

Welcome to [E-XFL.COM](https://www.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	LVD, POR, PWM
Number of I/O	13
Program Memory Size	4KB (4K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	128 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 4x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Through Hole
Package / Case	16-DIP (0.300", 7.62mm)
Supplier Device Package	16-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908qy4vpe

Chapter 1

General Description

1.1 Introduction

The MC68HC908QY4 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). The M68HC08 Family is a Complex Instruction Set Computer (CISC) with a Von Neumann architecture. All MCUs in the family use the enhanced M68HC08 central processor unit (CPU08) and are available with a variety of modules, memory sizes and types, and package types.

Table 1-1. Summary of Device Variations

Device	FLASH Memory Size	Analog-to-Digital Converter	Pin Count
MC68HC908QT1	1536 bytes	—	8 pins
MC68HC908QT2	1536 bytes	4 ch, 8 bit	8 pins
MC68HC908QT4	4096 bytes	4 ch, 8 bit	8 pins
MC68HC908QY1	1536 bytes	—	16 pins
MC68HC908QY2	1536 bytes	4 ch, 8 bit	16 pins
MC68HC908QY4	4096 bytes	4 ch, 8 bit	16 pins

1.2 Features

Features include:

- High-performance M68HC08 CPU core
- Fully upward-compatible object code with M68HC05 Family
- 5-V and 3-V operating voltages (V_{DD})
- 8-MHz internal bus operation at 5 V, 4-MHz at 3 V
- Trimmable internal oscillator
 - 3.2 MHz internal bus operation
 - 8-bit trim capability allows 0.4% accuracy⁽¹⁾
 - $\pm 25\%$ untrimmed
- Auto wakeup from STOP capability
- Configuration (CONFIG) register for MCU configuration options, including:
 - Low-voltage inhibit (LVI) trip point
- In-system FLASH programming
- FLASH security⁽²⁾

1. The oscillator frequency is guaranteed to $\pm 5\%$ over temperature and voltage range after trimming.

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

Memory

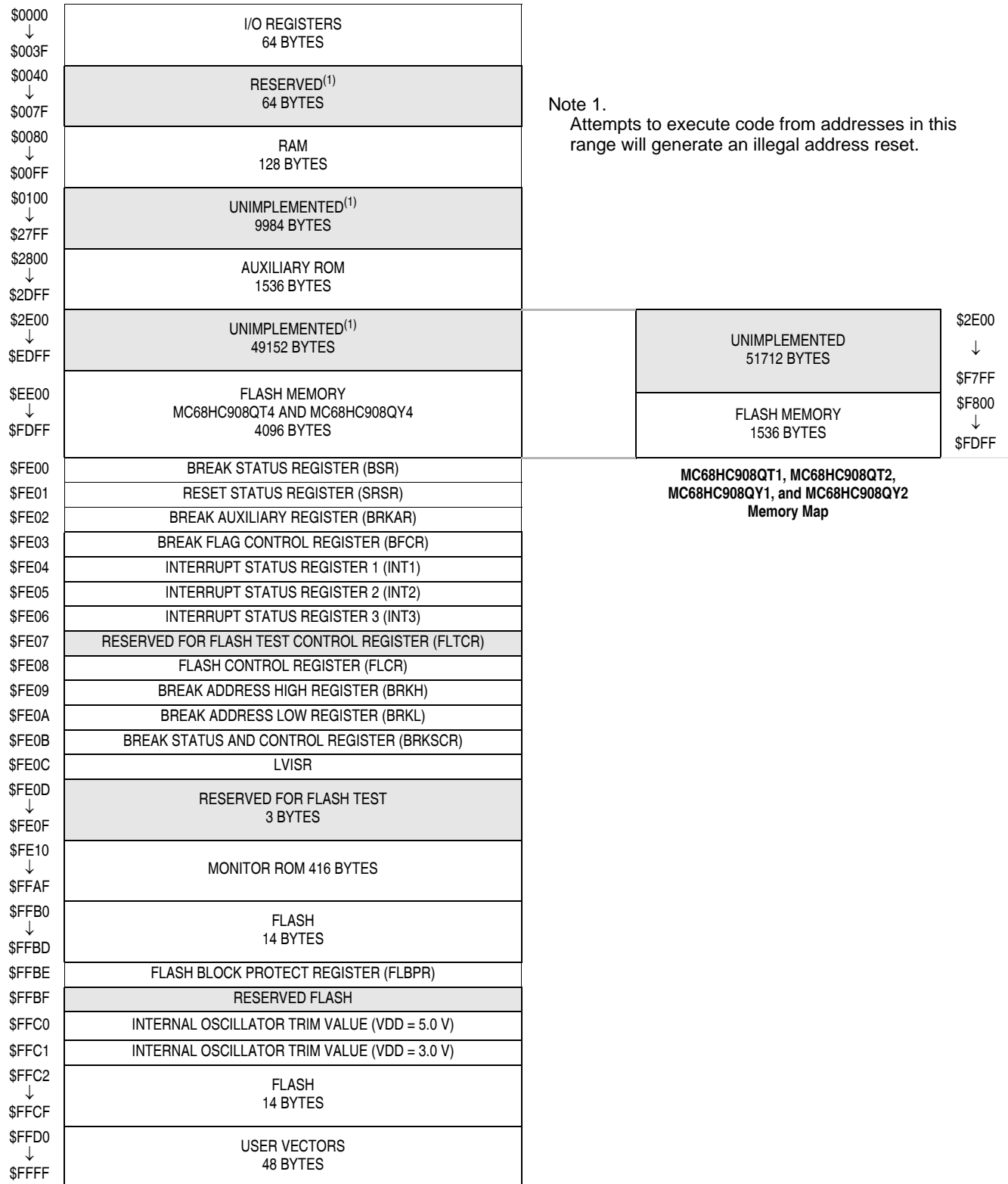


Figure 2-1. Memory Map

2.4 Input/Output (I/O) Section

Addresses \$0000–\$003F, shown in [Figure 2-2](#), contain most of the control, status, and data registers. Additional I/O registers have these addresses:

