MS89F/L 09Series

8051 High Speed Flash CPU With ADC

Hardware SPEC

MSHINE Technologies Corporation

HEAD QUARTER

TEL: +886-3-5833899
FAX: +886-3-5830858

SHENZHEN OFFICE

TEL:+86-755-88250870
FAX:+86-755-88250872

Contents

CONTENTS 2
FIGURES 5
TABLES 6
FEATURES 7
BLOCK DIAGRAM 8
PRODUCTS 9
PACKAGE 9
GENERAL DESCRIPTION 9
DOCUMENT CHANGE HISTORY 9
PINS CONFIGURATION 10
PIN DESCRIPTIONS 12
ABSOLUTE MAXIMUM RATINGS 14
8052 COMPATIBLE CPU CORE 14
CPU Architecture 15
Memory Types 15
8052 Compatible Special Function Registers (SFR) 16
I/O Registers 16
Register Banks 16
Accumulator (ACC) 16
Program Status Word (PSW) 16
Program Counter (PC) 16
Stack Pointer (SP) 16
Data Pointer (DPTR) 16
Interrupt Registers 16
Timer Registers 16
Serial Control Registers 16
B Register 17
PCON Register 17
Non-compatible Functions and Registers 17
Standard 8052 Addressing Modes 17
Implied Register Addressing Mode 17
Bit Addressing Mode 17
Immediate Addressing Mode 18
Direct Addressing Mode 18
Indexed (Indirect) Addressing Mode 18
MSHINE 8052 Instruction Execution 18
MS81/MS89 SERIES MEMORY MAP 18
SFR REGISTERS 19
Direct Addressing Mode registers 20
XMEM Addressing Mode registers 31
INTERRUPTS 34
Interrupt Vectors 35
Interrupt Priority 35
Interrupt Masking 35
SYSTEM RESET 36
CLOCK DISTRIBUTION 36

POWER CONTROL MODES 38
IDLE Mode 38
Power-Down Mode 38
Wake-Up Sources 39
ON-CHIP OSCILLATORS 39
LOW VOLTAGE STOP/DETECT 41
I/O PORTS 42
Port 0 42
Port 1 43
Port 3 43
I/O Pin Interrupts 43
CMOS I/O 45
TIMERS 47
Timer 0 and Timer 1 47
Timer 2 49
WATCH-DOG TIMER 54
SERIAL INTERFACE – UART 55
Receiving Data when RI is 1 55
PFD (PROGRAMMABLE FREQUENCY DIVIDER) 56
PULSE WIDTH MODULATION (PWM) 57
PWM Duty with PWTH SFR 58
ANALOG TO DIGITAL CONVERTER (ADC) 62
ADC Program Example 68
SPI 71
SPI Master Program Example 74
SPI Slave Program Example 75
I2C 77
I2C Slave Addresses 77
I2C Slave Clock Stretching 78
I2C Control Registers 78
I2C Master Example Program 82
I2C Slave Example Program 85
I2C Timing Diagram 87
Extern Master Writes MS89XX 87
External Master Read MS89XX 88
External Master Read/Write Other Device 88
IN APPLICATION PROGRAMMING (IAP) 89
Programmed by CPU Itself 90
Sector Erasing/Programming 90
Flash Data Protection 91
IN SYSTEM DEBUGGER 93
FLASH INFO SECTOR 95
AC & DC ELECTRICAL CHARACTERISTICS 97
DC Characteristics – I/O, CPU 97
AC Electrical Characteristics – Oscillators 98
AC ELECTRICAL CHARACTERISTICS – ADC 99
DC Characteristics – ADC 99

PRODUCT PROCUREMENT 100
PACKAGE SIZE 101
DIP20 101
SOP20 102
SOP16 103
SSOP 104

Figures

Figure 1. MS89F/L series Block Diagram. 8
Figure 2. Pins Configuration 11
Figure 3. MSHINE 8051 Core structure. 15
Figure 4. Memory Map. 19
Figure 5. Oscillator Source 37
Figure 6. Clock Hardware configuration 40
Figure 7. Port configuration. 42
Figure 8. IE0/IE1 Concept Diagram. 44
Figure 9. MS89F/L SERIES CMOS I/O. 45
Figure 10. Timer 0 and Timer 1 Mode 0 and 1 configuration. 47
Figure 11. Timer 0 and Timer 1 Mode 2, auto reload mode. 48
Figure 12. Timer 0 Mode 3 48
Figure 13. Timer 2 in auto-reload mode. 49
Figure 14. T2 in capture mode. 49
Figure 16. PWM Generation principle. 57
Figure 17. ADC and Reference Configuration 63
Figure 18. MS89F/L SERIES SPI Configuration 71
Figure 19. SPI Configuration. 71
Figure 20. MS89F/L SERIES I2C Configuration 77
Figure 21. I2C Slave SCL Stretching. 78
Figure 22. External Master Writes MS89XX. 87
Figure 23. External Master Reads MS89XX. 88
Figure 24. Flash Protection Scheme. 92
Figure 25. MS89FXX In System Debugger Configuration. 93
Figure 26. In System Debugger used recources. 94

Tables

Table 1. MS89F/L series DIP/SOP20 Pins Configuration. 11
Table 2. MS89F/L SERIES Pins Description. 13
Table 3. Port 1 multi-functions 43
Table 4. Port 3 multi-functions 43
Table 5. DC Characteristics – I/O, CPU 98
Table 6. AC Electrical Characteristics – Oscillators 98
Table 7. AC Characteristics – ADC 99
Table 8. ADC DC characteristics 99
Table 9. Product procurement 100

Features

- 8-bit High Speed micro-controller built in
- 8K bytes Flash Memory
  - In Application Programming (IAP) support.
  - 4x2K bytes sectors can be erased or programmed independently.
  - Program code can be protected by hardware.
  - CPU can upgrade FLASH by running programs in RAM.
- Total 512 bytes RAM. 256 bytes Internal DATA RAM (Random Access Memory), (IRAM). 256 Byes external RAM (XRAM). XRAM can be switched to Program area.
- Three selectable clock source for CPU:
  * External Oscillator (Up to 20MHz)
  * RC oscillator with external R (up to 20MHz)
  * Internal trimmed Oscillator 8MHz at 3.3V/5V).
- Maximum 18 Programmable I/O pins. (P0 ~ P3)
- 3 timers built in, compatible to Intel 80C32. Also an additional watch-dog timer to prevent system halt.
- Watchdog timer uses independent low speed internal low power oscillator that can be used as low power "wakeup" interrupt source.
- 10-bit ADC built in, with 16 selectable input sources.
- ADC can use internal regulated reference or external reference.
- Analog Comparator built in.
- Standard UART interface. With 11.059 MHZ Oscillator, the baud rate may be exact 115200 BPS.[1]
- TWO I2C Ports master/slave.
- One SPI Master/Slave Port.
- 3 Ch 10-bits PWM/CMP output pins. PWM can set change both period and duty.
- 5 interrupt input sources shared with I/O port.
- Low Voltage Stop/ Detect capability
- Low Voltage Reset
- APOR : Power on reset .
- PFD : Programmable Frequency Divider Output.
- Hardware Watch Dog Timer.
- Standby mode , Stop mode
- MS89F series Operating voltage range: 4.5V ~ 5.5V.
- MS89L series Operating voltage range: 2.4V ~ 3.6V.

[1] For UART using IRC at 115200 BPS, please contact MSHINE.

Block Diagram

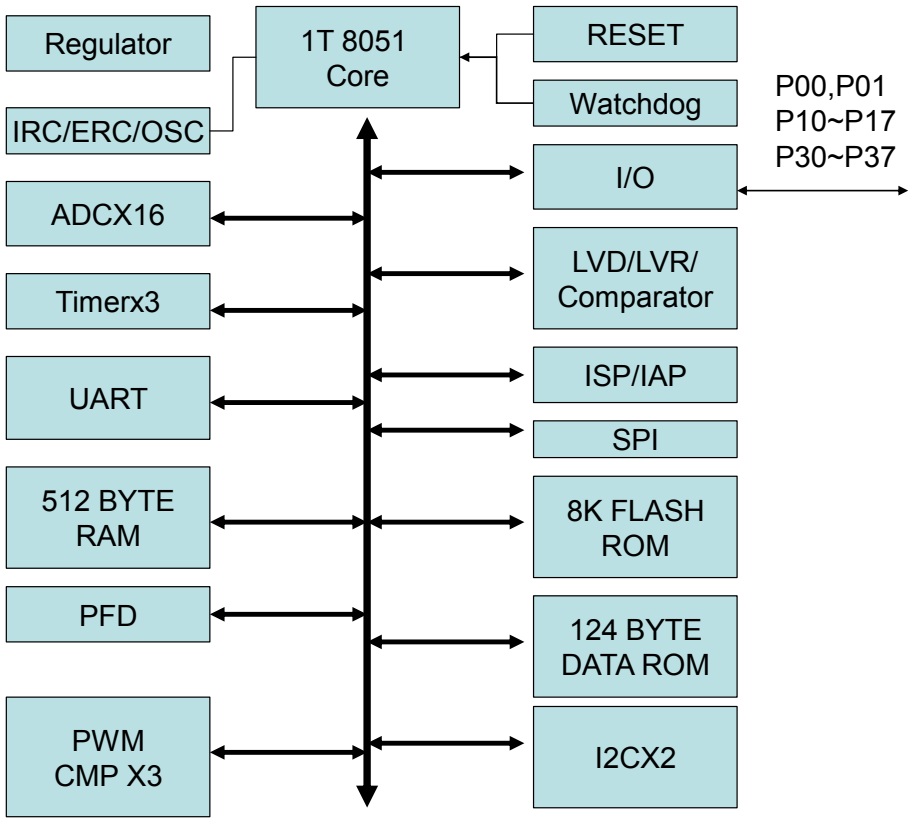


Figure 1. MS89F/L series Block Diagram.

Products

Item	Operating Voltage	Flash ROM	SRAM Byte	Data Flash (Option)		INFO Sector	A/D 10 Bit	Package
				Program Rom	Data Rom			
MS89F/L09	5.5V ~ 4.5V 3.6V ~ 2.4V	8K	512	8K, 6K, 4K, 2K	0K, 2K, 4K, 6K	124 Byte	V	DIP20 SOP20
MS89F/L109	5.5V ~ 4.5V 3.6V ~ 2.4V	8K	512	8K, 6K, 4K, 2K	0K, 2K, 4K, 6K	124 Byte	V	DIP16 SOP16
MS89F/L309	5.5V ~ 4.5V 3.6V ~ 2.4V	8K	512	8K, 6K, 4K, 2K	0K, 2K, 4K, 6K	124 Byte	V	DIP 8 SOP 8

Package

■ 20, 16, 14, 8 Pin package.

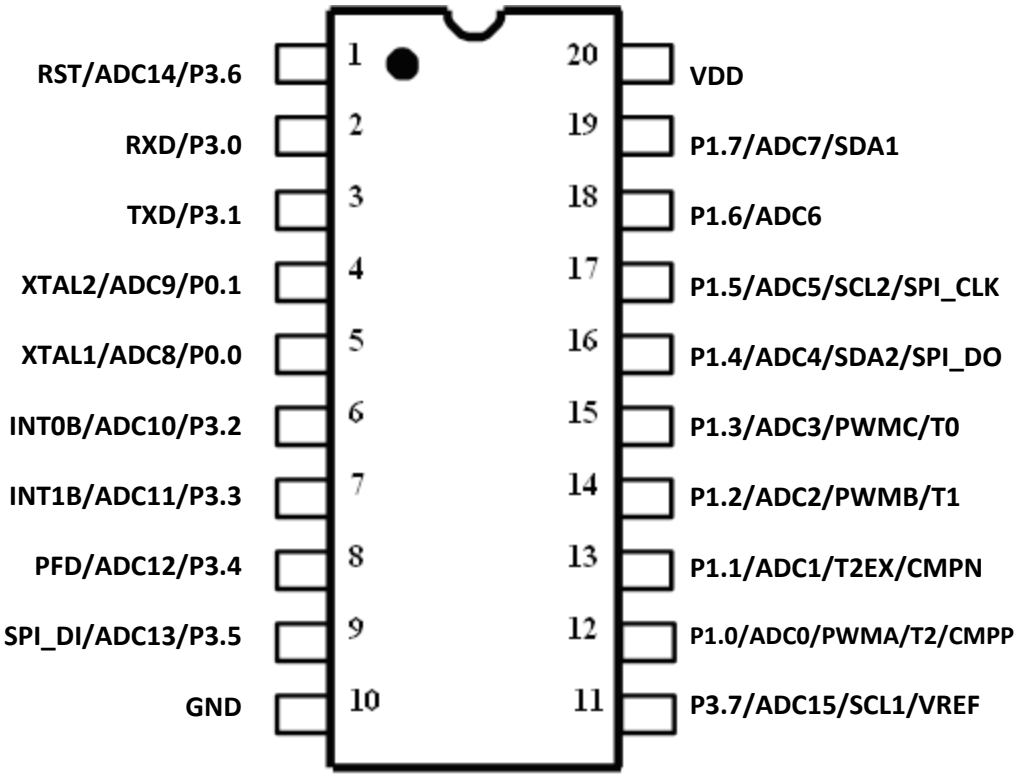
General Description

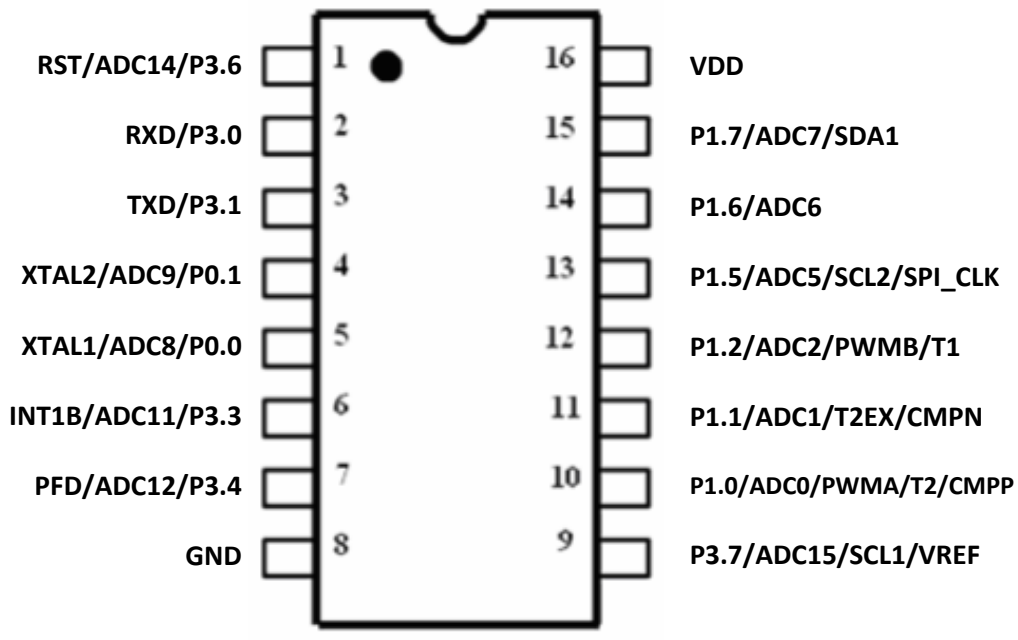
MS89F/L series is a general purpose microcontroller that is suitable for commercial or industrial applications like charger, industrial control, and other applications.

Document Change History

V001: First release.
V003: Add VREF.
V004: I/O modified.

Pins Configuration





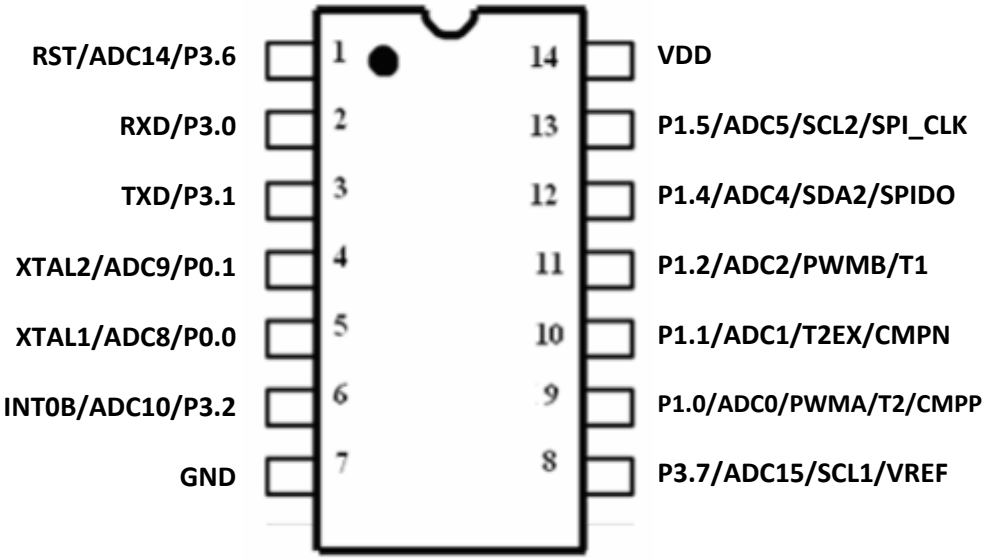


Figure 2. Pins Configuration

PIN #	Name	PIN #	Name
1	RESETB/P3.6/ADC14	20	VDD
2	P3.0/RXD	19	P1.7/SDA1/ADC7
3	P3.1/TXD	18	P1.6/ADC6
4	XTAL2/P0.1/ADC9	17	P1.5/SCL2/ADC5/SPI_CLK
5	XTAL1/P0.0/ADC8	16	P1.4/SDA2/ADC4/SPI_DO
6	P3.2/INT0B/ADC10	15	P1.3/PWMC/ADC3/T0
7	P3.3/INT1B/ADC11	14	P1.2/PWMB/ADC2/T1
8	P3.4/PFD/ADC12	13	P1.1/ T2EX/ADC1/CMPN
9	P3.5/SPI_DI/ADC13	12	P1.0/PWMA/T2/ADC0/CMPP
10	GND	11	P3.7/SCL1/VREF/ ADC15

Table 1. MS89F/L series DIP/SOP20 Pins Configuration.

Pin Descriptions

DIP20 Pin No.	Notation	I/O	Functional Description
1	RESETB/P3.6/ADC14	I/O	P3.6 or RESETB to reset MS89F/L SERIES to a known state.
2	P3.0/RXD	I/O	P3.0 or UART RXD function.
3	P3.1/TXD	I/O	P3.1 or UART TXD function.
4	XTAL2/P0.1/ADC9	I/O	Oscillator pin or P01.
5	XTAL1/P0.0/ADC8	I/O	Oscillator pin or P00. If ERC is used, this pin should connect to a resistor.
6	P3.2/INT0B/ADC10	I/O	P3.2 or INT0B
7	P3.3/INT1B/ADC11	I/O	P3.3 or INT1B
8	P3.4/PFD/ADC12	I/O	P3.4 or PFD(Programmable frequency output)
9	P3.5/SPI_DI/ADC13	I/O	P3.5 or SPI_DI (SPI data input from external device)
10	GND	PWR	Ground for digital circuits.
11	P3.7/SCL1/VREF/ADC15	I/O	P3.7, I2C Port 0 clock, or analog VREF.
12	P1.0/PWMA/T2/ADC0/CMPP	I/O	P1.0, PWM channel A, T2 pin, ADC channel 0 input, or analog comparator Positive input.
13	P1.1/ T2EX/ADC1/CMPN	I/O	P1.1, T2EX pin, or ADC channel 1, or analog comparator negative input.
14	P1.2/PWM2/ADC2/T1	I/O	P1.2, PWM channel B, or ADC channel 2, or T1.
15	P1.3/PWM3/ADC3/T0	I/O	P1.3, PWM channel C, or ADC channel 3, or T0.

16	P1.4/SDA2/ADC4/SPI_DDI	I/O	P1.4, I2C port 2 data I/O, or SPI data to external device, or ADC channel 4.
17	P1.5/SCL2/ADC5/SPI_CLK	I/O	P1.5, I2C port 2 clock, or SPI clock to/from external device.
18	P1.6/ADC6	I/O	P1.6 or ADC channel 6.
19	P1.7/SDA1/ADC7	I/O	P1.7 or I2C port 0 data I/O.
20	VDD	VDD	Digital VDD.

Table 2. MS89F/L SERIES Pins Description.

Absolute Maximum Ratings

Comments

MS89FXX DC Supply Voltage ....−0.5V to + 5.5 V
MS89LXX DC Supply Voltage ....−0.5V to + 4.0 V
Input Voltage.……..−0.5V to VDD + 0.5V
Output Voltage……−0.5V to VDD + 0.5V
Operating Temperature…….....−40° to 85° C
Storage Temperature……..−70° to 150° C

Never allow a stress to exceed the values listed under "Absolute Maximum Ratings", otherwise the device would suffer from a permanent damage. Nor is a stress at the listed value be allowed to persist over a period, since an extended exposure to the absolute maximum rating condition may also affect the reliability of the device, if not causing a damage thereof.

8052 Compatible CPU Core

MS81/89 Series uses 8052 compatible CPU Core. The CPU has the following features:

1. Harvard Machine, 8-bit[2] Data memory and 8-bit Program accessed at the same time, cause it the fastest general-purpose 8-bit CPU for complex controller applications.
2. CISC CPU, complex instruction set and bit-addressing mode may reduce program code size to the smallest. The features are listed below:
   a) Memory-mapped register bank. Each bank has 8 registers and totally 4 banks can be selected. Mapped registers can do most calculation and use the least number of code memory.
   b) Optimized Accumulator and memory-mapped register bank data movement. Only 1 byte instruction may move the data to/from the register and the accumulator.
   b) Bit-addressing mode, set/clear/toggle a bit of bit-addressable I/O register needs only 2 bytes.
   c) Optimized 2K subroutine call and branch, 2 bytes of program code can branch to another address in 2K bytes.
   e) Optimized “Decrease” and “branch if zero” into single byte instruction.
   f) Optimized “Compare” and “jump if not equal” into single byte instruction.
   g) 8-bit Multiplication and Division instruction supported.
   h) High-low 4-bit nimble operations supported.
   i) 3 bytes instruction to move data in non-register memory.
   j) Logic operation may be applied to mapped SFR directly.
3. Standard I/O, UART, Timers, Interrupt, and Event counters, make the application program portable between most 8051 compatible applications.
4. MSHINE 8051 core do lots of the instructions in 1 or 2 cycles, and is one of the fastest 8051 CPU on the market.

Comparing to other 8-bit CPU, especially VON-NEWMAN machines, the 8051 core will have worse performance only at 16-bit data addressing space. Since 8051 always need to use indirect addressing mode for 16-bit addressable data, it takes more cycles to do large data operations, or C/X/P memory operations. The following sub-sections will introduce the internals of the MSHINE 8051 CPU in more details.

[2] Data bus width is 8-bit. Address bus width can be modified. Address bus for IRAM is fixed to 8 bit. Address bus for P/X memory may be extend to 16-bit wide.

CPU Architecture

MSHINE 8052 CPU core has the following structure.

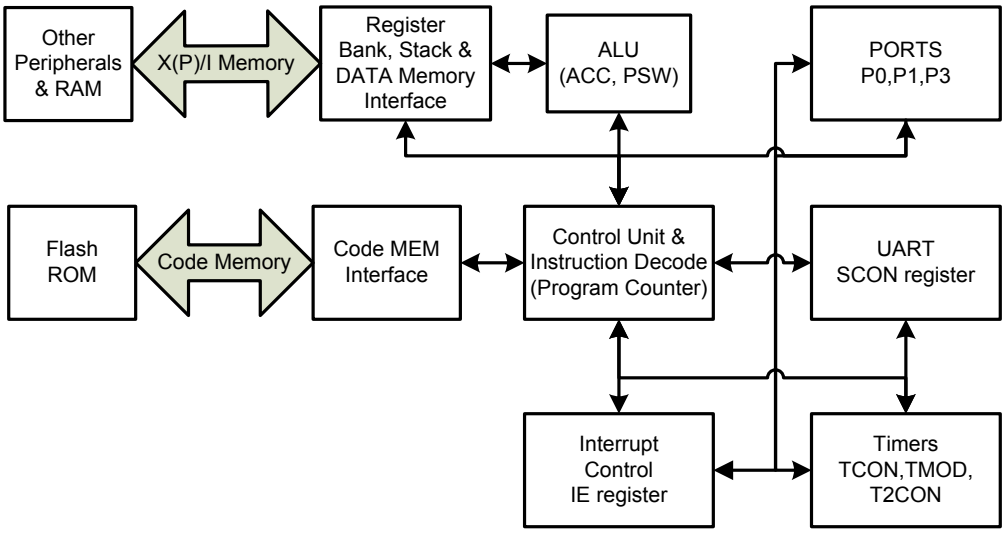


Figure 3. MSHINE 8051 Core structure.

It is shown that code memory is independent from register bank and registers, pipelined instruction execution make it possible to run an instruction per cycle.

Memory Types

Usually a standard 8052 CPU core may have the following memories connected:

1. SFR & direct accessing RAM. This area has 256 bytes of addressing space. However, it is usually divided into 2 parts. Low part direct accessing SRAM have 128 bytes, and high 128 bytes are for special function registers (SFR). In the low addresses, the first 32 bytes may be switch to register banks. Each bank has 8 bytes, named R0-R7, selected by PSW register bits. R0-R7 registers are used in Implied Addressing Mode instructions.
2. Bit Addressable Memory. Direct addressable addresses (RAM and SFR) 0x00, 0x08, 0x10, 0x18,…0xf8 with C flag are bit addressable.
3. Indirect RAM (IRAM) and STACK memory. IRAM and STACK may access up to 256 bytes of RAM, from 0x00~0xFF. While CPU using direct-addressing mode address 0x80~0xFF to access SFR, RAM at 0x80~0xFF may be used by STACK or indirect addressing mode like MOV A,@R0 instructions.
4. Code memory. 8052 CPU may "read" the code memory with special addressing mode. It cannot modify it directly since it uses FLASH on MS81/MS89 Series.
5. P-Memory and X-Memory. Formerly X-memory means the "External" memory of 8052. However, since the memory is "on-chip" with MS81/MS89 Series, "External" memory will cause confusing. Here it is called on-chip "X-memory" instead, and it just means the addressing space accessing with DPTR register with MOVX instructions. For different compilers, there is another "P" Memory for addressing. P-memory means the addressing space accessing with R0 and R1 registers with MOVX instructions, and P2 as the high address byte. MSHINE 8052 core mapped these memory spaces to the same address space. CPU

may use DPTR to access RAM from 0000~00FF and uses R0/P2 to access FF00~FFFF, or vice-versa.

