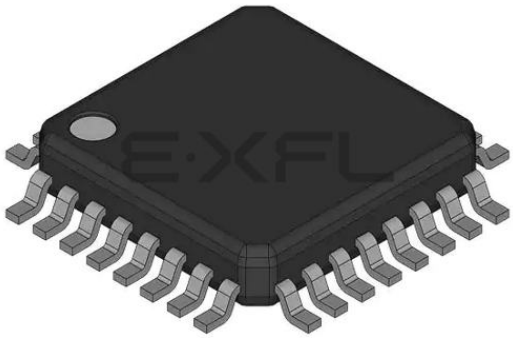


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

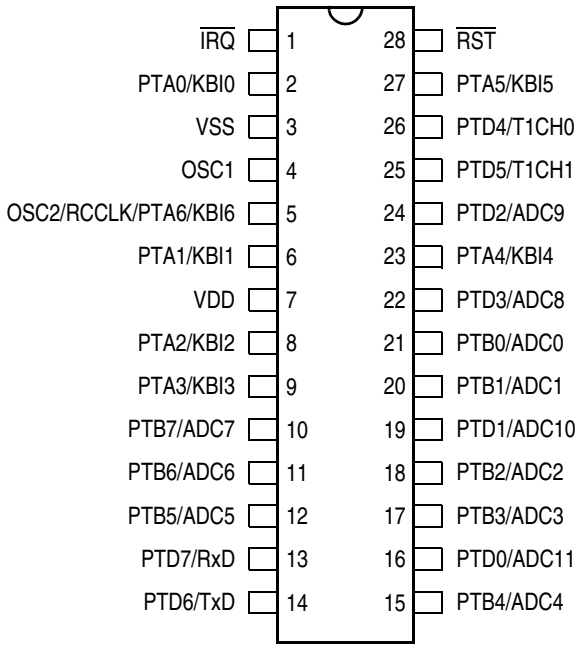
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	SCI
Peripherals	LED, LVD, POR, PWM
Number of I/O	26
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 13x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908jl8cfaer2

Chapter 6
Oscillator (OSC)

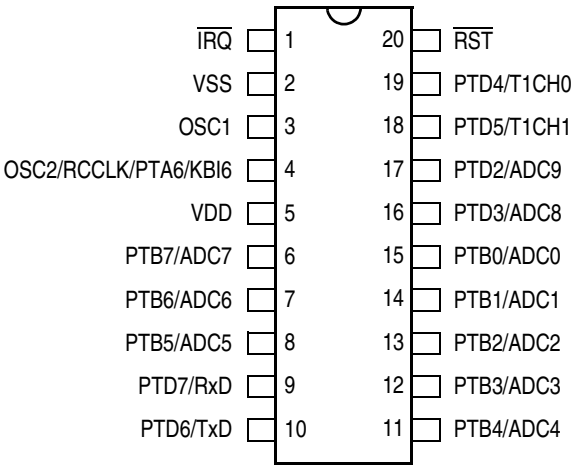
Chapter 7
Monitor ROM (MON)



Pins not available on 28-pin packages	
	PTE0/T2CH0
	PTE1/T2CH1
	ADC12/T2CLK
PTA7/KBI7	

Internal pads are unconnected.
Set these unused port I/Os to output low.

Figure 1-4. 28-Pin PDIP/SOIC Pin Assignment



The 20-pin MC68HC908JL8 is designated MC68HC908JK8.

Pins not available on 20-pin packages	
PTA0/KBI0	PTD0/ADC11
PTA1/KBI1	PTD1/ADC10
PTA2/KBI2	
PTA3/KBI3	PTE0/T2CH0
PTA4/KBI4	PTE1/T2CH1
PTA5/KBI5	
	ADC12/T2CLK
PTA7/KBI7	

Internal pads are unconnected.
Set these unused port I/Os to output low.

Figure 1-5. 20-Pin PDIP/SOIC Pin Assignment

1.5 Pin Functions

Description of the pin functions are provided in Table 1-2.

Chapter 3
Configuration and Mask Option Registers (CONFIG & MOR)

3.1 Introduction

This section describes the configuration registers, CONFIG1 and CONFIG2; and the mask option register (MOR).

The configuration registers enable or disable these options:

- Computer operating properly module (COP)
- COP timeout period ($2^{13}–2^4$ or $2^{18}–2^4$ ICLK cycles)
- Internal oscillator during stop mode
- Low voltage inhibit (LVI) module
- LVI module voltage trip point selection
- STOP instruction
- Stop mode recovery time (32 or 4096 ICLK cycles)
- Pull-up on $\overline{\text{IRQ}}$ pin

The mask option register selects the oscillator option:

- Crystal or RC

3.2 Functional Description

The configuration registers are used in the initialization of various options. The configuration registers can be written once after each reset. All of the configuration register bits are cleared during reset. Since the various options affect the operation of the MCU, it is recommended that these registers be written immediately after reset. The configuration registers are located at $001E and $001F. The configuration registers may be read at anytime.

NOTE

The options except LVIT[1:0] are one-time writable by the user after each reset. The LVIT[1:0] bits are one-time writable by the user only after each POR (power-on reset). The CONFIG registers are not in the FLASH memory but are special registers containing one-time writable latches after each reset. Upon a reset, the CONFIG registers default to predetermined settings as shown in Figure 3-1 and Figure 3-2.

The mask option register (MOR) is used to select the oscillator option for the MCU: crystal oscillator or RC oscillator. The MOR is implemented as a byte in FLASH memory. Hence, writing to the MOR requires programming the byte.

4.3 CPU Registers

Figure 4-1 shows the five CPU registers. CPU registers are not part of the memory map.

