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	I²C, IRSCI, SCI, SPI
Peripherals	LED, LVD, POR, PWM
Number of I/O	30
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	42-SDIP (0.600", 15.24mm)
Supplier Device Package	42-PDIP
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908ap8cbe

Chapter 14
Multi-Master IIC Interface (MMIIC)

Chapter 15
Analog-to-Digital Converter (ADC)

Chapter 1
General Description

1.1 Introduction

The MC68HC908AP64 is a member of the low-cost, high-performance M68HC08 Family of 8-bit microcontroller units (MCUs). 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	RAM Size (bytes)	FLASH Memory Size (bytes)
MC68HC908AP64	2,048	62,368
MC68HC908AP32	2,048	32,768
MC68HC908AP16	1,024	16,384
MC68HC908AP8	1,024	8,192

1.2 Features

Features of the MC68HC908AP64 include the following:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- Maximum internal bus frequency:
  - 8-MHz at 5V or 3V operating voltage
- Clock input options:
  - RC-oscillator
  - 32-kHz crystal-oscillator with 32MHz internal PLL
- User program FLASH memory with security[1] feature
  - 62,368 bytes for MC68HC908AP64
  - 32,768 bytes for MC68HC908AP32
  - 16,384 bytes for MC68HC908AP16
  - 8,192 bytes for MC68HC908AP8
- On-chip RAM
  - 2,048 bytes for MC68HC908AP64 and MC68HC908AP32
  - 1,024 bytes for MC68HC908AP16 and MC68HC908AP8
- Two 16-bit, 2-channel timer interface modules (TIM1 and TIM2) with selectable input capture, output compare, and PWM capability on each channel

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

- Timebase module
- Serial communications interface module 1 (SCI)
- Serial communications interface module 2 (SCI) with infrared (IR) encoder/decoder
- Serial peripheral interface module (SPI)
- System management bus (SMBus), version 1.0/1.1 (multi-master IIC bus)
- 8-channel, 10-bit analog-to-digital converter (ADC)
- $\overline{\text{IRQ1}}$ external interrupt pin with integrated pullup
- $\overline{\text{IRQ2}}$ external interrupt pin with programmable pullup
- 8-bit keyboard wakeup port with integrated pullup
- 32 general-purpose input/output (I/O) pins:
  - 31 shared-function I/O pins
  - 8 LED drivers (sink)
  - 6 × 25mA open-drain I/O with pullup
- Low-power design (fully static with stop and wait modes)
- Master reset pin (with integrated pullup) and power-on reset
- System protection features
  - Optional computer operating properly (COP) reset, driven by internal RC oscillator
  - Low-voltage detection with optional reset or interrupt
  - Illegal opcode detection with reset
  - Illegal address detection with reset
- 48-pin low quad flat pack (LQFP), 44-pin quad flat pack (QFP), and 42-pin shrink dual-in-line package (SDIP)
- Specific features of the MC68HC908AP64 in 42-pin SDIP are:
  - 30 general-purpose I/Os only
  - External interrupt on $\overline{\text{IRQ1}}$ only

Features of the CPU08 include the following:

- Enhanced HC05 programming model
- Extensive loop control functions
- 16 addressing modes (eight more than the HC05)
- 16-bit Index register and stack pointer
- Memory-to-memory data transfers
- Fast 8 × 8 multiply instruction
- Fast 16/8 divide instruction
- Binary-coded decimal (BCD) instructions
- Optimization for controller applications
- Efficient C language support

1.3 MCU Block Diagram

Figure 1-1 shows the structure of the MC68HC908AP64.

Table 5-1. CGMXCLK Clock Selection

OSCSEL1	OSCSEL0	CGMXCLK	OSC2 Pin	Comments
0	0	—	—	Not used
0	1	ICLK	f_{BUS}	Internal oscillator generates the CGMXCLK.
1	0	RCCLK	f_{BUS}	RC oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.
1	1	XCLK	Inverting output of X-TAL	X-tal oscillator generates the CGMXCLK. Internal oscillator is available after each POR or reset.

NOTE

The internal oscillator is a free running oscillator and is available after each POR or reset. It is turned-off in stop mode by setting the STOP_ICLKDIS bit in CONFIG2.

5.2.2 TBM Reference Clock Selection

The timebase module reference clock (OSCCLK) is selected by configuring two bits in the CONFIG2 register, at $001D. See Chapter 3 Configuration & Mask Option Registers (CONFIG & MOR).

