



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	Obsolete
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	-
Peripherals	LED, LVD, POR, PWM
Number of I/O	23
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.3V
Data Converters	A/D 12x8b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908jl3ecdw

1.5 Pin Functions

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

Table 1-2. Pin Functions

PIN NAME	PIN DESCRIPTION	IN/OUT	VOLTAGE LEVEL
$V_{DDJL3JL3}$	Power supply.	In	5V or 3V
V_{SS}	Power supply ground	Out	0V
\overline{RST}	RESET input, active low. With Internal pull-up and Schmitt trigger input.	Input	V_{DD} to V_{TST}
\overline{IRQ}	External IRQ pin. With software programmable internal pull-up and schmitt trigger input. This pin is also used for mode entry selection.	Input	V_{DD} to V_{TST}
OSC1	X-tal or RC oscillator input.	In	Analog
OSC2	MC68HC908JL3E/JK3E/JK1E: X-tal oscillator output, this is the inverting OSC1 signal.	Out	Analog
	MC68HRC908JL3E/JK3E/JK1E: Default is RC oscillator clock output, RCCLK. Shared with PTA6/KBI6, with programmable pull-up.	In/Out	V_{DD}
PTA[0:6]	7-bit general purpose I/O port.	In/Out	V_{DD}
	Shared with 7 keyboard interrupts KBI[0:6].	In	V_{DD}
	Each pin has programmable internal pull-up device.	In	V_{DD}
	PTA[0:5] have LED direct sink capability	In	V_{SS}
PTB[0:7]	8-bit general purpose I/O port.	In/Out	V_{DD}
	Shared with 8 ADC inputs, ADC[0:7].	In	Analog
PTD[0:7]	8-bit general purpose I/O port.	In/Out	V_{DD}
	PTD[3:0] shared with 4 ADC inputs, ADC[8:11].	Input	Analog
	PTD[4:5] shared with TIM channels, TCH0 and TCH1.	In/Out	V_{DD}
	PTD[2:3], PTD[6:7] have LED direct sink capability	In	V_{SS}
	PTD[6:7] can be configured as 25mA open-drain output with pull-up.	In/Out	V_{DD}

NOTE

On the MC68H(R)C908JK3E/JK1E, the following pins are not available:
PTA0, PTA1, PTA2, PTA3, PTA4, PTA5, PTD0, and PTD1.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0000	Port A Data Register (PTA)	Read:	0	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
\$0001	Port B Data Register (PTB)	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
\$0002	Unimplemented	Read:								
		Write:								
\$0003	Port D Data Register (PTD)	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
\$0004	Data Direction Register A (DDRA)	Read:	0	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB)	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0006	Unimplemented	Read:								
		Write:								
\$0007	Data Direction Register D (DDRD)	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0008 ↓ \$0009	Unimplemented	Read:								
		Write:								
\$000A	Port D Control Register (PDCR)	Read:	0	0	0	0	SLOWD7	SLOWD6	PTDPU7	PTDPU6
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000B ↓ \$000C	Unimplemented	Read:								
		Write:								
\$000D	Port A Input Pull-up Enable Register (PTAPUE)	Read:	PTA6EN	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000E ↓ \$0019	Unimplemented	Read:								
		Write:								
				= Unimplemented			R	= Reserved		

= Unimplemented

= Reserved

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 4)

2.8 Flash Page Erase Operation

Use the following procedure to erase a page of Flash memory. A page consists of 64 consecutive bytes starting from addresses \$XX00, \$XX40, \$XX80 or \$XXC0. The 48-byte user interrupt vectors area also forms a page. Any page within the 4K bytes user memory area (\$EC00–\$FBFF) can be erased alone. *The 48-byte user interrupt vectors cannot be erased by the page erase operation because of security reasons. Mass erase is required to erase this page.*

1. Set the ERASE bit and clear the MASS bit in the Flash Control Register.
2. Write any data to any Flash address within the page address range desired.
3. Wait for a time, t_{nvs} (10 μ s).
4. Set the HVEN bit.
5. Wait for a time t_{Erase} (1ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh} (5 μ s).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 μ s), the memory can be accessed in read mode again.

NOTE

Programming and erasing of Flash locations cannot be performed by code being executed from the Flash memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

2.9 Flash Mass Erase Operation

Use the following procedure to erase the entire Flash memory:

1. Set both the ERASE bit and the MASS bit in the Flash Control Register.
2. Write any data to any Flash location within the Flash memory address range.
3. Wait for a time, t_{nvs} (10 μ s).
4. Set the HVEN bit.
5. Wait for a time t_{MErase} (4ms).
6. Clear the ERASE bit.
7. Wait for a time, t_{nvh1} (100 μ s).
8. Clear the HVEN bit.
9. After time, t_{rcv} (1 μ s), the memory can be accessed in read mode again.

NOTE

Programming and erasing of Flash locations cannot be performed by code being executed from the Flash memory. While these operations must be performed in the order as shown, but other unrelated operations may occur between the steps.