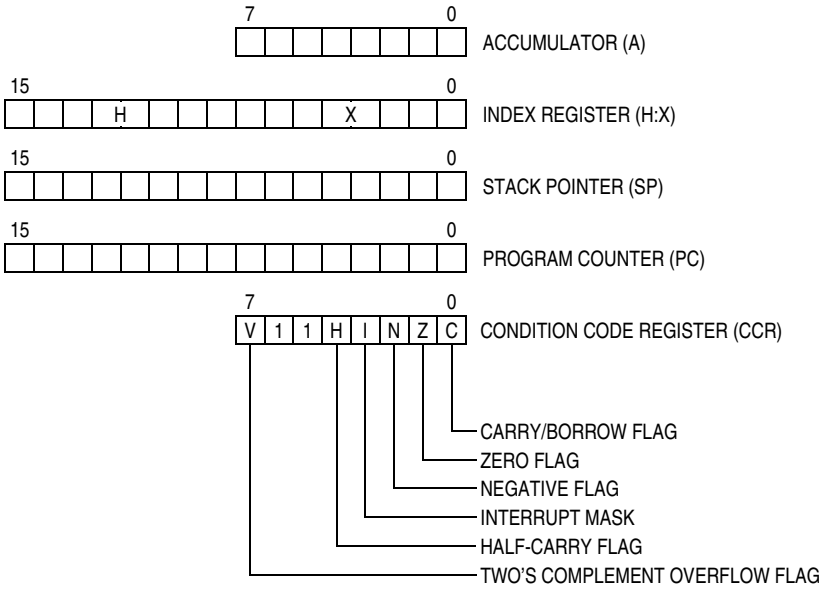


Figure 4-1. CPU Registers

4.3.1 Accumulator

The accumulator is a general-purpose 8-bit register. The CPU uses the accumulator to hold operands and the results of arithmetic/logic operations.

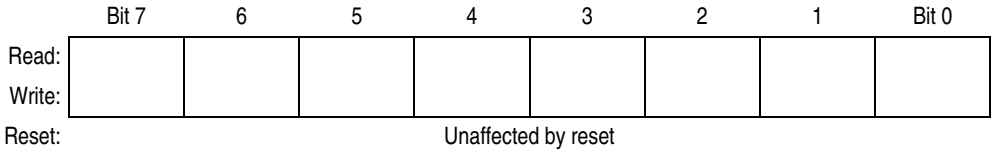


Figure 4-2. Accumulator (A)

4.3.2 Index Register

The 16-bit index register allows indexed addressing of a 64-Kbyte memory space. H is the upper byte of the index register, and X is the lower byte. H:X is the concatenated 16-bit index register.

In the indexed addressing modes, the CPU uses the contents of the index register to determine the conditional address of the operand.

The index register can serve also as a temporary data storage location.

Table 4-1. Instruction Set Summary

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
DBNZ *opr,rel* DBNZA *rel* DBNZX *rel* DBNZ *opr*,X,*rel* DBNZ X,*rel* DBNZ *opr*,SP,*rel*	Decrement and Branch if Not Zero	A ← (A)–1 or M ← (M)–1 or X ← (X)–1 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 3 + *rel* ? (result) ≠ 0 PC ← (PC) + 2 + *rel* ? (result) ≠ 0 PC ← (PC) + 4 + *rel* ? (result) ≠ 0	–	–	–	–	–	–	DIR INH INH IX1 IX SP1	3B 4B 5B 6B 7B 9E6B	dd rr rr rr ff rr rr ff rr	5 3 3 5 4 6
DEC *opr* DECA DECX DEC *opr*,X DEC ,X DEC *opr*,SP	Decrement	M ← (M) – 1 A ← (A) – 1 X ← (X) – 1 M ← (M) – 1 M ← (M) – 1 M ← (M) – 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3A 4A 5A 6A 7A 9E6A	dd ff ff	4 1 1 4 3 5
DIV	Divide	A ← (H:A)/(X) H ← Remainder	–	–	–	–	↕	↕	INH	52		7
EOR #*opr* EOR *opr* EOR *opr* EOR *opr*,X EOR *opr*,X EOR ,X EOR *opr*,SP EOR *opr*,SP	Exclusive OR M with A	A ← (A ⊕ M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A8 B8 C8 D8 E8 F8 9EE8 9ED8	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
INC *opr* INCA INCX INC *opr*,X INC ,X INC *opr*,SP	Increment	M ← (M) + 1 A ← (A) + 1 X ← (X) + 1 M ← (M) + 1 M ← (M) + 1 M ← (M) + 1	↕	–	–	↕	↕	–	DIR INH INH IX1 IX SP1	3C 4C 5C 6C 7C 9E6C	dd ff ff	4 1 1 4 3 5
JMP *opr* JMP *opr* JMP *opr*,X JMP *opr*,X JMP ,X	Jump	PC ← Jump Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BC CC DC EC FC	dd hh ll ee ff ff 	2 3 4 3 2
JSR *opr* JSR *opr* JSR *opr*,X JSR *opr*,X JSR ,X	Jump to Subroutine	PC ← (PC) + *n* (*n* = 1, 2, or 3) Push (PCL); SP ← (SP) – 1 Push (PCH); SP ← (SP) – 1 PC ← Unconditional Address	–	–	–	–	–	–	DIR EXT IX2 IX1 IX	BD CD DD ED FD	dd hh ll ee ff ff 	4 5 6 5 4
LDA #*opr* LDA *opr* LDA *opr* LDA *opr*,X LDA *opr*,X LDA ,X LDA *opr*,SP LDA *opr*,SP	Load A from M	A ← (M)	0	–	–	↕	↕	–	IMM DIR EXT IX2 IX1 IX SP1 SP2	A6 B6 C6 D6 E6 F6 9EE6 9ED6	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
LDHX #*opr* LDHX *opr*	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↕	↕	–	IMM DIR	45 55	ii jj dd	3 4

SBSW — SIM Break Stop/Wait

This status bit is useful in applications requiring a return to wait or stop mode after exiting from a break interrupt. Clear SBSW by writing a logic zero to it. Reset clears SBSW.