Address: $001D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	STOP_ ICLKDIS	STOP_ RCLKEN	STOP_ XCLKEN	OSCCLK1	OSCCLK0	0	0	SCIBD- SRC
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 5-3. Configuration Register 2 (CONFIG2)

Table 5-2. Timebase Module Reference Clock Selection

OSCCLK1	OSCCLK0	Timebase Clock Source
0	0	Internal oscillator (ICLK)
0	1	RC oscillator (RCCLK)
1	0	X-tal oscillator (XCLK)
1	1	Not used

NOTE

The RCCLK or XCLK is only available if that clock is selected as the CGM reference clock, whereas the ICLK is always available.

1. Choose the desired bus frequency, f_{BUSDES}, or the desired VCO frequency, $f_{VCLKDES}$; and then solve for the other.
   The relationship between f_{BUS} and f_{VCLK} is governed by the equation:

$$f_{VCLK} = 2^P \times f_{CGMPCLK} = 2^P \times 4 \times f_{BUS}$$

   where P is the power of two multiplier, and can be 0, 1, 2, or 3
2. Choose a practical PLL reference frequency, f_{RCLK}, and the reference clock divider, R. Typically, the reference is 32.768kHz and R = 1.

   Frequency errors to the PLL are corrected at a rate of f_{RCLK}/R. For stability and lock time reduction, this rate must be as fast as possible. The VCO frequency must be an integer multiple of this rate. The relationship between the VCO frequency, f_{VCLK}, and the reference frequency, f_{RCLK}, is

$$f_{VCLK} = \frac{2^P N}{R}(f_{RCLK})$$

   where N is the integer range multiplier, between 1 and 4095.

   In cases where desired bus frequency has some tolerance, choose f_{RCLK} to a value determined either by other module requirements (such as modules which are clocked by CGMXCLK), cost requirements, or ideally, as high as the specified range allows. See Chapter 22 Electrical Specifications.
   Choose the reference divider, R = 1.

   When the tolerance on the bus frequency is tight, choose f_{RCLK} to an integer divisor of f_{BUSDES}, and R = 1. If f_{RCLK} cannot meet this requirement, use the following equation to solve for R with practical choices of f_{RCLK}, and choose the f_{RCLK} that gives the lowest R.

$$R = \text{round}\left[R_{MAX} \times \left\{\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right) - \text{integer}\left(\frac{f_{VCLKDES}}{f_{RCLK}}\right)\right\}\right]$$

3. Calculate N:

$$N = \text{round}\left(\frac{R \times f_{VCLKDES}}{f_{RCLK} \times 2^P}\right)$$

4. Calculate and verify the adequacy of the VCO and bus frequencies f_{VCLK} and f_{BUS}.

$$f_{VCLK} = \frac{2^P N}{R}(f_{RCLK})$$

$$f_{BUS} = \frac{f_{VCLK}}{2^P \times 4}$$

Chapter 7
System Integration Module (SIM)

7.1 Introduction

This section describes the system integration module (SIM). Together with the CPU, the SIM controls all MCU activities. A block diagram of the SIM is shown in Figure 7-1. Figure 7-2 is a summary of the SIM input/output (I/O) registers. 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 COP timeout
- Interrupt control:
  - Acknowledge timing
  - Arbitration control timing
  - Vector address generation
- CPU enable/disable timing
- Modular architecture expandable to 128 interrupt sources

Table 7-1 shows the internal signal names used in this section.

Table 7-1. Signal Name Conventions

Signal Name	Description
ICLK	Internal oscillator clock
CGMXCLK	Selected oscillator clock from oscillator module
CGMVCLK, CGMPCLK	PLL output and the divided PLL output
CGMOUT	CGMPCLK-based or oscillator-based clock output from CGM module (Bus clock = CGMOUT ÷ 2)
IAB	Internal address bus
IDB	Internal data bus
PORRST	Signal from the power-on reset module to the SIM
IRST	Internal reset signal
R/$\overline{W}$	Read/write signal

7.4.3 SIM Counter and Reset States

External reset has no effect on the SIM counter. (See 7.6.2 Stop Mode for details.) The SIM counter is free-running after all reset states. (See 7.3.2 Active Resets from Internal Sources for counter control and internal reset recovery sequences.)

