{OL}	-	-	0.2	V	V_{DD}=4.5V,I_{OL}=0.5mA,P1,P3~8
High current	I_{OH}	9	11	14	mA	P1,P4,optional — V_{DD}=4.5V,V_{OH}=2.2V
Normal current		420	460	500	μA	P1,P3~8,optional
Low current		18	22	27	μA	P1,P3~8,optional
Leakage current	I_{LO}	-	-	1	μA	Open-drain,V_{DD}=5.5V,Vo=5.5V
Input resistor	R_{IN}	120	150	180	KΩ	P0,P2,pull-up,optional
		120	150	180	KΩ	P0,P2,pull-down,optional
		50	70	90	KΩ	$\overline{RESET}$
High Frequency Variation			20	30	%	V_{DD}=2.2~5.5V±10% RC OSC R=100K±2%, fc=4MHz
Low Frequency Variation			20	30	%	V_{DD}=2.2~5.5V±10% RC OSC R=1MΩ±2%, fs=32KHz

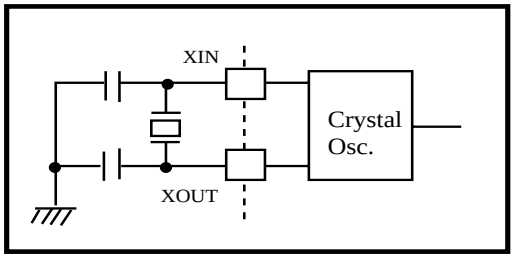
$\overline{\text{RESET}}$ PIN TYPE

TYPE RESET-A

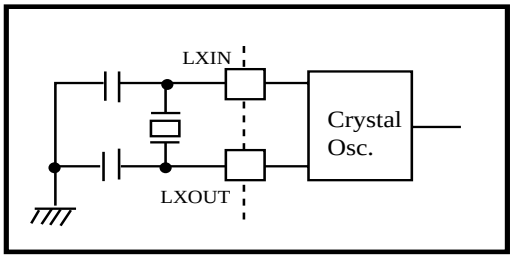


OSCILLATION PIN TYPE

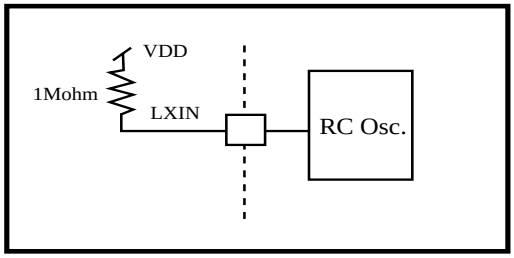
TYPE OSC-A



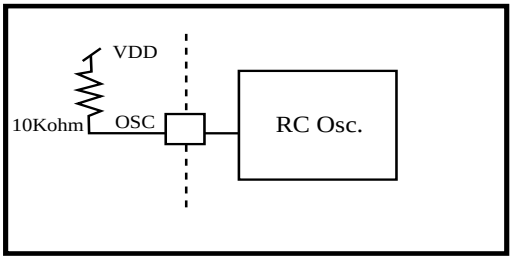
TYPE OSC-B



TYPE OSC-H1 (Low frequency)

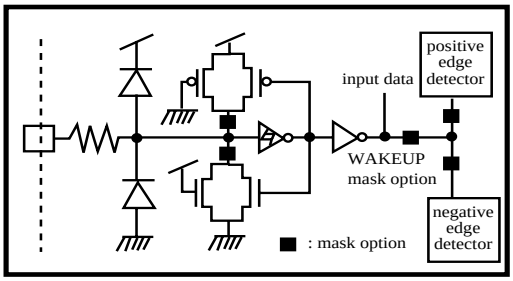


TYPE OSC-H2 (High frequency)