1 = Stop mode or wait mode was exited by break interrupt
0 = Stop mode or wait mode was not exited by break interrupt

SBSW can be read within the break state SWI routine. The user can modify the return address on the stack by subtracting one from it. The following code is an example of this. Writing zero to the SBSW bit clears it.

```
; This code works if the H register has been pushed onto the stack in the break
; service routine software. This code should be executed at the end of the
; break service routine software.

  HIBYTE   EQU    5

  LOBYTE   EQU    6

;          If not SBSW, do RTI

           BRCLR  SBSW,BSR, RETURN   ; See if wait mode or stop mode was exited
                                     ; by break.

           TST    LOBYTE,SP          ; If RETURNLO is not zero,

           BNE    DOLO               ; then just decrement low byte.

           DEC    HIBYTE,SP          ; Else deal with high byte, too.

  DOLO     DEC    LOBYTE,SP          ; Point to WAIT/STOP opcode.

  RETURN   PULH                      ; Restore H register.
           RTI
```

5.7.2 Reset Status Register (RSR)

This register contains six flags that show the source of the last reset. Clear the SIM reset status register by reading it. A power-on reset sets the POR bit and clears all other bits in the register.

Address: $FE01

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
Write:								
POR:	1	0	0	0	0	0	0	0

= Unimplemented

Figure 5-21. Reset Status Register (RSR)

POR — Power-On Reset Bit

1 = Last reset caused by POR circuit
0 = Read of RSR

6.2.2 RC Oscillator

The RC oscillator circuit is designed for use with external resistor and capacitor to provide a clock source with tolerance less than 10%.

In its typical configuration, the RC oscillator requires two external components, one R and one C. Component values should have a tolerance of 1% or less, to obtain a clock source with less than 10% tolerance. The oscillator configuration uses two components:

- C_{EXT}
- R_{EXT}

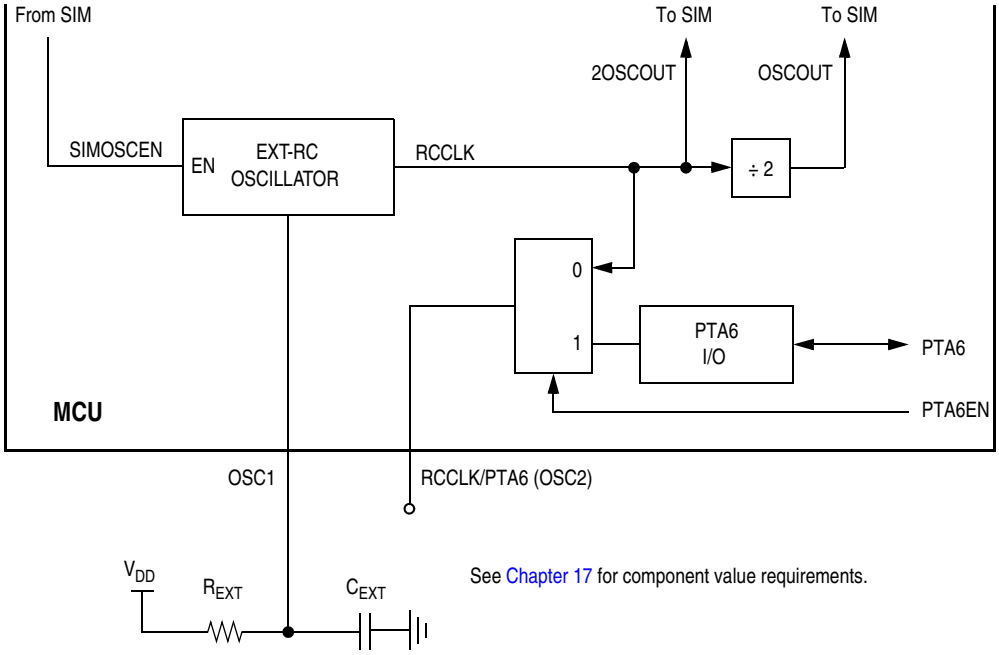


Figure 6-3. RC Oscillator External Connections

6.3 Internal Oscillator

The internal oscillator clock (ICLK) is a free running 50-kHz clock that requires no external components. It is used as the reference clock input to the computer operating properly (COP) module and the SIM.

The internal oscillator by default is always available and is free running after POR or reset. It can be stopped in stop mode by setting the STOP_ICLKDIS bit before executing the STOP instruction.

Figure 6-4 shows the logical representation of components of the internal oscillator circuitry.