Chapter 4

Central Processor Unit (CPU)

4.1 Introduction

The M68HC08 CPU (central processor unit) is an enhanced and fully object-code-compatible version of the M68HC05 CPU. The *CPU08 Reference Manual* (document order number CPU08RM/AD) contains a description of the CPU instruction set, addressing modes, and architecture.

4.2 Features

Features of the CPU include:

- Object code fully upward-compatible with M68HC05 Family
- 16-bit stack pointer with stack manipulation instructions
- 16-bit index register with x-register manipulation instructions
- 8-MHz CPU internal bus frequency
- 64-Kbyte program/data memory space
- 16 addressing modes
- Memory-to-memory data moves without using accumulator
- Fast 8-bit by 8-bit multiply and 16-bit by 8-bit divide instructions
- Enhanced binary-coded decimal (BCD) data handling
- Modular architecture with expandable internal bus definition for extension of addressing range beyond 64 Kbytes
- Low-power stop and wait modes

4.3 CPU Registers

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

Table 4-1. Instruction Set Summary (Sheet 2 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
BHS <i>rel</i>	Branch if Higher or Same (Same as BCC)	$PC \leftarrow (PC) + 2 + rel ? (C) = 0$	-	-	-	-	-	REL	24	rr	3
BIH <i>rel</i>	Branch if IRQ Pin High	$PC \leftarrow (PC) + 2 + rel ? \overline{IRQ} = 1$	-	-	-	-	-	REL	2F	rr	3
BIL <i>rel</i>	Branch if IRQ Pin Low	$PC \leftarrow (PC) + 2 + rel ? \overline{IRQ} = 0$	-	-	-	-	-	REL	2E	rr	3
BIT # <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> BIT <i>opr</i> ,X BIT <i>opr</i> ,X BIT ,X BIT <i>opr</i> ,SP BIT <i>opr</i> ,SP	Bit Test	(A) & (M)	0	-	-	†	†	IMM DIR EXT IX2 IX1 IX SP1 SP2	A5 B5 C5 D5 E5 F5 9EE5 9ED5	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
BLE <i>opr</i>	Branch if Less Than or Equal To (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (Z) \vee (N \oplus V) = 1$	-	-	-	-	-	REL	93	rr	3
BLO <i>rel</i>	Branch if Lower (Same as BCS)	$PC \leftarrow (PC) + 2 + rel ? (C) = 1$	-	-	-	-	-	REL	25	rr	3
BLS <i>rel</i>	Branch if Lower or Same	$PC \leftarrow (PC) + 2 + rel ? (C) \vee (Z) = 1$	-	-	-	-	-	REL	23	rr	3
BLT <i>opr</i>	Branch if Less Than (Signed Operands)	$PC \leftarrow (PC) + 2 + rel ? (N \oplus V) = 1$	-	-	-	-	-	REL	91	rr	3
BMC <i>rel</i>	Branch if Interrupt Mask Clear	$PC \leftarrow (PC) + 2 + rel ? (I) = 0$	-	-	-	-	-	REL	2C	rr	3
BMI <i>rel</i>	Branch if Minus	$PC \leftarrow (PC) + 2 + rel ? (N) = 1$	-	-	-	-	-	REL	2B	rr	3
BMS <i>rel</i>	Branch if Interrupt Mask Set	$PC \leftarrow (PC) + 2 + rel ? (I) = 1$	-	-	-	-	-	REL	2D	rr	3
BNE <i>rel</i>	Branch if Not Equal	$PC \leftarrow (PC) + 2 + rel ? (Z) = 0$	-	-	-	-	-	REL	26	rr	3
BPL <i>rel</i>	Branch if Plus	$PC \leftarrow (PC) + 2 + rel ? (N) = 0$	-	-	-	-	-	REL	2A	rr	3
BRA <i>rel</i>	Branch Always	$PC \leftarrow (PC) + 2 + rel$	-	-	-	-	-	REL	20	rr	3
BRCLR <i>n,opr,rel</i>	Branch if Bit <i>n</i> in M Clear	$PC \leftarrow (PC) + 3 + rel ? (Mn) = 0$	-	-	-	-	†	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	01 03 05 07 09 0B 0D 0F	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BRN <i>rel</i>	Branch Never	$PC \leftarrow (PC) + 2$	-	-	-	-	-	REL	21	rr	3
BRSET <i>n,opr,rel</i>	Branch if Bit <i>n</i> in M Set	$PC \leftarrow (PC) + 3 + rel ? (Mn) = 1$	-	-	-	-	†	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	00 02 04 06 08 0A 0C 0E	dd rr dd rr dd rr dd rr dd rr dd rr dd rr dd rr	5 5 5 5 5 5 5 5
BSET <i>n,opr</i>	Set Bit <i>n</i> in M	$Mn \leftarrow 1$	-	-	-	-	-	DIR (b0) DIR (b1) DIR (b2) DIR (b3) DIR (b4) DIR (b5) DIR (b6) DIR (b7)	10 12 14 16 18 1A 1C 1E	dd dd dd dd dd dd dd dd	4 4 4 4 4 4 4 4
BSR <i>rel</i>	Branch to Subroutine	$PC \leftarrow (PC) + 2$; push (PCL) $SP \leftarrow (SP) - 1$; push (PCH) $SP \leftarrow (SP) - 1$ $PC \leftarrow (PC) + rel$	-	-	-	-	-	REL	AD	rr	4
CBEQ <i>opr,rel</i> CBEQA # <i>opr,rel</i> CBEQX # <i>opr,rel</i> CBEQ <i>opr,X+,rel</i> CBEQ <i>X+,rel</i> CBEQ <i>opr,SP,rel</i>	Compare and Branch if Equal	$PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (X) - (M) = \00 $PC \leftarrow (PC) + 3 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 2 + rel ? (A) - (M) = \00 $PC \leftarrow (PC) + 4 + rel ? (A) - (M) = \00	-	-	-	-	-	DIR IMM IMM IX1+ IX+ SP1	31 41 51 61 71 9E61	dd rr ii rr ii rr ff rr rr ff rr	5 4 4 5 4 6
CLC	Clear Carry Bit	$C \leftarrow 0$	-	-	-	-	0	INH	98		1
CLI	Clear Interrupt Mask	$I \leftarrow 0$	-	-	0	-	-	INH	9A		2