7.5 Exception Control

Normal, sequential program execution can be changed in three different ways:

- Interrupts:
  - Maskable hardware CPU interrupts
  - Non-maskable software interrupt instruction (SWI)
- Reset
- Break interrupts

7.5.1 Interrupts

At the beginning of an interrupt, the CPU saves the CPU register contents on the stack and sets the interrupt mask (I bit) to prevent additional interrupts. At the end of an interrupt, the RTI instruction recovers the CPU register contents from the stack so that normal processing can resume. Figure 7-8 shows interrupt entry timing, and Figure 7-9 shows interrupt recovery timing.

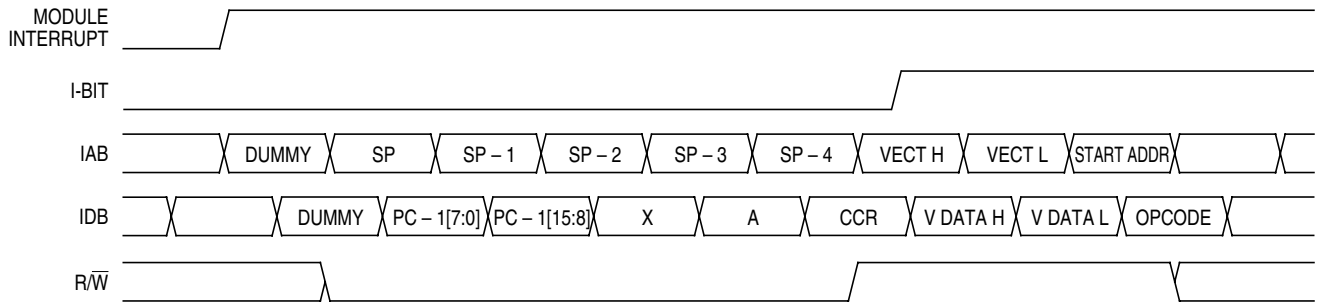


Figure 7-8. Interrupt Entry Timing

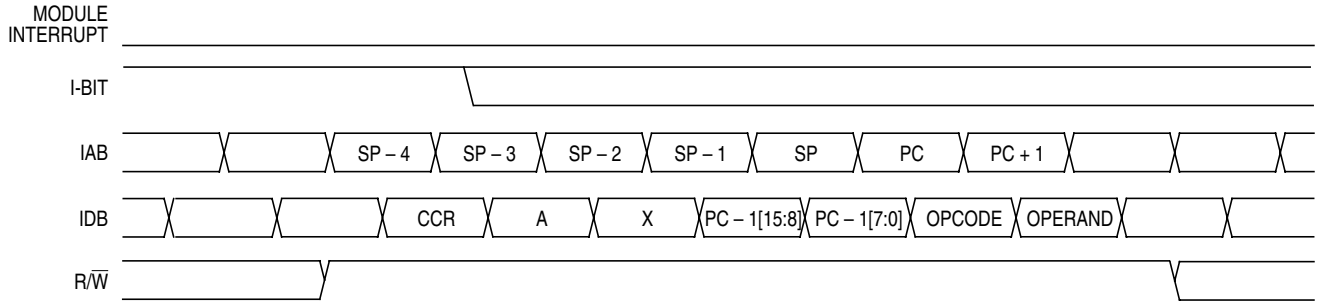


Figure 7-9. Interrupt Recovery Timing

Interrupts are latched, and arbitration is performed in the SIM at the start of interrupt processing. The arbitration result is a constant that the CPU uses to determine which vector to fetch. Once an interrupt is latched by the SIM, no other interrupt can take precedence, regardless of priority, until the latched interrupt is serviced (or the I bit is cleared).
(See Figure 7-10.)

7.5.3 Reset

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

7.5.4 Break Interrupts

The break module can stop normal program flow at a software-programmable break point by asserting its break interrupt output. (See Chapter 21 Break Module (BRK).) 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.

7.5.5 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 SIM break flag control register (SBFCR).

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 2-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.

7.6 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 in the following subsections. Both STOP and WAIT clear the interrupt mask (I) in the condition code register, allowing interrupts to occur.

7.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. Figure 7-15 shows the timing for wait mode entry.

A module that is active during wait mode can wake up the CPU with an interrupt if the interrupt is enabled. Stacking for the interrupt begins one cycle after the WAIT instruction during which the interrupt occurred. In wait mode, the CPU clocks are inactive. Refer to the wait mode subsection of each module to see if the module is active or inactive in wait mode. Some modules can be programmed to be active in wait mode.

Wait mode also can be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (SBSR). If the COP disable bit, COPD, in the mask option register is logic 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.