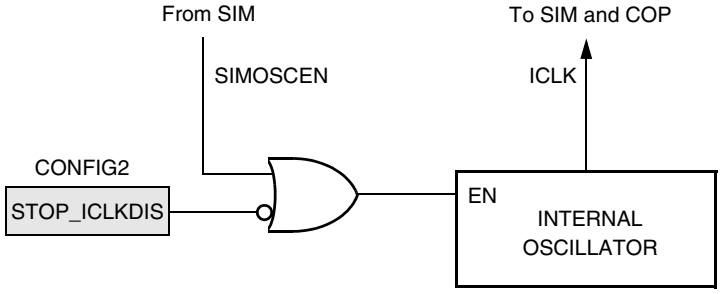


Figure 6-4. Internal Oscillator

Table 7-4. READ (Read Memory) Command

Description	Read byte from memory
Operand	Specifies 2-byte address in high byte:low byte order
Data Returned	Returns contents of specified address
Opcode	$4A
Command Sequence	

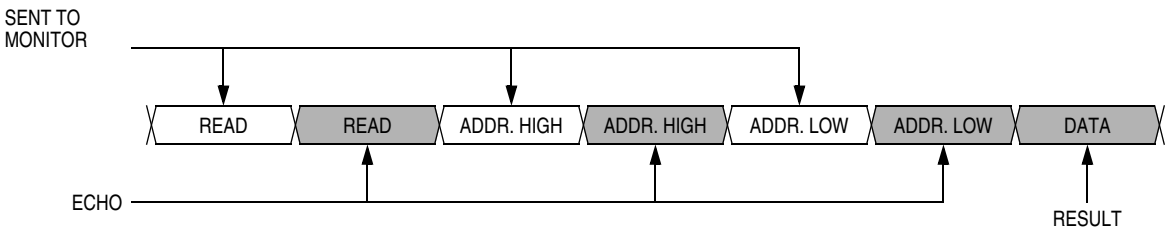
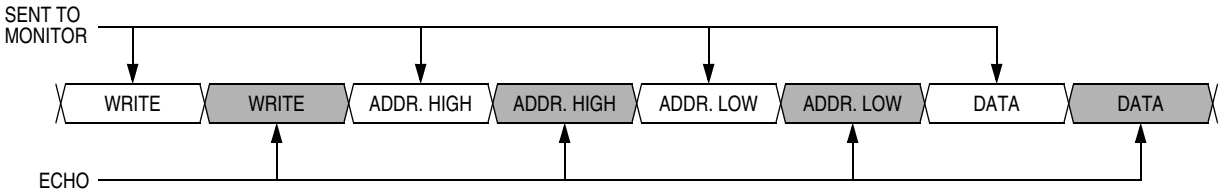


Table 7-5. WRITE (Write Memory) Command

Description	Write byte to memory
Operand	Specifies 2-byte address in high byte:low byte order; low byte followed by data byte
Data Returned	None
Opcode	$49
Command Sequence	



The coding example below is to perform a page erase, from $EF00–$EF3F. The Initialization subroutine is the same as the coding example for PRGRNGE (see 7.5.1 PRGRNGE).

```
ERARNGE         EQU     $FCBE
MAIN:
        BSR     INITIALISATION
        :
        :
        LDHX    #FILE_PTR
        JSR     ERARNGE
        :
```

7.5.3 LDRNGE

LDRNGE is used to load the data array in RAM with data from a range of FLASH locations.

Table 7-13. LDRNGE Routine

Routine Name	LDRNGE
Routine Description	Loads data from a range of locations
Calling Address	$FF30
Stack Used	9 bytes
Data Block Format	Bus speed (BUS_SPD) Data size (DATASIZE) Starting address (ADDRH) Starting address (ADDRL) Data 1 : Data N

The start location of FLASH from where data is retrieved is specified by the address ADDRH:ADDRL and the number of bytes from this location is specified by DATASIZE. The maximum number of bytes that can be retrieved in one routine call is 128 bytes. The data retrieved from FLASH is loaded into the data array in RAM. Previous data in the data array will be overwritten. User can use this routine to retrieve data from FLASH that was previously programmed.

The coding example below is to retrieve 32 bytes of data starting from $EF00 in FLASH. The Initialization subroutine is the same as the coding example for PRGRNGE (see 7.5.1 PRGRNGE).

```
LDRNGE          EQU     $FF30
MAIN:
        BSR     INITIALIZATION
        :
        :
        LDHX    #FILE_PTR
        JSR     LDRNGE
        :
```

8.4 Functional Description

Figure 8-1 shows the structure of the TIM. The central component of the TIM is the 16-bit TIM counter that can operate as a free-running counter or a modulo up-counter. The TIM counter provides the timing reference for the input capture and output compare functions. The TIM counter modulo registers, TMODH:TMODL, control the modulo value of the TIM counter. Software can read the TIM counter value at any time without affecting the counting sequence.

The two TIM channels (per timer) are programmable independently as input capture or output compare channels.