8052 Compatible Special Function Registers (SFR)

Standard 8052 CPU has the following SFR that is compatible among them.

I/O Registers

Standard 8052 CPU may access to P0, P1, P2, and P3 at fixed address. Please see I/O ports at later sections.

Register Banks

Direct Accessed RAM 00~1F may be divided into 4 register banks. Each bank has registers R0~R7 that can be used for implied addressing mode instructions. Bank selection is by <RS1,RS0> of PSW register.

Accumulator (ACC)

ACC register is used for arithmatic operations, like addition, substraction, and logic operations.

Program Status Word (PSW)

PSW register has "Carry", "Overflow", "Parity", and "Carry for low 4-bit" are available. Also <RS1,RS0> flags are used to select register bank.

Program Counter (PC)

The 16-bit program counter means the Code memory address that the CPU is executing. It always increases 1 except the following conditions:

1. Branch.
2. Interrupt.
3. Longer execution cycles like MUL (multiplication), etc.

Stack Pointer (SP)

The 8-bit SP register points to the IRAM and it will "increase 2" when CPU enters a function call or interrupt service routine, because CPU will save 2 bytes of (next) program counter to the stack. And CPU will pop the program counter from the stack after the routine is end by RET or RETI instruction.

Data Pointer (DPTR)

The 16-bit DPTR register is used for accessing X-memory and C memory that needs 16-bits address bus. It may be seperate as DPL/DPH as 8-bit registers.

Interrupt Registers

The IE register is used for interrupt "ENABLE" control and IP register is for priority control. Totally 2 priorities can be set for different events.

Timer Registers

Timer 0,1, and 2 are standard timers, corresponding registers are TCON, TMOD, T2CON, RCAP2H/L, TH0, TL0, TH1, TL1, TH2, and TL2.

Serial Control Registers

SCON register used for SERIAL communication. See "Serial Communication" at later sections.

B Register

B register is used for multiplication and division.

PCON Register

Power down mode and IDLE mode are standard operating mode for 8052 compatible CPU. See "power control modes" at later sections.

Non-compatible Functions and Registers

MS81/MS89 Series thus have the following non-compatible functions/SFR to standard 8052 CPU.

a) TDIV register. Formerly 8052 a cycle needs 12 clocks.
b) PCON register bits 4,5. MS89 Series may use different clock for power saving.
c) ADC related functions and registers. Standard 8052 has no ADC function.
d) PWM, PFD, SPI, I2C, EFLASH, Watchdog Timer related functions and registers. Formerly 8052 has none of these functions.

Standard 8052 Addressing Modes

The standard 8052 has the following addressing mode that are also used in MS89 and MS81 Series.

Implied Register Addressing Mode

The implied addressing mode is fast and usually needs only 1 bytes. It include the following instructions:

a) MOV instruction with ACC and R0~7 as operands.
b) ADD/ADDC/SUBB/ANL/ORL/XRL/DEC with ACC and R0~R7 as operands.
c) INC with ACC and R0~R7, or DPTR as operands.
d) Rotate and shift (RR/RL/RRC/RLC) instructions with ACC as its operands.
e) CPL (complementory) and CLR with ACC/Carry flag as its operands.
f) MUL and DIV instructions.
g) DJNZ with R0~R7, ACC as its operands.

The implied instruction usually is very fast executed and saves a lot of program memory.

Bit Addressing Mode

There are 256 bits of RAM/SFR can be direct accessed by CPU, with the following instructions:

a) SETB, set a bit.
b) CLR, clear a bit.
c) CPL, complement a bit.
d) "MOV <bit-address>,C" or "MOV C,<bit-address>"

The bit-addressable area has 2 region, first is memory from 20~3F, which maps to bit address 00-7F, and the other is SFR 80,88,90,98…F8.

Bit-addressing mode has following benefit comparing to normal parallel addressing modes:

a) Save time and speed. If no bit-addressing mode, CPU should READ a byte, AND/OR a byte, and then WRITE it back for the same function.
b) CPU and Hardware can set/clear the same bit. The above READ/AND-OR/WRITE will have a risk that some other bits may be toggled by hardware after READ but before WRITE. If no bit-addressing mode is provided, those bits can only be "READ-ONLY", which means CPU may NOT write to those bits to prevent hardware toggled bits modified by software. Bit addressing mode make the CPU has the ability to toggle the bits that can be set or cleared also by hardware. That is, some hardware interrupt can also be triggered by CPU itself, also named "fake" interrupts. It is not possible if no bit-addressing mode.

Immediate Addressing Mode

All the registers, IRAM and SFR may store "constant" to it. All constants are known as Immediate Addressing Mode.

Direct Addressing Mode

The "Direct Addressing Mode" can access all SFR at address 0x80~0xFF, while accessing RAM at 0x00~7F. All the movement, arithmetic, and stack operations support this addressing mode. The instructions include MOV, ADD, ADDC, SUBB, ORL, ANL, XRL, PUSH, POP.

Indexed (Indirect) Addressing Mode

Indexed addressing mode may access 3 kinds memories:

a) IRAM, using R0, R1 as index, with MOV A,@R0 instruction, or ADC A,@R0.
b) P-Memory, also R0, R1 as index, with instructions like MOVX A,@R0 instruction. Note that P2 will be used for high byte address for P-Memory. Here it is also the X-Memory.
c) X-Memory, with DPTR as 16-bit index, uses the instruction like MOV A,@DPTR.
d) Code Memory, with DPTR/PC as base, ACC as index. MOV A, @A+DPTR

MSHINE 8052 Instruction Execution

MSHINE 8052 executes the instructions most in 2 clock cycles, since most instructions have 2 bytes. Because it is Harvard machine, operation result write-back is overlapped with first instruction fetch cycle. The exception is when CPU is running at X-Memory. Because it becomes "VON-NEWMAN" machine when CPU runs X-RAM.

Some 1-byte instructions like "SETB C", may be pipelined to 1 clock cycle, that is what "1-T" means.

MSHINE 8052 CPU needs more cycles to read X and P memory. For details, please contact MSHINE Technologies Corp.

MS81/MS89 Series Memory Map

MS89F/L SERIES has 256 bytes XMEM that can be switched to Program memory 0x0000~0x00FF. Total 512 bytes of RAM and 8192 bytes of ROM. Note that XMEM SFR and SRAM may be accessed by instructions like "MOVX A,@R0", with P2 set to 0x00 or 0xff.

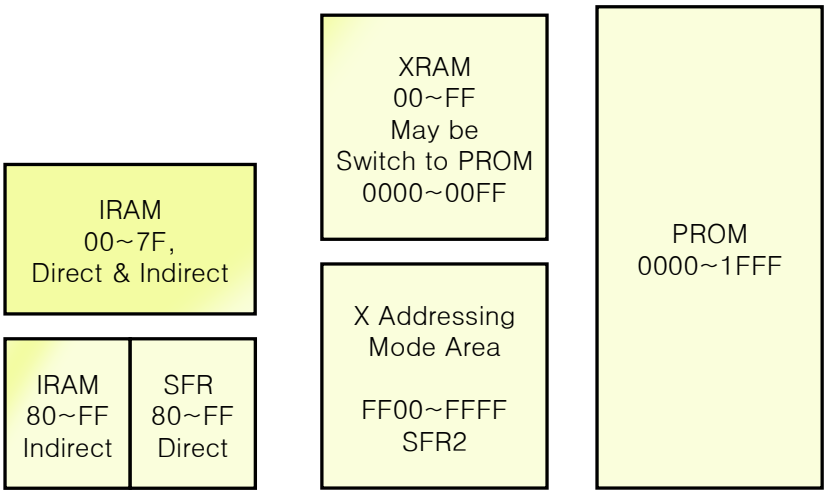


Figure 4. Memory Map.

SFR Registers

Special Function registers are listed in the following tables. Detailed descriptions are in later chapters.

Part A, Direct Addressing Mode SFR registers.

HIGH/LOW ADDRESS	0 (BIT ADDRESSABLE)	1	2	3	4	5	6	7
80H	P0	SP	DPL	DPH				PCON
88H	TCON	TMOD	TL0	TL1	TH0	TH1		
90H	P1							
98H	SCON	SBUF						
A0H								
A8H	IE							
B0H	P3							
B8H	IP	PFDRLD	PWH	RCCTL	XTLCTL	TOPT		PWCNTH
C0H	SYSC	SYSC2	PWTH	PWPH	PWPL	PWEN	PWCON	PWCNT
C8H	T2CON		RCAP2L	RCAP2H	TL2	TH2	ADH	ADL
D0H	PSW		P0DIR	P1DIR		P3DIR	TDIV	
D8H	ADCON		ADCSRC		ANAIO1	ANAIO2	ADCREF	CMPCON
E0H	ACC							
E8H	LVDR							
F0H	B							

F8H								

Part B. XMEM Addressing Mode.

HIGH/LOW ADDRESS	0	1	2	3	4	5	6	7
FF00H	XMEMCON	DEVID						
FF08H	SPIINT	SPICON	SPIDAT					
FF10H	I2C0INT	I2C0CON	I2C0DAT	I2C0DIV	I2C1INT	I2C1CON	I2C1DAT	I2C1DIV
FF18H								
FF20H	EFLASHCON	PRGAL	PRGAH	PRGDAT				
FF28H								
FF30H								
FF38H								

Direct Addressing Mode registers

Address	Notation	Function	Default value
80H	P0	Port 0 control register note that only 2 bits are accessible here.	00H
81H	SP	Stack pointer	81H
82H	DPL	Data pointer Lower address	00H
83H	DPH	Data pointer High address	00H
87H	PCON	Power Control Register Bit 7 SMOD Bit 6 – Bit 5 CK1 Bit 4 CK0 Bit 3 GF1 Bit 2 GF0 Bit 1 PD Bit 0 IDL Bit 7: SMOD, set 1 for double baud-rate. Bit <5, 4>: <CK1, CK0>, set for CPU clock rate. Note default values of these 2 bits are also OTP Programmable. R/W of these 2 bits is available. =<0, 0> select CPUCLK=system clock (RC or OSC) =<0, 1> select CPUCLK=system clock (RC or OSC) /2 =<1, 0> select CPUCLK=system clock (RC or OSC) /4 =<1, 1> select CPUCLK=system clock (RC or OSC) /8 Note that CPUCLK is the only clock source of CPU, and the clock will give to timers, too. Bit 3: GF1, General bit addressable bit. Bit 2: GF0, General bit addressable bit. Bit 1: PD, Power down this chip by setting this bit to 1. Bit 0: IDL. Set this bit to 1 to enter idle mode.	0-NN0000 B
88H	TCON	Timer Control register,	00H

		Bit 7 TF1 Bit 6 TR1 Bit 5 TF0 Bit 4 TR0 Bit 3 IE1 Bit 2 IT1 Bit 1 IE0 Bit 0 IT0 Bit 7: TF1, Timer 1 overflow flag. Set by hardware when timer 1 overflow, and cleared by hardware when leaves the interrupt routine automatically. Bit 6: TR1, Timer 1 Run control bit, set/clear by software to enable/disable the timer/counter 1. Bit 5: TF0, Timer 0 overflow flag. Set by hardware when timer 1 overflow, and cleared by hardware when enters the interrupt routine automatically. Bit 4: TR0, Timer 0 Run control bit, set/clear by software to enable/disable the timer/counter1. Bit 3: IE1, External Interrupt 1 Flag. Set also by hardware when EX1(P3.3) interrupt happens. Cleared by hardware with RETI in corresponding service routine. Also, this bit can be set by CPU to cause software interrupt and clear it by CPU manually. Bit 2: IT1, interrupt 1 control. Set this bit 1 to enable INT1B EDGE interrupt. Bit 1: IE0, External interrupt 0 flag. Set by hardware when external interrupt (INT0B, P3.2) source goes from high to low. Cleared by hardware as it enters the interrupt service routine automatically. Also, this bit can be set by CPU to cause software interrupt and clear it by CPU manually. Bit 0: IT0, interrupt 0 control bit (FLAG). Set this bit 1 to enable INT0B EDGE interrupt.	
89H	TMOD	Mode control of timer 0 and timer 1. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 Timer 1 control Timer 0 control GATE C/TB M1 M0 GATE C/TB M1 M0 High 4 bits are used to control Timer1, and low 4 bits are used to control timer 0 Bit 7, 4: GATE, When TR1/0 is 1 and GATE=1, Timer/Counter1/0 will run only when INTx is high. When GATE=0, Timer/Counter1/0 will run only when TR1/0=1. Bit 6,3: C/TB, Timer or counter selector. This bit is cleared by software for timer operation, and set 1 for event counter operation. Bit <5,4>/<1,0>: M1,M0, Operation modes, listed below <M1, M0>= <0, 0>: 13 bit timer/counter. (8048 compatible) <0, 1>: 16 bit timer/counter. <1, 0>: 8 bit auto-reload timer/counter. <1, 1>: TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bit, TH0 is another timer controlled by timer 1 control bits, and TL1/TH1 is stopped.	00H

8AH	TL0	Timer/Counter 0 low byte	00H
8BH	TL1	Timer/Counter 1 low byte	00H
8CH	TH0	Timer/Counter 0 high byte	00H
8DH	TH1	Timer/Counter 1 high byte	00H
90H	P1	Port 1 control bits.	FFH
98H	SCON	Serial Port control register, Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 SM0 SM1 SM2 REN TB8 RB8 TI RI Bit <7,6>: SM0, SM1, Serial port mode control, =<0,0>, Mode 0, serial port is used as shift register, clock is FOSC/12 =<0,1>, Mode 1, serial port is used as 8-bit UART, clock is variable. =<1,0>, Mode 2, serial port is used as 9-bit UART, clock is Fosc/64 or Fosc/32. =<1,1>, Mode 3, serial port is used as 9-bit UART, clock is variable. Bit 5: SM2, Enable the multiprocessor communication feature in Mode 2 and 3. In Mode 0, this bit should set to 0. In mode 1, SM2=1 let RI is activated only when stop bit is received. In Mode 2 and 3, SM2=1 let RI is activated only when the 9th bit is 0. Bit 4: REN, set/clear by software to enable/disable the serial data receiving. Bit 3: TB8, the 9th bit value that will be send in mode 2 and 3. Bit 2: RB8, the 9th bit value of mode 2 and 3. In mode 1, this bit will be the received stop bit if SM2=0. In Mode 0, RB8 is not used. Bit 1: TI, Transmit interrupt flag, this bit will be set by hardware at the end of 8th bit time in mode 0. Or at the beginning of the stop bit of other modes. This bit must be cleared by software. Bit 0: RI, Receive interrupt flag, This bit will be set by hardware at the end of 8th bit time at Mode 0, or half way through the stop bit time in other modes. (Except see SM2) This bit must be cleared by software.	00H
99H	SBUF	Serial Port buffer, read this byte for data received, write this byte to transmit data.	XXXXXXXX B
A0	P2	Used for PMEM accessing. P2=0x00 will access SRAM of XMEM, and P2=0xff will access SFR of XMEM.	0FFH
A8H	IE	Interrupt enable register. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 EA ADC ET2 ES ET1 EX1 ET0 EX0 Bit 7: EA, all interrupts are disabled if this bit is 0.	00H

		Bit 6: ADC, Enable or disable ADC interrupt. Bit 5: ET2, Enable timer 2 interrupt if this bit set to 1. Bit 4: ES, Enable or disable serial port interrupt. Bit 3: ET1, Enable or disable timer 1 overflow interrupt. Bit 2: EX1, Enable or disable P3.3 (INT1) interrupt. Bit 1: ET0, Enable or disable timer 0 overflow interrupt. Bit 0: EX0, Enable or disable P3.2 (INT0) interrupt.	
B0H	P3	Port 3 control register	00H
B8H	IP	Interrupt Priority register, 0 is lower priority and 1 is higher priority. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 PADC PT2 PS PT1 PX1 PT0 PX0 Bit 6: PADC, ADC interrupt priority level. Bit 5: PT2, Timer 2 priority bit. Bit 4: PS, Serial Port priority level. Bit 3: PT1, Timer 1 priority level. Bit 2: PX1, external INT1B interrupt priority level. Bit 1: PT0, Timer 0 interrupt priority level. Bit 0: PX0, external INT0B interrupt priority level.	--00 0000 B
B9H	PFDRLD	PFD period reload register This register will load to a 8-bit counter, which make PFD toggle when that counter overflows.	XX
BAH	PWH	MSB settings of PWM channel n Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 PWL1 PWL0 – – – – PWT9 PWT8 Bit<7,6>: <PWL1,0> PWM Channel N resolution selection <0, 0> PWM has 7-bit resolution <0, 1> PWM has 8-bit resolution <1, 0> PWM has 9-bit resolution <1, 1> PWM has 10-bit resolution Bit <1, 0>: <PWT9,8>, PWM Duty threshold value setting, MSB. LSB is set/read from PWTH Register.	0000 0000B
BBH	RCCTL	RC oscillator control register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 RCOE ERCEN IRCEN OSCF[4:0] This register is mainly for test mode: Bit <7>: RCOE, output RC clock to P3.7 Bit <6>: ERCEN. When test mode, this bit set to 1 may enable External RC. Bit <5>: IRCEN. When using external RC, this bit set 1 may enable Bit <4:0>: OSCF. IRC frequency may be re-tuned by this register.	000nnnnn

BCH	OSCCTL	Crystal Oscillator Control register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 WDTEN XTLOE XTLEN Bit<4>, WDTEN, enable watchdog manually by setting this bit to 1. Bit<1>, XTLOE. Output XTL clock to P3.7 Bit<0>, XTLEN. When using IRC, enable this bit may enable XTL connected.	---0—00B
BDH	TOPT	Testing Option Register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 LVSEN LVDEN WDTEN WDTCK_SEL[2:0] This register is for testing purpose. When the option is not correctly write, the register may force the code to WDT or LVDR by corresponding bits.	00000000
BFH	PWCNTH	PWM Counter High Byte Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 PWCNT9 PWCNT8 PWM Counter value, of the channel specified by PWEN.	
C0H	SYSC	System Control Register, mainly for the special functions. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 PCC PBC ADCE PAC PFDCK1 PFDCK0 PFDE Bit 0: PFDE, set 1 to enable PFD function. Bit <2,1>: PFDCK <1,0>, select PFD clock source from <0,0>: oscillator <0,1>: oscillator/2 <1,0>: oscillator/4 <1,1>: oscillator/8 Bit 5: ADCE, set 1 to enable ADC function. Bit 3: PAC, set 1 change PWM channel A to CMP mode. Bit 6: PBC, set 1 change PWM channel B to CMP mode. Bit 7: PCC, set 1 change PWM channel C to CMP mode.	00H
C1H	SYSC2	System control register 2, mainly for IO and PFD Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 P3CM P1CM P0CM WTIE WTCLR WTIF Bit 1: Watchdog interrupt flag. Bit 2: WTCLR, set this bit 1 will clear the watch-dog counter. Bit 3: WTIE, set this bit 1 will make watchdog in interrupt mode.	0000-000B

		Bit 4: P0CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 0. Bit 5: P1CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 2. Bit 7: P3CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 3.	
C2H	**PWTH**	PWM toggle value. MSB is PWH.bit <0> Note that PWEN will select which channel PWM is active.	00H
C3H	**PWMPH**	PWM Period High Byte. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 PWM P9 Bit 0 PWM P8 This register can read/write the PWM period register high 2 bits of the channel specified by PWEN.	0xH
C4H	**PWMPL**	PWM Period Low Byte This register can read/write the period of PWM channel specified by PWEN.	XXH
C5H	**PWEN**	PWM enable control. Bit 7 PWACT1 Bit 6 PWACT0 Bit 5 – Bit 4 – Bit 3 PWCEN Bit 2 PWBEN Bit 1 – Bit 0 PWAEN PWXEN set 1 enable PWM channel X . PWACT<1,0>: select the channel active for PWCNT,PWTH,PWH,PWMPH,PWHPL, and PWCNTH.	X0H
C6H	**PWCON**	PWM Control register Bit 7 PCINV Bit 6 PBINV Bit 5 Bit 4 PAINV Bit 3 PCKB1 Bit 2 PCKB0 Bit 1 PCKA1 Bit 0 PCKA0 Bit <1,0>: <PCKA1, PCKA0>, select the clock source of PWM channel A <0, 0>: Channel A uses clock from OSC <0, 1>: Channel A uses clock from OSC/2 <1, 0>: Channel A uses clock from OSC/4 <1, 1>: Channel A uses clock from OSC/8 Bit <3,2>: <PCKB1, PCKB0>, select the clock source of PWM channel B and C. <0, 0>: Channel B,C uses clock from OSC <0, 1>: Channel B,C uses clock from OSC/2 <1, 0>: Channel B,C uses clock from OSC/4 <1, 1>: Channel B,C uses clock from OSC/8 Bit <7~4>: P<X>INV, Set 1 to make the PWM output inverted. This is useful for Push-pull operation.	00H

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td>C7H</td><td>PWCNT</td><td>PWM counter value setting. PWM channel n, MSB is PWH <4, 3>. Note that this register is to also set the initial value of the counter. This register is R/W.</td><td>00H</td></tr>
<tr><td>C8H</td><td>T2CON</td><td>
<table>
<tr><td>Bit</td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td></tr>
<tr><td>Notation</td><td>TF2</td><td>EXF2</td><td>RCLK</td><td>TCLK</td></tr>
<tr><td>Default</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>Bit</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>Notation</td><td>EXEN2</td><td>TR2</td><td>C/T2B</td><td>CP/RL2B</td></tr>
<tr><td>Default</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
<table>
<tr><td>BIT</td><td>NAME</td><td>BIT ADDRESS</td><td>DESCRIPTION</td></tr>
<tr><td>7</td><td>TF2</td><td>CFh</td><td>Timer 2 Overflow. This bit is set when T2 overflows. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered. This bit will not be set if either TCLK or RCLK bits are set.</td></tr>
<tr><td>6</td><td>EXF2</td><td>CEh</td><td>Timer 2 External Flag. Set by a reload or capture caused by a 1−0 transition on T2EX (P1.1), but only when EXEN2 is set. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered.</td></tr>
<tr><td>5</td><td>RCLK</td><td>CDh</td><td>Timer 2 Receive Clock. When this bit is set, Timer 2 will be used to determine the serial port receive baud rate. When clear, Timer 1 will be used.</td></tr>
<tr><td>4</td><td>TCLK</td><td>CCh</td><td>Timer 2 Receive Clock. When this bit is set, Timer 2 will be used to determine the serial port transmit baud rate. When clear, Timer 1 will be used.</td></tr>
<tr><td>3</td><td>EXEN2</td><td>CBh</td><td>Timer 2 External Enable. When set, a 1−0 transition on T2EX (P1.1) will cause a capture or reload to occur.</td></tr>
<tr><td>2</td><td>TR2</td><td>CAh</td><td>Timer 2 Run. When set,</td></tr>
</table>
</td><td></td></tr>
</table>

		timer 2 will be turned on. Otherwise, it is turned off. 1 C/T2B C9h Timer 2 Counter/Interval Timer. If clear, Timer 2 is an interval counter. If set, Timer 2 is incremented by 1-0 transition on T2 (P1.0). 0 CP/RL2B C8h Timer 2 Capture/Reload. If clear, auto reload occurs on timer 2 overflow, or T2EX 1-0 transition if EXEN2 is set. If set, a capture will occur on a 1-0 transition of T2EX if EXEN2 is set.	
CAH	RCAP2L	Capture register, low byte, see Timer 2.	00
CBH	RCAP2H	Capture register, high byte.	00
CCH	T2L	Timer 2 low byte	00
CDH	T2H	Timer 2 high byte	00
CEH	ADH	ADC result 8 MSBs, Note that ADC result is 12 bit, and this register can select the 8 bit result from ADC result, controlled at bit <7,6> of ADSRC register. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 ADC9 ADC8 ADC7 ADC6 ADC5 ADC4 ADC3 ADC2 Note that this register is read only.	00000000B
CFH	ADL	ADC result 2 LSBs, Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 ADC1 ADC0 – – – – – – Note that this register is also read only.	0000----B
D0H	PSW	Program status word, Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 CY AC F0 RS1 RS0 OV – P Bit 0: P, Parity flag. Bit 2: OV, Overflow flag. Bit <4, 3>: <RS1, RS0>, Register bank select bits. =<0, 0> for 00~07H =<0, 1> for 08~0FH	00H