- \$FE00 — Break status register, BSR
- \$FE01 — Reset status register, SRSR
- \$FE02 — Break auxiliary register, BRKAR
- \$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 — FLASH control register, FLCR
- \$FE09 — Break address register high, BRKH
- \$FE0A — Break address register low, BRKL
- \$FE0B — Break status and control register, BRKSCR
- \$FE0C — LVI status register, LVISR
- \$FE0D — Reserved
- \$FFBE — FLASH block protect register, FLBPR
- \$FFC0 — Internal OSC trim value (factory programmed, VDD = 5.0 V)
- \$FFC1 — Internal OSC trim value (factory programmed, VDD = 3.0 V)
- \$FFFF — COP control register, COPCTL

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0000	Port A Data Register (PTA) See page 98.	Read:	R	AWUL	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
		\$0001	Port B Data Register (PTB) See page 100.	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2
Write:										
Reset:	Unaffected by reset									
\$0002	Unimplemented									
\$0003	Unimplemented									
\$0004	Data Direction Register A (DDRA) See page 98.	Read:	R	R	DDRA5	DDRA4	DDRA3	0	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
		\$0005	Data Direction Register B (DDRB) See page 101.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2
Write:										
Reset:	0			0	0	0	0	0	0	0
				= Unimplemented			R	= Reserved	U = Unaffected	

Figure 2-2. Control, Status, and Data Registers (Sheet 1 of 5)

3.7.2 ADC Data Register

One 8-bit result register is provided. This register is updated each time an ADC conversion completes.

Address: \$003E

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0
Write:								
Reset:	Indeterminate after reset							


 = Unimplemented

Figure 3-4. ADC Data Register (ADR)

3.7.3 ADC Input Clock Register

This register selects the clock frequency for the ADC.

Address: \$003F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:				0	0	0	0	0
Write:	ADIV2	ADIV1	ADIV0					
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 3-5. ADC Input Clock Register (ADICLK)

ADIV2–ADIV0 — ADC Clock Prescaler Bits

ADIV2, ADIV1, and ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. [Table 3-2](#) shows the available clock configurations. The ADC clock frequency should be set between $f_{\text{ADIC(MIN)}}$ and $f_{\text{ADIC(MAX)}}$. The analog input level should remain stable for the entire conversion time (maximum = 17 ADC clock cycles).

Table 3-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	Bus clock ÷ 1
0	0	1	Bus clock ÷ 2
0	1	0	Bus clock ÷ 4
0	1	1	Bus clock ÷ 8
1	X	X	Bus clock ÷ 16

X = don't care

Chapter 4

Auto Wakeup Module (AWU)

4.1 Introduction

This section describes the auto wakeup module (AWU). The AWU generates a periodic interrupt during stop mode to wake the part up without requiring an external signal. [Figure 4-1](#) is a block diagram of the AWU.

4.2 Features

Features of the auto wakeup module include:

- One internal interrupt with separate interrupt enable bit, sharing the same keyboard interrupt vector and keyboard interrupt mask bit
- Exit from low-power stop mode without external signals
- Selectable timeout periods
- Dedicated low-power internal oscillator separate from the main system clock sources

4.3 Functional Description

The function of the auto wakeup logic is to generate periodic wakeup requests to bring the microcontroller unit (MCU) out of stop mode. The wakeup requests are treated as regular keyboard interrupt requests, with the difference that instead of a pin, the interrupt signal is generated by an internal logic.

Writing the AWUIE bit in the keyboard interrupt enable register enables or disables the auto wakeup interrupt input (see [Figure 4-1](#)). A logic 1 applied to the AWUIREQ input with auto wakeup interrupt request enabled, latches an auto wakeup interrupt request.

Auto wakeup latch, AWUL, can be read directly from the bit 6 position of port A data register (PTA). This is a read-only bit which is occupying an empty bit position on PTA. No PTA associated registers, such as PTA6 data direction or PTA6 pullup exist for this bit.

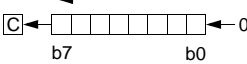
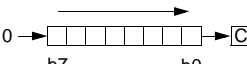
Entering stop mode will enable the auto wakeup generation logic. An internal RC oscillator (exclusive for the auto wakeup feature) drives the wakeup request generator. Once the overflow count is reached in the generator counter, a wakeup request, AWUIREQ, is latched and sent to the KBI logic. See [Figure 4-1](#).

Wakeup interrupt requests will only be serviced if the associated interrupt enable bit, AWUIE, in KBIER is set. The AWU shares the keyboard interrupt vector.