8.3.1 Entering Monitor Mode

Table 8-1 shows the pin conditions for entering monitor mode. As specified in the table, monitor mode may be entered after a POR and will allow communication at 9600 baud provided one of the following sets of conditions is met:

1. If $FFFE and $FFFF do not contain $FF (programmed state):
   - The external clock is 4.9152 MHz with PTB0 low or 9.8304 MHz with PTB0 high
   - $\overline{IRQ1} = V_{TST}$
2. If $FFFE and $FFFF both contain $FF (erased state):
   - The external clock is 9.8304 MHz
   - $\overline{IRQ1} = V_{DD}$ (this can be implemented through the internal $\overline{IRQ1}$ pullup
3. If $FFFE and $FFFF both contain $FF (erased state):
   - The external clock is 32.768 kHz (crystal)
   - $\overline{IRQ1} = V_{SS}$ (this setting initiates the PLL to boost the external 32.768 kHz to an internal bus frequency of 2.4576 MHz

If V_{TST} is applied to $\overline{IRQ1}$ and PTB0 is low upon monitor mode entry (above condition set 1), the bus frequency is a divide-by-two of the input clock. If PTB0 is high with V_{TST} applied to $\overline{IRQ1}$ upon monitor mode entry, the bus frequency will be a divide-by-four of the input clock. Holding the PTB0 pin low when entering monitor mode causes a bypass of a divide-by-two stage at the oscillator only if V_{TST} is applied to $\overline{IRQ1}$. In this event, the CGMOUT frequency is equal to the CGMXCLK frequency, and the OSC1 input directly generates internal bus clocks. In this case, the OSC1 signal must have a 50% duty cycle at maximum bus frequency.

If entering monitor mode without high voltage on $\overline{IRQ1}$ (above condition set 2, where applied voltage is either V_{DD}), then all port A pin requirements and conditions, including the PTB0 frequency divisor selection, are not in effect. This is to reduce circuit requirements when performing in-circuit programming.

NOTE

If the reset vector is blank and monitor mode is entered, the chip will see an additional reset cycle after the initial POR reset. Once the part has been programmed, the traditional method of applying a voltage, V_{TST}, to $\overline{IRQ1}$ must be used to enter monitor mode.

The COP module is disabled in monitor mode based on these conditions:

- If monitor mode was entered as a result of the reset vector being blank (above condition set 2 or 3), the COP is always disabled regardless of the state of $\overline{IRQ1}$ or $\overline{RST}$.
- If monitor mode was entered with V_{TST} on $\overline{IRQ1}$ (condition set 1), then the COP is disabled as long as V_{TST} is applied to either $\overline{IRQ1}$ or $\overline{RST}$.

The second condition states that as long as V_{TST} is maintained on the $\overline{IRQ1}$ pin after entering monitor mode, or if V_{TST} is applied to $\overline{RST}$ after the initial reset to get into monitor mode (when V_{TST} was applied to $\overline{IRQ1}$), then the COP will be disabled. In the latter situation, after V_{TST} is applied to the $\overline{RST}$ pin, V_{TST} can be removed from the $\overline{IRQ1}$ pin in the interest of freeing the $\overline{IRQ1}$ for normal functionality in monitor mode.

Figure 8-2 shows a simplified diagram of the monitor mode entry when the reset vector is blank and just V_{DD} voltage is applied to the $\overline{IRQ1}$ pin. An external oscillator of 9.8304 MHz is required for a baud rate of 9600, as the internal bus frequency is automatically set to the external frequency divided by four.

11.8.7 SCI Baud Rate Register

The baud rate register (SCBR) selects the baud rate for both the receiver and the transmitter.

Address:	$0019	6	5	4	3	2	1	Bit 0
Read:	0	0	SCP1	SCP0	R	SCR2	SCR1	SCR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented R = Reserved

Figure 11-16. SCI Baud Rate Register (SCBR)

SCP1 and SCP0 — SCI Baud Rate Prescaler Bits

These read/write bits select the baud rate prescaler divisor as shown in Table 11-6. Reset clears SCP1 and SCP0.

Table 11-6. SCI Baud Rate Prescaling

SCP1 and SCP0	Prescaler Divisor (PD)
00	1
01	3
10	4
11	13

SCR2–SCR0 — SCI Baud Rate Select Bits

These read/write bits select the SCI baud rate divisor as shown in Table 11-7. Reset clears SCR2–SCR0.