		=<1, 0> for 10~17H =<1, 1> for 18~1FH Bit 5: F0, general purpose bit. Bit 6: AC, Auxiliary Carry flag. Bit 7: CY, Carry flag.	
D2H	P0DIR	Port 0 direction register, valid only when SYS2.P0CM=1. 0 means input to CPU, 1 means output from CPU.	00H
D3H	P1DIR	Port 1 direction register, valid only when SYS2.P1CM=1. 0 means input to CPU, 1 means output from CPU.	00H
D5H	P3DIR	Port 3 direction register, valid only when SYS2.P3CM=1. 0 means input to CPU, 1 means output from CPU.	00H
D6H	TDIV	Timer pre-division value Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 – – T2C1 T2C0 T1C1 T1C0 T0C1 T0C0 Bit <1, 0>: <T0C1, T0C0> select clock to timer 0 <0, 0>: Use CPUCLK/12, (Standard CPU) <0, 1>: Use CPUCLK/8 <1, 0>: Use CPUCLK/4 <1, 1>: Use CPUCLK/2 Bit <3, 2>: <T1C1, T1C0> select clock to timer 1 <0, 0>: Use ,CPUCLK/12, (Standard CPU) <0, 1>: Use CPUCLK/8, <1, 0>: Use CPUCLK/4 <1, 1>: Use CPUCLK/2 Bit <5, 4>: <T2C1, T2C0> select clock to timer 2 <0, 0>: Use ,CPUCLK/12, (Standard CPU) <0, 1>: Use CPUCLK/8, <1, 0>: Use CPUCLK/4 <1, 1>: Use CPUCLK/2	---- 0000B
D8H	ADCON	Analog to digital Converter control register, Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 ADCK1 ADCK0 – – – RDY Bit 0: RDY, this bit will set 1 by hardware after a sample of ADC is finished conversion, and will be cleared automatically by hardware at interrupt vector. Bit <7, 6>: <ADCK1, ADCK0>, ADC clock source selection. ADCON.ADCK1,0 ADC Clock Source 00 OSC/16 01 OSC/32 10 OSC/64 11 OSC/8	00H

DAH	ADCSRC	ADC INPUT SOURCE SELECT REGISTER Bit 7 ADSH 1 Bit 6 ADSH 0 Bit 5 SHEN Bit 4 SRC4 Bit 3 SRC3 Bit 2 SRC2 Bit 1 SRC1 Bit 0 SRC0 SHEN=0: ADH is MSB of ADC result. SHEN=0, ADH will have AD result shift as below. ADSH<1,0>: Decide which bits of ADC result will come to ADH register: =0,0: ADH is ONE BIT shift of ADC result =0,1: ADH is 2 bit shift of ADC result =1,0: ADH is 3 bit shift of result. =1,1, ADH is 4 bit shift of ADC result. Bit<4..0>: SRC<4..0>, select the ADC source signal Setting Source <0,0,0,0,0> P1.0 <0,0,0,0,1> P1.1 <0,0,0,1,0> P1.2 <0,0,0,1,1> P1.3 <0,0,1,0,0> P1.4 <0,0,1,0,1> P1.5 <0,0,1,1,0> P1.6 <0,0,1,1,1> P1.7 <0,1,0,0,0> P0.0 <0,1,0,0,1> P0.1 <0,1,0,1,0> P3.2 <0,1,0,1,1> P3.3 <0,1,1,0,0> P3.4 <0,1,1,0,1> P3.5 <0,1,1,1,0> P3.6 <0,1,1,1,1> P3.7 <1,X,X,X,> Internal voltage 1.2V	00 – 0000B
DCH	ANAIO1	Analog I/O Control Register 1. Bit 7 AIO7 Bit 6 AIO6 Bit 5 AIO5 Bit 4 AIO4 Bit 3 AIO3 Bit 2 AIO2 Bit 1 AIO1 Bit 0 AIO0 Set the corresponding I/O to analog purpose: Bit <7>: AIO7. Write 1 to set P1.7 as analog pin. Bit <6>: AIO6. Write 1 to set P1.6 as analog pin. Bit <5>: AIO5. Write 1 to set P1.5 as analog pin. Bit <4>: AIO4. Write 1 to set P1.4 as analog pin. Bit <3>: AIO3. Write 1 to set P1.3 as analog pin. Bit <2>: AIO2. Write 1 to set P1.2 as analog pin. Bit <1>: AIO1. Write 1 to set P1.1 as analog pin. Bit <0>: AIO0. Write 1 to set P1.0 as analog pin.	00H

DDH	ANAIO2	Analog I/O Control Register 1. Bit 7 AIO15 Bit 6 AIO14 Bit 5 AIO13 Bit 4 AIO12 Bit 3 AIO11 Bit 2 AIO10 Bit 1 AIO9 Bit 0 AIO8 Set the corresponding I/O to analog purpose: Bit <7>: AIO15, Write 1 to set P3.7 as analog pin. Bit <6>: AIO14. Write 1 to set P3.6 as analog pin. Bit <5>: AIO13. Write 1 to set P3.5 as analog pin. Bit <4>: AIO12. Write 1 to set P3.4 as analog pin. Bit <3>: AIO11. Write 1 to set P3.3 as analog pin. Bit <2>: AIO10. Write 1 to set P3.2 as analog pin. Bit <1>: AIO9. Write 1 to set P0.1 as analog pin. Bit <0>: AIO8. Write 1 to set P0.0 as analog pin.	00H
DEH	ADCREF	ADC Reference Source configuration ADCREF=0xA, ADC uses external reference voltage from P3.7. =0xB, MS89FXX will output 2.1 V reference at P3.7. =0x4, ADC uses VDD as its reference. =0x9, MS89FXX will output LDO 2.1 V at P3.7. If this register is set to 0x0A, ADC will use external supplied reference at P3.7. If this register is set to 0x0B, MS89FXX will output 2.1 V reference at P3.7 and take it as voltage reference. Note that external capacitor to GND is required at P3.7 is used for voltage reference.	-----0000
DFH	CMPCON	Analog Comparator Control register Bit 7 BGTR<2:0> Bit 6 Bit 5 Bit 4 BGSW Bit 3 Bit 2 VREFV<1:0> Bit 1 COP/CO[3] Bit 0 CMPEN BGTR<2:0>: ADC voltage trimming value. Correct value is at IFREN2 byte 5. BGSW: Set 1 will change LVD/LVR from BG reference to low power reference. COP: Set 1 will output comparator output to PAD. Set 0 (default) the comparator result (CO) is read by this bit only. That is, read/write this bit has different meaning. Write is to control if comparator is output to P3.6, read means read this result of comparator. CMPEN: Set 1 to enable comparator. VREF<1:0>, ADC VREF selection, 00: 1.9V 01: 2.3V 10: 2.7V 11: 3.6V For 5V parts, IFREN2 byte 5 will note the correct BGTR value for 3.6V. For 3V parts, IFREN2 byte 5 will have the BGTR	000000X0

[3] Because R/W means different function, CMPCON MUST not use instructions like ORL/ANL.

<table>
<tr><td></td><td></td><td>value for 2.7V.</td><td></td></tr>
<tr><td>E0H</td><td>ACC</td><td>Accumulator of CPU</td><td>00H</td></tr>
<tr><td>E8H</td><td>LVDR</td><td>Low Voltage Detect and Stop control.

Bit 7 RSTEN | Bit 6 LVSEN | Bit 5 LVSEL | Bit 4 LVD1 | Bit 3 LVD0 | Bit 2 LVDOUT | Bit 1 LVDEN | Bit 0 LVREN

Bit 0, LVREN, set this bit 1 to enable low-voltage-reset (LVR).

Bit 1, LVDEN, set this bit 1 to enable Low voltage detection. The result will come out at bit 2.

Bit 2, LVDOUT, if this bit read 1, then VDD has voltage higher than specified by LVD1 and LVD0.

Bit 5, set 1 for selecting LVD voltage to modify, as following table.

<table>
<tr><td>LVDR<4:3></td><td>Toggle voltage (V) for CMPCON.4=0</td><td>Toggle voltage for CMPCON.4=1</td></tr>
<tr><td>00</td><td>4.1</td><td>X</td></tr>
<tr><td>01</td><td>3.6</td><td>X</td></tr>
<tr><td>10</td><td>2.8</td><td>3.7</td></tr>
<tr><td>11</td><td>2.5</td><td>3.1</td></tr>
</table>

If bit 5 set to 0, it selects LVR voltage.

<table>
<tr><td>LVDR<4:3></td><td>Reset voltage (V) for CMPCON.4=0</td><td>Reset voltage for CMPCON.4=1</td></tr>
<tr><td>00</td><td>3.9</td><td>X</td></tr>
<tr><td>01</td><td>3.4</td><td>X</td></tr>
<tr><td>10</td><td>2.8</td><td>3.3</td></tr>
<tr><td>11</td><td>2.4</td><td>2.9</td></tr>
</table>

Bit 6: LVSEN, set 1 the comparator output will come out to stop CPU, prevent in-accurate access of memory. Release of clock should wait 1024 CPU clock cycles.

BIT 7: RSTEN. This bit set 1 will make P3.6 as reset pin. P3.6 is default input pin after power on reset. Note that this bit will back to simple input pin only after next power on.</td><td>-n000?0n</td></tr>
<tr><td>F0H</td><td>B</td><td>B register, mainly for MUL and DIV instructions.</td><td>00H</td></tr>
</table>

XMEM Addressing Mode registers

Address	Notation	Function	Default value
FF00	XMEMCON	XMEM control register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 XRP Set XRP to 1 will switch XMEM to PROM.	0
FF01	DEVID	Device ID for I2C checking.	0x2a
FF08	SPIINT	SPI Interrupt Register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 SPITIE **SPIRIE** **SPITIF** **SPIRIF** SPITIE: Tx Interrupt Enable SPIRIE: Rx Interrupt Enable SPITIF: Tx Flag, SPIRIF: Rx Flag	0
FF09	SPICON	SPI Control Register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 SPIEN **MASTER** **CKE** **CKP** SPIRST CKOE DR1 DR0 SPIEN: Set 1 to enable SPI. MASTER: Set 1 to be master. CKE: Set 0 to sample data at falling edge. Set 1 to sample data at rising edge. CKP: Set 0 to idle at low level, set 1 to idle at high level. SPIRST: Write 1 to reset SPI internal clock counters. CLKOE: Set 1 to enable clock out to receive/transmit a byte of data. DR1,0: Master clock out <00> FOSC/2 <0,1> FOSC/4 <1,0> FOSC/8 <1,1> WDT Clock.	00
FF0A	SPIDAT	SPI Data Register	xx
FF10	I2C0INT	I2C Interrupt Register Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 **START** **STOP** SCI2FLG ASLV ACKF SLRW Bit 5, Start: Read 1 means start condition sensed. This will also wake up CPU. Write 0 will clear it. Bit 4, Stop: Read 1 means stop condition sensed. Write 0 will clear it. Bit 3, SCI2FLG: In Master mode, read 1 when a byte received/sent. Write 0 to clear it. Slave mode: Read 1 when a byte received / read back from master. ASLV: Address from master matches slave. Our slave address	00

		ACKF: ACK flag is received or sent. SLRW: In slave mode, this bit denotes if external master want to read or write MS89F/L SERIES. This bit reads 1 if external master want to read MS89XX.	
FF11	I2C0CON	I2C-0 Control register Bit 7 SCIEN; Bit 6 **MASTER**; Bit 5 **RX_TX**; Bit 4 **ACK_GEN**; Bit 3 INTEN; Bit 2 TR_START; Bit 1 TR_STOP; Bit 0 SCIEN: Set 1 to enable this I2C port MASTER: Set 1 to be master. RX_TX: Set 1 to be TX mode in master. ACKGEN: Set 1 to enable ACK at RX mode. INTEN: Enable interrupt. TR_START: Send start condition at master, next byte. TR_STOP: Send stop condition at master, next byte.	10010000 B
FF12	I2C0DAT	I2C Data	XX
FF13	I2C0DIV	I2C Clock Division. From OSC clock /2.	FF
FF14	I2C1INT	I2C Interrupt Register Bit 7; Bit 6; Bit 5 **START**; Bit 4 **STOP**; Bit 3 SCI2FLG; Bit 2 ASLV; Bit 1 ACKF; Bit 0 Bit 5, Start: Read 1 means start condition sensed. This will also wake up CPU. Write 0 will clear it. Bit 4, Stop: Read 1 means stop condition sensed. Write 0 will clear it. Bit 3, SCI2FLG: In Master mode, read 1 when a byte received/sent. Write 0 to clear it. Slave mode: Read 1 when a byte received / read back from master. ASLV: Address from master matches slave. Our slave address ACKF: ACK flag is received or sent.	00
FF15	I2C1CON	I2C-0 Control register Bit 7 SCIEN; Bit 6 **MASTER**; Bit 5 **RX_TX**; Bit 4 **ACK_GEN**; Bit 3 INTEN; Bit 2 TR_START; Bit 1 TR_STOP; Bit 0 SCIEN: Set 1 to enable this I2C port MASTER: Set 1 to be master. RX_TX: Set 1 to be TX mode in master. ACKGEN: Set 1 to enable ACK at RX mode. INTEN: Enable interrupt. TR_START: Send start condition at master, next byte. TR_STOP: Send stop condition at master, next byte.	10010000 B
FF16	I2C1DAT	I2C Data	XX
FF17	I2C1DIV	I2C Clock Division. From OSC clock /2. Note that lower 4 bits are fixed to 0.	FF
FF20	EFLASHCON	EFLASH control register	00

		Bit 7 CPURST Bit 6 **AUTOINC** Bit 5 **BUSY** Bit 4 **RD** Bit 3 SE Bit 2 MASE Bit 1 IFREN1 Bit 0 START CPURST: Set 1 to reset CPU. CPU will not access ROM/RAM when this bit is set. AUTOINC: Set 1 to auto increase address when read/write program data. BUSY: IAP is busy when read 1. RD: Set 1 to read the byte back to PRGDAT SE: Sector Erase enable when set to 1. MASE: Mass Erase enable when set to 1. IFREN1: Info sector 1 used if set to 1. Start: Operation start when set to 1.	
FF21	PRGAL	EFlash Address low byte	55H
FF22	PRGAH	EFlash Address high byte, etc. Bit 7 IFREN2 Bit 6 **OSC1** Bit 5 **OSC0** Bit 4 **PRGA12** Bit 3 PRGA11 Bit 2 PRGA10 Bit 1 PRGA9 Bit 0 PRGA8 IFREN2: Set 1 to enable reading of the second information sector[4]. OSC<1,0>: Timing to set for FLASH erase/programming: 00: FOSC <=10 MHz 01: FOSC <= 20 MHz 10: FOSC <= 40 MHz	00H
FF23	PRGDAT	EFlash Data byte	xx
FF24	EFLASHPROT	Only this byte write as 0xa5 will un-protect the flash memory.	00H

Interrupts

When CPU is running, special events may "interrupt" current program execution and jump to special service routine. Standard 80C32 has 6 interrupt routines. MS89F/L SERIES adds ADC, WDT, SPI and I2C service routines. The interrupt sources are listed below.

1. Timer 0 Overflow
2. Timer 1 Overflow
3. INT0B,
4. INT1B,
5. Timer 2 Overflow.
6. Serial Port Transmission and Receiving.
7. ADC
8. WDT
9. SPI

[4] The second information sector is written by MSHINE before shipping. It cannot be erased or programmed to other value.

10. I2C

Interrupt Vectors

When interrupts happen, the CPU will go to the vector specified below.

EVENT	VECTOR ADDRESS
Reset	0000H
IE0 (INT0B pin)	0003H
TF0 (Timer 0 overflow)	000BH
IE1 (INT1B pin)	0013H
TF1 (Timer 1 overflow)	001BH
RI & TI (Serial Port events)	0023H
TF2, EXF2	002BH
ADC (ADC data ready)	0033H
WDT(Watch-Dog) in Interrupt Mode	003BH
SPI	0043H
I2C	004BH

Interrupt Priority

MS89F/L SERIES has 2 level priorities like standard 80C32, and can be set by IP register. High level priority interrupt may interrupt low level interrupt service routine.

ADDRESS	NOTATION	FUNCTION	DEFAULT VALUE
B8H	IP	Interrupt Priority register, 0 is lower priority and 1 is higher priority. Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 **PADC** **PT2** PS PT1 PX1 PT0 PX0 Bit 6: ***PADC***, ADC interrupt priority level. Bit 4: PS, Serial Port priority level. Bit 3: PT1, Timer 1 priority level. Bit 2: PX1, external INT1B interrupt priority level. Bit 1: PT0, Timer 0 interrupt priority level. Bit 0: PX0, external INT0B interrupt priority level.	--00 0000 B

Note that WDT, I2C and SPI are high priority level.

Interrupt Masking

MS89F/L SERIES has interrupts that can be enabled or disabled by IE register.

ADDRESS	NOTATION	FUNCTION	DEFAULT VALUE
A8H	IE	Interrupt enable register.	00H

		Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0 EA **ADC** ET2 ES ET1 EX1 ET0 EX0 Bit 7: EA, all interrupts are disabled if this bit is 0. Bit 6: ***ADC***, Enable or disable ADC interrupt. Bit 5: ET2, Set 1 to enable T2/T2EX interrupts. Bit 4: ES, Enable or disable serial port interrupt. Bit 3: ET1, Enable or disable timer 1 overflow interrupt. Bit 2: EX1, Enable or disable external interrupt 1. Bit 1: ET0, Enable or disable timer 0 overflow interrupt. Bit 0: EX0, Enable or disable external interrupt 0.	

Note that SPI & I2C related interrupt is not controlled by above register but SPIINT and I2CXCON register.

System Reset

MS89F/L SERIES has 3 sources that can make it into the initial known state:

1. Power-ON Reset (POR). MS89F/L SERIES has POR active at 1.9V.
2. Low-Voltage Reset (LVR). MS89F/L Series can reset itself with specified voltage.
3. Watchdog Overflow (WDTOV). Watchdog can be enabled by information sector.
4. Reset pin with reset pin enabled (P3.6). This pin is enabled by CPU itself at LVDR register (0E8H) bit 7. Note that this pin is high active and external PULL LOW resistor must be connected if RST function is used.

After reset, CPU will run the program from address 0x0000. Its program counter may be changed to other values by branch instructions, or by interrupt, which is described in later sections.

Clock Distribution

MS89F/L SERIES has 3 clock sources. The clock distribution is as below.

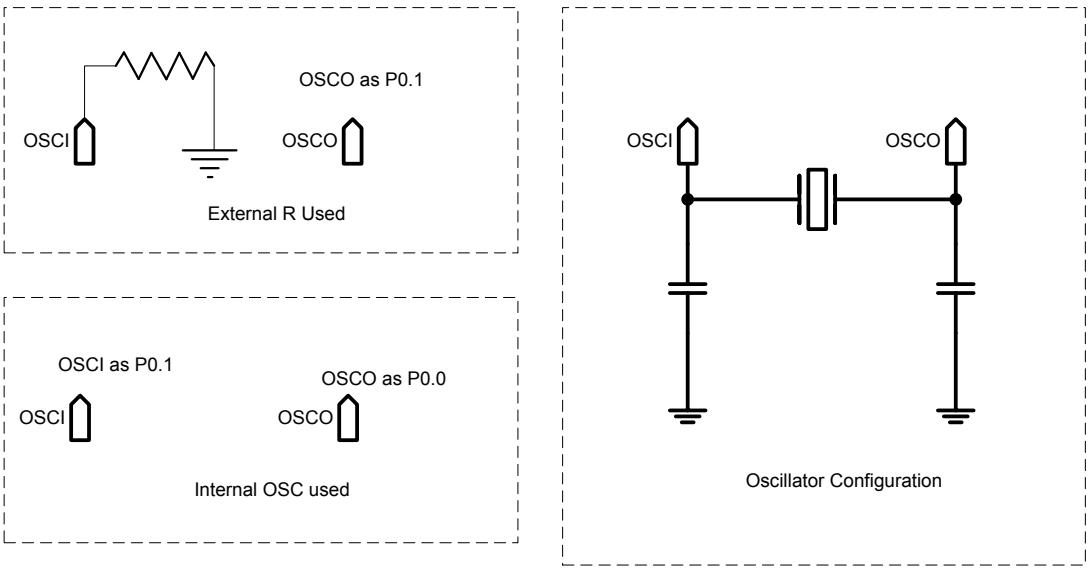


Figure 5. Oscillator Source

If internal oscillator is around 8 MHz, MS89FXX will read out Flash options at information sectors after around 100ms that system reset is released, and CPU will start running after another 100ms or around. MS89FXX cannot start running “immediately”, this measure is to prevent un-expected events while power is not stable

Note that CPU and its timers use the same clock source, or timers can use the clock source of slower. TDIV register can select MS89F/L Series’ timer clock source to CPU clock divided by 2, 4, 8, or 12. This make functions like UART comes to higher baud-rate.

Other functions not compatible to 80C32 can use clock sources faster than CPU. The faster clock sources make the function like PWM more precise.

When external OSC is selected, CPU must NOT write any value to P00/P01, or it will hang. Write any value to P00/P01 will cause the oscillator stop.

Power Control Modes

MS89F/L SERIES has power modes specified by PCON register. Using OSC clock / 4 or OSC clock / 8 may get more immunity from system noises.

Address	Notation	Function	Default
87H	PCON	Power Control Register	0-11 0000 B

REG.87H **PCON**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SMOD	-	CK1	CK0	GF1	GF0	PD	IDL

Bit 7: SMOD
1 = Enable double baud-rate
0 = Disable double baud-rate

Bit [5, 4]: [CK1, CK0]
Set for CPU clock rate. Note default values of these 2 bits are also Flash Programmable.
R/W of these 2 bits is available.
00 = select CPUCLK = SYS_CLK/1
01 = select CPUCLK = SYS_CLK/2
10 = select CPUCLK = SYS_CLK/4
11 = select CPUCLK = SYS_CLK/8 **(DEFAULT)**

IDLE Mode

At Idle mode, the clock does not enter the CPU but other peripherals still works. Any enabled interrupt of the CPU will wake it up and enters the service routine.

NOTE that 3 NOP instructions are required following PD or IDL instruction.

Power-Down Mode

When the chip enters the Power-Down Mode, all peripherals and clocks are stopped.

NOTE that 3 NOP instructions are required following PD or IDL instruction.

The following program enters power down mode:

```
PUSH    PCON
ORL     CMPCON,#10H         ; BG AT ADC MUST BE POWER DOWN
MOV     PCON,#0B0H
NOP
NOP
NOP
MOV     PCON,#0B2H;
NOP
```

```
NOP
NOP
POP     PCON
...
```

Wake-Up Sources

All enabled interrupt events can wake up CPU. If CPU is in power-down mode, the following sources can wake CPU up from Deep Sleep mode:

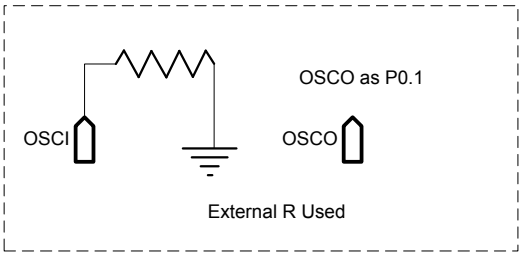
1. Reset => restart.
2. Watch-dog interrupt.
3. INT0B if INT0B interrupt enabled.
4. INT1B if INT1B interrupt enabled.
5. T2/T2EX if ET2 is set.
6. P3.0 if UART & UART interrupt enabled.

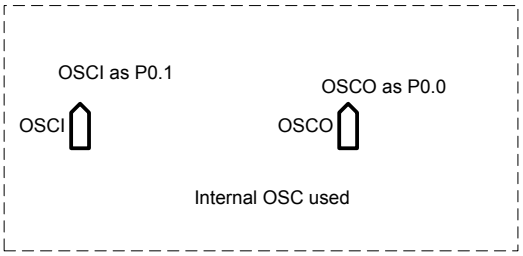
MSHINE has an additional document for interrupt/wake-up operation. Please contact MSHINE for further information.

On-Chip Oscillators

MS89F/L SERIES has on-chip RC oscillator (No external component required) or it can have external Resonator/Oscillator/RC connected that has better precision. Selection is done by information sector of MS89F/L SERIES. The configuration is as follows.

Resistor is 200K~10K Ohms. 200K ohm is around 400Hz.





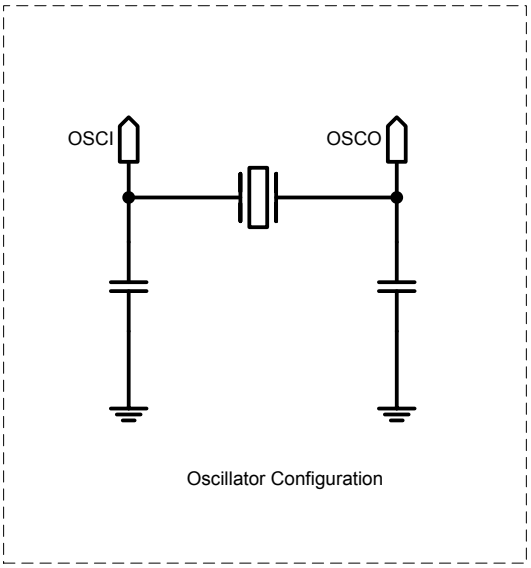


Figure 6. Clock Hardware configuration

Oscillator selection is done by information sector programming, and the writer tool may be used.

BTW, RCCTL register has default trimming values that tuned to 8MHZ+/- 2.5%. CPU may directly modify RCCTL to other frequency.

Low Voltage Stop/Detect/Reset

MS89F/L SERIES has low voltage [stop/detect] function that can stop or detect at voltage specified. The corresponding register is LVDR below.
Note that this register will be set at POR with info-sector data.

Address	Notation	Function	Default
E8H	LVDR	Low Voltage Detect and Stop control	0n011?1n

REG.E8H LVDR

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTEN	LVSEN	LVSEL	LVDL1	LVDL0	LVDOUT	LVDEN	LVREN

Bit 0, LVREN, set this bit 1 to enable LVR.
Bit 1, LVDEN.
0 = Disable LVD
1 = Enable LVD
Bit 2, LVDOUT
If this bit read 1, then VDD has voltage higher than specified by LVD1 and LVD0.
Bit 5, LVSEL, set 1 to select LVS/LVD, and set 0 for LVR voltage. LVD/LVR voltage may be different.
Bit [4:3], LVDL[1:0]
Set LVD/LVS compare supply voltage level[5].