The overflow count can be selected from two options defined by the COPRS bit in CONFIG1. This bit was “borrowed” from the computer operating properly (COP) using the fact that the COP feature is idle (no MCU clock available) in stop mode. The typical values of the periodic wakeup request are (at room temperature):

- COPRS = 0: 650 ms @ 5 V, 875 ms @ 3 V
- COPRS = 1: 16 ms @ 5 V, 22 ms @ 3 V

Table 7-1. Instruction Set Summary (Sheet 4 of 6)

Source Form	Operation	Description	Effect on CCR					Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z				
JMP <i>opr</i> JMP <i>opr</i> JMP <i>opr</i> ,X JMP <i>opr</i> ,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 <i>opr</i> JSR <i>opr</i> JSR <i>opr</i> ,X JSR <i>opr</i> ,X JSR ,X	Jump to Subroutine	PC ← (PC) + <i>n</i> (<i>n</i> = 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 # <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> LDA <i>opr</i> ,X LDA <i>opr</i> ,X LDA ,X LDA <i>opr</i> ,SP LDA <i>opr</i> ,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 # <i>opr</i> LDHX <i>opr</i>	Load H:X from M	H:X ← (M:M + 1)	0	–	–	↑	↑	IMM DIR	45 55	ii jj dd	3 4
LDX # <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> LDX <i>opr</i> ,X LDX <i>opr</i> ,X LDX ,X LDX <i>opr</i> ,SP LDX <i>opr</i> ,SP	Load X from M	X ← (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	AE BE CE DE EE FE 9EEE 9EDE	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
LSL <i>opr</i> LSLA LSLX LSL <i>opr</i> ,X LSL ,X LSL <i>opr</i> ,SP	Logical Shift Left (Same as ASL)		↑	–	–	↑	↑	DIR INH INH IX1 IX SP1	38 48 58 68 78 9E68	dd ff ff	4 1 1 4 3 5
LSR <i>opr</i> LSRA LSRX LSR <i>opr</i> ,X LSR ,X LSR <i>opr</i> ,SP	Logical Shift Right		↑	–	–	0	↑	DIR INH INH IX1 IX SP1	34 44 54 64 74 9E64	dd ff ff ff	4 1 1 4 3 5
MOV <i>opr</i> , <i>opr</i> MOV <i>opr</i> ,X+ MOV # <i>opr</i> , <i>opr</i> MOV X+, <i>opr</i>	Move	(M) _{Destination} ← (M) _{Source} H:X ← (H:X) + 1 (IX+D, DIX+)	0	–	–	↑	↑	DD DIX+ IMD IX+D	4E 5E 6E 7E	dd dd dd ii dd dd	5 4 4 4
MUL	Unsigned multiply	X:A ← (X) × (A)	–	0	–	–	–	INH	42		5
NEG <i>opr</i> NEGA NEGX NEG <i>opr</i> ,X NEG ,X NEG <i>opr</i> ,SP	Negate (Two's Complement)	M ← –(M) = \$00 – (M) A ← –(A) = \$00 – (A) X ← –(X) = \$00 – (X) M ← –(M) = \$00 – (M) M ← –(M) = \$00 – (M)	↑	–	–	↑	↑	DIR INH INH IX1 IX SP1	30 40 50 60 70 9E60	dd ff ff ff	4 1 1 4 3 5
NOP	No Operation	None	–	–	–	–	–	INH	9D		1
NSA	Nibble Swap A	A ← (A[3:0]:A[7:4])	–	–	–	–	–	INH	62		3
ORA # <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ORA <i>opr</i> ,X ORA <i>opr</i> ,X ORA ,X ORA <i>opr</i> ,SP ORA <i>opr</i> ,SP	Inclusive OR A and M	A ← (A) (M)	0	–	–	↑	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	AA BA CA DA EA FA 9EEA 9EDA	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
PSHA	Push A onto Stack	Push (A); SP ← (SP) – 1	–	–	–	–	–	INH	87		2
PSHH	Push H onto Stack	Push (H); SP ← (SP) – 1	–	–	–	–	–	INH	8B		2
PSHX	Push X onto Stack	Push (X); SP ← (SP) – 1	–	–	–	–	–	INH	89		2