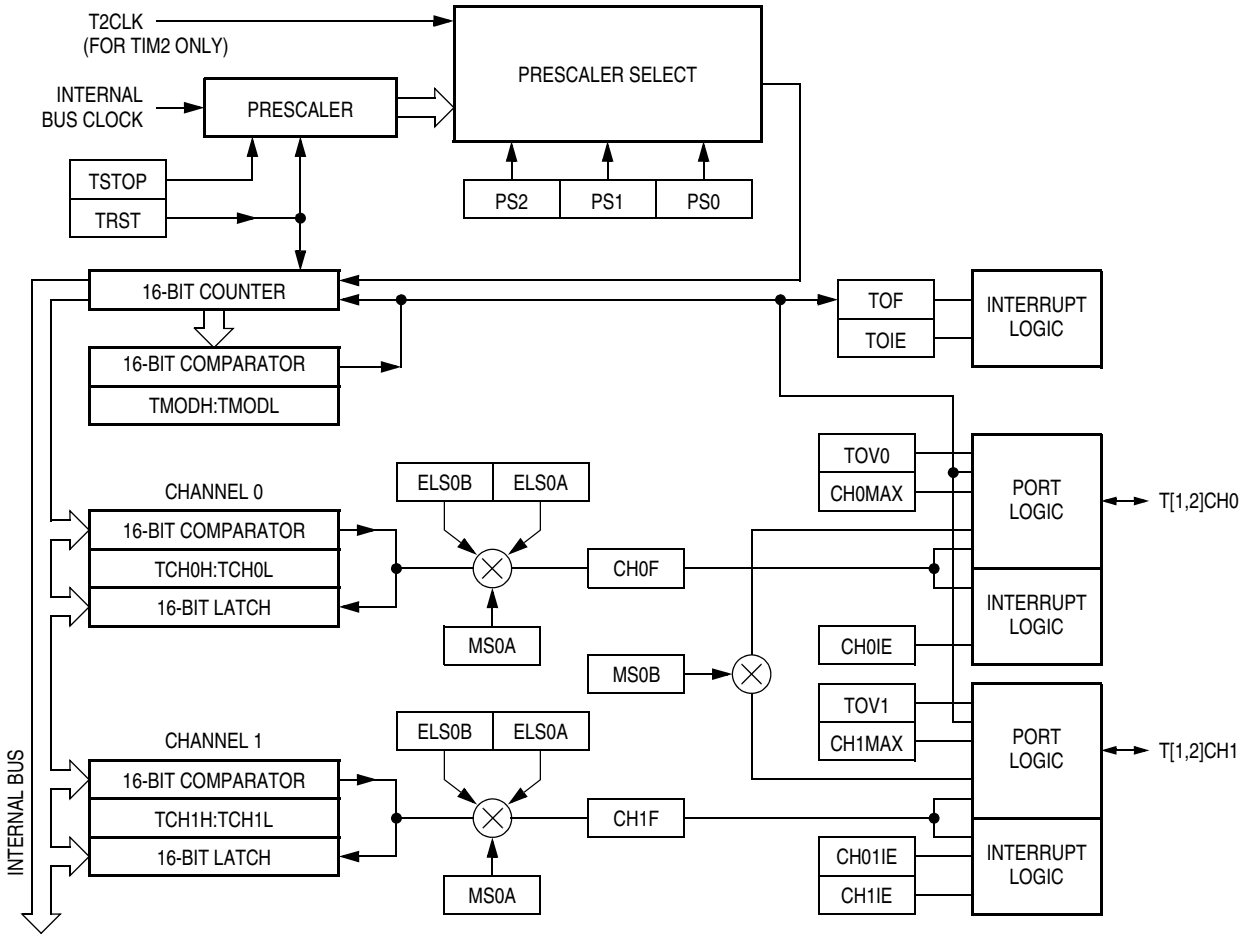


Figure 8-1. TIM Block Diagram

Figure 8-2 summarizes the timer registers.

NOTE

References to either timer 1 or timer 2 may be made in the following text by omitting the timer number. For example, TSC may generically refer to both T1SC and T2SC.

8.4.3.1 Unbuffered Output Compare

Any output compare channel can generate unbuffered output compare pulses as described in 8.4.3 Output Compare. The pulses are unbuffered because changing the output compare value requires writing the new value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change an output compare value could cause incorrect operation for up to two counter overflow periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that counter overflow period. Also, using a TIM overflow interrupt routine to write a new, smaller output compare value may cause the compare to be missed. The TIM may pass the new value before it is written.

Use the following methods to synchronize unbuffered changes in the output compare value on channel x:

- When changing to a smaller value, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current output compare pulse. The interrupt routine has until the end of the counter overflow period to write the new value.
- When changing to a larger output compare value, enable TIM overflow interrupts and write the new value in the TIM overflow interrupt routine. The TIM overflow interrupt occurs at the end of the current counter overflow period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same counter overflow period.

8.4.3.2 Buffered Output Compare

Channels 0 and 1 can be linked to form a buffered output compare channel whose output appears on the TCH0 pin. The TIM channel registers of the linked pair alternately control the output.

Setting the MS0B bit in TIM channel 0 status and control register (TSC0) links channel 0 and channel 1. The output compare value in the TIM channel 0 registers initially controls the output on the TCH0 pin. Writing to the TIM channel 1 registers enables the TIM channel 1 registers to synchronously control the output after the TIM overflows. At each subsequent overflow, the TIM channel registers (0 or 1) that control the output are the ones written to last. TSC0 controls and monitors the buffered output compare function, and TIM channel 1 status and control register (TSC1) is unused. While the MS0B bit is set, the channel 1 pin, TCH1, is available as a general-purpose I/O pin.

NOTE

In buffered output compare operation, do not write new output compare values to the currently active channel registers. User software should track the currently active channel to prevent writing a new value to the active channel. Writing to the active channel registers is the same as generating unbuffered output compares.

8.4.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As Figure 8-3 shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM

9.8 I/O Registers

These I/O registers control and monitor SCI operation:

- SCI control register 1 (SCC1)
- SCI control register 2 (SCC2)
- SCI control register 3 (SCC3)
- SCI status register 1 (SCS1)
- SCI status register 2 (SCS2)
- SCI data register (SCDR)
- SCI baud rate register (SCBR)

9.8.1 SCI Control Register 1

SCI control register 1:

- Enables loop mode operation
- Enables the SCI
- Controls output polarity
- Controls character length
- Controls SCI wakeup method
- Controls idle character detection
- Enables parity function
- Controls parity type

Address: $0013

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
Reset:	0	0	0	0	0	0	0	0

Figure 9-9. SCI Control Register 1 (SCC1)

LOOPS — Loop Mode Select Bit

This read/write bit enables loop mode operation. In loop mode the RxD pin is disconnected from the SCI, and the transmitter output goes into the receiver input. Both the transmitter and the receiver must be enabled to use loop mode. Reset clears the LOOPS bit.