LVDL[1:0]	LVR release voltage	LVD/LVS Release Voltage
00	2.4	2.5
01	2.55	2.8
10	3.4	3.6
11	3.85	4.1

Bit 6: LVSEN
To comparator output will come out to stop CPU, prevent in-accurate access of memory. Release of clock should wait 1024 CPU clock cycles
0 = Disable LVD
1 = Enable LVD
BIT 7: RSTEN.
This bit set 1 will make P3.6 as reset pin. P3.6 is default input pin after power on reset. Note that this bit will be reset only by APOR function.
0 = Disable external reset
1 = Enable external reset (low active)

[5] CMPCON should be set to IF2 byte 1, and additionally set CMPCON.4 to 0.

I/O Ports

MS89F/L SERIES has I/O up to 18 pins. Basically all can be configured as 8051 like standard I/O. The very old 8051 standard has a concept that CPU shall write 1 to read external H/L state, since the I/O pins is “almost” open-drain in old N-MOS 8051. After 80C32, the style is kept except that 2 cycles of strong high pulse (4mA) will output with stronger pull high (50uA) will be enabled if pin’s state is high. That is, in addition to TTL logic support old 8051 standard, 80C32 and MS81/MS89 Series can also drive external CMOS logic circuits directly and fast.

In addition to 80C32 style, if MS81/MS89 Series may have strong high output kept if CMOS mode of that port is enabled and DIR (direction) is set to output. In CMOS mode, input and output shall be set independently.

Port 0

Port 0 control register is located at 80H and must be accessed with direct addressing.

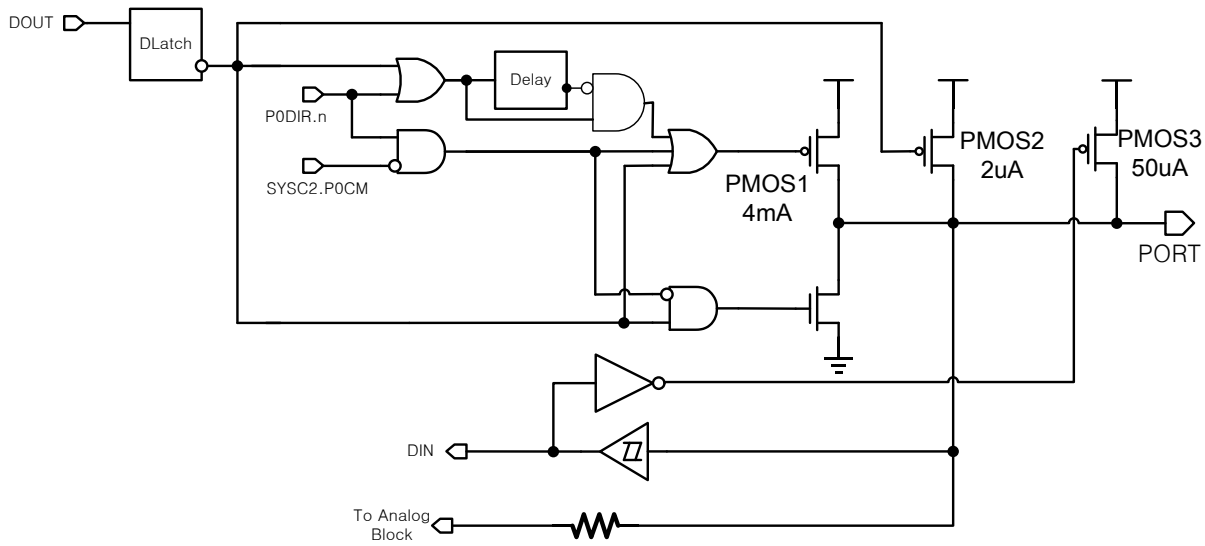


Figure 7. Port configuration.

In the figure above, when QB is from 1 to 0, PMOS1 will provide 4 mA for 2 cycles, and PMOS2 will provide 2uA constantly. If pin state is high, PMOS3 will provide another 50 uA additionally.

Note that PMOS1,2, and 3 will be disabled when setting to analog pin by ANAIO1/ANAIO2 registers.

P0 is shared with oscillator.

P0.0: OSCI, oscillator in, when external oscillator is selected.
P0.1: OSCO, oscillator out or RC oscillator resistor connection. Used as the special function when

external oscillator or RC oscillator is selected by programming INFO sector of the flash memory.

Port 1

P1 control register is at 90H of SFR registers. Port 1 is a general purpose port.

PORT FUNCTION	SPECIAL FUNCTION
P1.0	ADC0/PWM0/T2
P1.1	ADC1/PWM1/T2EX
P1.2	ADC2/PWM2
P1.3	ADC3/PWM3
P1.4	ADC4/SDA2/SPI_DO
P1.5	ADC5/SCL2/SPI_CLK
P1.6	ADC6
P1.7	ADC7/SDA1

Table 3. Port 1 multi-functions

Port 3

Port 3 has the same the same configuration as port 0.

Note: that there are many multi-function pins as below:

PORT FUNCTION	SPECIAL FUNCTION
P3.0	RXD
P3.1	TXD
P3.2	INT0B
P3.3	INT1B
P3.4	T0
P3.5	T1/SPI_DI
P3.6	PFD
P3.7[6]	SCL1/VREF/LDO

Table 4. Port 3 multi-functions

I/O Pin Interrupts

MS89 Series has 2 interrupt pins INT0B(P3.2) and INT1B(P3.3) can be used for interrupt sources. The interrupt source can be configured as LEVEL mode or EDGE mode. When it configured as LEVEL, CPU enters the interrupt routine whenever the source is LOW state. For “EDGE” mode, CPU enters the interrupt routine only when HIGH to LOW transition occurred.

[6] P3.7 has internal resistor around 17 K Ohm connect to GND, CPU must output it to low to enter low power mode.

That is, when TCON.IT0 or IT1 is set, corresponding INT0B(IE0) or INT1B(IE1) will be set as EDGE mode. CPU will enter the interrupt routine if EA=1, EXn=1, and IEn=1. The concept diagram of setting and clear IE0/IE1 is as Figure 8.

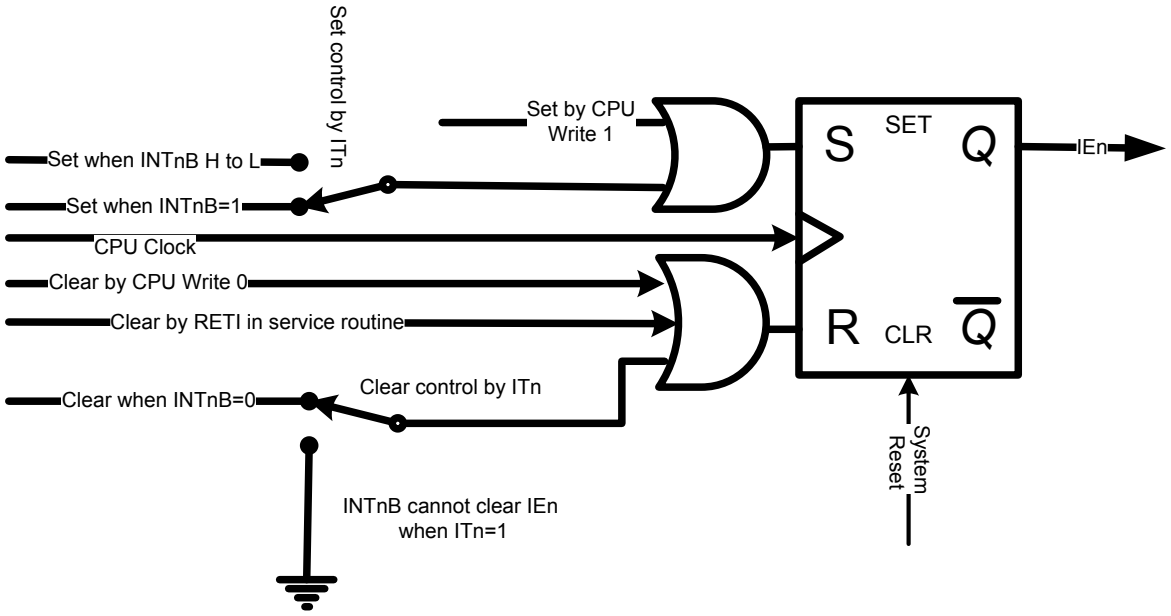


Figure 8. IE0/IE1 Concept Diagram.

It is shown above that IE0/IE1 can be modified by many resources. When IT0/IT1 is 0, it is mainly controlled by P3.2/P3.3 directly. If IT0/IT1 is 1, it can be set by EDGE of INT0B/INT1B, and cleared by CPU or RETI instruction in corresponding interrupt routine.

For key-button detection, using EDGE mode is a proper good idea but input bouncing is not easy to prevent. Usually CPU will clear EX0/EX1 if it needs to de-bouncing the input, by polling with timer interrupt. After the button is confirmed and the corresponding required functions are executed, EX0/EX1 will be set back again. In this kinds of scenario, IT0/IT1 set to 0 may still be a good method, because CPU will not always goes to the interrupt routine because EX0/EX1 is cleared.

CMOS I/O

MS89F/L SERIES is set to NMOS IO like 80C31. However, for large current driving applications, MS89F/L SERIES also provided CMOS style IO to use. To set port n as CMOS IO, just set corresponding SYSC2.PnCM bits to 1, and remember to set the direction register PNDIR.

CMOS I/O means that the direction of the port is fixed, like below.

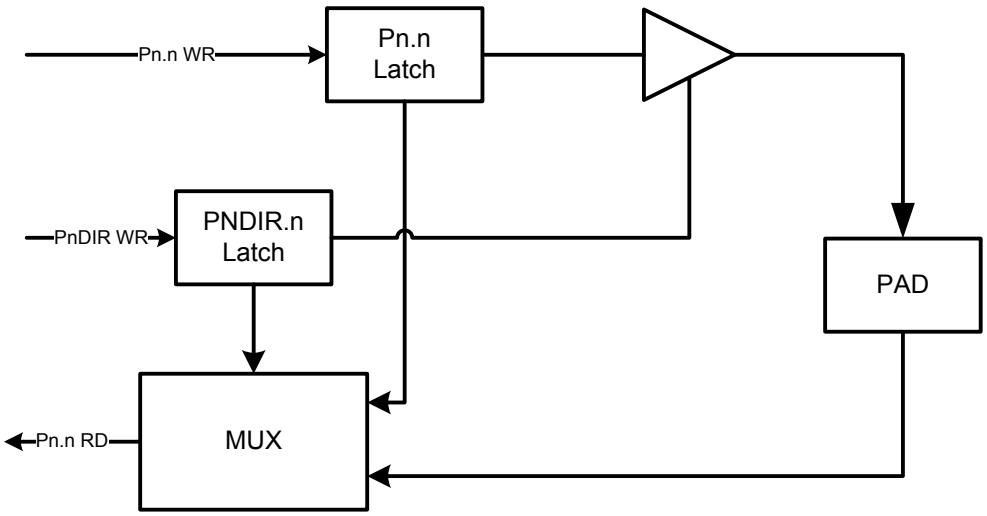


Figure 9. MS89F/L SERIES CMOS I/O.

When the port is set to CMOS I/O, PMOS1 will ON always when output high. That is, its direction should be set by corresponding PNDIR registers, and can output level only when corresponding DIR bit is 1. This makes MS89F/L SERIES drive motor/LED with HIGH logic level.

When I/O is set to CMOS-style, PMOS2 and PMOS3 in Figure 5 still works. That is, PMOS2 will still provides 2 uA unconditionally and PMOS3 will still provide up to 50 uA when the pin is high.

Note that P1.3 need special operation if need output high at CMOS Mode. Please contact MSHINE if P1.3 needs to output HIGH at CMOS mode.

Address	Notation	Function	Default
C1H	SYSC2	System control register 2, mainly for IO and PFD	0000-000B
D2H	P0DIR	Port 0 direction register, valid only when SYS2.P0CM=1. 0 means input to CPU, 1 means output from CPU.	00H
D3H	P1DIR	Port 1 direction register, valid only when SYS2.P2CM=1. 0 means input to CPU, 1 means output from CPU	00H
D5H	P3DIR	Port 3 direction register, valid only when SYS2.P3CM=1. 0 means input to CPU, 1 means output from CPU.	00H

REG.C1H

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CM		P1CM	P0CM	WTIE	WTCLR	WTIF	

Bit 1: WTIF, Watchdog Interrupt flag, read 1 if watchdog timer overflows, and interrupt enabled.
Bit 2: WTCLR, this bit is write-only. Write this bit 1 will clear watchdog timer.
Bit 3: WTIE, write this bit 1 will set watchdog timer to interrupt mode, but not reset MS89F/L SERIES CPU.
Bit 4: P0CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 0.
Bit 5: P1CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 1.
Bit 7: P3CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 3.

Timers

MS89F/L SERIES has the same timers as 80C32, with an additional watch-dog timer. The details are described as follows.

Timer 0 and Timer 1

MS89F/L SERIES has the standard timer configuration as below. Timer 0 and timer 1 can all be set as timers and counters, and has 4 different operating modes.

Timer 0 and Timer 1 have both 16-bits SFR named TH0/TL0 and TH1/TL1. These SFR may count-up automatically at a constant rate or when a HIGH-LOW transition on T0/T1 pin happened. It is shown in Figure 10 that it counts up at the following conditions:

c) GATE=0, CTB=0, TRn=1. This is a free count up timer.
d) GATE=1, CTB=0, TRn=1. This setting is used for count up gating. INT0B/INT1B may be used to gate the timer count up, and HIGH pulse width may be found by checking TH/TL registers.
e) CTB=1. At this setting, the timer will be used as event counter. It count up 1 when T0/T1 pin has high to low transition.

At MODE 0, T0/T1 is used as 13-bit counter that TL0/TL1 has only 5-bit width. In this mode, TL0/TL1 is count from 0 to 31, and reset to 0 with TH0/TH1 count up 1, from 0~255. If TH1/TL1 counts to 0, it continues counting up. This Arch is mainly to be compatible for … 8048, and people seldom uses it now.

At MODE 1, T0/T1 is used as 16-bit counter. Both TH and TL registers are 8-bit wide.

When TH1/TH0 overflows, they will set the flags corresponding TF1/TF0, which may cause CPU enter the interrupt routine if EA bit is 1. TF1/TF0 bits will be cleared by CPU manually or by an RETI instruction in the corresponding service routine.

Note that TF0/TF1 also can be set by CPU itself, and software interrupt may be generated.

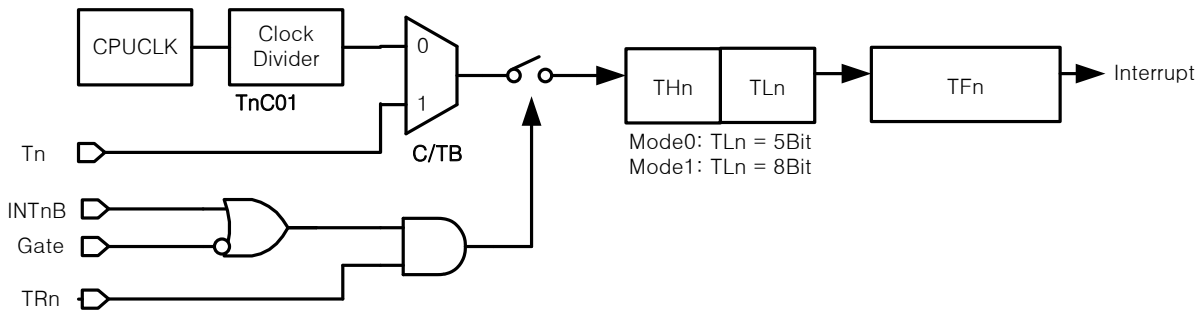


Figure 10. Timer 0 and Timer 1 Mode 0 and 1 configuration.

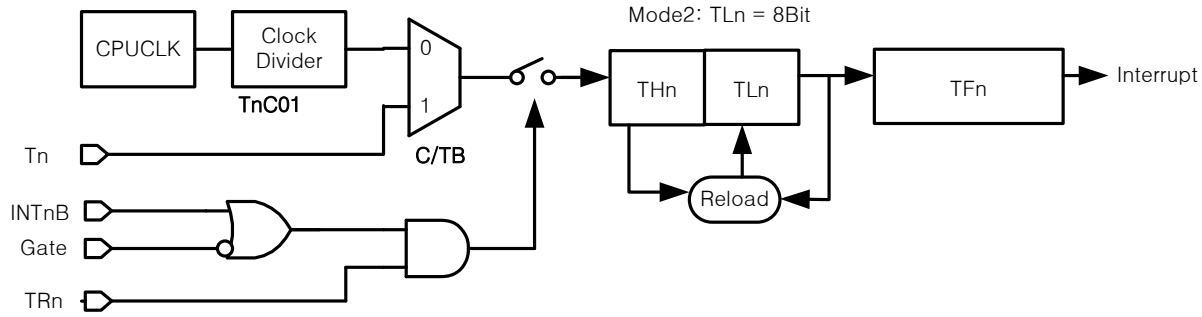


Figure 11. Timer 0 and Timer 1 Mode 2, auto reload mode.

As shown above, when [MODE1, MODE0] = [1, 0], the timer is set as auto-reload mode at MODE 2. At this mode, TF0/TF1 will be set when TL0/TL1 overflows. However, it will also load start-up value of TL0/TL1 from TH0/TH1 after overflow. That is, a periodic timer may be configured with different period.

For example, if a 1-ms period timer is required, and CPU is running 8MHZ/4. With TDIV bits sets to 01 (Divide of 8), the timer reload value should be set to 256-(2MHz * 0.001 S)/8 = 6. That is, if TH0 is set to 6 at CPU running 2 MHZ and TDIV.01 set to 00, with TR1 set to 1, 1-ms period interrupt may be generated.

When [MODE1,MODE0]=[1,1], timer 0 will be configured as MODE3. Mode 3 is used usually when TIMER1 is used for BAUD-RATE generator. At this mode, 2 timers can be configured by TH0 and TL0, like Figure 12.

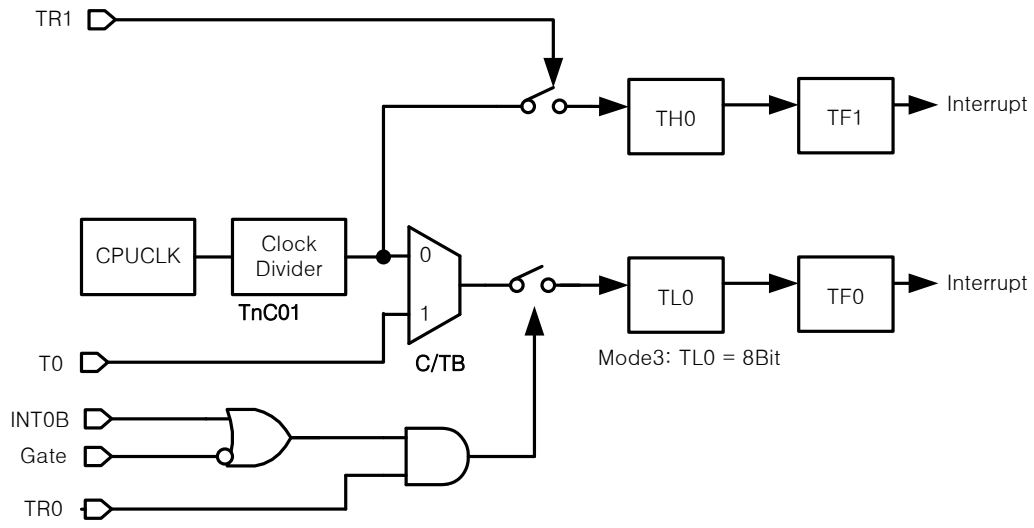


Figure 12. Timer 0 Mode 3

Timer 2

MS89F/L SERIES has the same timer 2 as 80C32. It may work at auto-reload mode or capture mode.

When working in 16-bit auto-reload mode, its configuration is as following

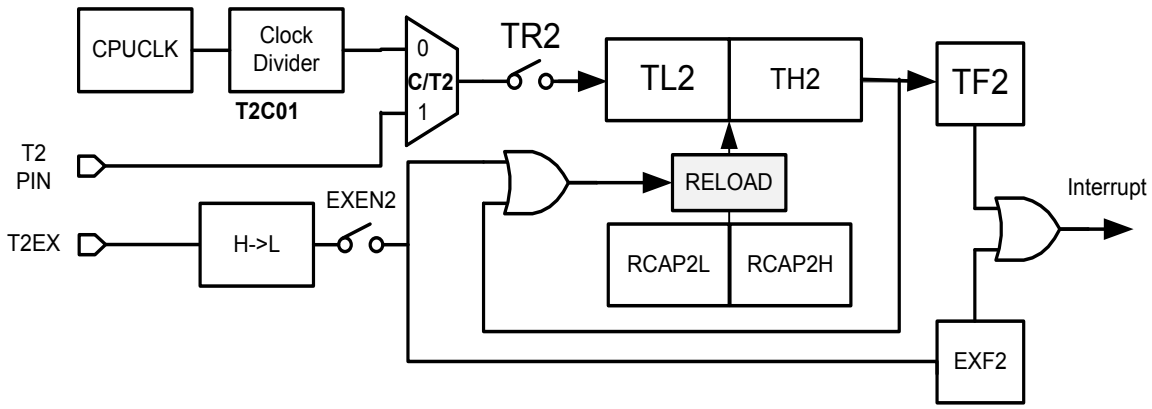


Figure 13. Timer 2 in auto-reload mode.

When timer 2 in auto-reload mode, TL2 and TH2 will reload its value when overflow, or EXEN2=1 and T2EX has transition H->L. And the interrupt will happen when TF2 is set or EXF2 detected.

Also Timer 2 may work in “capture mode”. In this mode, the counter value is captured to RCAP2H and RCAP2L when T2EX high->low transition detected. The configuration is as follows:

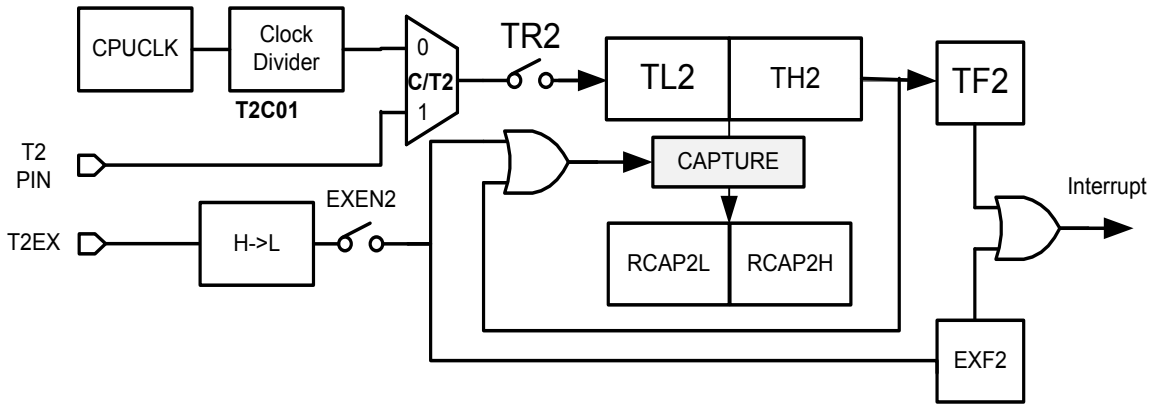


Figure 14. T2 in capture mode.

The corresponding SFR are listed as below.

Address	Notation	Function	Default
88H	TCON	Timer Control register,	00H

REG.88H **TCON**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit 7: TF1, Timer 1 overflow flag.
1 = timer 1 overflow event.

Note: When timer overflow, the interrupt routine automatically.

Bit 6: TR1, Timer 1 Run control bit
1 = enable timer/counter 1
0 = disable timer/counter 1

Bit 5: TF0, Timer 0 overflow flag.
1 = timer 0 overflow event.

Note: When timer overflow, the interrupt routine automatically.

Bit 4: TR0, Timer 0 Run control bit.
1 = enable timer/counter 0
0 = disable timer/counter 0

Bit 3: IE1, External Interrupt 1 edge flag.
1 = external interrupt (INT1B, P3.3) source goes from high to low.

Bit 2: IT1, interrupt 1 control bit.
1 = INT1B interrupt happened
0 = INT1B interrupt not happened yet.

Bit 1: IE0, External interrupt 0 edge flag.
1 = external interrupt (INT0B, P3.2) source goes from high to low.

Bit 0: IT0, interrupt 0 control bit.
1 = INT0B interrupt happened.
0 = INT0B interrupt not happened yet.

Address	Notation	Function	Default
89H	TMOD	MODE CONTROL OF TIMER 0 AND TIMER 1.	00H
8AH	TL0	Timer/Counter 0 low byte	00H
8BH	TL1	Timer/Counter 1 low byte	00H
8CH	TH0	Timer/Counter 0 high byte	00H
8DH	TH1	Timer/Counter 1 high byte	00H
D6H	TDIV	Timer pre-division value	---- 0000B

REG.89H TMOD

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Timer 1 control				Timer 0 control			
GATE	C/TB	M1	M0	GATE	C/TB	M1	M0

High 4 bits are used to control Timer1, and low 4 bits are used to control timer 0

Bit 7,4: GATE

When TR1/0 is 1 and GATE=1, Timer/Counter1/0 will run only when INTx is high.
When GATE=0, Timer/Counter1/0 will run only when TR1/0=1.

Bit 6,3: C/TB, Timer or counter selector.

This bit is cleared by software for timer operation, and set 1 for event counter operation.

Bit [5,4]/[1,0]: M1,M0,

00 = 13 bit timer/counter. (8048 compatible)
01 = 16 bit timer/counter.
10 = 8 bit auto-reload timer/counter.
11 = TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bit, TH0 is another timer controlled by timer 1 control bits, and TL1/TH1 is stopped.