5.7.3 Break Flag Control Register (BFCR)

The break control register contains a bit that enables software to clear status bits while the MCU is in a break state.

Address:	\$FE03							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	BCFE	R	R	R	R	R	R	R
Write:								
Reset:	0							
	<div>R</div> = Reserved							

Figure 5-22. Break Flag Control Register (BFCR)

BCFE — Break Clear Flag Enable Bit

This read/write bit enables software to clear status bits by accessing status registers while the MCU is in a break state. To clear status bits during the break state, the BCFE bit must be set.

1 = Status bits clearable during break

0 = Status bits not clearable during break

Chapter 7

Monitor ROM (MON)

7.1 Introduction

This section describes the monitor ROM (MON) and the monitor mode entry methods. The monitor ROM allows complete testing of the MCU through a single-wire interface with a host computer. This mode is also used for programming and erasing of Flash memory in the MCU. Monitor mode entry can be achieved without use of the higher test voltage, V_{TST} , as long as vector addresses \$FFFE and \$FFFF are blank, thus reducing the hardware requirements for in-circuit programming.

7.2 Features

Features of the monitor ROM include the following:

- Normal user-mode pin functionality
- One pin dedicated to serial communication between monitor ROM and host computer
- Standard mark/space non-return-to-zero (NRZ) communication with host computer
- Execution of code in RAM or Flash
- Flash memory security feature⁽¹⁾
- Flash memory programming interface
- 960 bytes monitor ROM code size
- Monitor mode entry without high voltage, V_{TST} , if reset vector is blank (\$FFFE and \$FFFF contain \$FF)
- Standard monitor mode entry if high voltage, V_{TST} , is applied to \overline{IRQ}

7.3 Functional Description

The monitor ROM receives and executes commands from a host computer. Figure 7-1 shows a example circuit used to enter monitor mode and communicate with a host computer via a standard RS-232 interface.

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute host-computer code in RAM while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTB0 pin. A level-shifting and multiplexing interface is required between PTB0 and the host computer. PTB0 is used in a wired-OR configuration and requires a pull-up resistor.

1. No security feature is absolutely secure. However, Freescale's strategy is to make reading or copying the Flash difficult for unauthorized users.