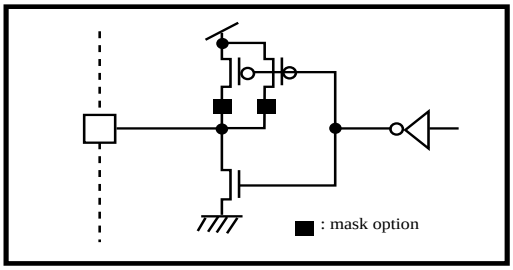
INPUT PIN TYPE

TYPE INPUT-K

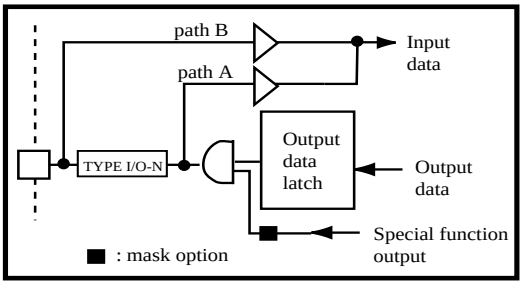


I/O PIN TYPE

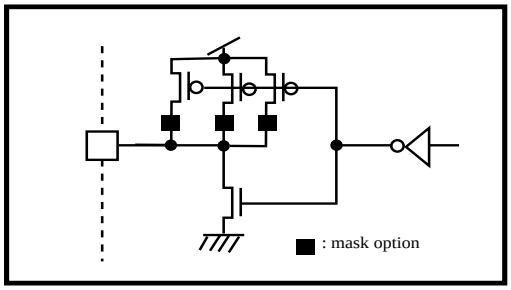
TYPE I/O-N



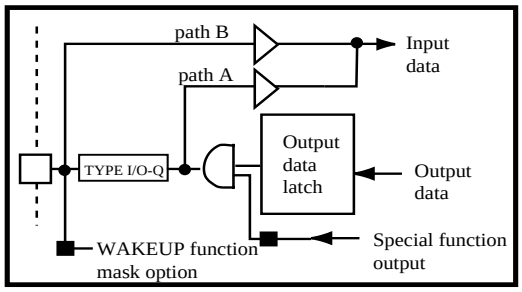
TYPE I/O-O



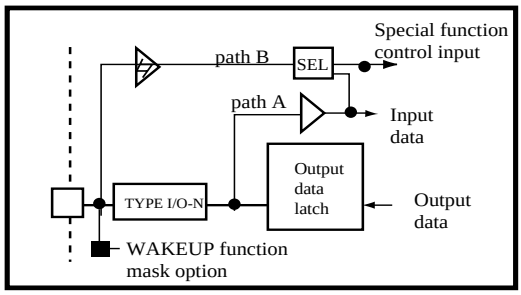
TYPE I/O-Q



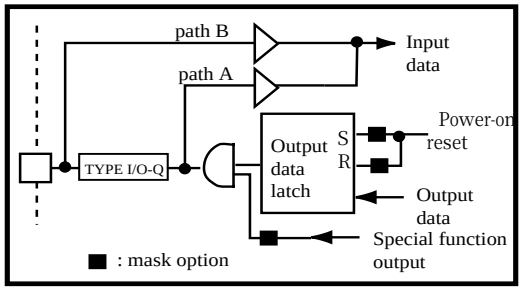
TYPE I/O-R1



TYPE I/O-S



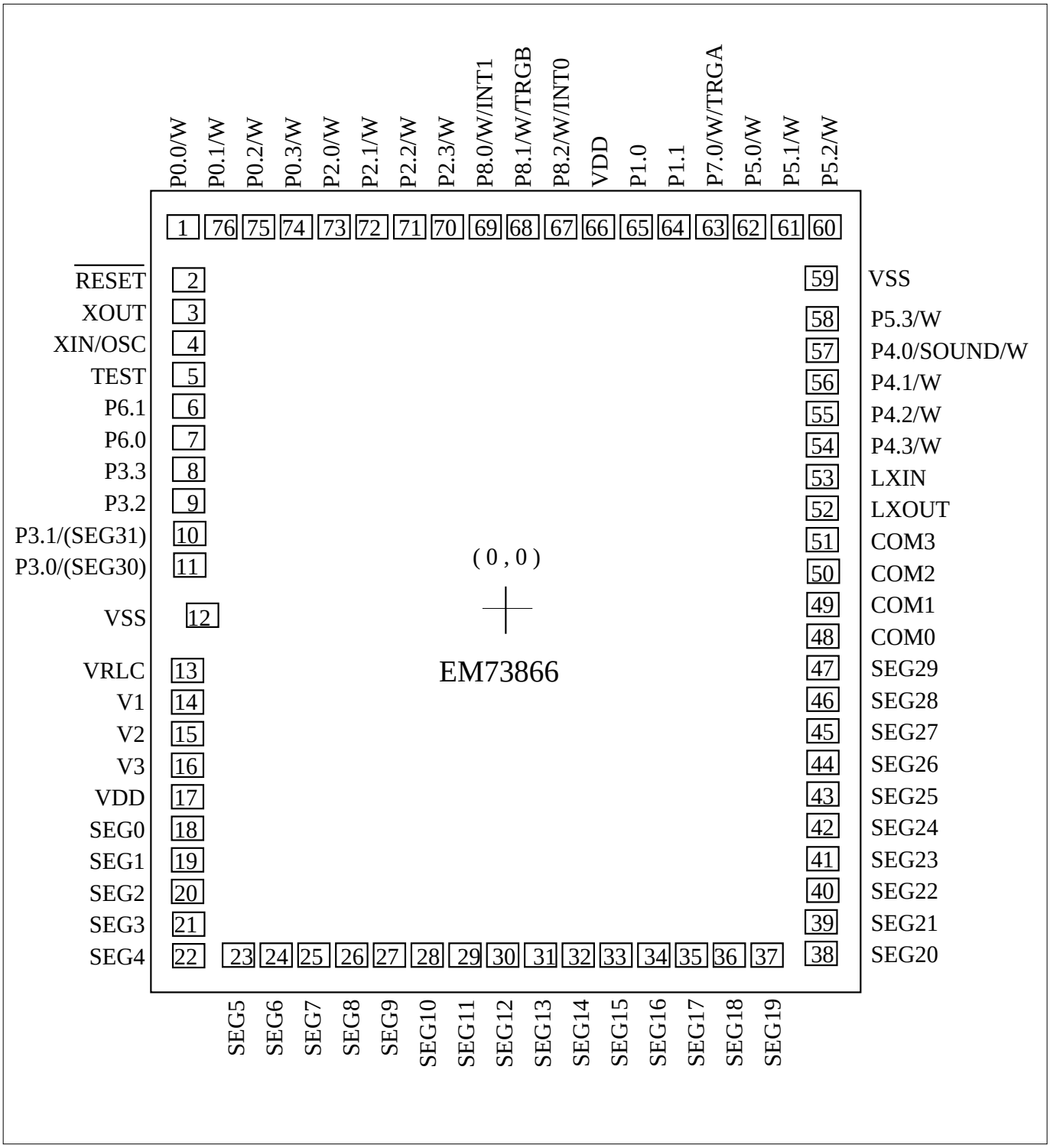
TYPE I/O-Z



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.

PAD DIAGRAM


P0.0/W
P0.1/W
P0.2/W
P0.3/W
P2.0/W
P2.1/W
P2.2/W
P2.3/W
P8.0/W/INT1
P8.1/W/TRGB
P8.2/W/INT0
VDD
P1.0
P1.1
P7.0/W/TRGA
P5.0/W
P5.1/W
P5.2/W
1 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60
RESET 2
XOUT 3
XIN/OSC 4
TEST 5
P6.1 6
P6.0 7
P3.3 8
P3.2 9
P3.1/(SEG31) 10
P3.0/(SEG30) 11
VSS 12
VRLC 13
V1 14
V2 15
V3 16
VDD 17
SEG0 18
SEG1 19
SEG2 20
SEG3 21
SEG4 22
59 VSS
58 P5.3/W
57 P4.0/SOUND/W
56 P4.1/W
55 P4.2/W
54 P4.3/W
53 LXIN
52 LXOUT
51 COM3
50 COM2
49 COM1
48 COM0
47 SEG29
46 SEG28
45 SEG27
44 SEG26
43 SEG25
42 SEG24
41 SEG23
40 SEG22
39 SEG21
38 SEG20
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
(0 , 0)
EM73866
SEG5
SEG6
SEG7
SEG8
SEG9
SEG10
SEG11
SEG12
SEG13
SEG14
SEG15
SEG16
SEG17
SEG18
SEG19