REG.D6H TDIV

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	–	T2C1	T2C0	T1C1	T1C0	T0C1	T0C0

Bit [1, 0]: [T0C1, T0C0] select clock to timer 0
[0, 0]: Use CPUCLK/12, (Standard CPU)
[0, 1]: Use CPUCLK/8
[1, 0]: Use CPUCLK/4
[1, 1]: Use CPUCLK/2

Bit [3, 2]: [T1C1, T1C0] select clock to timer 1
[0, 0]: Use ,CPUCLK/12, (Standard CPU)
[0, 1]: Use CPUCLK/8,
[1, 0]: Use CPUCLK/4
[1, 1]: Use CPUCLK/2

Bit [3, 2]: [T2C1, T2C0] select clock to timer 2
[0, 0]: Use ,CPUCLK/12, (Standard CPU)
[0, 1]: Use CPUCLK/8,
[1, 0]: Use CPUCLK/4
[1, 1]: Use CPUCLK/2

Address	Notation	Function	Default
C8H	T2CON	MODE CONTROL OF TIMER 0 AND TIMER 1.	00H
CAH	TL0	Timer/Counter 0 low byte	00H
CBH	TL1	Timer/Counter 1 low byte	00H
CCH	TH0	Timer/Counter 0 high byte	00H
CDH	TH1	Timer/Counter 1 high byte	00H

REG.C8H **T2CON**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2B	CP/RL 2B

Bit 7: TF2,

Timer 2 Overflow. This bit is set when T2 overflows. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered. This bit will not be set if either TCLK or RCLK bits are set.

Bit 6: EXF2,

Timer 2 External Flag. Set by a reload or capture caused by a 1-0 transition on T2EX (P1.1), but only when EXEN2 is set. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered.

Bit 5: RCLK,

Timer 2 Receive Clock. When this bit is set, Timer 2 will be used to determine the serial port receive baud rate. When clear, Timer 1 will be used.

Bit 4: TCLK,

Timer 2 Transmit Clock. When this bit is set, Timer 2 will be used to determine the serial port transmit baud rate. When clear, Timer 1 will be used.

Bit 3: EXEN2,

Timer 2 External Enable. When set, a 1-0 transition on T2EX (P1.1) will cause a capture or reload to occur.

Bit 2: TR2,

Timer 2 Run. When set, timer 2 will be turned on. Otherwise, it is turned off.

Bit 1: C/T2B,

Timer 2 Counter/Interval Timer. If clear, Timer 2 is an interval counter. If set, Timer 2 is incremented by 1-0 transition on T2 (P1.0).

Bit 0: CP/RL2B,

Timer 2 Capture/Reload. If clear, auto reload occurs on timer 2 overflow, or T2EX 1-0 transition if EXEN2 is set. If set, a capture will occur on a 1-0 transition of T2EX if EXEN2 is set.

Watch-dog Timer

MS89F/L SERIES has an additional watch-dog timer that is different from 80C32. This timer is enabled by Flash option.

If this option is enabled, the software must clear the watch-dog timer by setting SYSC2.2 constantly. Other wise the watch-dog timer will force MS89F/L SERIES to reset or interrupted after a period of clock cycles. The period can be configured by Info Sector.

OTP have the options to reset MS89F/L SERIES after 2^(11+n) cycles and n is from 5 second to 15ms.

This timer cannot be used as other purpose. The only special register PSW is as below.

Address	Notation	Function	Default
C1H	SYSC2	System control register 2, mainly for IO and PFD	0000000XB

REG.C1H

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3CM		P1CM	P0CM	WTIE	WTCLR	WTIF	

Bit 1: WTIF, Watchdog Interrupt flag, read 1 if watchdog timer overflows, and interrupt enabled.
Bit 2: WTCLR, this bit is write-only. Write this bit 1 will clear watchdog timer.
Bit 3: WTIE, write this bit 1 will set watchdog timer to interrupt mode, but not reset MS89F/L SERIES CPU.
Bit 4: P0CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 0.
Bit 5: P1CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 1.
Bit 7: P3CM, set 0 for default 80C31 I/O structure, set 1 for CMOS I/O structure on Port 3.

If the watchdog timer is used to reset, CPU should set WTCLR to 1 periodically. If Watchdog Timer is used for interrupt source, the interrupt routine is usually like below, which can clear the interrupt.

```
?interrupt_3b:
                ANL SYSC2,#0F8H
                RETI
```

Note that if Watchdog timer is used to wake CPU up from IDLE/PD mode, please contact M-SHINE Technologies Corp. for corresponding examples.

Serial Interface – UART

MS89F/L series has an UART that is suitable for serial data transmission. Note that the interface has only ONE-byte buffer that the data must be checked by the software before the second byte finished receiving. The clock source may be generated by T1 or T2, and its configuration is via register SCON described below.

Address	Notation	Function	Default
98H	SCON	Serial Port control register,	00H

REG.98H SCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit [7,6]: SM0, SM1, Serial port mode control,
00 = Mode 0, serial port is used as shift register, clock is CPU Clock/12
01 = Mode 1, serial port is used as 8-bit UART, clock is variable.
10 = Mode 2, serial port is used as 9-bit UART; clock is CPU Clock/64 or CPU Clock/32.
11 = Mode 3, serial port is used as 9-bit UART, clock is variable.
Bit 5: SM2
1 = Enable multiprocessor communication feature in Mode 2 and 3
0 = Disable multiprocessor communication feature in Mode 2 and 3

If T1 is used for UART clock source, its baud-rate is as the following equation:

Baud-Rate= CPU Clock Rate / TDIV[2,3] Setting / 32 or 16 / (256-TH1)

For example, CPU uses 11.059 MHz, TDIV[3,2]=[1,1], PCON.7=1, and TH1=253, the baud-rate will be

11059000 Hz / 2 / 16 / (256-253) = 115198 BPS,

The baud-rate is very close to 115200 BPS.

Receiving Data when RI is 1

For standard 80C31, incoming start bit will be skipped if RI is not cleared yet. UART incoming data is usually processed by interrupt routine and RI will be cleared in a short time. However, if RI is not cleared before next data bit, the received data will be wrong. MS89/L series has an FLASH option that data bits may come even RI is not cleared. If only RI is cleared before next stop bit, the data may be received correctly.

PFD (Programmable Frequency Divider)

PFD is used with the PFDRLD below. A counter loads PFDRLD, and count up. After overflow, P3.4 toggles if PFD is enabled.

Address	Notation	Function	Default
C0H	SYSC	System control register, mainly for the special functions.	00H
B9H	PFDRLD	PFD period reload register This register will load to an 8-bit (DOWN) counter, which makes PFD toggle when that counter underflows.	XX

REG.C0H SYSC

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCC	PBC	ADCE		PAC	PFDCK1	PFDCK0	PFDE

Bit 0: PFDE
1 = enable PFD function.
0 = disable PFD function
Bit [2:1]: PFDCK [1,0], select PFD clock source from
00 = oscillator
01 = oscillator/2
10 = oscillator/4
11 = oscillator/8
Bit 5: ADCE
1 = enable ADC function.
0 = disable ADC function
Bit 3,6,7: PXC
1= Set PWM Channel X to [digital] comparator mode
0= Set PWM Channel X to PWM mode.

Pulse Width Modulation (PWM)

MS89F/L SERIES supports 3 channel PWM sources. The output is shared with P1.0, P1.2, and P1.3. The PWM resolution is 10 bits. It has PWM mode and comparator mode. In comparator mode, they all generate a 50% duty frequency. However, 3 of them mean 3 frequencies with 3 different phases, which will be described later. The basic concept is like Figure 15.

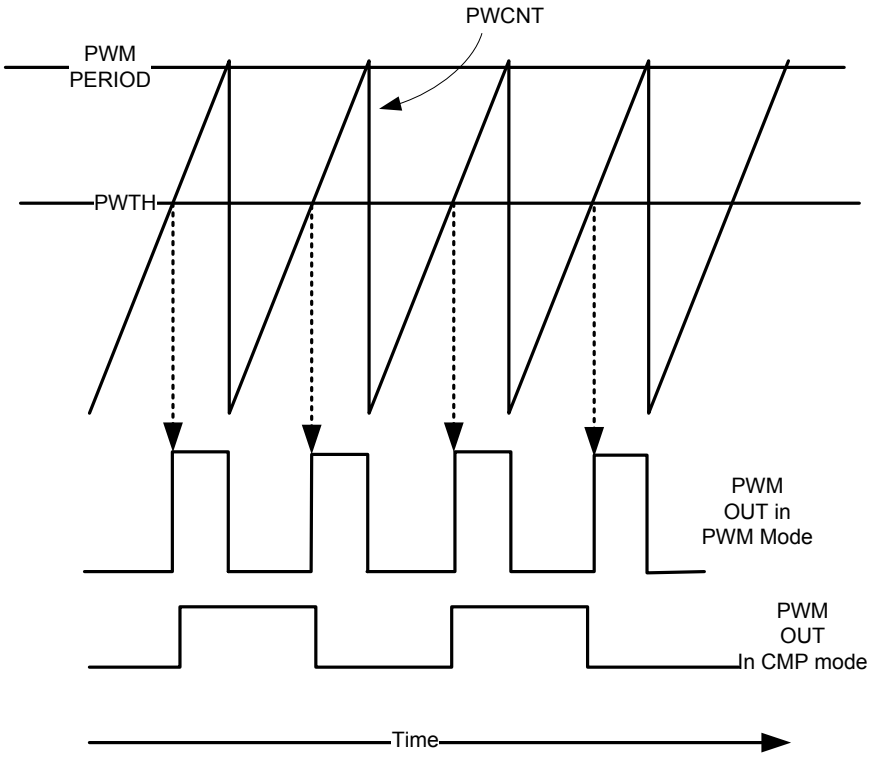


Figure 15. PWM Generation principle.

As Figure 15 shows, there are 3 variables in a PWM channel. One is the “period” of PWM output, which controls also the output frequency, is set by PWPH and PWPL registers. The other is PWMTH which controls the duty/phase of the output, is set by PWTH and PWH registers. Finally, the free up running PWCNT registers can be read and write by PWCNT and PWCNTH registers. PWCNT is freely count up, the circuit will compare the counter value with the value PWMTH. If PWMTH is equal to PWMCNT, the PWM output will be set (If PWMINV is 0)/or toggle. And PWM output will reset to 0 if counter is back to 0 in PWM mode.

Whenever a new PWTH is set, MS89F SERIES will use the new PWTH value after the PWCNT is back to 0. This measure is used to prevent high/low byte not set at the same time.

However, that is still possible high low byte of PWTH may be set before and after PWCNT is back to 0. CPU is better to check PWCNT to a farther value from 0, before set PWTH high and low bytes.

PWM Duty with PWTH SFR

PWM output has a special case of all 1 and all zero case. We recommend using "INV" bit in PWMCON register to achieve.

For example, if the PWM set to be 8-bit resolution, period is FF, and corresponding INV bit is 0, the PWM duty[7] may be set to **1/256~256/256** by setting PWTH 00~0xFF. If INV bit is set, the PWM duty will become **255/256~0/256** by setting PWTH 00~0xFF.

It is easier to get the concept by the following table.

PWTH	0	1	2	0xFF (255)
PWM DUTY when INV=0	1/256	2/256	3/256	256/256
PWM DUTY when INV=1	255/256	254/256	253/256	0/256

Table 3. PWM Duty with PWTH at 8-bit PWM setting.

Generate Signals in Same Frequency Different Phase

MS89F/L Series PWM module may generate up to 3 signals with same frequencies and different phase. Since all PWMCNT may start at the same time, all channels may have same CNT value and same frequency. With different PWTH settings in CMP mode, different channels may have different PHASE. An example is shown in **Figure 16**.

[7] Duty means HIGH period / total period.

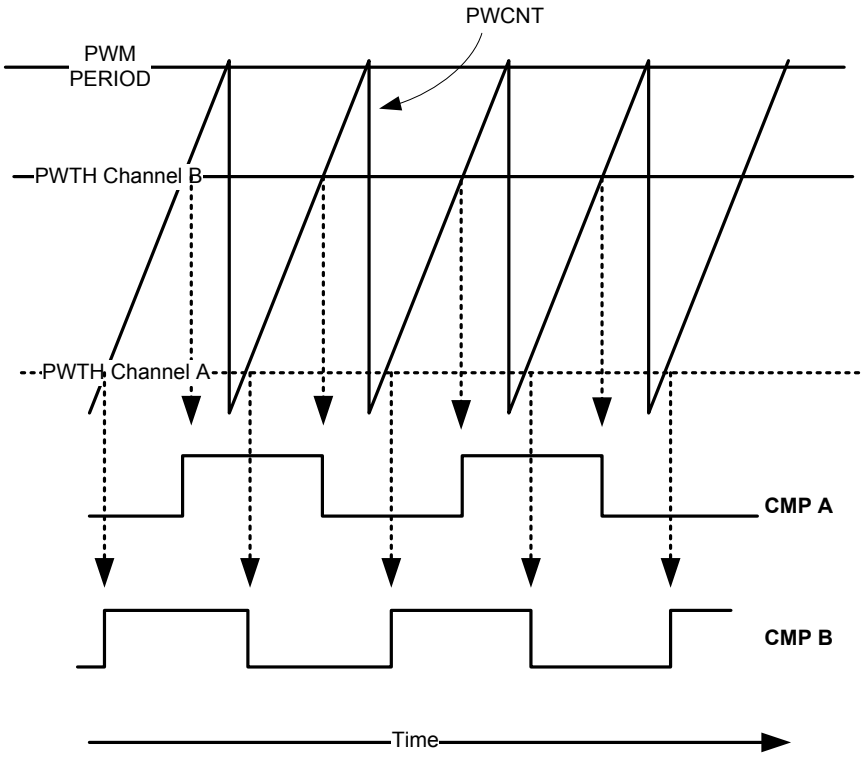


Figure 16. PWM Channels with CMP of different phase.

The corresponding registers are listed below:

Address	Notation	Function	Default
BFH	**PWCNTH**	PWM counter value, 2 MSB at bit <1,0>	00H
C0H	**SYSC**	System control register, CMP mode of PWM settings.	
C2H	**PWTH**	PWM toggle value. MSB is at PWH register.	00H
C3H	**PWPH**	PWM Period 2 MSB at bit <1,0>	03H
C4H	**PWPL**	PWM Period bit <7..0>	FFH
C5H	**PWEN**	PWM enable control.	X0H
C6H	**PWCON**	PWM Control register	00H
C7H	**PWCNT**	PWM Counter value, bit <7..0>.	00H
BAH	**PWH**	MSB settings of PWM	0000 0000B

REG.C5H, PWEN

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWACT1	PWACT0	–	–	PWCEN	PWBEN		PWAEN

PWEN bits [7,6] is used to set which channel PWM is active, and the registers PWTH/PWH/PWCNT will reference to that channel.
When bits <7,6>=<0,0>, PWTH/PWH/PWCNT will map to channel A registers.
When bits <7,6>=<1,0>, PWTH/PWH/PWCNT will map to channel B registers.
When bits <7,6>=<1,1>, PWTH/PWH/PWCNT will map to channel C registers.

Bits [3..0] are used to set if the PWM is enabled. Set corresponding bit to 1 will enable that PWM channel.

REG.C0H **SYSC**

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCC	**PBC**	**ADCE**		**PAC**	**PFDCK1**	**PFDCK0**	**PFDE**

Bit 0: PFDE
1 = enable PFD function.
0 = disable PFD function
Bit [2:1]: PFDCK [1,0], select PFD clock source from
00 = oscillator
01 = oscillator/2
10 = oscillator/4
11 = oscillator/8
Bit 5: ADCE
1 = enable ADC function.
0 = disable ADC function
Bit 3,6,7: PXC
1= Set PWM Channel X to [digital] comparator mode
0= Set PWM Channel X to PWM mode.

REG.C6H, PWCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCINV	PBINV		PAINV	PCKB 1	PCKB 0	PCKA1	PCKA0

Bit [1,0]: [PCKA1, PCKA0], select the clock source of PWM channel 0,1.
[0, 0]: Channel A uses clock from OSC
[0, 1]: Channel A uses clock from OSC/2
[1, 0]: Channel A uses clock from OSC/4
[1, 1]: Channel A uses clock from OSC/8

Bit [3,2]: [PCKB1, PCKB0], select the clock source of PWM channel 2,3.
[0, 0]: Channel B,C uses clock from OSC
[0, 1]: Channel B,C uses clock from OSC/2
[1, 0]: Channel B,C uses clock from OSC/4
[1, 1]: Channel B,C uses clock from OSC/8

Bit [4]: P[A]INV, Set 1 to make the PWM channel A output inverted.

Bit [6]: P[B]INV, Set 1 to make the PWM channel B output inverted.
Bit [7]: P[C]INV, Set 1 to make the PWM channel C output inverted.

REG.BAH, PWH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWL1	PWL0	–			–	PWT9	PWT8

Bit[7,6]: [PWL1,0] PWM resolution selection
[0, 0] PWM has 7-bit resolution
[0, 1] PWM has 8-bit resolution
[1, 0] PWM has 9-bit resolution
[1, 1] PWM has 10-bit resolution

Bit [1,0]: [PWT9,8], PWM Duty threshold value setting, MSB. LSB is set/read from PWTH Register.

Analog Comparator

MS89F/L Series has an independent comparator that can be used to check external voltage levels. The configuration is as below:

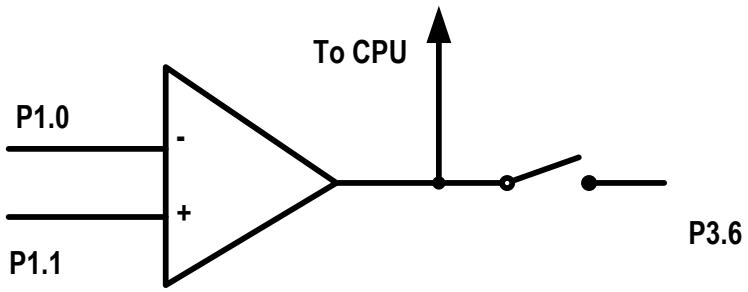


Figure 17. MS89F/L Comparator Configuration.

The comparator has output that can be read by CPU. If the result need to be de-glitch, P36 with external R/C filter to another input pin may work correctly. The configuration register is CMPCON .

REG.BFH, CMPCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
BGTR<2:0>			BGSW	VREFV<1:0>		COP/CO	CMPEN

Bit[7,6,5]: BGTR<2:0> ADC reference voltage trimming value.
Bit [4]: BGSW, set 1 to use low power LVD/LVR.
Bit [3,2], Select ADC internal generated reference voltage.

Bit 1: COP/CO, write 1 means comparator output to PAD3.6. Read 1 means P1.0 voltage >= P1.1, and read 0 means P1.1 voltage >= P1.0.

Analog to Digital Converter (ADC)

MS89F/L SERIES has an ADC with 16-channel input. The configuration is as below:

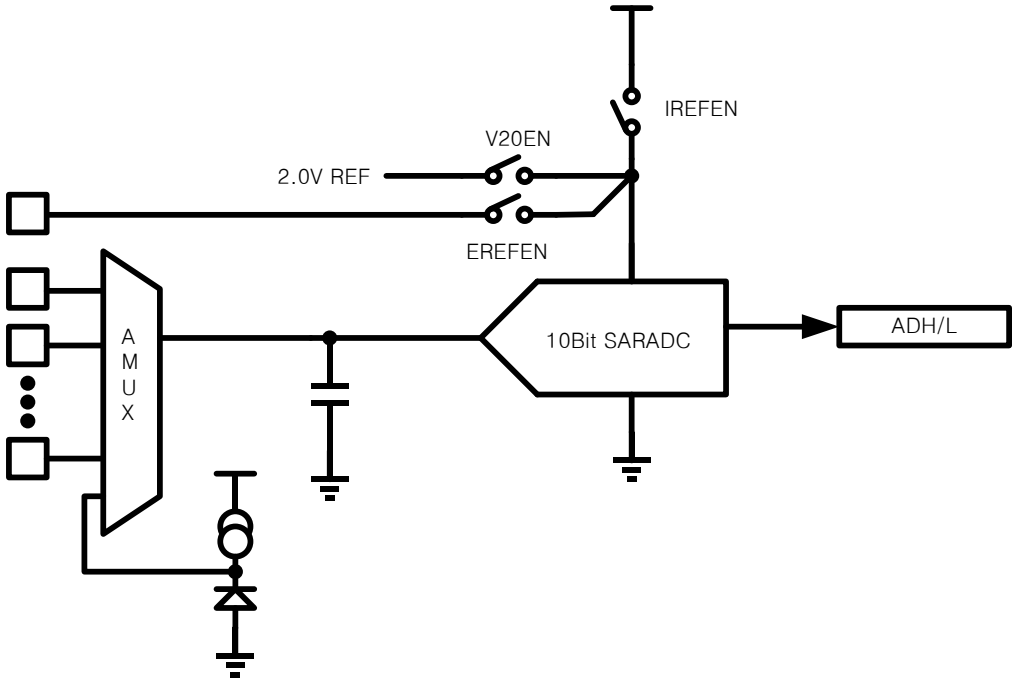


Figure 18. ADC and Reference Configuration

MS89F/L SERIES has built-in analog to digital converter that can have 10-bit result after one time of conversion. The corresponding registers are as below.

Address	Notation	Function	Default
C0H	SYSC	System control registers, mainly for the special functions.	00H
D8H	ADCON	Analog to digital Converter control register	00H

REG.C0H SYSC

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCC	PBC	ADCE		PAC	PFDCK1	PFDCK0	PFDE

Bit 0: PFDE
1 = enable PFD function.
0 = disable PFD function
Bit [2:1]: PFDCK [1,0], select PFD clock source from
00 = oscillator
01 = oscillator/2
10 = oscillator/4
11 = oscillator/8
Bit 5: ADCE
1 = enable ADC function.
0 = disable ADC function
Bit 3,6,7: PXC
1= Set PWM Channel X to [digital] comparator mode
0= Set PWM Channel X to PWM mode.

REG.D8H ADCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCK1	ADCK0			RST	START	MODE	RDY

Bit 0: RDY
1 Means ADC data ready. CPU Write 0 may clear it, or RETI in ADC interrupt routine may also clear it.
Bit 1: Mode, 0 is continuous mode, 1 is one-shot mode.
Bit 2: START, Write 1 to start ADC one-shot cycle if MODE=1.
Bit 3: RST, Write 1 will Reset ADC ONE-SHOT cycle if MODE=1.

Bit [7, 6]: ADCK[0:1]
ADC clock source selection

ADCK[1:0]	ADC Clock Source
00	OSC/16
01	OSC/32
10	OSC/64
11	OSC/8

Note that sample rate is ADC clock rate / 16. That is, when OSC is 8 MHz, ADCK[1:0]=00 will uses sample rate 31250 Hz.

Address	Notation	Function	Default
DAH	ADCSRC	ADC INPUT SOURCE SELECT REGISTER	00 – 0000B

REG.DAH ADCSRC

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADSH1	ADSH0	SHEN	SRC4	SRC3	SRC2	SRC1	SRC0

SHEN

0 = ADH is MSB of ADC result.

1 = ADH will have AD result shift as below. (See ADSH[1,0])

ADSH[1,0]:

Decide which bits of ADC result will come to ADH register:

00 = ADH is ONE BIT shift of ADC result

01 = ADH is 2 bit shift of ADC result

10 = ADH is 3 bit shift of result.

11 = ADH is 4 bit shift of ADC result.

Bit[4..0]: SRC[4..0],

Select the ADC source signal

Setting	Source
00000	P1.0
00001	P1.1
00010	P1.2
00011	P1.3
00100	P1.4
00101	P1.5
00110	P1.6
00111	P1.7
01000	P0.0
01001	P0.1
01010	P3.2
01011	P3.3
01100	P3.4
01101	P3.5
01110	P3.6
01111	P3.7
1XXX	Band-GAP VREF

Address	Notation	Function	Default
DCH	ANAIO1	Analog I/O Control Register 1.	00000000B
DDH	ANAIO2	Analog I/O Control Register 2.	
DEH	ADCREF	ADC Reference control	

REG.DCH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AIO7	AIO6	AIO5	AIO4	AIO3	AIO2	AIO1	AIO0

Set the corresponding I/O to analog purpose:
Bit [7]: AIO7. Write 1 to set P1.7 as analog pin.
Bit [6]: AIO6. Write 1 to set P1.6 as analog pin.
Bit [5]: AIO5. Write 1 to set P1.5 as analog pin.
Bit [4]: AIO4. Write 1 to set P1.4 as analog pin.
Bit [3]: AIO3. Write 1 to set P1.3 as analog pin.
Bit [2]: AIO2. Write 1 to set P1.2 as analog pin.
Bit [1]: AIO1. Write 1 to set P1.1 as analog pin.
Bit [0]: AIO0. Write 1 to set P1.0 as analog pin.

REG.DDH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AIO15	AIO14	AIO13	AIO12	AIO11	AIO10	AIO9	AIO8

Set the corresponding I/O to analog purpose:
Bit [7]: AIO15. Write 1 to set P3.7 as analog pin.
Bit [6]: AIO14. Write 1 to set P3.6 as analog pin.
Bit [5]: AIO13. Write 1 to set P3.5 as analog pin.
Bit [4]: AIO12. Write 1 to set P3.4 as analog pin.
Bit [3]: AIO11. Write 1 to set P3.3 as analog pin.
Bit [2]: AIO10. Write 1 to set P3.2 as analog pin.
Bit [1]: AIO9. Write 1 to set P0.1 as analog pin.
Bit [0]: AIO8. Write 1 to set P0.0 as analog pin.

REG. DEH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
				ADCREF3	ADCREF2	ADCREF1	ADCREF0

Set the corresponding I/O to analog purpose:
ADCREF=0xA, ADC uses external reference voltage from P3.7.
=0xB, MS89FLXX will output 3.6 V reference at P3.7.
=0x4, ADC uses VDD as its reference.
.
Do not set to other values.