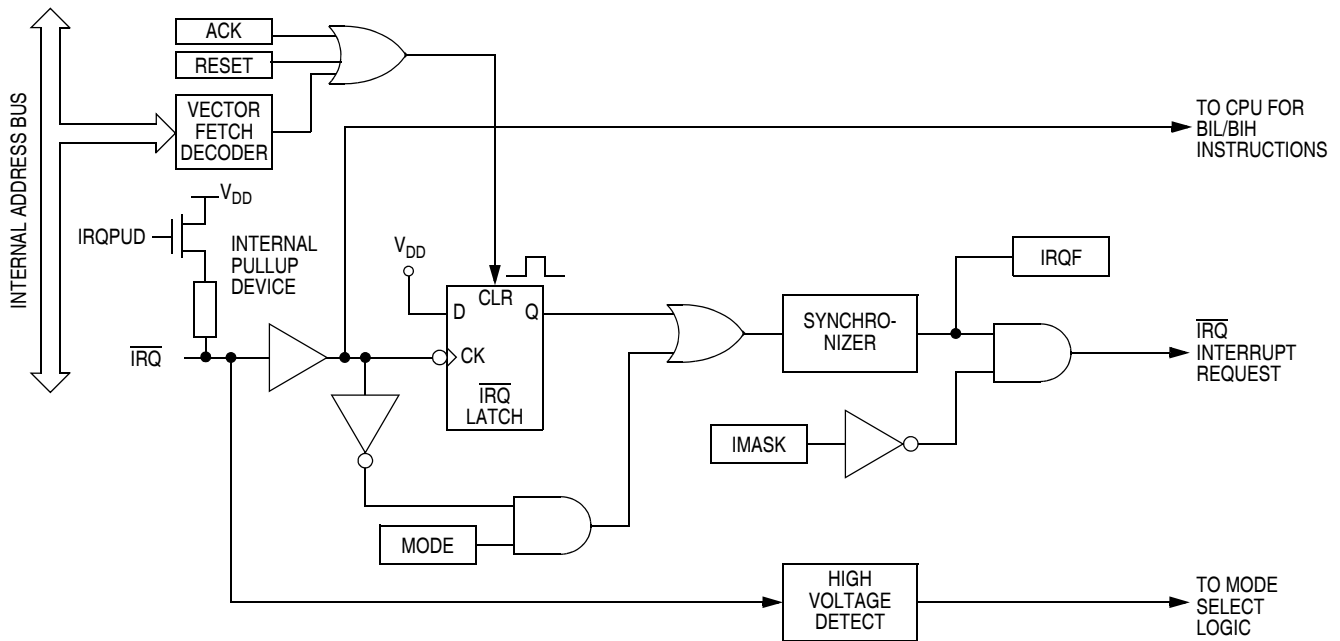


Figure 8-2. IRQ Module Block Diagram

8.3.1 MODE = 1

If the MODE bit is set, the $\overline{\text{IRQ}}$ pin is both falling edge sensitive and low level sensitive. With MODE set, both of the following actions must occur to clear the $\overline{\text{IRQ}}$ interrupt request:

- Return of the $\overline{\text{IRQ}}$ pin to a high level. As long as the $\overline{\text{IRQ}}$ pin is low, the IRQ request remains active.
- IRQ vector fetch or software clear. An IRQ vector fetch generates an interrupt acknowledge signal to clear the IRQ latch. Software generates the interrupt acknowledge signal by writing a 1 to ACK in INTSCR. The ACK bit is useful in applications that poll the $\overline{\text{IRQ}}$ pin and require software to clear the IRQ latch. Writing to ACK prior to leaving an interrupt service routine can also prevent spurious interrupts due to noise. Setting ACK does not affect subsequent transitions on the $\overline{\text{IRQ}}$ pin. A falling edge that occurs after writing to ACK latches another interrupt request. If the IRQ mask bit, IMASK, is clear, the CPU loads the program counter with the IRQ vector address.

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

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

8.3.2 MODE = 0

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

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

NOTE

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



External Interrupt (IRQ)

10.4 LVI Status Register

The LVI status register (LVISR) indicates if the V_{DD} voltage was detected below the V_{TRIPF} level while LVI resets have been disabled.

Address: \$FE0C

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

 = Unimplemented
 R = Reserved

Figure 10-2. LVI Status Register (LVISR)

LVIOUT — LVI Output Bit

This read-only flag becomes set when the V_{DD} voltage falls below the V_{TRIPF} trip voltage and is cleared when V_{DD} voltage rises above V_{TRIPR} . The difference in these threshold levels results in a hysteresis that prevents oscillation into and out of reset (see [Table 10-1](#)). Reset clears the LVIOUT bit.

Table 10-1. LVIOUT Bit Indication

V_{DD}	LVIOUT
$V_{DD} > V_{TRIPR}$	0
$V_{DD} < V_{TRIPF}$	1
$V_{TRIPF} < V_{DD} < V_{TRIPR}$	Previous value

10.5 LVI Interrupts

The LVI module does not generate interrupt requests.

10.6 Low-Power Modes

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

10.6.1 Wait Mode

If enabled, the LVI module remains active in wait mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of wait mode.

10.6.2 Stop Mode

When the LVIPWRD bit in the configuration register is cleared and the LVISTOP bit in the configuration register is set, the LVI module remains active in stop mode. If enabled to generate resets, the LVI module can generate a reset and bring the MCU out of stop mode.



Input/Output Ports (PORTS)

PTAPUE[5:0] — Port A Input Pullup Enable Bits

These read/write bits are software programmable to enable pullup devices on port A pins.

1 = Corresponding port A pin configured to have internal pull if its DDRA bit is set to 0

0 = Pullup device is disconnected on the corresponding port A pin regardless of the state of its DDRA bit

Table 12-1 summarizes the operation of the port A pins.

Table 12-1. Port A Pin Functions

PTAPUE Bit	DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
				Read/Write	Read	Write
1	0	X ⁽¹⁾	Input, V _{DD} ⁽²⁾	DDRA5–DDRA0	Pin	PTA5–PTA0 ⁽³⁾
0	0	X	Input, Hi-Z ⁽⁴⁾	DDRA5–DDRA0	Pin	PTA5–PTA0 ⁽³⁾
X	1	X	Output	DDRA5–DDRA0	PTA5–PTA0	PTA5–PTA0 ⁽⁵⁾

1. X = don't care
2. I/O pin pulled to V_{DD} by internal pullup.
3. Writing affects data register, but does not affect input.
4. Hi-Z = high impedance
5. Output does not apply to PTA2

12.3 Port B

Port B is an 8-bit general purpose I/O port. Port B is only available on the MC68HC908QY1, MC68HC908QY2, and MC68HC908QY4.