Pad No.	Symbol	X	Y
1	P0.0/W	-1056.5	1391.1
2	$\overline{\text{RESET}}$	-1030.3	1186.6
3	XOUT	-1030.3	1066.7
4	XIN/OSC	-1030.3	946.7
5	TEST	-1030.3	826.7
6	P6.1	-1030.3	703.3
7	P6.0	-1030.3	583.4
8	P3.3	-1030.3	459.5
9	P3.2	-1030.3	339.6
10	P3.1/(SEG31)	-1030.3	215.8
11	P3.0/(SEG30)	-1030.3	95.8
12	VSS	-991.7	-105.5
13	VRLC	-1030.3	-271.5
14	V1	-1030.3	-391.5
15	V2	-1030.3	-511.5
16	V3	-1030.3	-631.5
17	VDD	-1030.3	-782.8
18	SEG0	-1030.3	-902.8
19	SEG1	-1030.3	-1022.8
20	SEG2	-1030.3	-1142.8
21	SEG3	-1030.3	-1262.8
22	SEG4	-1030.3	-1382.8
23	SEG5	-839.4	-1390.2
24	SEG6	-719.4	-1390.2
25	SEG7	-599.4	-1390.2
26	SEG8	-479.4	-1390.2
27	SEG9	-359.4	-1390.2
28	SEG10	-239.4	-1390.2
29	SEG11	-119.4	-1390.2
30	SEG12	0.6	-1390.2
31	SEG13	120.6	-1390.2
32	SEG14	240.6	-1390.2
33	SEG15	360.6	-1390.2
34	SEG16	480.6	-1390.2
35	SEG17	600.6	-1390.2
36	SEG18	720.6	-1390.2
37	SEG19	840.6	-1390.2
38	SEG20	1030.2	-1390.7
39	SEG21	1030.2	-1270.7
40	SEG22	1030.2	-1150.7

Pad No.	Symbol	X	Y
41	SEG23	1030.2	-1030.7
42	SEG24	1030.2	-910.7
43	SEG25	1030.2	-790.7
44	SEG26	1030.2	-670.7
45	SEG27	1030.2	-550.7
46	SEG28	1030.2	-430.7
47	SEG29	1030.2	-310.7
48	COM0	1030.2	-190.7
49	COM1	1030.2	-70.7
50	COM2	1030.2	49.3
51	COM3	1030.2	169.3
52	LXOUT	1030.2	289.3
53	LXIN	1030.2	409.2
54	P4.3/W	1030.2	531.2
55	P4.2/W	1030.2	651.2
56	P4.1/W	1030.2	775.0
57	P4.0/SOUND/W	1030.2	895.0
58	P5.3/W	1030.2	1018.8
59	VSS	1030.2	1181.6
60	P5.2/W	1061.1	1391.1
61	P5.1/W	941.0	1391.1
62	P5.0/W	817.2	1391.1
63	P7.0/W/TRGA	697.2	1391.1
64	P1.1	573.4	1391.1
65	P1.0	453.4	1391.1
66	VDD	332.0	1391.1
67	P8.2/W/INT0	180.7	1391.1
68	P8.1/WTRGB	56.9	1391.1
69	P8.0/W/INT1	-63.1	1391.1
70	P2.3/W	-189.5	1391.1
71	P2.2/W	-309.5	1391.1
72	P2.1/W	-438.5	1391.1
73	P2.0/W	-558.5	1391.1
74	P0.3/W	-687.5	1391.1
75	P0.2/W	-807.5	1391.1
76	P0.1/W	-936.5	1391.1

Chip size : 2420 * 3140 UM
Unit : μm
For PCB layout, IC substrate must be floated or connected at Vss.

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) 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'

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

(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

**** 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