The reference voltage is trimmed by MSHINE and recorded in IF2 byte #1. For MS89F Series, the typical voltage is 3.75V, for MS89L series, the voltage is 2.65V. Setting the voltage is by register CMPCON. Please use the reference program to set the voltage.

CEH	ADH	ADC result 8 MSBs,	00000000B
CFH	ADL	ADC RESULT 2 LSBS,	0000----B

REG.CEH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2

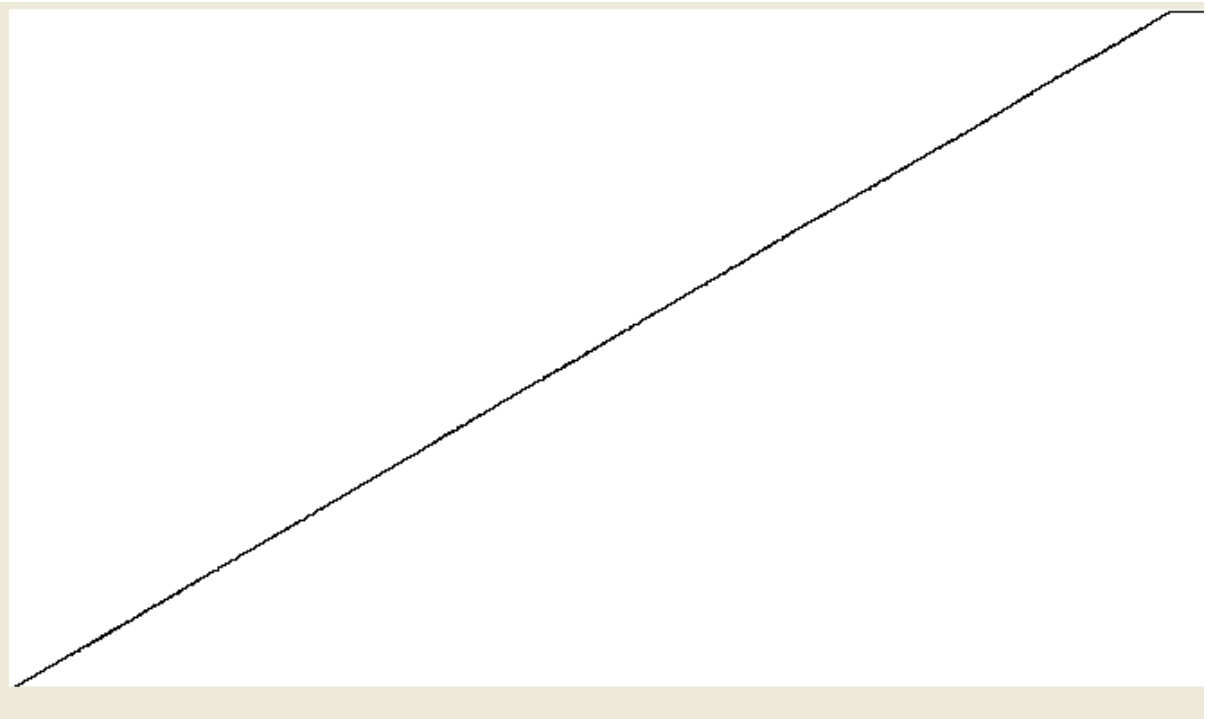
Note: That ADC result is 10 bit, and this register can select the 8 bit result from ADC result, controlled at bit [7,6] of ADSRC register.
Note: This register is read only.

REG.CFH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADC1	ADC0			–	–	–	–

Note that this register is also read only.

Typical ADC Curve is like below. The curve is measured with P3.7 generated VREF and CPU using IRC 8.0 MHZ at VDD=5.0V.



It is shown that offset is around 10~15MV at GND side and FULL-CODE out at 3.709V, while VREF measured at 3.706V, INL <= 2LSB if offset removed. INL <= 6 LSB if offset is not removed.

ADC Program Example

The following program will get out ADC data, and send out to UART. Using the P3.6 pin as analog. ADC uses external reference voltage from P3.7.

```
#include<reg52.h>
#include"f09reg.h"
void ADC_init(void);
void UART_init(void);
void UART_TX(unsigned char);
main()
{

        PCON=0x80;
        UART_init();
        ADC_init();
        SYSC=0x20;
        while(1)
        {
                ADCRDY=0;
                ADCSTART=1;
                while(!ADCRDY);
                UART_TX(ADL);
        }
}
void ADC_init()
{
        I2C0CON = 0;
        PRGAH=0X80;
        PRGAL=0X01;
        EFLASHCON=0X11; // READ THE VALUE
        CMPCON=PRGDAT;   // set the P37 VREF!!
        ANAIO1 = 0x0; //
        ANAIO2 = 0xC0; //P3.6 P3.7 analog pin   // P3.7 VREF
        ADCREF =      0x0B; //use vref
        ADCSRC = 0x0E; //use P3.6
        SYSC = 0x20;
        ADCON = 0x02; //one shot mode, osc/16
}
void UART_init(void)
{
        TDIV = 0x0D;
        SCON = 0x50;
        TMOD = 0x21;
        TH1 = 243;
        TL1 = 243;
        ET0 = 0;
        TR1 = 1;
        TI = 1;   //timer1 start
        EX0 = 0;
```

```
        EX1 = 0;
        RI = 0;
}
void UART_TX(unsigned char ch)
{
        SBUF = ch;
        TI = 0;
        while(TI==0);
}
```

And for KEIL compiler, the header file is like below:

```
#ifndef __F09REG_H__
#define __F09REG_H__

#ifndef __C51__
#define sfr extern volatile unsigned char
#define xdata
#define code
#define interrupt
#define sbit extern volatile unsigned char
#define bit unsigned char
#endif

sfr PFDRLD     = 0xB9;
sfr PWH        = 0xBA;
sfr RCCTL      = 0xBB;
sfr XTLCTL     = 0xBC;
sfr TOPT       = 0xBD;
sfr PWMCNTH = 0xBF;
sfr SYSC       = 0xC0;
sfr SYSC2      = 0xC1;
sfr PWTH       = 0xC2;
sfr PWMPH      = 0xC3;
sfr PWMPL      = 0xC4;
sfr PWEN       = 0xC5;
sfr PWCON      = 0xC6;
sfr PWCNT      = 0xC7;
sfr ADH        = 0xCE;
sfr ADL= 0xCF;
sfr P0DIR      = 0xD2;
sfr P1DIR      = 0xD3;
sfr P3DIR      = 0xD5;
sfr TDIV       = 0xD6;
sfr ADCON      = 0xD8;
sfr ADCSRC   = 0xDA;
sfr ANAIO1   = 0xDC;
sfr ANAIO2   = 0xDD;
sfr ADCREF     = 0xDE;
```

```
sfr CMPCON   = 0xDF;

sfr LVDR         = 0xE8;

#define XMEMCON (*((volatile unsigned char xdata *)0xff00))
#define DEVID (*((volatile unsigned char xdata *)0xff01))

#define SPIINT (*((volatile unsigned char xdata *)0xff08))
#define SPICON (*((volatile unsigned char xdata *)0xff09))
#define SPIDAT (*((volatile unsigned char xdata *)0xff0A))

#define I2C0INT (*((volatile unsigned char xdata *)0xff10))
#define I2C0CON (*((volatile unsigned char xdata *)0xff11))
#define I2C0DAT (*((volatile unsigned char xdata *)0xff12))
#define I2C0DIV (*((volatile unsigned char xdata *)0xff13))
#define I2C1INT (*((volatile unsigned char xdata *)0xff14))
#define I2C1CON (*((volatile unsigned char xdata *)0xff15))
#define I2C1DAT (*((volatile unsigned char xdata *)0xff16))
#define I2C1DIV (*((volatile unsigned char xdata *)0xff17))

#define EFLASHCON (*((volatile unsigned char xdata *)0xff20))
#define PRGAL (*((volatile unsigned char xdata *)0xff21))
#define PRGAH (*((volatile unsigned char xdata *)0xff22))
#define PRGDAT (*((volatile unsigned char xdata *)0xff23))
#define PRGPROT (*((volatile unsigned char xdata *)0xff24))

sbit P00 = P0^0;
sbit P01 = P0^1;
sbit P10 = P1^0;
sbit P11 = P1^1;
sbit P12 = P1^2;
sbit P13 = P1^3;
sbit P14 = P1^4;
sbit P15 = P1^5;
sbit P16 = P1^6;
sbit P17 = P1^7;
sbit P30 = P3^0;
sbit P31 = P3^1;
sbit P32 = P3^2;
sbit P33 = P3^3;
sbit P34 = P3^4;
sbit P35 = P3^5;
sbit P36 = P3^6;
sbit P37 = P3^7;
sbit ADCEN = IE^6;
sbit ADCRDY = ADCON^0;
sbit ADCSTART= ADCON^2;

#endif
```

SPI

MS89F/L SERIES has a SPI port that can communicate with external SPI device. Though the registers use X memory, MS89F/L SERIES can still access those register in a short time.

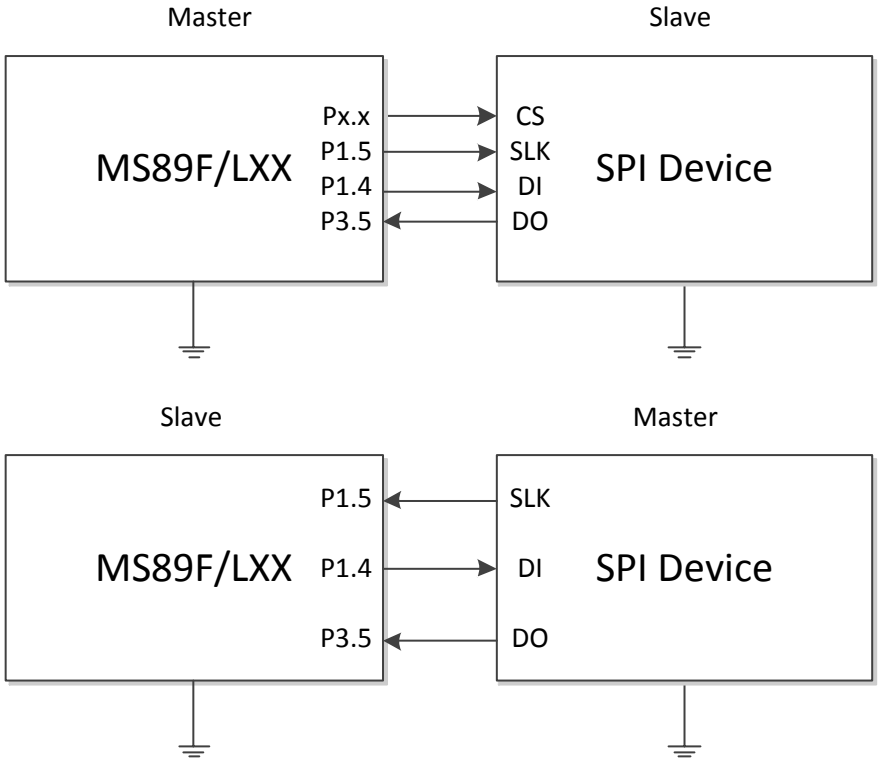


Figure 19. MS89F/L SERIES SPI Configuration

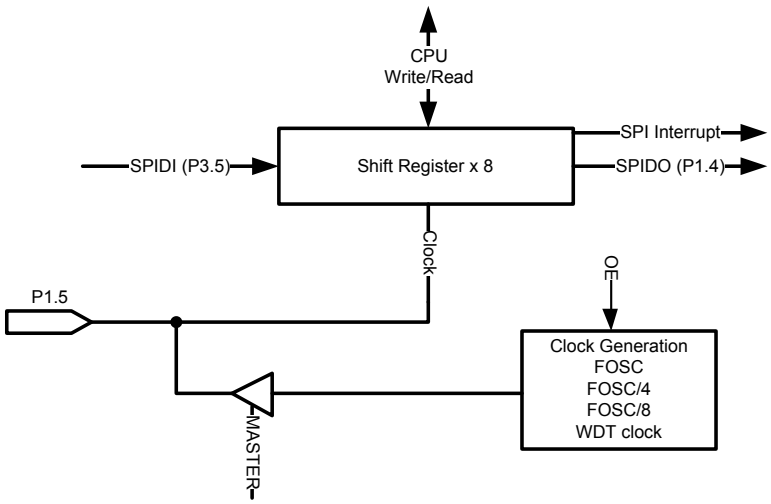


Figure 20. SPI Configuration.

As above figure, SPI clock source may be generated by itself or use external one. The master clock

can also set the phase and the idle state level. The configuration registers are as follows.

Note that SPI pins are shared with I2C port 1. I2C port 1 must be disabled before using SPI port.

Address	Notation	Function	Default
FF08	SPIINT	SPI interrupt control register	00H
FF09	SPICON	SPI control register	00H
FF0A	SPIDAT	SPI data register. Note that in master mode, data will shift out after SPICON.SPIOE is set 1.	00H

REG.FF08H, SPIINT

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPITIE	SPIRIE	SPITIF	SPIRIF				

SPIINT BIT[7], SPITIE:
Enable SPI Transmit interrupt if this bit is set.

SPIINT BIT[6], SPIRIE
Enable SPI Receive interrupt if this bit is set.

SPIINT BIT[5], SPITIF
This bit will read 1 if 8 bits of data is shift out. This bit must clear to 0 manually. “RETI” instruction in interrupt routine will not clear this bit.

SPIINT BIT[4], SPIRIF
This bit will read 1 if 8 bits of data is shift in. This bit must clear to 0 manually. “RETI” instruction in interrupt routine will not clear this bit.

Note that to discriminate TX and RX, MS89F/L SERIES check if data is written before clock is output/received by CPU. If data write to SPIDAT before clock is out/received, TX interrupt will happen. If no data is written to SPIDAT by CPU before clock is out/received, RX interrupt will happen.

REG.FF09H, SPICON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPIEN	MASTER	CKE	CKP	SPIRST	CKOE	DR1	DR0

SPIINT BIT[7], SPIEN:
Enable SPI function, P1.5, P1.4 and P3.5 will be switched to SPI function if this bit is set.

SPIINT BIT[6], MASTER
Set this bit will make MS89F/L SERIES as SPI Master. Or it will be slave. P1.5 will output clock if MS89F/L SERIES is master.

SPIINT BIT[5], CKE,
Set 0 to sample data at falling edge, set 1 to sample data at rising edge of clock.

SPIINT BIT[4], CKP
Set 0 make master clock idle at low level. Set 1 to make master clock idle at high level. This bit is valid for master only.

SPIINT BIT[3], SPIRST:
This bit is write only, set this bit 1 will reset internal SPI clock counter.

SPIINT BIT[2], CLKOE
Set this bit 1 to start output clock if MS89F/L SERIES is master mode.

SPIINT BIT[1,0], DR[1,0],
These bits are used to select SPI Master clock rate:
[0,0]: FORBIDDEN, do not use it.
[0,1]: FOSC/4 is used
[1,0]: FOSC/8 is used.
[1,1]: WDT clock is used.[8]

[8] Watchdog must be enabled by Info Sector Flash option first.

SPI Master Program Example

The following program will send out SPI data 0~99, and receive all the 100 data bytes, send out to UART.

```
unsigned char flag;
void spiint(void) interrupt {    // at 0x0043 vector, depends
                                 // on different compiler, the declaration should be modified
        flag=1;
        TI=0;
        SBUF=SPIDAT;
        while(!TI);
        TI=0;
}
main()
{
        unsigned char i;
        PCON=0x80;
        TH1=256-1;
        TH0=0;
        TMOD=0X22;
        SCON=0X50;
        TR1=1;
        ET0=1;

        I2C1CON=0;// DISABLE I2C
        SPICON=0xf8;
        SPIINT=0xc0;
        EA=1;
        for(i=0;i!=100;i++)
        {
                flag=0;
                SPIDAT=i;
                SPICON=0xf5;          // output clock, and data
                while(!flag);
        }
        return 0;
}
```

SPI Slave Program Example

Following is the program example for SPI Slave operation. MS89FXX will send out SPI data of last received increase 1. It will also output SPI received data to UART.

```
bit flag;
unsigned char pdata fifo[100];
unsigned char fifoi,fifoo;

void spiint(void) interrupt {    // at 0x0043 vector, depends
                                 // on different compiler, the declaration should be modified
        flag=1;
        SPIINT=0xc0;
        fifo[fifoi++]=SPIDAT;
        if(fifoi==100)
                fifoi=0;
        SPIDAT=SPIDAT+1;
}
main()
{
        unsigned char i;
        PCON=0x80;
        TH1=256-1;
        TH0=0;
        TMOD=0X22;
        SCON=0X50;
        TR1=1;
        ET0=1;

        fifoi=0;
        fifoo=0;
        I2C1CON=0;
        SPIDAT=0;

        SPICON=0XB8;
        SPIINT=0xc0; // enable interrupt now
        EA=1;

        while(1)
        {
                while(fifoi!=fifoo)
                {
                        TI=0;
                        SBUF=fifo[fifoo++];
                        if(fifoo==100)
                                fifoo=0;
                        while(!TI);
                }
                i++;
```

```
                TI=0;
        }
        return 0;
}
```

I2C

MS89F/L SERIES has 2 I2C port that can be used for device communication.

After reset, all the ports are active with slave mode enabled. That is, CPU can be programmed by external I2C master by either port. With special programmed I2C behavior, external master may communicate with MS89FXX with I2C standards. The details are as below.

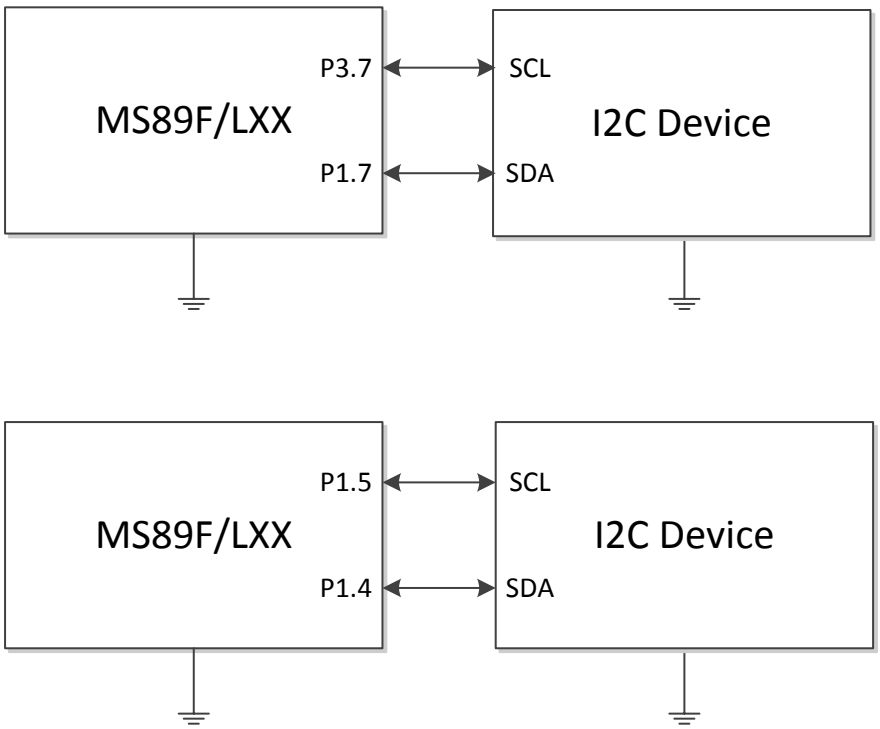


Figure 21. MS89F/L SERIES I2C Configuration

I2C Slave Addresses

MS89F/L SERIES needs special operation from I2C port. Special address 0x55 and 0x5a is reserved for special operation.

Slave Address	Function
0x55	This address is for internal FLASH testing purpose. User should not use it.
0x5A	This address is for I2C. External Master may R/W this address to perform register controlling.
DEV_ID	This address cannot be 0x55 or 0x5A. The additional address is used for master/slave communication for software.

I2C Slave Clock Stretching

MS89FXX I2C slave do clock stretching when it is busy. Clock stretching means MS89FXX is busy. That is, SCL will be pulled low by MS89FXX every time 8 bits are transferred. MS89FXX will release SCL signal only after interrupt is cleared. The method is as follows:

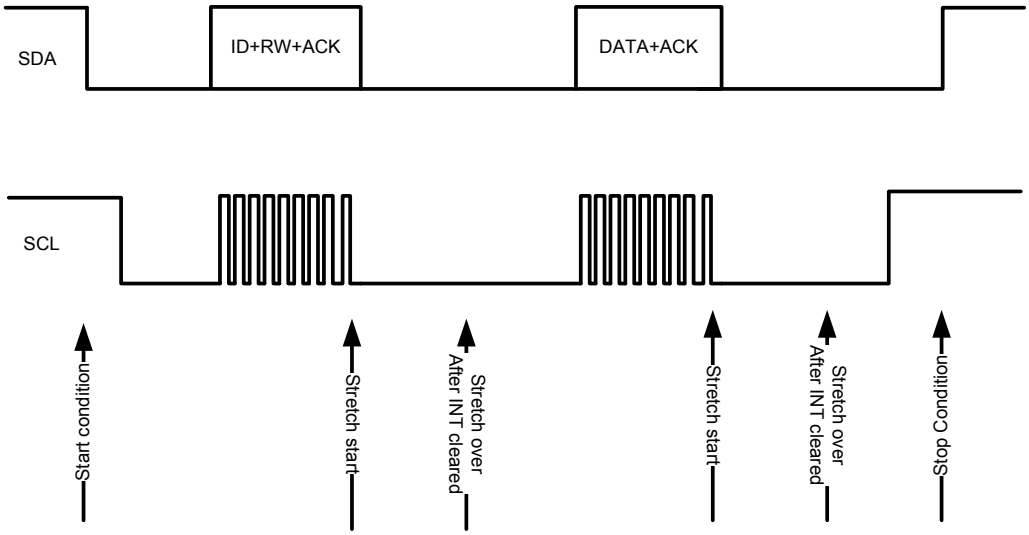


Figure 22. I2C Slave SCL Stretching.

I2C Control Registers

The control registers of I2C ports are listed as following.

Address	Notation	Function	Default
FF01	**DEVID**	Device ID of I2C address.	00H
FF10	**I2C0INT**	I2C Port 0 Interrupt register	00H
FF11	**I2C0CON**	I2C Port 0 Control register	10010000B
FF12	**I2C0DAT**	I2C Data register	00H
FF13	**I2C0DIV**	I2C clock division register. I2C Master will use FOSC/2/I2C0DIV as the clock rate or I2C Port 0.	
FF14	**I2C1INT**	I2C Port 1 Interrupt register	00H
FF15	**I2C1CON**	I2C Port 1 Control register	10010000B
FF16	**I2C1DAT**	I2C Data register	00H
FF17	**I2C1DIV**	I2C clock division register. I2C Master will use FOSC/2/I2C0DIV as the clock rate or I2C Port 1.	

REG.FF01H, DEVID

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
–	Device ID <6:0>						

Except to 0x5a and 0x55, DEVID may have other different values for slave operation.

REG.FF10/FF14, I2CXINT,

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	ARBL	START	STOP	SCI2FLG	ASLV	ACKF	SLRW

Note that bits 3~6 of I2CXINT can be cleared by CPU write 0, but will not be changed if CPU write 1 to it. It is a good method that the CPU directly writes 0xF7 to I2CXINT immediately. Bits 0~2 are read only bits. It will not change when CPU write to it.

Bit 6, ARBL,

Arbitration Lost. If I2C master does not read SDA high when it outputs high at data byte sending cycles, it means arbitration lost. It will generate interrupt, and this bit will set to 1. It will be cleared only when CPU write 0 to it.

Bit 5, START,

read 1 means start condition sensed. CPU will also be waked up if the I2C port enabled. CPU may clear it by write it to 0. CPU write 1 has no effect.

Bit 4, STOP,

read 1 means STOP condition sensed. CPU may clear it by write it to 0. Write 1 has no effect.

Bit 3, SCI2FLG,

This bit read 1 means a byte is send or received by I2C port. Interrupt will happen when a byte is send/received (with ACK), or arbitration lost. Note that SCL will be stretched if this bit is high. Interrupt will not happen when start/stop sensed.

Bit 2, ASLV,

This bit read 1 means the address from master matches ID. Usually MS89F/L SERIES will send ACK at this time. Note that this bit is read only for CPU.

Bit 1, ACKF

ACKF read 1 means ACK bit is received or sent. This bit is also read only.

Bit 0, SLRW

In slave mode, this bit denotes if external master want to read or write MS89F/L SERIES. If master want to read MS89XX, it will sent first byte LSB to 1,

and this bit will read 1. This bit is also read only.

REG.FF11/FF15, I2CXCON,

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCIEN	MASTER	RX_TX	ACK_GEN	INTEN	TR_START	TR_STOP	

Bit 7, SCI2EN,
This bit set to 1 will enable the corresponding I2C Port.

Bit 6, MASTER,
This bit set to 1 will set the corresponding I2C Port as master.

Bit 5, RX_TX,
This bit set to 1 will set to TX mode. Note that this bit means “transaction R/W”.

Bit 4, ACK_GEN,
This bit set 1 will generate ACK to external I2C device.

Bit 3, INTEN, set this bit to 1 will enable corresponding interrupt to CPU.

Bit 2, TR_START, set 1 will generate start condition for master.

Bit 1, TR_STOP, set 1 will generate stop condition for master.