1 = Loop mode enabled
0 = Normal operation enabled

ENSCI — Enable SCI Bit

This read/write bit enables the SCI and the SCI baud rate generator. Clearing ENSCI sets the SCTE and TC bits in SCI status register 1 and disables transmitter interrupts. Reset clears the ENSCI bit.

1 = SCI enabled
0 = SCI disabled

TXINV — Transmit Inversion Bit

This read/write bit reverses the polarity of transmitted data. Reset clears the TXINV bit.

1 = Transmitter output inverted
0 = Transmitter output not inverted

NOTE

Setting the TXINV bit inverts all transmitted values, including idle, break, start, and stop bits.

When DDREx is a logic 1, reading address $0008 reads the PTEx data latch. When DDREx is a logic 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 11-5 summarizes the operation of the port E pins.

Table 11-5. Port E Pin Functions

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X(1)	Input, Hi-Z(2)	DDRE[1:0]	Pin	PTE[1:0](3)
1	X	Output	DDRE[1:0]	PTE[1:0]	PTE[1:0]

1. X = don't care.
2. Hi-Z = high impedance.
3. Writing affects data register, but does not affect the input.

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the latch. Software may generate the interrupt acknowledge signal by writing a logic one to the ACK bit in the interrupt status and control register (INTSCR). The ACK bit is useful in applications that poll the $\overline{IRQ}$ pin and require software to clear the IRQ latch. Writing to the ACK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{IRQ}$ pin. A falling edge that occurs after writing to the ACK bit latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the vector address at locations $FFFA and $FFFB.
- Return of the $\overline{IRQ}$ pin to logic one — As long as the $\overline{IRQ}$ pin is at logic zero, IRQ remains active.

The vector fetch or software clear and the return of the $\overline{IRQ}$ pin to logic one may occur in any order. The interrupt request remains pending as long as the $\overline{IRQ}$ pin is at logic zero. A reset will clear the latch and the MODE control bit, thereby clearing the interrupt even if the pin stays low.

If the MODE bit is clear, the $\overline{IRQ}$ pin is falling-edge-sensitive only. With MODE clear, a vector fetch or software clear immediately clears the IRQ latch.

The IRQF bit in the INTSCR register can be used to check for pending interrupts. The IRQF bit is not affected by the IMASK bit, which makes it useful in applications where polling is preferred.

Use the BIH or BIL instruction to read the logic level on the $\overline{IRQ}$ pin.

NOTE

When using the level-sensitive interrupt trigger, avoid false interrupts by masking interrupt requests in the interrupt routine.

NOTE

An internal pull-up resistor to V_{DD} is connected to the $\overline{IRQ}$ pin; this can be disabled by setting the IRQPUD bit in the CONFIG2 register ($001E).

12.4 IRQ Module During Break Interrupts

The system integration module (SIM) controls whether the IRQ latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear the latches during the break state. (See Chapter 5 System Integration Module (SIM).)

To allow software to clear the IRQ latch during a break interrupt, write a logic one to the BCFE bit. If a latch is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the latches during the break state, write a logic zero to the BCFE bit. With BCFE at logic zero (its default state), writing to the ACK bit in the IRQ status and control register during the break state has no effect on the IRQ latch.

12.5 IRQ Status and Control Register (INTSCR)

The IRQ status and control register (INTSCR) controls and monitors operation of the IRQ module. The INTSCR has the following functions:

- Shows the state of the IRQ flag
- Clears the IRQ latch
- Masks IRQ and interrupt request
- Controls triggering sensitivity of the $\overline{IRQ}$ interrupt pin

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	IRQF		IMASK	MODE
Write:						ACK		
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 12-3. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag Bit

This read-only status bit is high when the IRQ interrupt is pending.

1 = $\overline{IRQ}$ interrupt pending
0 = $\overline{IRQ}$ interrupt not pending

ACK — IRQ Interrupt Request Acknowledge Bit

Writing a logic one to this write-only bit clears the IRQ latch. ACK always reads as logic zero. Reset clears ACK.

IMASK — IRQ Interrupt Mask Bit

Writing a logic one to this read/write bit disables IRQ interrupt requests. Reset clears IMASK.

1 = IRQ interrupt requests disabled
0 = IRQ interrupt requests enabled

MODE — IRQ Edge/Level Select Bit

This read/write bit controls the triggering sensitivity of the $\overline{IRQ}$ pin. Reset clears MODE.

1 = $\overline{IRQ}$ interrupt requests on falling edges and low levels
0 = $\overline{IRQ}$ interrupt requests on falling edges only

Address: $001E

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	IRQPUD	R	R	LVIT1	LVIT0	R	R	R
Reset:	0	0	0	Not affected	Not affected	0	0	0
POR:	0	0	0	0	0	0	0	0

R = Reserved

Figure 12-4. Configuration Register 2 (CONFIG2)

IRQPUD — $\overline{IRQ}$ Pin Pull-Up Disable Bit

IRQPUD disconnects the internal pull-up on the $\overline{IRQ}$ pin.

1 = Internal pull-up is disconnected
0 = Internal pull-up is connected between $\overline{IRQ}$ pin and V_{DD}

14.3.7 COPRS (COP Rate Select)

The COPRS signal reflects the state of the COP rate select bit (COPRS) in the configuration register 1.

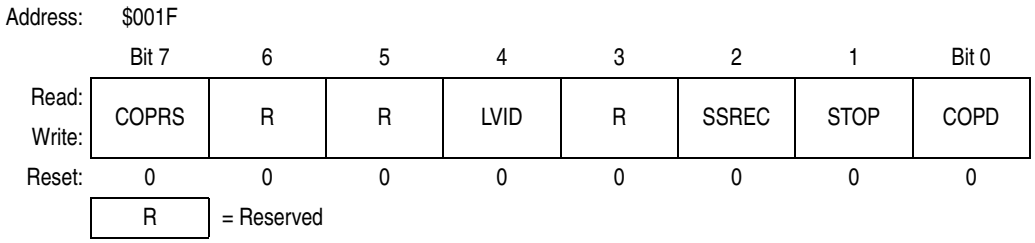


Figure 14-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS.