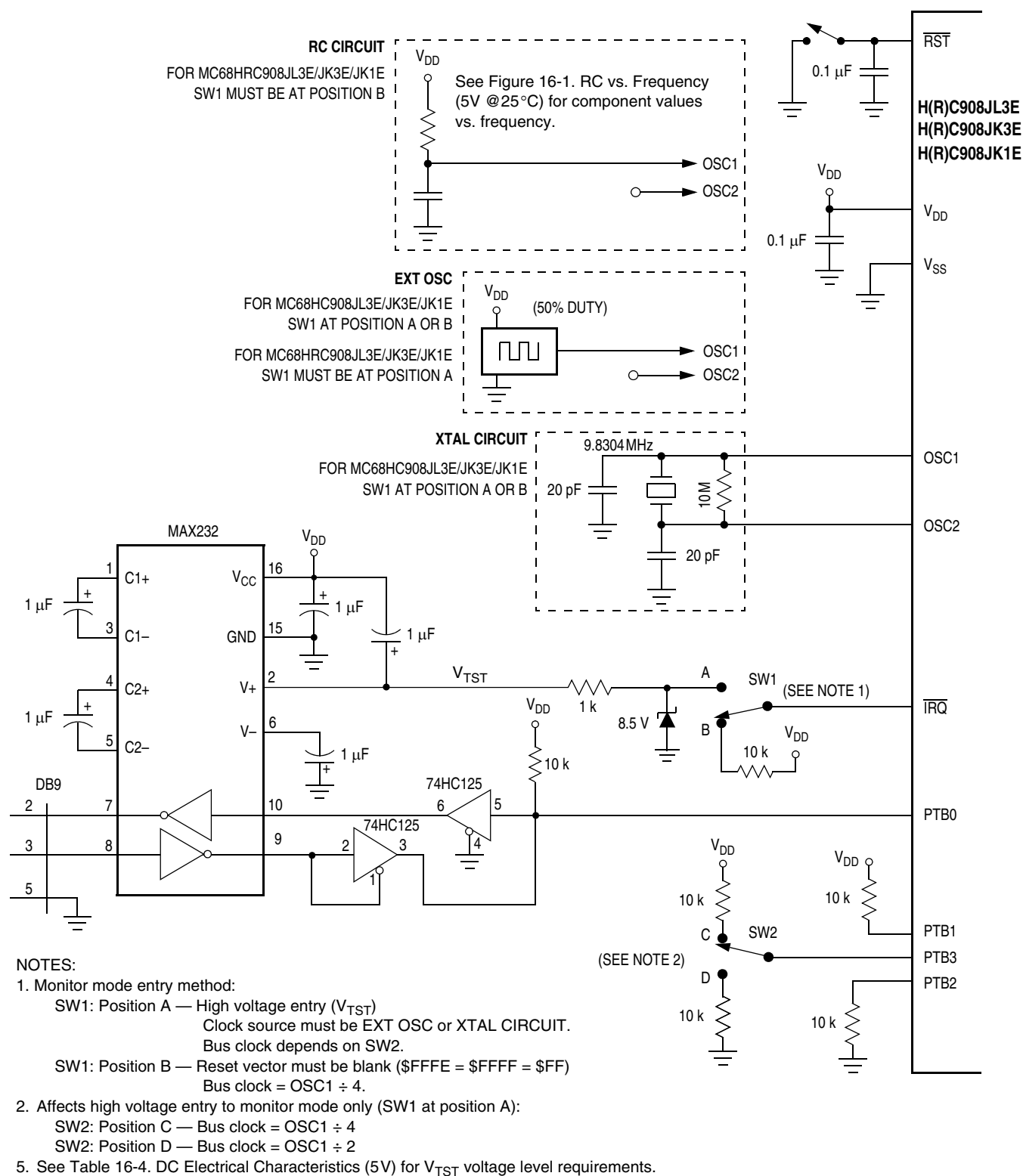


Figure 7-1. Monitor Mode Circuit

8.9.4 TIM Channel Status and Control Registers (TSC0:TSC1)

Each of the TIM channel status and control registers does the following:

- Flags input captures and output compares
- Enables input capture and output compare interrupts
- Selects input capture, output compare, or PWM operation
- Selects high, low, or toggling output on output compare
- Selects rising edge, falling edge, or any edge as the active input capture trigger
- Selects output toggling on TIM overflow
- Selects 0% and 100% PWM duty cycle
- Selects buffered or unbuffered output compare/PWM operation

Address:	\$0025	TSC0						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0

Address:	\$0028	TSC1						
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
Write:	0							
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 8-7. TIM Channel Status and Control Registers (TSC0:TSC1)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM counter registers matches the value in the TIM channel x registers.

When TIM CPU interrupt requests are enabled (CHxIE=1), clear CHxF by reading the TIM channel x status and control register with CHxF set and then writing a zero to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing zero to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a one to CHxF has no effect.

- 1 = Input capture or output compare on channel x
- 0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM CPU interrupt service requests on channel x. Reset clears the CHxIE bit.

- 1 = Channel x CPU interrupt requests enabled
- 0 = Channel x CPU interrupt requests disabled