I2C Master Example Program

The following program code can program and read back 24C02 (I2C flash) by MS89F/L SERIES I2C circuits, and send out read back data to UART P3.1. The program is as follows. Note that if interrupt is enabled and EA is set, MS89F/L SERIES will jump to interrupt vector at 004BH.

```
unsigned char i2c_send_byte_with_start(unsigned char dataout);
unsigned char i2c_send_byte_with_stop(unsigned char dataout);
unsigned char i2c_read_byte_send_ack(void);
unsigned char i2c_read_byte_no_ack(void);
unsigned char i2c_send_byte_wait_ack(unsigned char dataout);
void i2c_err(void);

main()
{
        unsigned char i;
        PCON=0x80;
        TH1=256-1;
        TH0=0;
        TMOD=0X22;
        SCON=0X50;
        TR1=1;
        ET0=1;
        //  ….
// TEST WRITE EXTERNAL 24C02
        // 24C02 ADDRESS 000, FIRST BYTE IS A0 OR A1
        I2C1DIV=0X10;
        i=i2c_send_byte_with_start(0xa0);
        if(i) i2c_err();
        // next byte is word address
        i=i2c_send_byte_wait_ack(0x00);
        if(i) i2c_err();
        // following is dataout
        for(i=0;i<6; i++)
                i2c_send_byte_wait_ack(255-i);
        i2c_send_byte_with_stop(0x5a);
        // now read back
        // re -send address first

        do{ // wait 24c12 busy, it may need 5 ms
                i=i2c_send_byte_with_start(0xa0);
                // time-out processing may be add here
        }while(i);
        i=i2c_send_byte_wait_ack(0x00); // address
        if(i) i2c_err();

        // now read, restart
        i=i2c_send_byte_with_start(0xa1);
        if(i) i2c_err();

        i=i2c_read_byte_send_ack();
        TI=0;
        SBUF=i;
        while(!TI);

        i=i2c_read_byte_send_ack();
        TI=0;
        SBUF=i;
        while(!TI);
        i=i2c_read_byte_send_ack();
        TI=0;
```

```
		SBUF=i;
		while(!TI);
		i=i2c_read_byte_send_ack();
		TI=0;
		SBUF=i;
		while(!TI);
		i=i2c_read_byte_send_ack();
		TI=0;
		SBUF=i;
		while(!TI);
		i=i2c_read_byte_send_ack();
		TI=0;
		SBUF=i;
		while(!TI);

		i=i2c_read_byte_no_ack(); // give stop here
		TI=0;
		SBUF=i;
		while(!TI);
		TI=0;
}
void i2c_err(void)
{
		while(1)
			P36^=1;
}

unsigned char i2c_send_byte_with_start(unsigned char dataout)
{

		I2C1DAT=dataout;
		if(dataout&0x01)// read
			I2C1CON=0XC4; // MEANS READ AT LATER ...
		else
			I2C1CON=0xf4;
		I2C1INT=0; //start transmit

		while(!(I2C1INT&0x08)); //byte transmit
		P00^=1;
		if(!(I2C1INT&0x02)) // wait ack
			return 1;
		P00^=1;
		return 0;

}
unsigned char i2c_send_byte_with_stop(unsigned char dataout)
{

		I2C1DAT=dataout;
		I2C1INT=0; //start transmit

		while(!(I2C1INT&0x08)); //byte transmit
		P00^=1;
		if(!(I2C1INT&0x02)) // wait ack
			return 1;
		P00^=1;
		I2C1CON=0xf2;
		I2C1INT=0; // OUT BY CLEAR INT
		while(!(I2C1INT&0x10)); //stop detect
		return 0;

}

unsigned char i2c_read_byte_send_ack(void)
```

```
{
        I2C1CON=0XD0; // ACK AND READ BACK
        I2C1INT=0;
        while(!(I2C1INT&0x08));
        return I2C1DAT;
}
unsigned char i2c_read_byte_no_ack(void)
{
        I2C1CON=0XC0; // ACK AND READ BACK, GIVE STOP
        I2C1INT=0;
        while(!(I2C1INT&0x08));
        I2C1CON=0XF2;
        I2C1INT=0; // SEND STOP
        while(!(I2C1INT&0x10)); //byte transmit
        return I2C1DAT;

}

unsigned char i2c_send_byte_wait_ack(unsigned char dataout)
{
        I2C1DAT=dataout;
        I2C1INT=0;                      // START TRANSMIT BY CLEAR BIT 3
        while(!(I2C1INT&0x08)); //byte transmit
        P00^=1;
        if(!(I2C1INT&0x02))
                return 1; // no ack
        P00^=1;
        return 0;
}
```

Some important rules for using I2C Master are as below:

1. Start condition with data send out together.
2. Stop condition may send after data was send and ACK received.
3. Data bytes and “stop” condition goes out with I2CXINT.4 set to 0.
4. Interrupt generated after ACK is checked or sent.
5. Start or stop condition received will not generate interrupt, unless arbitration lost.

I2C Slave Example Program

MS89F/L SERIES can act as an I2C slave part that can be accessed with external device. Device ID can be configured as well. However, data transmitting and receiving should follow I2C command rule. The example program is as below.

```
unsigned char i2c_state_machine(void);

main()
{
        unsigned char i;
        PCON=0x80;
        TH1=256-1;
        TH0=0;
        TMOD=0X22;
        SCON=0X50;
        TR1=1;
        ET0=1;

        DEVID=0X0a;// default value is 2A, change id to 0x0a
        I2C0INT=0;
        P36=0; // MEANS START
        while(1)
        {
                if(i2c_state_machine())
                {
                        i++;
                }
        }
        return 0;
}
#define I2C_WAIT_IDMATCH 0
#define I2C_WAIT_MASTER_WRITE    1
#define I2C_WAIT_MASTER_READ     2
unsigned char i2c_state_machine(void)
{
        static unsigned char state=0;

        // if stop , back to first state
        P0=state;
        if(!(I2C0INT&0x08))// wait interrupt
                        return 0;

        if(I2C0INT&0X30) // if stop or start found ==> state from 0
                state=0;
        P0=state;
        switch(state)
        {
                case I2C_WAIT_IDMATCH:
                        if(!(I2C0INT&0X04))// if not ADDR match?
                        {
                                // if not match, just clear the flag
                                I2C0INT=0;
                                return 0;
                        }
                        if(I2C0INT&0X01)// 1 MEANS READ, 0 MEANS WRITE
                        {
                                state=I2C_WAIT_MASTER_READ;
                                if(fifoo!=fifoi)
                                {
                                        I2C0DAT=fifo[fifoo++];
```

```
                                        if(fifoo==100)
                                                fifoo=0;
                                }else
                                        I2C0DAT=0;
                        }
                        else
                        {
                                state=I2C_WAIT_MASTER_WRITE;
                        }
                        I2C0INT=0;
                        break;
                case I2C_WAIT_MASTER_WRITE:
                         fifo[fifoi++]=I2C0DAT;
                         I2C0INT=0;
                         return 1;
                case I2C_WAIT_MASTER_READ:
                        if(fifoo!=fifoi)
                        {
                                I2C0DAT=fifo[fifoo++];
                                if(fifoo==100)
                                        fifoo=0;
                        }else
                                I2C0DAT=0;
                        I2C0INT=0; // clear interrupt
                        return 1;
        }
        return 0;
}
```

The related rules are as follows:

1. After start condition, the ID byte will not be read by CPU. CPU can check if it matches after interrupt.
2. Start and stop condition will not generate interrupt. Interrupt happens only after ACK is checked or sent.
3. I2CxINT bit 3 set to 0 (clear interrupt) make the external master can continue next byte.
4. For read operation, if master must send ACK if it need to read following bytes. If the master no sending ACK, interrupt will generate until a new byte with “start” condition is sent.
5. If MS89F/L SERIES is busy, I2CXCON bit 4 may clear to 0, and ACK will not be sent.
6. If interrupt is not cleared, SCL will hold low.
7. Interrupt will happen even the address is 0x5a or 0x55. (Special addresses) The bit ASLV will not be set.

I2C Timing Diagram

This section shows some detailed timing of I2C slave of the register bits for reference.

Extern Master Writes MS89XX

If external master want to write MS89XX, CPU will get the flag as below.

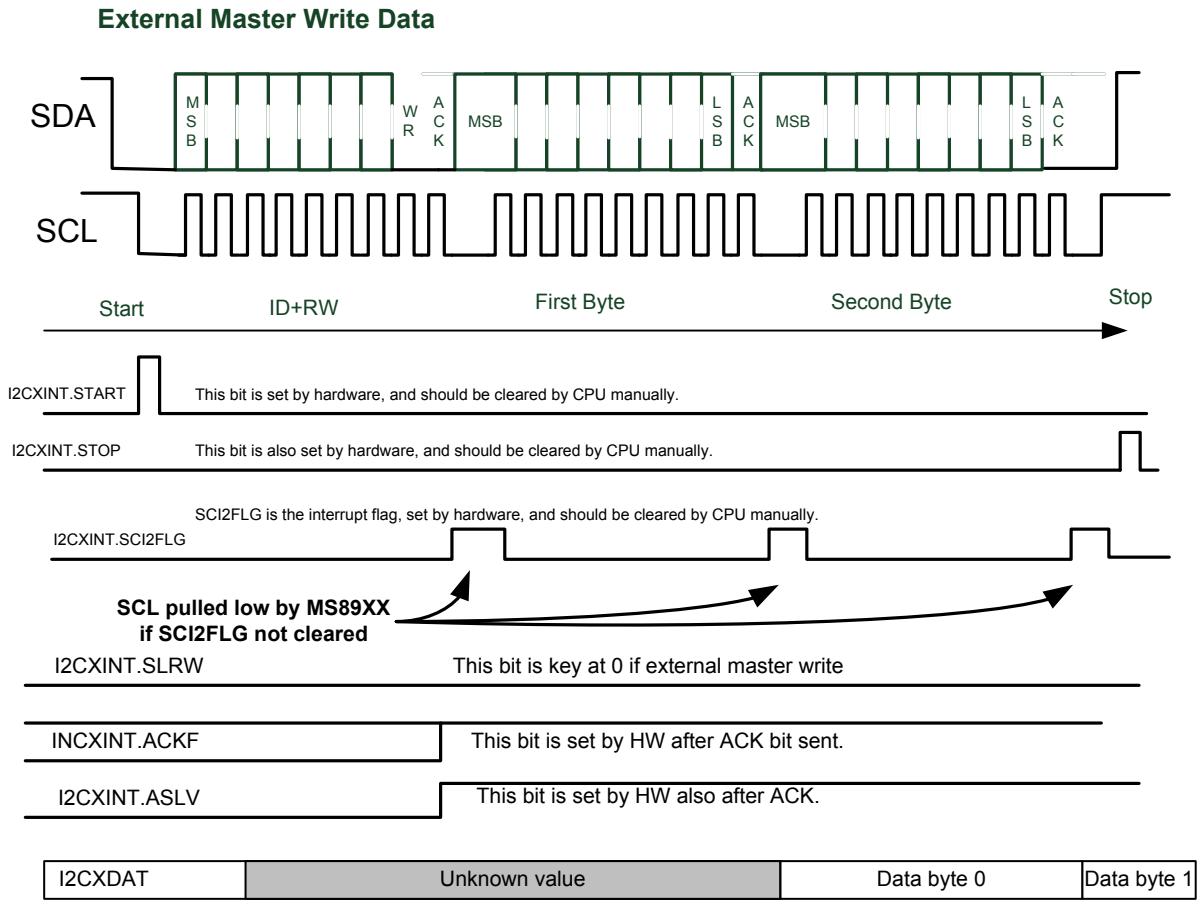


Figure 23. External Master Writes MS89XX.

From above it is shown that CPU shall clear I2CXINT.STOP/START/SCI2FLG bits. However, other bits like ASLV, ACKF, SLRW are read only bits, CPU write data will not change it.

Also, SCL is pulled low by MS89XX when SCI2FLG is 1. CPU should clear it ASAP.

External Master Read MS89XX

When external master want to read MS89XX, it shall follow I2C transaction rule and the timing is as follows.

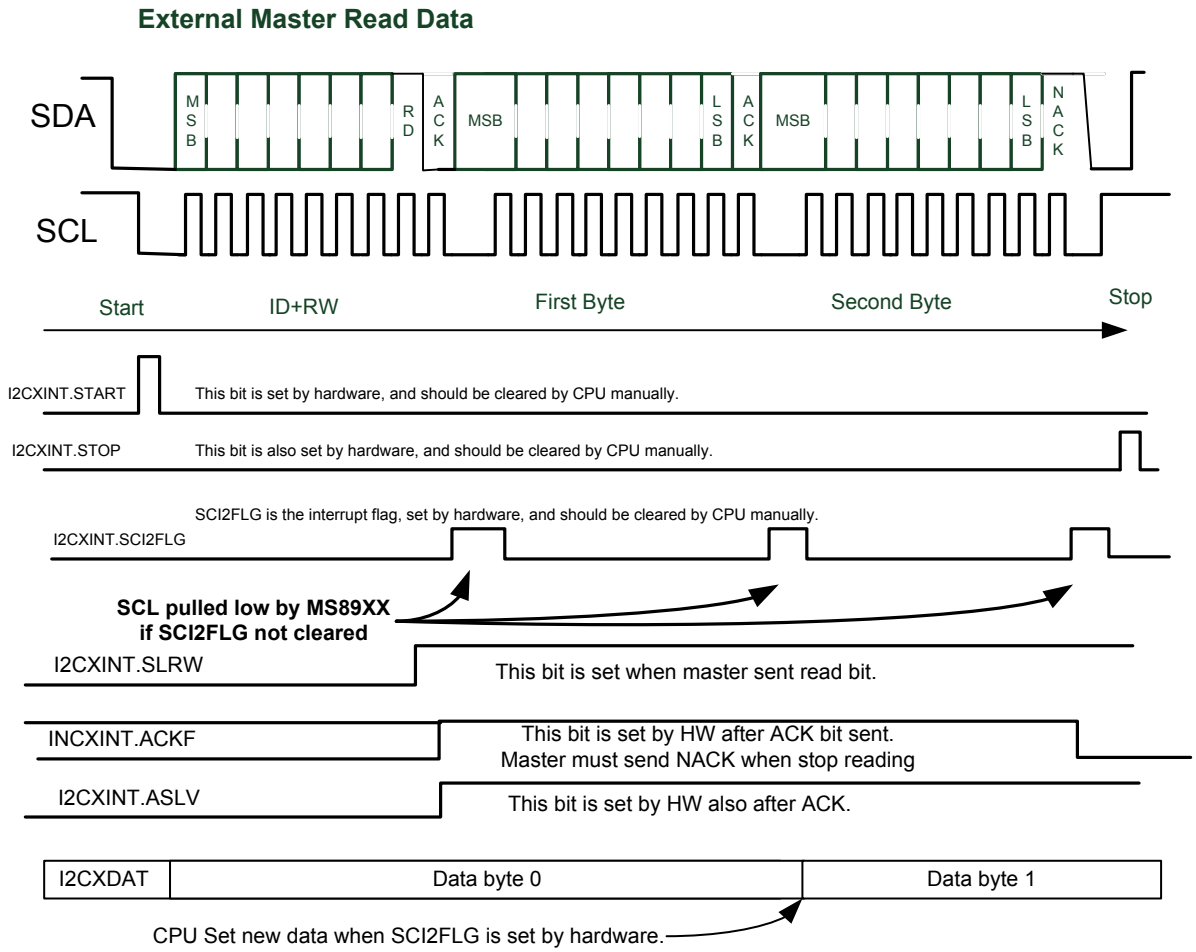


Figure 24. External Master Reads MS89XX.

From above figure, it is shown that CPU is better to write the new I2CDAT at the interrupt routine (or when SCI2FLG is 1 by polling method), and clear SCI2 FLG as soon as possible.

External Master Read/Write Other Device

When external master is reading/writing other device, which is connect to MS89XX in parallel, MS89XX have also interrupt flags. CPU must check ASLV bit when interrupt comes (SCI2FLG is read 1).

Other flags like START/STOP/ASLV/ACKF/SLRW will also respond correctly, but I2CXDAT will have no effect, and ASLV is low if address is not equal to DEV_ID.

In Application Programming (IAP)

MS89F/L SERIES may be programmed by itself. CPU may program the flash memory by itself, with CPU halt or not halt[9] when erasing/programming.

Note that the CPU has the possibilities to write/erase wrong data to FLASH accidentally as below:

1. VDD is too low and CPU runs to WRONG program code, which cause the data to be erased or modified.
2. While erasing/programming, the VDD dropped and erase/write to wrong address.

That is, if a program code need to erase/program the FLASH, corresponding LVR (Low voltage reset) MUST be enabled as long as possible, which makes sure CPU will not erase/program the flash accidentally.

Please see LVR chapter for detailed settings.

The corresponding resisters of FLASH operations are as below:

Address	Notation	Function	Default
FF00	XMEMCON	If bit 0 set to 1, x-memory 10~ff will be mapped to program ROM 10~FF.	00H
FF20	EFLASHCON	Flash control register	0
FF21	PRGAL	Programming address, low byte.	0
FF22	PRGAH	Programming address, high byte, and speed bits.	0
FF23	PRGDAT	Programming data.	XX
FF24	FLASHPROT	Only this byte set to 0xa5, write and erase can proceed. This byte is to prevent FLASH be erase or programmed accidentally.	00

REG.FF20H, EFLASHCON

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPURST	AUTOINC	BUSY	RD	SE	MASE	IFREN1	START

EFLASHCON BIT[7], CPURST:
Set this bit to 1 will keep CPU in reset state.

EFLASHCON BIT[6], AUTOINC
Set thjs bit 1 the read/write procedure will increase the (programming/reading pointer) address 1 automatically.

EFLASHCON BIT[5], BUSY
Read this bit 1 means the flash operation is busy.

EFLASHCON BIT[4], RD
Set this bit 1 will read back the flash data to PRGDAT. CPU can check

[9] Only running PROGRAM in SRAM can make CPU continue running.

PRGDAT for the new data.

EFLASHCON BIT[3], SE
Set this bit 1 will erase the corresponding 2K sector of flash if IFREN1 is 0. Note that if IFREN1 is set, this operation will "ONLY" erase the information sector without erasing the program sector.

EFLASHCON BIT[2], MASE
Set this bit 1 will ERASE whole flash block. Note that if IFREN1 is set, this operation will "ALSO" erase the information sector.

EFLASHCON BIT[1], IFREN1
Set this bit 1 the Flash operation will be active to the information sector, 128 bytes.

EFLASHCON BIT[0], START
Set this bit 1 will start the required flash operation.

REG.FF22H, PRGAH

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IFREN2	**OSC1**	**OSC0**	PRGA12	PRGA11	PRGA10	PRGA9	PRGA8

BIT <4:0>, PRGA<12:8>: The address high bytes of IAP programming.
BIT <6:5>: FLASH erase/programming timing setting for different oscillators.
00: Oscillator is <= 10 MHz.
01: Oscillator is <= 20 MHz
10: Oscillator is <= 40 MHz
BIT <7>: IFREN2. Set this bit 1 to read information sector 2. Info-sector 2 is used for ADC trimming data storage, and can only be read by CPU.

Programmed by CPU Itself

MS89F/L SERIES may be programmed by CPU itself. Only EFLASHPROT set to 0xa5, write and erase can proceed. If CPU is running the program of FLASH area, CPU will halt for the program/erase duration. If CPU is running program of RAM (XMEMCON.0==1), it will continue running when flash is programming/erasing. Only CPU is using the XMEM addressing mode to registers 0xff20~0xff24.

Sector Erasing/Programming

MS89F/L SERIES has 4x 2KB Sectors. CPU may run the program at a special sector or program at RAM to erase other sectors. The specified sector will be erased after CPU sets SE to 1 with correct A[12,11] at PRGAH or IFREN1 set to 1.

To erase "only" information sector, CPU may write SE bit and IFREN bit at the same time. Note that the first 16 bytes must be written back after the info-sector is erased before next reset.

For example, if CPU is running program at 0x0000~0x07FF, it may erase the other 3 sectors of 0x800~0x1FFF. During the ~100ms flash-erasing period, CPU will be halt.

If CPU is running the program in SRAM (XMENCON=1), CPU may erase all the sectors. During the ~100ms flash erasing period, the BUSY bit of EFLASHCON will be 1, and CPU may still run the program and interrupt acceptable. Only that the program on SRAM can be 256 bytes maximum.

For flash programming, CPU will just halt for around 100 us, and busy bit is also valid that NO HALT if CPU is running the program on SRAM (XMEMCON.0 set to 1).

Flash Data Protection

MS89XX Flash Data protection has 2 meanings. The first one is to protect from ESD, power source stability issue, or other jamming signals. MS89XX will not erase/write the flash data if 0xff24 is not set to 0xa5. That is, the software should clear 0xff24 to 0 after it has erase/program the Flash, this measure is to prevent the flash be erased/modified accidentally.

The other meaning of protection is to prevent the program be read back by external [devices] for copy-right issue. As described, if information sector byte 2 is set to 0x55, read back & byte program from external I2C master will be disabled. However, external I2C master can still "erase" the flash sectors with corresponding I2C commands. The only difference is that when erasing the Information Sector by external I2C master, the main program will also be erased together at the same time.

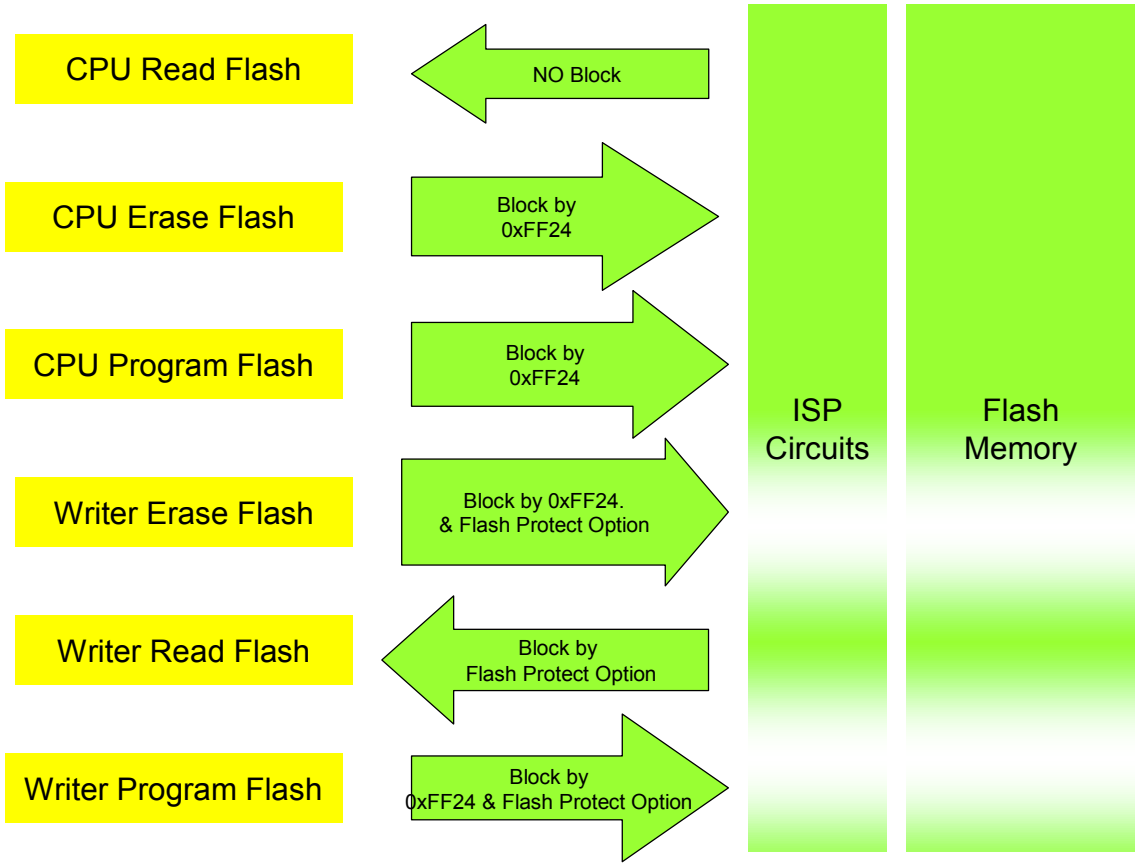


Figure 25. Flash Protection Scheme.

In System Debugger

MS89F/L08[10] supports "In System Debugger" function that user may debug the program easily and fast. When ICE bit is programmed by external ISP, MS89FXX will use the upper most 2K bytes (0x1800~0x1FFF) as the "monitor" code, and user may debug the program with the following configuration.

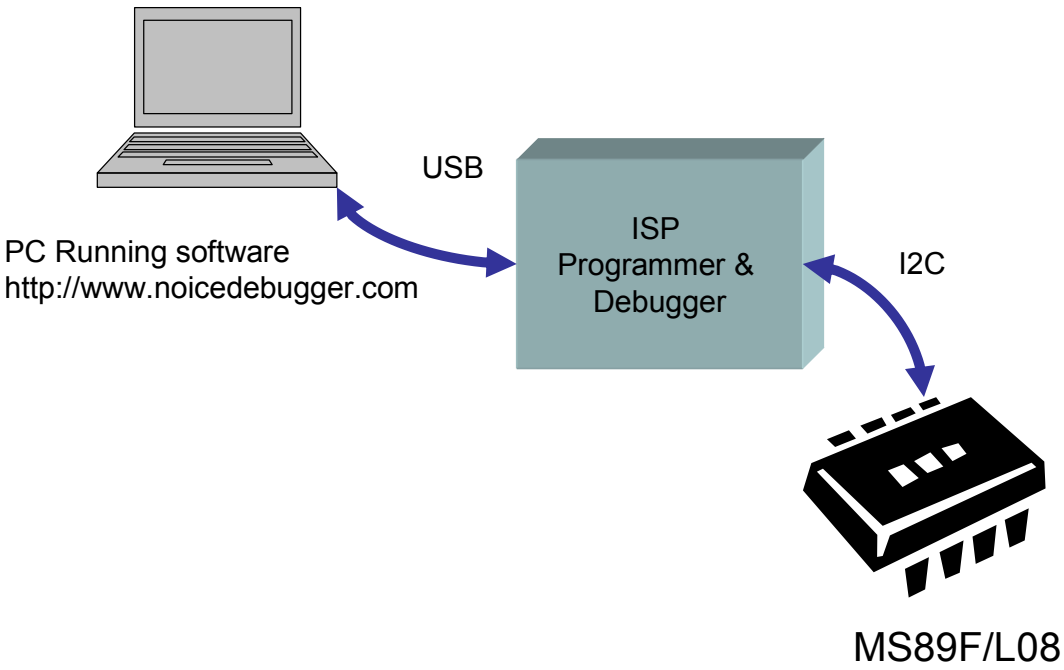


Figure 26. MS89FXX In System Debugger Configuration.