1 = COP timeout period is ($2^{13} – 2^4$) ICLK cycles
0 = COP timeout period is ($2^{18} – 2^4$) ICLK cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled
0 = COP module enabled

14.4 COP Control Register

The COP control register is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

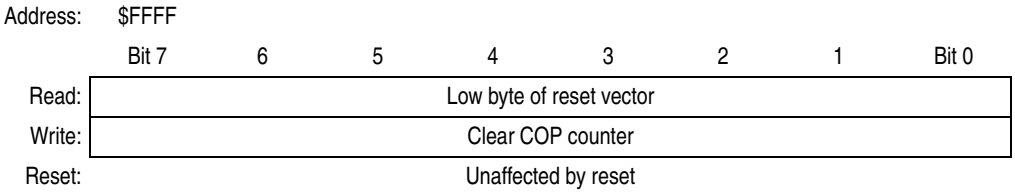


Figure 14-3. COP Control Register (COPCTL)

14.5 Interrupts

The COP does not generate CPU interrupt requests.

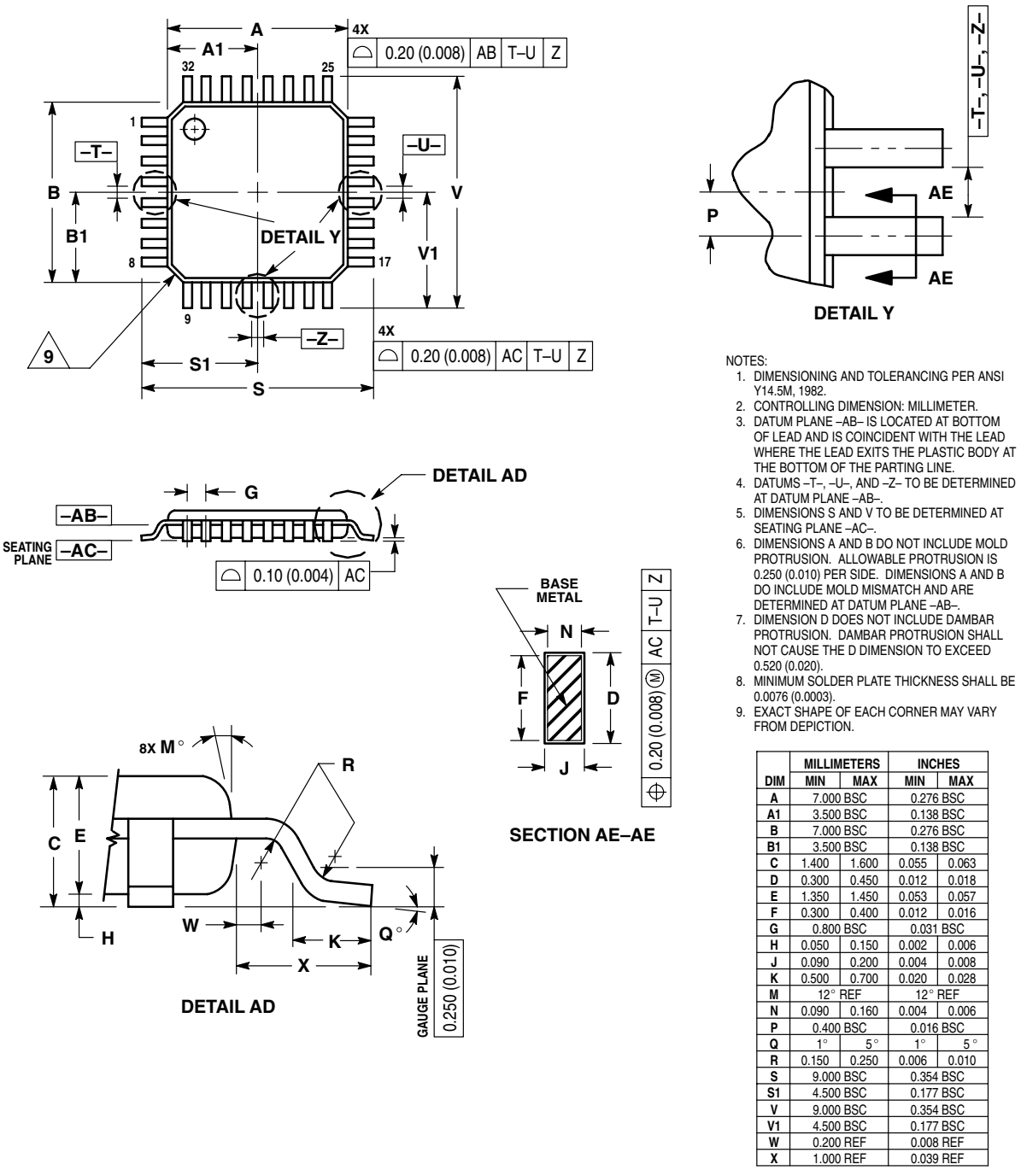
14.6 Monitor Mode

The COP is disabled in monitor mode when V_{TST} is present on the $\overline{IRQ}$ pin or on the $\overline{RST}$ pin.

14.7 Low-Power Modes

The WAIT and STOP instructions put the MCU in low-power consumption standby modes.