12.3.1 Port B Data Register

The port B data register (PTB) contains a data latch for each of the eight port B pins.

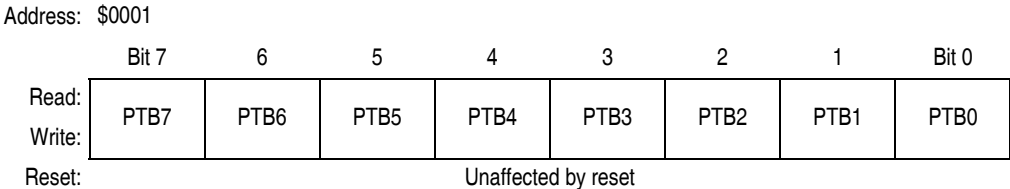


Figure 12-5. Port B Data Register (PTB)

PTB[7:0] — Port B Data Bits

These read/write bits are software programmable. Data direction of each port B pin is under the control of the corresponding bit in data direction register B. Reset has no effect on port B data.

Chapter 13

System Integration Module (SIM)

13.1 Introduction

This section describes the system integration module (SIM), which supports up to 24 external and/or internal interrupts. Together with the central processor unit (CPU), the SIM controls all microcontroller unit (MCU) activities. A block diagram of the SIM is shown in [Figure 13-1](#). The SIM is a system state controller that coordinates CPU and exception timing.

The SIM is responsible for:

- Bus clock generation and control for CPU and peripherals
 - Stop/wait/reset/break entry and recovery
 - Internal clock control
- Master reset control, including power-on reset (POR) and computer operating properly (COP) timeout
- Interrupt control:
 - Acknowledge timing
 - Arbitration control timing
 - Vector address generation
- CPU enable/disable timing

Table 13-1. Signal Name Conventions

Signal Name	Description
BUSCLKX4	Buffered clock from the internal, RC or XTAL oscillator circuit.
BUSCLKX2	The BUSCLKX4 frequency divided by two. This signal is again divided by two in the SIM to generate the internal bus clocks (bus clock = $BUSCLKX4 \div 4$).
Address bus	Internal address bus
Data bus	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/ \overline{W}	Read/write signal

13.4.2.2 Computer Operating Properly (COP) Reset

An input to the SIM is reserved for the COP reset signal. The overflow of the COP counter causes an internal reset and sets the COP bit in the SIM reset status register (SRSR). The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

To prevent a COP module time out, write any value to location \$FFFF. Writing to location \$FFFF clears the COP counter and stages 12–5 of the SIM counter. The SIM counter output, which occurs at least every 4080 BUSCLKX4 cycles, drives the COP counter. The COP should be serviced as soon as possible out of reset to guarantee the maximum amount of time before the first time out.

The COP module is disabled during a break interrupt with monitor mode when BDCOP bit is set in break auxiliary register (BRKAR).

13.4.2.3 Illegal Opcode Reset

The SIM decodes signals from the CPU to detect illegal instructions. An illegal instruction sets the ILOP bit in the SIM reset status register (SRSR) and causes a reset.

If the stop enable bit, STOP, in the mask option register is 0, the SIM treats the STOP instruction as an illegal opcode and causes an illegal opcode reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources.

13.4.2.4 Illegal Address Reset

An opcode fetch from an unmapped address generates an illegal address reset. The SIM verifies that the CPU is fetching an opcode prior to asserting the ILAD bit in the SIM reset status register (SRSR) and resetting the MCU. A data fetch from an unmapped address does not generate a reset. The SIM actively pulls down the $\overline{\text{RST}}$ pin for all internal reset sources. See [Figure 2-1. Memory Map](#) for memory ranges.

13.4.2.5 Low-Voltage Inhibit (LVI) Reset

The LVI asserts its output to the SIM when the V_{DD} voltage falls to the LVI trip voltage V_{TRIPF} . The LVI bit in the SIM reset status register (SRSR) is set, and the external reset pin ($\overline{\text{RST}}$) is held low while the SIM counter counts out 4096 BUSCLKX4 cycles after V_{DD} rises above V_{TRIPR} . Sixty-four BUSCLKX4 cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur. The SIM actively pulls down the ($\overline{\text{RST}}$) pin for all internal reset sources.

13.5 SIM Counter

The SIM counter is used by the power-on reset module (POR) and in stop mode recovery to allow the oscillator time to stabilize before enabling the internal bus (IBUS) clocks. The SIM counter also serves as a prescaler for the computer operating properly module (COP). The SIM counter uses 12 stages for counting, followed by a 13th stage that triggers a reset of SIM counters and supplies the clock for the COP module. The SIM counter is clocked by the falling edge of BUSCLKX4.

13.5.1 SIM Counter During Power-On Reset

The power-on reset module (POR) detects power applied to the MCU. At power-on, the POR circuit asserts the signal PORRST. Once the SIM is initialized, it enables the oscillator to drive the bus clock state machine.

13.6.3 Reset

All reset sources always have equal and highest priority and cannot be arbitrated.

13.6.4 Break Interrupts

The break module can stop normal program flow at a software programmable break point by asserting its break interrupt output. (See [Chapter 15 Development Support](#).) The SIM puts the CPU into the break state by forcing it to the SWI vector location. Refer to the break interrupt subsection of each module to see how each module is affected by the break state.

13.6.5 Status Flag Protection in Break Mode

The SIM controls whether status flags contained in other modules can be cleared during break mode. The user can select whether flags are protected from being cleared by properly initializing the break clear flag enable bit (BCFE) in the break flag control register (BFCR).