12.4 Functional Description

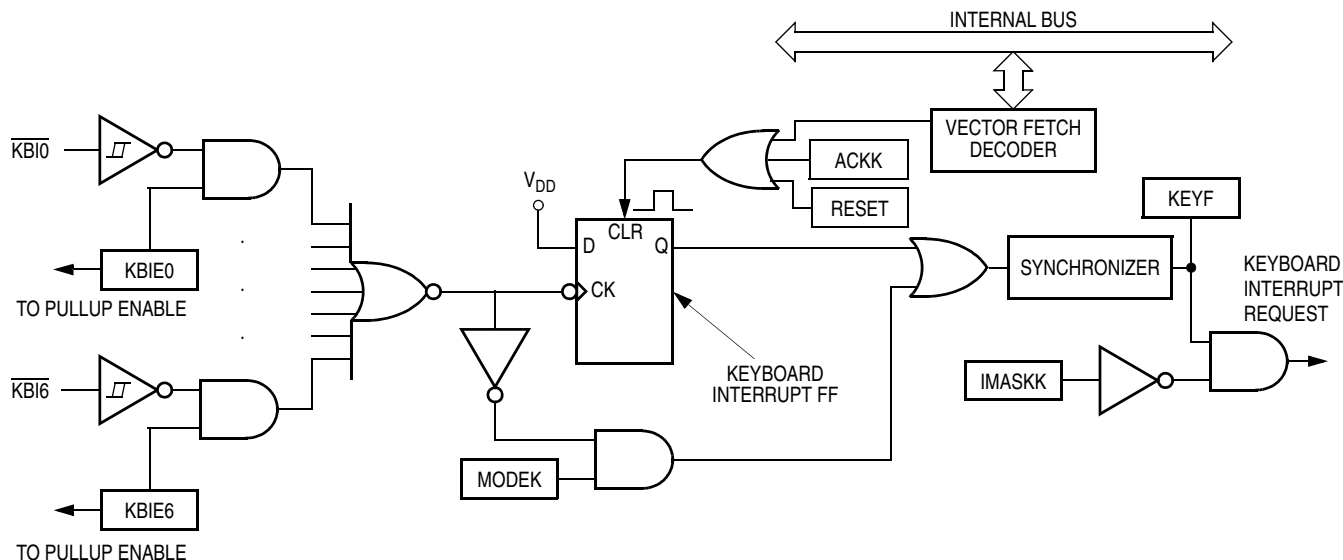


Figure 12-2. Keyboard Interrupt Block Diagram

Writing to the KBIE6–KBIE0 bits in the keyboard interrupt enable register independently enables or disables each port A pin as a keyboard interrupt pin. Enabling a keyboard interrupt pin in port A also enables its internal pull-up device irrespective of PTAPUE_x bits in the port A input pull-up enable register (see 10.2.3 Port A Input Pull-up Enable Register (PTAPUE)). A logic 0 applied to an enabled keyboard interrupt pin latches a keyboard interrupt request.

A keyboard interrupt is latched when one or more keyboard pins goes low after all were high. The MODEK bit in the keyboard status and control register controls the triggering mode of the keyboard interrupt.

- If the keyboard interrupt is edge-sensitive only, a falling edge on a keyboard pin does not latch an interrupt request if another keyboard pin is already low. To prevent losing an interrupt request on one pin because another pin is still low, software can disable the latter pin while it is low.
- If the keyboard interrupt is falling edge- and low level-sensitive, an interrupt request is present as long as any keyboard pin is low.

If the MODEK bit is set, the keyboard interrupt pins are both falling edge- and low level-sensitive, and both of the following actions must occur to clear a keyboard interrupt request:

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a 1 to the ACKK bit in the keyboard status and control register KBSCR. The ACKK bit is useful in applications that poll the keyboard interrupt pins and require software to clear the keyboard interrupt request. Writing to the ACKK bit prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACKK does not affect subsequent transitions on the keyboard interrupt pins. A falling edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations \$FFE0 and \$FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at 0, the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to 1 may occur in any order.

If the MODEK bit is clear, the keyboard interrupt pin is falling-edge-sensitive only. With MODEK clear, a vector fetch or software clear immediately clears the keyboard interrupt request.

Reset clears the keyboard interrupt request and the MODEK bit, clearing the interrupt request even if a keyboard interrupt pin stays at 0.

The keyboard flag bit (KEYF) in the keyboard status and control register can be used to see if a pending interrupt exists. The KEYF bit is not affected by the keyboard interrupt mask bit (IMASKK) which makes it useful in applications where polling is preferred.

To determine the logic level on a keyboard interrupt pin, disable the pull-up device, use the data direction register to configure the pin as an input and then read the data register.

NOTE

Setting a keyboard interrupt enable bit (KBIE_x) forces the corresponding keyboard interrupt pin to be an input, overriding the data direction register. However, the data direction register bit must be a 0 for software to read the pin.

12.4.1 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pull-up to reach a logic 1. Therefore a false interrupt can occur as soon as the pin is enabled.

To prevent a false interrupt on keyboard initialization:

1. Mask keyboard interrupts by setting the IMASKK bit in the keyboard status and control register.
2. Enable the KBI pins by setting the appropriate KBIE_x bits in the keyboard interrupt enable register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in the data direction register A.
2. Write 1s to the appropriate port A data register bits.
3. Enable the KBI pins by setting the appropriate KBIE_x bits in the keyboard interrupt enable register.

12.5 Keyboard Interrupt Registers

Two registers control the operation of the keyboard interrupt module:

- Keyboard status and control register
- Keyboard interrupt enable register

13.3.7 COPRS (COP Rate Select)

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

Address:	\$001F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	COPRS	R	R	LVID	R	SSREC	STOP	COPD
Write:								
Reset:	0	0	0	0	0	0	0	0
	<div style="border: 1px solid black; padding: 2px; display: inline-block;">R</div> = Reserved							

Figure 13-2. Configuration Register 1 (CONFIG1)

COPRS — COP Rate Select Bit

COPRS selects the COP timeout period. Reset clears COPRS.

1 = COP timeout period is $8176 \times 2\text{OSCOU}$ cycles

0 = COP timeout period is $262,128 \times 2\text{OSCOU}$ cycles

COPD — COP Disable Bit

COPD disables the COP module.

1 = COP module disabled

0 = COP module enabled

13.4 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.

Address:	\$FFFF							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 13-3. COP Control Register (COPCTL)

13.5 Interrupts

The COP does not generate CPU interrupt requests.