The debugger has the following constrains:

1. I2C1 (with pins) will be used by ICE, main program should not use it.
2. I2C0 Slave address must be fixed to 0x2A.
3. Monitor code will use 0x1800~0x1FFF, user can use only 6K bytes (0x0000~0x17FF) of ROM.
4. Monitor code will use 8 bytes of stack space (operating with SP), and X-Memory 0x00E0~0x00FF.

The debugger may set breakpoints and checking/modifying the content of different memory spaces. If user want to use advance ICE functions like "Stop with specified event", or "check back all the CPU cycles when stop", please contact MSHINE Technologies Corp.

[10] F/L02,04 does not support this function.

Also, because setting breakpoints need to modify the contents of Flash ROM. External I2C master need to “read back the sector”, and “erase the sector”, then “program the sector with the new value” even only 1 breakpoint to be insert. And each sector is 2K bytes. That is, setting/clearing the breakpoints may need up to 1 second for Debugger operation.

If the user wants to share the I2C1 interface with SPI master or slave operations, please contact MSHINE Technologies Corp.

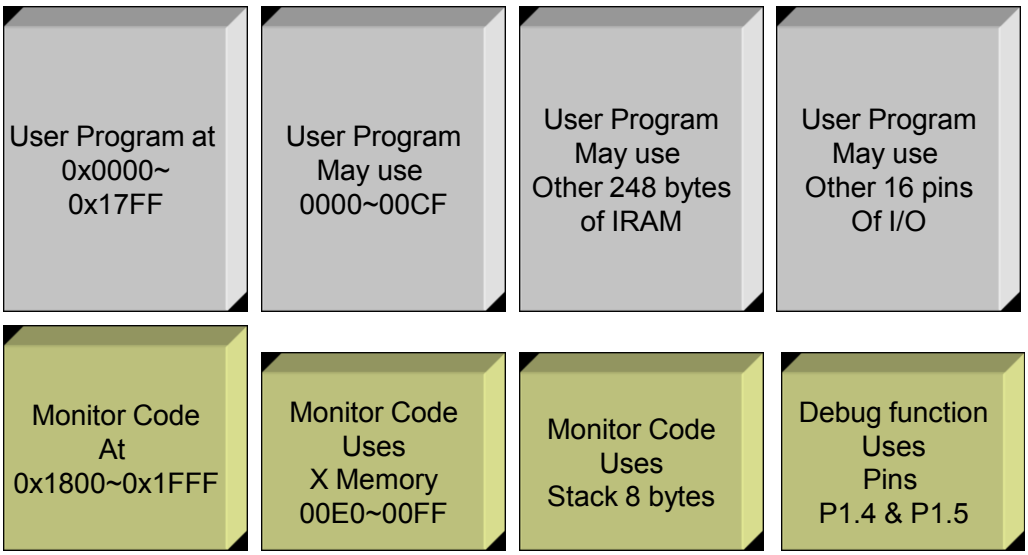


Figure 27. In System Debugger used recources.

Note that the In System Debugger may be used with all the clock sources, IRC/ERC and external oscillator.

Flash Info Sector

MS89F/L SERIES has a 128 byte information sector that can be used for special data. However, the first 16 bytes are used for special purpose, users should restore after programming. The 4 first bytes are for hardware purpose, described below. The other 12 bytes are reserved for writer use. User may use the 112 bytes as will, only the 16 byte must write back after Info Sector is erased.

Byte 0:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPUCKDIV[1:0]		WDTCK[2:0]			LVRENB	OSCSEL[1:0]	

Byte 1:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved					LVSENB	FASTCK	TEST_ENB

Byte 2: Read protect byte.

When this byte is 0xAA, I2C read and program will be forbidden, and I2C info-sector erase will also erase the main sectors.

Byte 3:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			URX	WDTENB	LVSLEV[2:0]		

Other bytes of the flash sector can be freely used. If the sector is erased, the first 4 bytes must be restored before next reset/power-on. The bits are described below:

OSCSEL[1,0]: 11 (default): IRC is used,
10: ERC oscillator used.
0X: External oscillator used.

LVSENB: Low voltage stop enable if set to 0.
LVRENB: Low voltage reset enable if set to 0.

WDTCK[2,0]: Watchdog clock division of 2^7+WDTCK

CPUCKDIV[1,0]: Default CPU speed.

TEST_ENB: Set 0 to enable test mode clock.

FASTCK: Set 0 for oscillator frequency < 8 MHz, that CPU may use Oscillator clock directly.

LVSLEV[2:0]: LVS Level Selection.

WDTENB: Set 0 to enable Watch-dog timer.

URX: Set 0 will pass first bits of RX data to enter the serial port even if RI=1.[11]

Byte 3~ Byte 15: WRITER protects the IC .Set 0xFF will disable protection. Users should restore after programming. If you do not want to use protection, please contact MSHINE

[11] For standard 80C31, start bit will be skipped if RI is not cleared to 0. If this bit is 0, MS89F/L series will enable UART data receiving even if RI is 1. That is, this bit set to 0 will make MS89F/L series easier to receive UART data, but will be not compatible standard 80C31.

AC & DC Electrical Characteristics

DC Characteristics – I/O, CPU

MS89F09						
Parameters	Conditions (5V)	Symbol	Min.	Typ.	Max.	Unit
Supply voltage	MCU operating voltage	V_{DD}	4.5	5.0	5.5	V
	ROM operating voltage	V_{ROMH}	4.5	5.0	5.5	V
	RAM operating voltage	V_{RAM}	4.5	5.0	5.5	V
Supply Current	Internal RC 8 MHZ/1			9.5		mA
	Internal RC 8 MHZ/2			6.5		mA
	Internal RC 8 MHZ/4			4.5		mA
	Internal RC 8 MHZ/8			3.8		mA
	ERC 1 MHZ/1			2.5		mA
	Idle Mode[12], Internal RC only, 5V			2		mA
	WDT Power Down at 5V			65		uA
	Power-Down Mode, 5V			18	30	uA
Output voltage	I_{OH}=1, Push-pull pins.	V_{OH1}	VDD	–	–	V
	I_{OL}=2 mA, push-pull pins	V_{OL1}	0.2	–	–	V
	I_{OL}=2.2 mA, open-drain pins	V_{OL2}	0.2	–	–	V
Input voltage	All IO Pins	V_{IH1}	0.8	–	V_{DD}+0.	V
	All IO Pins	V_{IL1}	-0.3	–	0.2	V
Output current	Default IO, low state, pull up	I_{OH1}	1			uA
	Default IO, 0.7VDD, pull up	I_{OH2}	20			uA
	CMOS Output, 4.0V	I_{OH3}	8			mA
	Sink current VOL=1V	I_{OL}			-8	mA
MS89L09						
Parameters	Conditions (3.3V)	Symbol	Min.	Typ.	Max.	Unit
Supply voltage	MCU operating voltage	V_{DD}	2.4	3.3	3.6	V
	ROM operating voltage	V_{ROMH}	2.4	3.3	3.6	V
	RAM operating voltage	V_{RAM}	1.8	3.3	3.6	V
Supply Current	Internal RC @ 8MHz/1,			7		mA
	Internal RC @ 8MHz/8			2.7		mA
	ERC @ 1MHZ/1			1.5		mA
	Idle Mode IRC/8			1.5		mA
	WDT only at PD mode			26		uA
	Power-Down Mode, 3.3V			5	10	uA
Output voltage	I_{OH}=1, Push-pull pins.	V_{OH1}	VDD	–	–	V
	I_{OL}=2 mA, push-pull pins	V_{OL1}	0.2	–	–	V
	I_{OL}=2.2 mA, open-drain pins	V_{OL2}	0.2	–	–	V
Input voltage	All IO Pins	V_{IH1}	0.8	–	V_{DD}+0.	V
	All IO Pins	V_{IL1}	-0.3	–	0.2	V
Output current	Default IO, low state, pull up	I_{OH1}	1			uA
	Default IO, 0.7VDD, pull up	I_{OH2}	20			uA

[12] PCON=0xB1.

	CMOS Output, 2.5V	I_{OH3}	8			mA
	Sink current VOL=1V	I_{OL}			-8	mA

Table 5. DC Characteristics – I/O, CPU

AC Electrical Characteristics – Oscillators

MS89F09					
Parameter	Symbol	Min.	Typ.	Max.	Unit
Operation Voltage	V_{DD}	3.8		5.5	Volt.
IHRC Frequency	F_{RC}		8		MHz
IHRC Frequency Shift[13]		-2.1		+2.1	%
ILRC Frequency			16		KHz
ILRC Frequency Shift		-50		+50	%
RC Power Consumption	I_{RC}		1.5		mA
OSC frequency	F_{OSC}			20	MHz
MS89L09					
Parameter	Symbol	Min.	Typ.	Max.	Unit
Operation Voltage	V_{DD}	2.4		3.6	Volt.
IHRC Frequency	F_{RC}		8		MHz
IHRC Frequency Shift		-2.1		+2.1	%
ILRC Frequency			32		KHz
ILRC Frequency Shift		-50		+50	%
RC Power Consumption	I_{RC}		1.5		mA
OSC frequency	F_{OSC}			20	MHz

Table 6. AC Electrical Characteristics – Oscillators

[13] Trimmed with specified voltage 4.75~5.25V or 3.0~3.6V, not for wide range operation.

AC Electrical Characteristics – ADC

Parameter	Symbol	Min.	Typ.	Max.	Unit
Conversion Rate	t_{CONV}		8.0		us
INL	INL		+/−2		LSB
DNL	DNL		+/−1		LSB
Offset Error			4		LSB
Gain Error			4		LSB
SNDR			60		DB
T/H Acquisition Time	t_{ACQ}			400	ns
Channel Switching time	t_{SW}			400	ns
Aperture Jitter				200	ps
Operation Frequency				12	MHz
Duty Cycle		30		70	%
Channel Isolation	I_{SOCH}	60			dB
VDD isolation	I_{SOVDD}	40			dB
Input Capacitance	C_{IN}		10		pF
Capacitive Bypass at REF	C_{BYPASS}	1.0		4.7	uF

Table 7. AC Characteristics – ADC

DC Characteristics – ADC

MS89F09					
Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DD}	3.6		5.5	V
Input voltage Range[14]	V_{IR}	0		VREF	V
Vref Range	V_{REF}			AVDD	V
Multiplexer Leakage Current				±1	uA
Operation Current	I_{ADC}		2.5		mA
Shutdown Current			2	10	uA
Operating Temperature Ranges	T_{OP}	−40		+85	°C
MS89L09					
Parameter	Symbol	Min.	Typ.	Max.	Unit
Supply Voltage	V_{DD}	2.4		3.6	V
Input voltage Range	V_{IR}	0		VREF	V
Vref Range	V_{REF}			AVDD	V
Multiplexer Leakage Current				±1	uA
Operation Current	I_{ADC}		2.5		mA
Shutdown Current			2	10	uA
Operating Temperature Ranges	T_{OP}	−40		+85	°C

Table 8. ADC DC characteristics

[14] Its output code is unipolar.

Product procurement

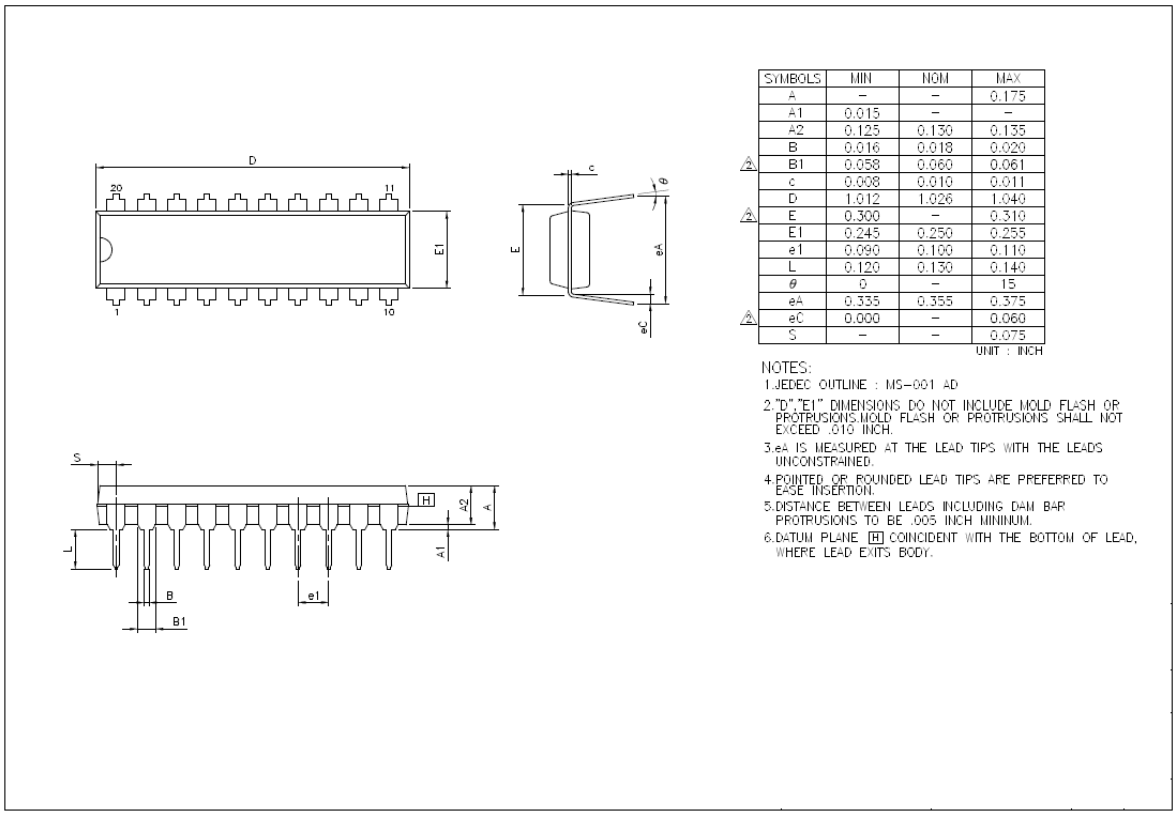
PKG TYPE	Operating	Flash
Part NO.	Voltage	ROM
SOP 20 Pin		
MS89F09S20	5.5V ~ 3.8V	8K
DIP 20 Pin		
MS89F09D20	5.5V ~ 3.8V	8K

PKG TYPE	Operating	Flash
Part NO.	Voltage	ROM
SOP 20 Pin		
MS89L09S20	3.8V ~ 2.4V	8K
DIP 20 Pin		
MS89L09D20	3.6V ~ 2.4V	8K

Table 9. Product procurement

Package Size

DIP20

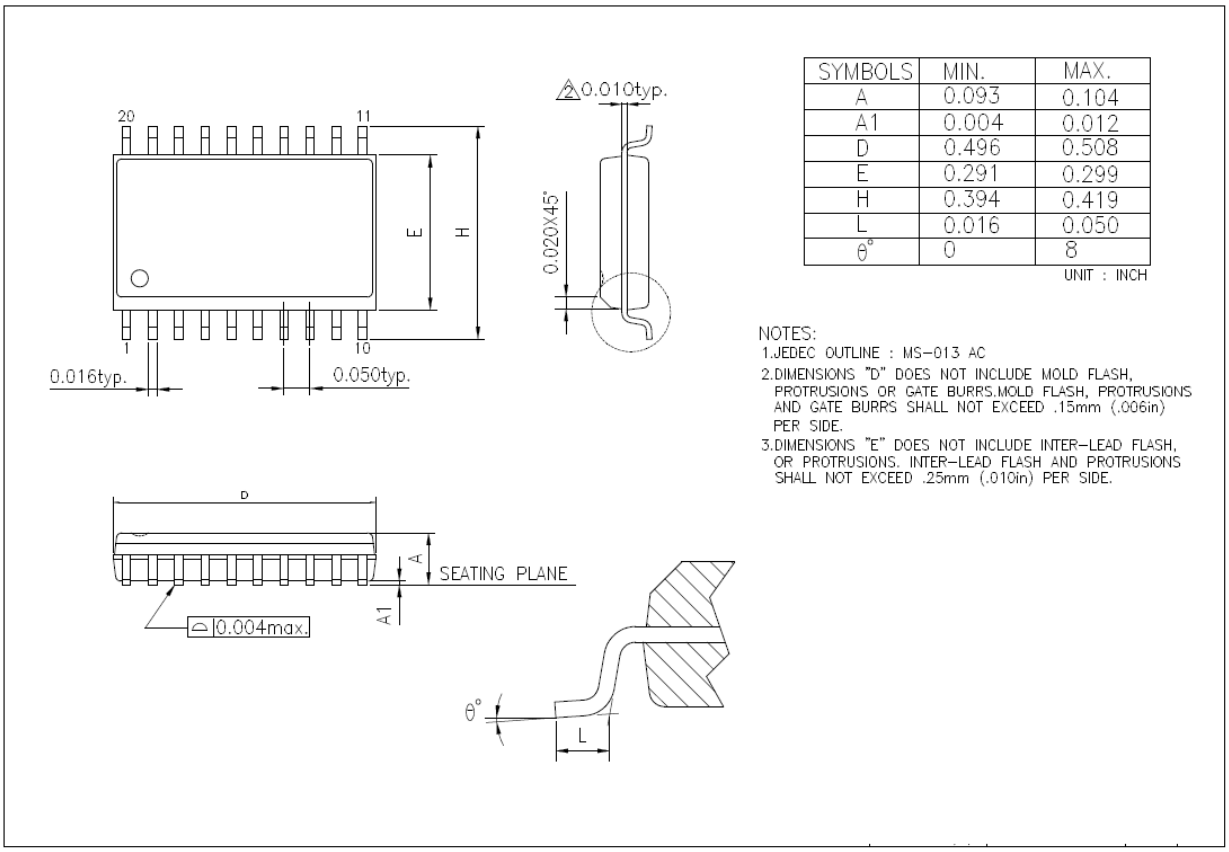


SYMBOLS	MIN	NOM	MAX
A	–	–	0.175
A1	0.015	–	–
A2	0.125	0.130	0.135
B	0.016	0.018	0.020
B1	0.058	0.060	0.061
c	0.008	0.010	0.011
D	1.012	1.026	1.040
E	0.300	–	0.310
E1	0.245	0.250	0.255
e1	0.090	0.100	0.110
L	0.120	0.130	0.140
θ	0	–	15
eA	0.335	0.355	0.375
eC	0.000	–	0.060
S	–	–	0.075

UNIT : INCH

NOTES:
1. JEDEC OUTLINE : MS-001 AD
2. "D","E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS.MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 INCH.
3. eA IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.
4. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.
5. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MININUM.
6. DATUM PLANE [H] COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY.

SOP20



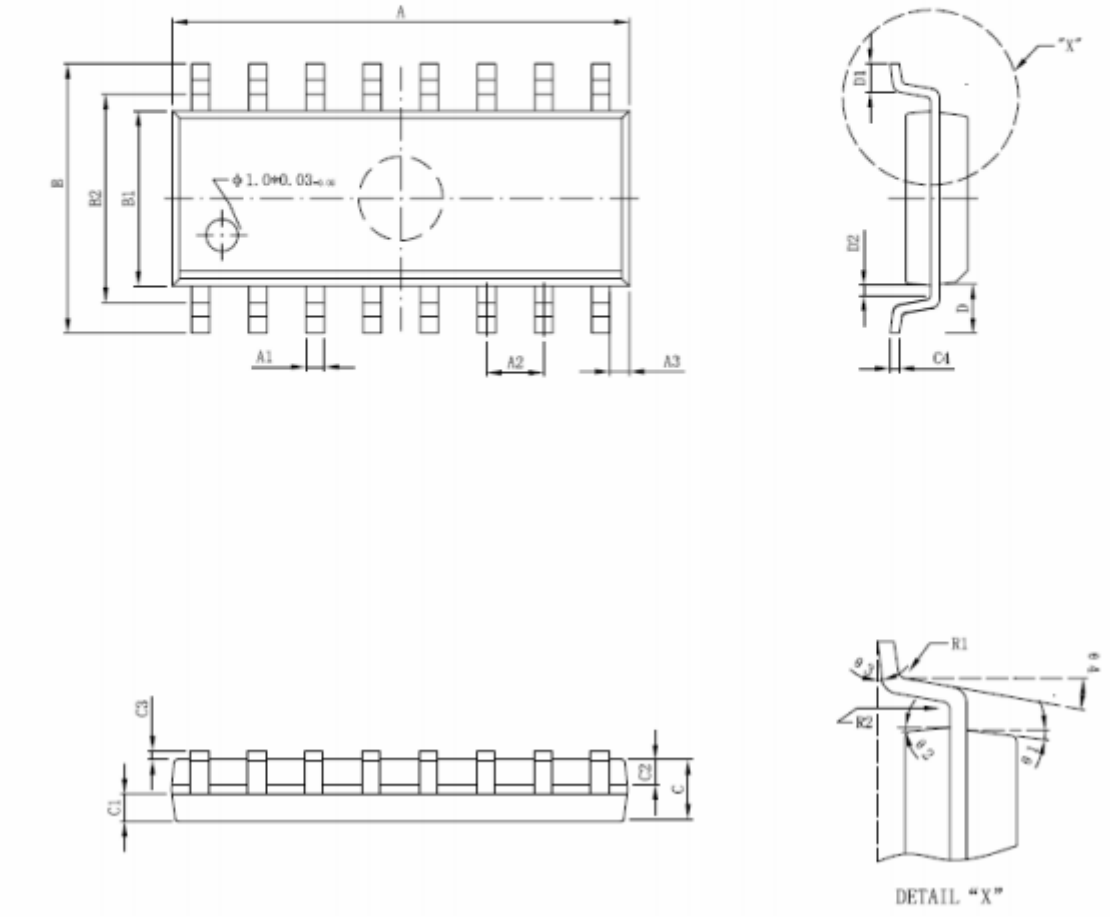
SYMBOLS	MIN.	MAX.
A	0.093	0.104
A1	0.004	0.012
D	0.496	0.508
E	0.291	0.299
H	0.394	0.419
L	0.016	0.050
θ°	0	8

UNIT : INCH

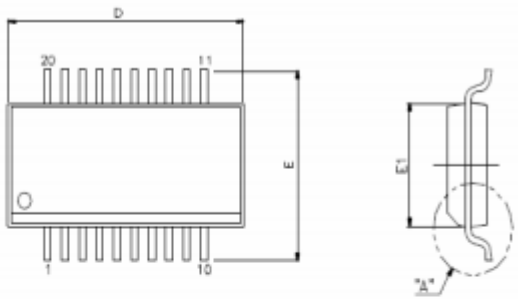
NOTES:
1.JEDEC OUTLINE : MS-013 AC
2.DIMENSIONS "D" DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS.MOLD FLASH, PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED .15mm (.006in) PER SIDE.
3.DIMENSIONS "E" DOES NOT INCLUDE INTER-LEAD FLASH, OR PROTRUSIONS. INTER-LEAD FLASH AND PROTRUSIONS SHALL NOT EXCEED .25mm (.010in) PER SIDE.

SOP16

尺寸 / 标注	最 小（mm）	最 大（mm）	尺寸 / 标注	最 小（mm）	最 大（mm）
A	9.9	10.10	C4	0.2TYP	
A1	0.356	0.456	D	1.05TYP	
A2	1.27TYP		D1	0.40	0.70
A3	0.35TYP		D2	0.22	0.42
B	5.84	6.24	R1	0.15TYP	
B1	3.84	4.04	R2	0.15TYP	
B2	5.0TYP		θ1	8° TYP	
C	1.35	1.55	θ2	8° TYP	
C1	0.61	0.71	θ3	4° TYP	
C2	0.54	0.64	θ4	15° TYP	
C3	0.10	0.25			

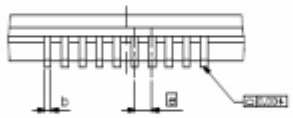


SSOP



SYMBOLS	MIN.	NOM.	MAX.
A	0.053	0.064	0.069
A1	0.004	0.006	0.010
A2	–	–	0.059
b	0.008	–	0.012
C	0.007	–	0.010
D	0.337	0.341	0.344
E	0.228	0.236	0.244
E1	0.150	0.154	0.157
e	0.025 BASIC		
L	0.016	0.025	0.050
L1	0.041 BASIC		
θ°	0°	–	8°

UNIT : INCH

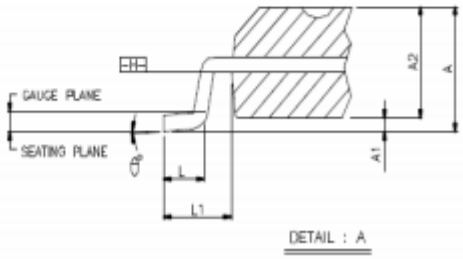


NOTES:

1.JEDEC OUTLINE : MO-137 AD

2.DIMENSION D DOES NOT INCLUDE MOLD PROTRUSIONS OR GATE BURRS. MOLD PROTRUSIONS AND GATE BURRS SHALL NOT EXCEED 0.006" PER SIDE. DIMENSION E1 DOES NOT INCLUDE INTERLEAD MOLD PROTRUSIONS. INTERLEAD MOLD PROTRUSIONS SHALL NOT EXCEED 0.010" PER SIDE.

3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION/INTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.004" TOTAL IN EXCESS OF b DIMENSION AT MAXIMUM MATERIAL CONDITION. DAMBAR INTRUSION SHALL NOT REDUCE DIMENSION b BY MORE THAN 0.002" AT LEAST



DETAIL : A