Protecting flags in break mode ensures that set flags will not be cleared while in break mode. This protection allows registers to be freely read and written during break mode without losing status flag information.

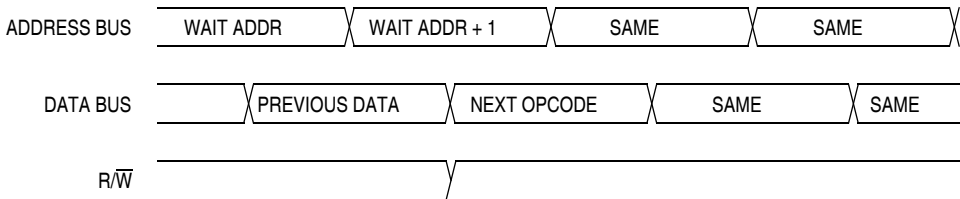
Setting the BCFE bit enables the clearing mechanisms. Once cleared in break mode, a flag remains cleared even when break mode is exited. Status flags with a two-step clearing mechanism — for example, a read of one register followed by the read or write of another — are protected, even when the first step is accomplished prior to entering break mode. Upon leaving break mode, execution of the second step will clear the flag as normal.

13.7 Low-Power Modes

Executing the WAIT or STOP instruction puts the MCU in a low power- consumption mode for standby situations. The SIM holds the CPU in a non-clocked state. The operation of each of these modes is described below. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur.

13.7.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. [Figure 13-14](#) shows the timing for wait mode entry.



NOTE: Previous data can be operand data or the WAIT opcode, depending on the last instruction.

Figure 13-14. Wait Mode Entry Timing

Chapter 14

Timer Interface Module (TIM)

14.1 Introduction

This section describes the timer interface module (TIM). The TIM is a two-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. [Figure 14-2](#) is a block diagram of the TIM.

14.2 Features

Features of the TIM include the following:

- Two input capture/output compare channels
 - Rising-edge, falling-edge, or any-edge input capture trigger
 - Set, clear, or toggle output compare action
- Buffered and unbuffered pulse width modulation (PWM) signal generation
- Programmable TIM clock input
 - 7-frequency internal bus clock prescaler selection
 - External TIM clock input
- Free-running or modulo up-count operation
- Toggle any channel pin on overflow
- TIM counter stop and reset bits

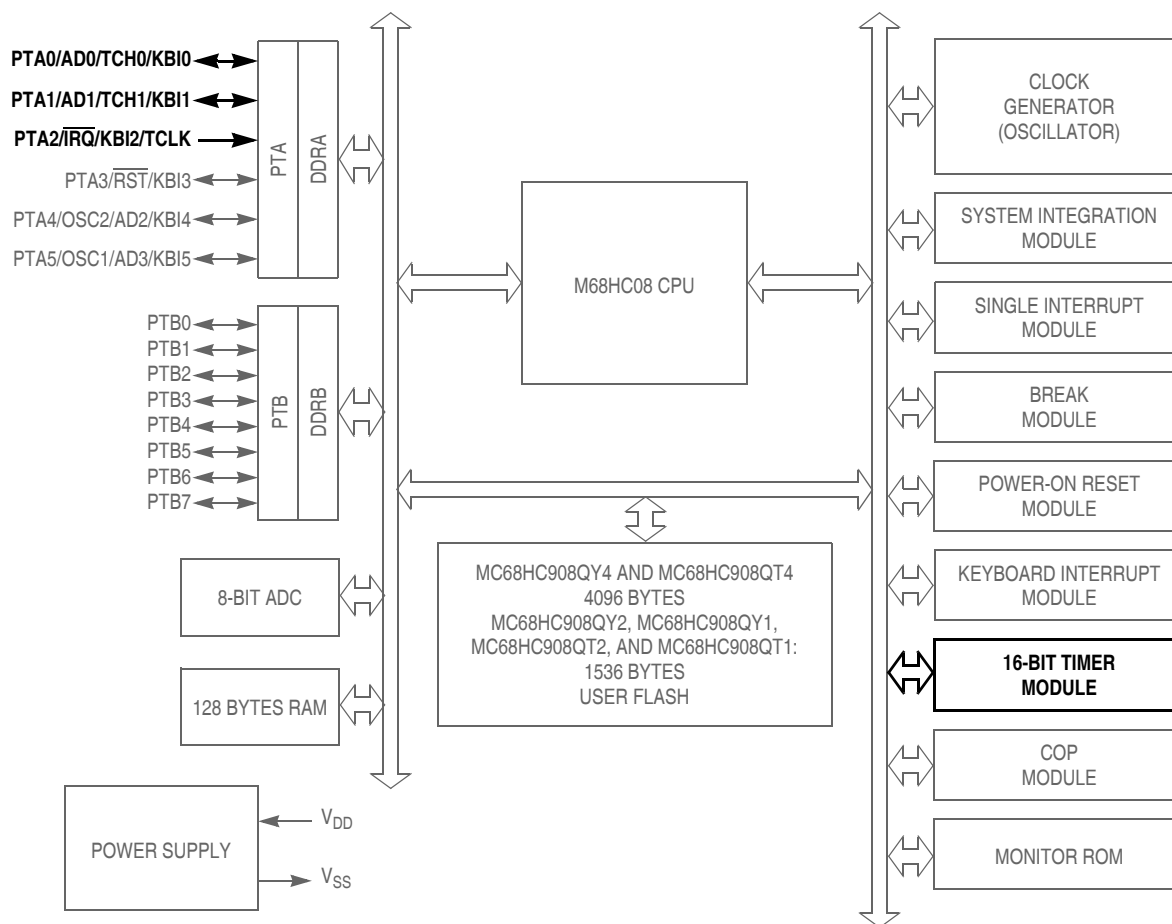
14.3 Pin Name Conventions

The TIM shares two input/output (I/O) pins with two port A I/O pins. The full names of the TIM I/O pins are listed in [Table 14-1](#). The generic pin name appear in the text that follows.