13.6 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.

13.7 Low-Power Modes

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

15.4.2 Break Address Registers

The break address registers contain the high and low bytes of the desired breakpoint address. Reset clears the break address registers.

Address: \$FE0C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 15	14	13	12	11	10	9	Bit 8
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 15-4. Break Address Register High (BRKH)

Address: \$FE0D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	6	5	4	3	2	1	Bit 0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 15-5. Break Address Register Low (BRKL)

15.4.3 Break Status Register

The break status register contains a flag to indicate that a break caused an exit from wait mode.

Address: \$FE00

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R	R	R	R	R	R	SBSW	R
Write:							Note ⁽¹⁾	
Reset:							0	

R = Reserved 1. Writing a zero clears SBSW.

Figure 15-6. Break Status Register (BSR)

SBSW — SIM Break Stop/Wait

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.

- 1 = Wait mode was exited by break interrupt
- 0 = Wait mode was not exited by break interrupt

16.8 3V DC Electrical Characteristics

Table 16-7. DC Electrical Characteristics (3V)

Characteristic ⁽¹⁾	Symbol	Min	Typ ⁽²⁾	Max	Unit
Output high voltage ($I_{LOAD} = -1.0\text{mA}$) PTA0–PTA6, PTB0–PTB7, PTD0–PTD7	V_{OH}	$V_{DD} - 0.4$	—	—	V
Output low voltage ($I_{LOAD} = 0.8\text{mA}$) PTA6, PTB0–PTB7, PTD0, PTD1, PTD4, PTD5	V_{OL}	—	—	0.4	V
Output low voltage ($I_{LOAD} = 20\text{mA}$) PTD6, PTD7	V_{OL}	—	—	0.5	V
LED drives ($V_{OL} = 1.8\text{V}$) PTA0–PTA5, PTD2, PTD3, PTD6, PTD7	I_{OL}	3	6	10	mA
Input high voltage PTA0–PTA6, PTB0–PTB7, PTD0–PTD7, \overline{RST} , \overline{IRQ} , OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage PTA0–PTA6, PTB0–PTB7, PTD0–PTD7, \overline{RST} , \overline{IRQ} , OSC1	V_{IL}	V_{SS}	—	$0.3 \times V_{DD}$	V
V_{DD} supply current, $f_{OP} = 2\text{MHz}$ Run ⁽³⁾ MC68HC908JL3E/JK3E/JK1E MC68HRC908JL3E/JK3E/JK1E Wait ⁽⁴⁾ MC68HC908JL3E/JK3E/JK1E MC68HRC908JL3E/JK3E/JK1E Stop ⁽⁵⁾ (–40°C to 85°C) MC68HC908JL3E/JK3E/JK1E MC68HRC908JL3E/JK3E/JK1E	I_{DD}	— — — — — —	3 1.5 1.5 0.2 1 1	3.5 2 2 0.3 5 5	mA mA mA mA μA μA
Digital I/O ports Hi-Z leakage current	I_{IL}	—	—	± 10	μA
Input current	I_{IN}	—	—	± 1	μA
Capacitance Ports (as input or output)	C_{OUT} C_{IN}	— —	— —	12 8	pF
POR rearm voltage ⁽⁶⁾	V_{POR}	0	—	100	mV
POR rise time ramp rate ⁽⁷⁾	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	$1.5 \times V_{DD}$	—	8.5	V
Pullup resistors ⁽⁸⁾ PTD6, PTD7 \overline{RST} , \overline{IRQ} , PTA0–PTA6	R_{PU1} R_{PU2}	1.8 16	3.3 26	4.8 36	k Ω k Ω

Table continued on next page

NOTES:

1. POSITIONAL TOLERANCE OF LEADS, SHALL BE WITHIN 0.25 MM (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION DOES NOT INCLUDE MOLD FLASH.
4. 710-01 OBSOLETE, NEW STD 710-02.
5. CONTROLLING DIMENSION: INCH

DIM	INCH		MILLIMETER		DIM	INCH		MILLIMETER						
	MIN	MAX	MIN	MAX		MIN	MAX	MIN	MAX					
A	1.435	1.465	36.45	37.21										
B	0.540	0.560	13.72	14.22										
C	0.155	0.200	3.94	5.08										
D	0.014	0.022	0.36	0.56										
F	0.040	0.060	1.02	1.52										
G	0.100	BSC	2.54	BSC										
H	0.065	0.085	1.65	2.16										
J	0.008	0.015	0.20	0.38										
K	0.115	0.135	2.92	3.43										
L	0.600	BSC	15.24	BSC										
M	0°	15°	0°	15°										
N	0.020	0.040	0.51	1.02										
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.				MECHANICAL OUTLINE						PRINT VERSION NOT TO SCALE				
TITLE: 28 LD PDIP										DOCUMENT NO: 98ASB42390B			REV: D	
					CASE NUMBER: 710-02			24 MAY 2005						
					STANDARD: NON-JEDEC									