18.7 32-Pin Low-Profile Quad Flat Pack (LQFP)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
2. CONTROLLING DIMENSION: MILLIMETER.
3. DATUM PLANE –AB– IS LOCATED AT BOTTOM OF LEAD AND IS COINCIDENT WITH THE LEAD WHERE THE LEAD EXITS THE PLASTIC BODY AT THE BOTTOM OF THE PARTING LINE.
4. DATUMS –T–, –U–, AND –Z– TO BE DETERMINED AT DATUM PLANE –AB–.
5. DIMENSIONS S AND V TO BE DETERMINED AT SEATING PLANE –AC–.
6. DIMENSIONS A AND B DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.250 (0.010) PER SIDE. DIMENSIONS A AND B DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –AB–.
7. DIMENSION D DOES NOT INCLUDE DAMBAR PROTRUSION. DAMBAR PROTRUSION SHALL NOT CAUSE THE D DIMENSION TO EXCEED 0.520 (0.020).
8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076 (0.0003).
9. EXACT SHAPE OF EACH CORNER MAY VARY FROM DEPICTION.

	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	7.000 BSC		0.276 BSC	
A1	3.500 BSC		0.138 BSC	
B	7.000 BSC		0.276 BSC	
B1	3.500 BSC		0.138 BSC	
C	1.400	1.600	0.055	0.063
D	0.300	0.450	0.012	0.018
E	1.350	1.450	0.053	0.057
F	0.300	0.400	0.012	0.016
G	0.800 BSC		0.031 BSC	
H	0.050	0.150	0.002	0.006
J	0.090	0.200	0.004	0.008
K	0.500	0.700	0.020	0.028
M	12° REF		12° REF	
N	0.090	0.160	0.004	0.006
P	0.400 BSC		0.016 BSC	
Q	1°	5°	1°	5°
R	0.150	0.250	0.006	0.010
S	9.000 BSC		0.354 BSC	
S1	4.500 BSC		0.177 BSC	
V	9.000 BSC		0.354 BSC	
V1	4.500 BSC		0.177 BSC	
W	0.200 REF		0.008 REF	
X	1.000 REF		0.039 REF	

Figure 18-6. 32-Pin LQFP (Case #873A)

Appendix A
MC68HC08JL8

A.1 Introduction

This section introduces the MC68HC08JL8, the ROM part equivalent to the MC68HC908JL8/JK8. The entire data book applies to this ROM device, with exceptions outlined in this appendix.

Table A-1. Summary of MC68HC08JL8 and MC68HC908JL8 Differences

	MC68HC08JL8	MC68HC908JL8
Memory ($DC00–$FBFF)	8,192 bytes ROM	8,192 bytes FLASH
User vectors ($FFDC–$FFFF)	36 bytes ROM	36 bytes FLASH
Registers at $FE08 and $FFCF	Not used; locations are reserved.	FLASH related registers. $FE08 — FLCR $FFCF — FLBPR
Mask option register ($FFD0)	Defined by mask; read only.	Read/write FLASH register.
Monitor ROM ($FC00–$FDFF and $FE10–$FFCE)	$FC00–$FDFF: Not used. $FE10–$FFCE: Used for testing purposes only.	Used for testing and FLASH programming/erasing.
Available Packages	20-pin PDIP (MC68HC08JK8) 20-pin SOIC (MC68HC08JK8) 28-pin PDIP 28-pin SOIC 32-pin SDIP 32-pin LQFP	20-pin PDIP (MC68HC908JK8) 20-pin SOIC (MC68HC908JK8) 28-pin PDIP 28-pin SOIC 32-pin SDIP 32-pin LQFP

A.2 MCU Block Diagram

Figure A-1 shows the block diagram of the MC68HC08JL8.

A.3 Memory Map

The MC68HC08JL8 has 8,192 bytes of user ROM from $DC00 to $FBFF, and 36 bytes of user ROM vectors from $FFDC to $FFFF. On the MC68HC908JL8, these memory locations are FLASH memory.

Figure A-2 shows the memory map of the MC68HC08JL8.

A.5 Mask Option Register

The mask option register at $FFD0 is read only. The value is defined by mask option (hard-wired connections) specified at the time as the ROM code submission.

On the MC68HC908JL8, the MOR is implemented as a FLASH, which can be programmed, erased, and read.

A.6 Monitor ROM

The monitor program (monitor ROM: $FE10–$FFCE) on the MC68HC08JL8 is for device testing only. $FC00–$FDFF are unused.

A.7 Electrical Specifications

Electrical specifications for the MC68HC908JL8 apply to the MC68HC08JL8, except for the parameters indicated below.

A.7.1 DC Electrical Characteristics

Table A-2. DC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
V_{DD} supply current, f_{OP} = 8MHz RC oscillator option	I_{DD}	Values same as, and characterized from MC68HC908JL8, but not tested.			
Low-voltage inhibit, trip falling voltage	V_{TRIPF}	3.55 (3.60)[3]	4.02 (4.25)	4.48 (4.48)	V
Low-voltage inhibit, trip rising voltage	V_{TRIPR}	3.66 (3.75)	4.13 (4.40)	4.59 (4.63)	V

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. The numbers in parenthesis are MC68HC908JL8 values.

Table A-3. DC Electrical Characteristics (3V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
V_{DD} supply current, f_{OP} = 4MHz RC oscillator option	I_{DD}	Values same as, and characterized from MC68HC908JL8, but not tested.			
Low-voltage inhibit, trip voltage (No hysteresis implemented for 3V LVI)	V_{LVI3}	2.1 (2.18)[3]	2.4 (2.49)	2.69 (2.68)	V

1. V_{DD} = 2.7 to 3.3 Vdc, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted.
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.
3. The numbers in parenthesis are MC68HC908JL8 values.