Table 14-1. Pin Name Conventions

TIM Generic Pin Names:	TCH0	TCH1	TCLK
Full TIM Pin Names:	PTA0/TCH0	PTA1/TCH1	PTA2/TCLK

Timer Interface Module (TIM)



RST, IRQ: Pins have internal (about 30K Ohms) pull up

PTA[0:5]: High current sink and source capability

PTA[0:5]: Pins have programmable keyboard interrupt and pull up

PTB[0:7]: Not available on 8-pin devices – MC68HC908QT1, MC68HC908QT2, and MC68HC908QT4 (see note in [12.1 Introduction](#))

ADC: Not available on the MC68HC908QY1 and MC68HC908QT1

Figure 14-1. Block Diagram Highlighting TIM Block and Pins

14.6 Wait Mode

The WAIT instruction puts the MCU in low power-consumption standby mode.

The TIM remains active after the execution of a WAIT instruction. In wait mode the TIM registers are not accessible by the CPU. Any enabled CPU interrupt request from the TIM can bring the MCU out of wait mode.

If TIM functions are not required during wait mode, reduce power consumption by stopping the TIM before executing the WAIT instruction.

14.7 TIM During Break Interrupts

A break interrupt stops the TIM counter.

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state. See [13.8.2 Break Flag Control Register](#).

To allow software to clear status bits during a break interrupt, write a 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a two-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

14.8 Input/Output Signals

Port A shares three of its pins with the TIM. Two TIM channel I/O pins are PTA0/TCH0 and PTA1/TCH1 and an alternate clock source is PTA2/TCLK.

14.8.1 TIM Clock Pin (PTA2/TCLK)

PTA2/TCLK is an external clock input that can be the clock source for the TIM counter instead of the prescaled internal bus clock. Select the PTA2/TCLK input by writing 1s to the three prescaler select bits, PS[2–0]. (See [14.9.1 TIM Status and Control Register](#).) When the PTA2/TCLK pin is the TIM clock input, it is an input regardless of port pin initialization.

14.8.2 TIM Channel I/O Pins (PTA0/TCH0 and PTA1/TCH1)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. PTA0/TCH0 can be configured as a buffered output compare or buffered PWM pin.

14.9 Input/Output Registers

The following I/O registers control and monitor operation of the TIM:

- TIM status and control register (TSC)
- TIM counter registers (TCNTH:TCNTL)
- TIM counter modulo registers (TMODH:TMODL)
- TIM channel status and control registers (TSC0 and TSC1)
- TIM channel registers (TCH0H:TCH0L and TCH1H:TCH1L)