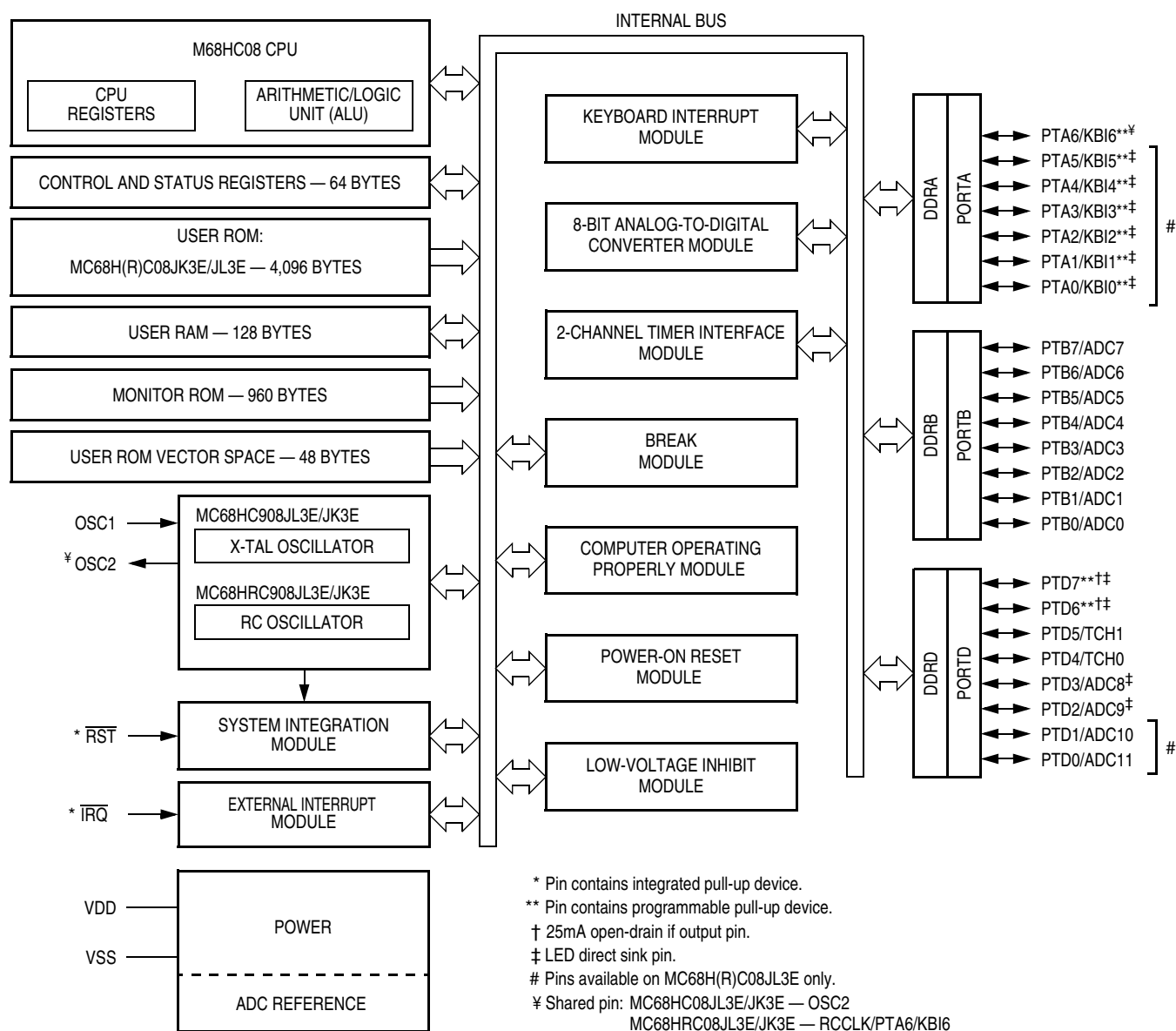


Figure B-1. MC68H(R)C08JL3E/JK3E Block Diagram

B.3 Memory Map

The MC68H(R)C08JL3E/JK3E has 4,096 bytes of user ROM from \$EC00 to \$FBFF, and 48 bytes of user ROM vectors from \$FFD0 to \$FFFF. On the MC68H(R)C908JL3E/JK3E, these memory locations are Flash memory.

Figure B-2 shows the memory map of the MC68H(R)C08JL3E/JK3E.

\$0000 ↓ \$003F	I/O REGISTERS 64 BYTES
\$0040 ↓ \$007F	RESERVED 64 BYTES
\$0080 ↓ \$00FF	RAM 128 BYTES
\$0100 ↓ \$EBFF	UNIMPLEMENTED 60,160 BYTES
\$EC00 ↓ \$FBFF	ROM MC68H(R)C08JL3E/JK3E 4,096 BYTES
\$FC00 ↓ \$FDFF	MONITOR ROM 512 BYTES
\$FE00	BREAK STATUS REGISTER (BSR)
\$FE01	RESET STATUS REGISTER (RSR)
\$FE02	RESERVED (UBAR)
\$FE03	BREAK FLAG CONTROL REGISTER (BFCR)
\$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
\$FE05	INTERRUPT STATUS REGISTER 2 (INT2)
\$FE06	INTERRUPT STATUS REGISTER 3 (INT3)
\$FE07	RESERVED
\$FE08	RESERVED
\$FE09	RESERVED
\$FE0A	RESERVED
\$FE0B	RESERVED
\$FE0C	BREAK ADDRESS HIGH REGISTER (BRKH)
\$FE0D	BREAK ADDRESS LOW REGISTER (BRKL)
\$FE0E	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
\$FE0F	RESERVED
\$FE10 ↓ \$FFCF	MONITOR ROM 448 BYTES
\$FFD0 ↓ \$FFFF	USER ROM VECTORS 48 BYTES

Figure B-2. MC68H(R)C08JL3E/JK3E Memory Map

B.7 Electrical Specifications

Electrical specifications for the MC68H(R)C908JL3E/JK3E apply to the MC68H(R)C08JL3E/JK3E, except for the parameters indicated below.

B.7.1 DC Electrical Characteristics

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

Characteristic ⁽¹⁾	Symbol	Min	Typ ⁽²⁾	Max	Unit
V _{DD} supply current, f _{OP} = 4 MHz					
Run ⁽³⁾					
MC68HC08JL3E/JK3E		—	9	11	mA
MC68HRC08JL3E/JK3E		—	4.3	5	mA
Wait ⁽⁴⁾					
MC68HC08JL3E/JK3E		—	5.5	6.5	mA
MC68HRC08JL3E/JK3E		—	0.8	1.5	mA
Stop ⁽⁵⁾	I _{DD}				
(–40°C to 85°C)					
MC68HC08JL3E/JK3E		—	1.8	5	μA
MC68HRC08JL3E/JK3E		—	1.8	5	μA
(–40°C to 125°C)					
MC68HC08JL3E/JK3E		—	5	10	μA
MC68HRC08JL3E/JK3E		—	5	10	μA
Pullup resistors ⁽⁶⁾					
PTD6, PTD7	R _{PU1}	1.8	4.3	4.8	kΩ
RST, IRQ, PTA0–PTA6	R _{PU2}	16	31	36	kΩ

1. V_{DD} = 4.5 to 5.5 Vdc, V_{SS} = 0 Vdc, T_A = T_L to T_H, unless otherwise noted.

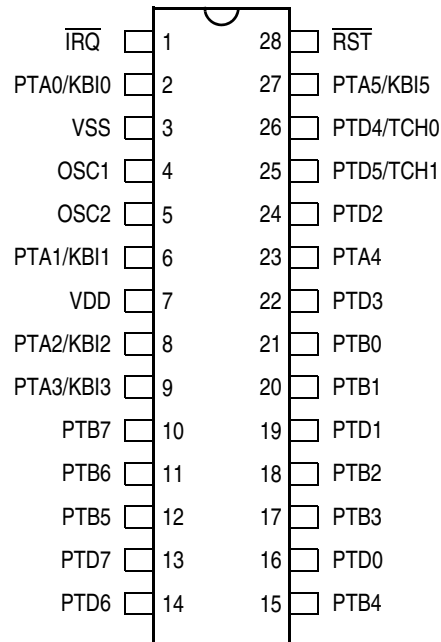
2. Typical values reflect average measurements at midpoint of voltage range, 25 °C only.

3. Run (operating) I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects run I_{DD}. Measured with all modules enabled.

4. Wait I_{DD} measured using external square wave clock source (f_{OP} = 4 MHz). All inputs 0.2V from rail. No dc loads. Less than 100 pF on all outputs. C_L = 20 pF on OSC2. All ports configured as inputs. OSC2 capacitance linearly affects wait I_{DD}.

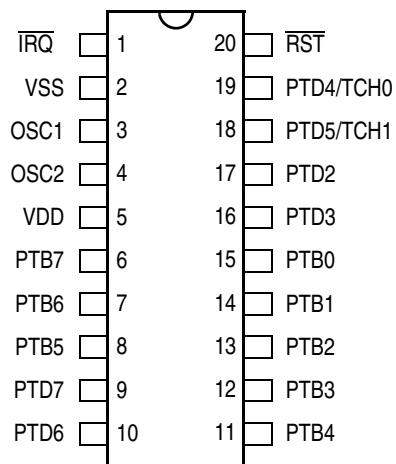
5. Stop I_{DD} measured with OSC1 grounded; no port pins sourcing current. LVI is disabled.

6. R_{PU1} and R_{PU2} are measured at V_{DD} = 5.0V.



MC68HC908KL3E

Figure C-2. 28-Pin PDIP/SOIC Pin Assignment



Pins not available on 20-pin packages	
PTA0/KBI0	PTD0
PTA1/KBI1	PTD1
PTA2/KBI2	
PTA3/KBI3	
PTA4/KBI4	
PTA5/KBI5	
Internal pads are unconnected.	

MC68HC908KK3E

Figure C-3. 20-Pin PDIP/SOIC Pin Assignment