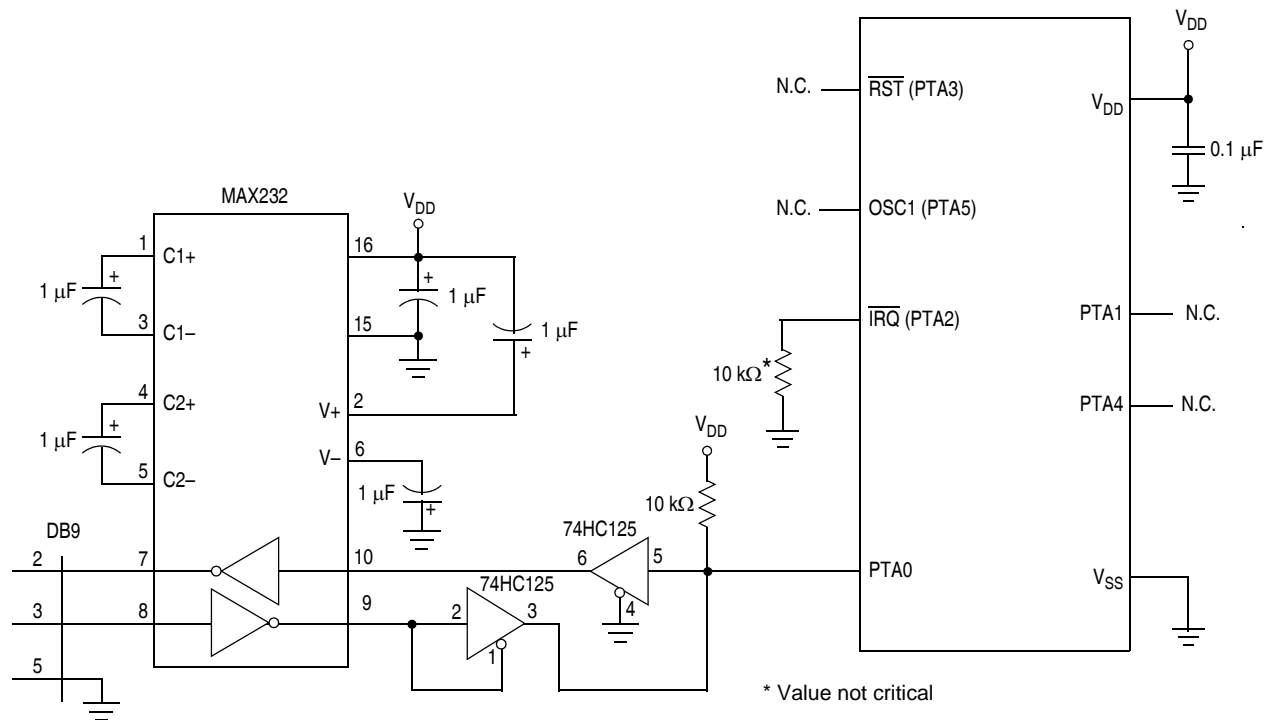


Figure 15-12. Monitor Mode Circuit (Internal Clock, No High Voltage)

Simple monitor commands can access any memory address. In monitor mode, the MCU can execute code downloaded into RAM by a host computer while most MCU pins retain normal operating mode functions. All communication between the host computer and the MCU is through the PTA0 pin. A level-shifting and multiplexing interface is required between PTA0 and the host computer. PTA0 is used in a wired-OR configuration and requires a pullup resistor.

The monitor code has been updated from previous versions of the monitor code to allow enabling the internal oscillator to generate the internal clock. This addition, which is enabled when $\overline{\text{IRQ}}$ is held low out of reset, is intended to support serial communication/programming at 9600 baud in monitor mode by using the internal oscillator, and the internal oscillator user trim value OSCTRIM (FLASH location \$FFC0, if programmed) to generate the desired internal frequency (3.2 MHz). Since this feature is enabled only when $\overline{\text{IRQ}}$ is held low out of reset, it cannot be used when the reset vector is programmed (i.e., the value is not \$FFFF) because entry into monitor mode in this case requires V_{TST} on $\overline{\text{IRQ}}$. The $\overline{\text{IRQ}}$ pin must remain low during this monitor session in order to maintain communication.

Table 15-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a power-on reset (POR) and will allow communication at 9600 baud provided one of the following sets of conditions is met:

- If \$FFFE and \$FFFF do not contain \$FF (programmed state):
 - The external clock is 9.8304 MHz
 - $\overline{\text{IRQ}} = V_{\text{TST}}$
- If \$FFFE and \$FFFF contain \$FF (erased state):
 - The external clock is 9.8304 MHz
 - $\overline{\text{IRQ}} = V_{\text{DD}}$ (this can be implemented through the internal $\overline{\text{IRQ}}$ pullup)
- If \$FFFE and \$FFFF contain \$FF (erased state):
 - $\overline{\text{IRQ}} = V_{\text{SS}}$ (internal oscillator is selected, no external clock required)

Chapter 17

Ordering Information and Mechanical Specifications

17.1 Introduction

This section contains order numbers for the MC68HC908QY1, MC68HC908QY2, MC68HC908QY4, MC68HC908QT1, MC68HC908QT2, and MC69HC908QT4. Dimensions are given for:

- 8-pin plastic dual in-line package (PDIP)
- 8-pin small outline integrated circuit (SOIC) package
- 8-pin dual flat no lead (DFN) package
- 16-pin PDIP
- 16-pin SOIC
- 16-pin thin shrink small outline package (TSSOP)

17.2 MC Order Numbers

Table 17-1. MC Order Numbers

MC Order Number	ADC	FLASH Memory	Package
MC908QY1	—	1536 bytes	16-pins PDIP, SOIC, and TSSOP
MC908QY2	Yes	1536 bytes	
MC908QY4	Yes	4096 bytes	
MC908QT1	—	1536 bytes	8-pins PDIP, SOIC, and DFN
MC908QT2	Yes	1536 bytes	
MC908QT4	Yes	4096 bytes	

Temperature and package designators:

- C = -40°C to +85°C
- V = -40°C to +105°C
- M = -40°C to +125°C
- P = Plastic dual in-line package (PDIP)
- DW = Small outline integrated circuit package (SOIC)
- DT = Thin shrink small outline package (TSSOP)
- FQ = Dual flat no lead (DFN)

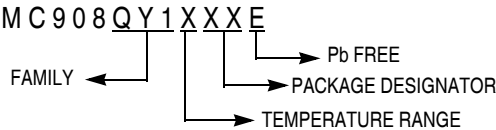


Figure 17-1. Device Numbering System

17.3 Package Dimensions

Refer to the following pages for detailed package dimensions.



NOTES:

- 1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M – 1994.
- 2. ALL DIMENSIONS ARE IN INCHES.
- 3. 626-03 TO 626-06 OBSOLETE. NEW STANDARD 626-07.

△ 4. DIMENSION TO CENTER OF LEAD WHEN FORMED PARALLEL.

△ 5. PACKAGE CONTOUR OPTIONAL (ROUND OR SQUARE CONERS).

STYLE 1:

PIN	1.	AC IN	5.	GROUND
	2.	DC + IN	6.	OUTPUT
	3.	DC – IN	7.	AUXILIARY
	4.	AC IN	8.	VCC

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: 8 LD PDIP			DOCUMENT NO: 98ASB42420B		REV: N
			CASE NUMBER: 626-06		19 MAY 2005
			STANDARD: NON-JEDEC		