Table 11-7. SCI Baud Rate Selection

SCR2, SCR1, and SCR0	Baud Rate Divisor (BD)
000	1
001	2
010	4
011	8
100	16
101	32
110	64
111	128

Use this formula to calculate the SCI baud rate:

$$\text{baud rate} = \frac{\text{SCI clock source}}{64 \times \text{PD} \times \text{BD}}$$

where:

SCI clock source = f_{BUS} or CGMXCLK
(selected by SCIBDSRC bit in CONFIG2 register)
PD = prescaler divisor
BD = baud rate divisor

R8 — Received Bit 8

When the SCI is receiving 9-bit characters, R8 is the read-only ninth bit (bit 8) of the received character. R8 is received at the same time that the IRSCDR receives the other 8 bits.
When the SCI is receiving 8-bit characters, R8 is a copy of the eighth bit (bit 7). Reset has no effect on the R8 bit.

T8 — Transmitted Bit 8

When the SCI is transmitting 9-bit characters, T8 is the read/write ninth bit (bit 8) of the transmitted character. T8 is loaded into the transmit shift register at the same time that the IRSCDR is loaded into the transmit shift register. Reset has no effect on the T8 bit.

DMARE — DMA Receive Enable Bit

CAUTION

The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.

1 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)
0 = DMA not enabled to service SCI receiver DMA service requests generated by the SCRF bit (SCI receiver CPU interrupt requests enabled)

DMATE — DMA Transfer Enable Bit

CAUTION

The DMA module is not included on this MCU. Writing a logic 1 to DMARE or DMATE may adversely affect MCU performance.

1 = SCTE DMA service requests enabled; SCTE CPU interrupt requests disabled
0 = SCTE DMA service requests disabled; SCTE CPU interrupt requests enabled

ORIE — Receiver Overrun Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the receiver overrun bit, OR. Reset clears ORIE.

1 = SCI error CPU interrupt requests from OR bit enabled
0 = SCI error CPU interrupt requests from OR bit disabled

NEIE — Receiver Noise Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the noise error bit, NE. Reset clears NEIE.

1 = SCI error CPU interrupt requests from NE bit enabled
0 = SCI error CPU interrupt requests from NE bit disabled

FEIE — Receiver Framing Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the framing error bit, FE. Reset clears FEIE.

1 = SCI error CPU interrupt requests from FE bit enabled
0 = SCI error CPU interrupt requests from FE bit disabled

PEIE — Receiver Parity Error Interrupt Enable Bit

This read/write bit enables SCI error CPU interrupt requests generated by the parity error bit, PE. (See 12.9.4 IRSCI Status Register 1.) Reset clears PEIE.

1 = SCI error CPU interrupt requests from PE bit enabled
0 = SCI error CPU interrupt requests from PE bit disabled

12.9.4 IRSCI Status Register 1

SCI status register 1 contains flags to signal these conditions:

- Transfer of IRSCDR data to transmit shift register complete
- Transmission complete
- Transfer of receive shift register data to IRSCDR complete
- Receiver input idle
- Receiver overrun
- Noisy data
- Framing error
- Parity error

Address: $0043

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
Write:								
Reset:	1	1	0	0	0	0	0	0

= Unimplemented

Figure 12-15. IRSCI Status Register 1 (IRSCS1)

SCTE — SCI Transmitter Empty Bit

This clearable, read-only bit is set when the IRSCDR transfers a character to the transmit shift register. SCTE can generate an SCI transmitter CPU interrupt request. When the SCTIE bit in IRSCC2 is set, SCTE generates an SCI transmitter CPU interrupt request. In normal operation, clear the SCTE bit by reading IRSCS1 with SCTE set and then writing to IRSCDR. Reset sets the SCTE bit.

1 = IRSCDR data transferred to transmit shift register
0 = IRSCDR data not transferred to transmit shift register

TC — Transmission Complete Bit

This read-only bit is set when the SCTE bit is set, and no data, preamble, or break character is being transmitted. TC generates an SCI transmitter CPU interrupt request if the TCIE bit in IRSCC2 is also set. TC is automatically cleared when data, preamble or break is queued and ready to be sent. There may be up to 1.5 transmitter clocks of latency between queueing data, preamble, and break and the transmission actually starting. Reset sets the TC bit.

1 = No transmission in progress
0 = Transmission in progress

SCRF — SCI Receiver Full Bit

This clearable, read-only bit is set when the data in the receive shift register transfers to the SCI data register. SCRF can generate an SCI receiver CPU interrupt request. When the SCRIE bit in IRSCC2 is set, SCRF generates a CPU interrupt request. In normal operation, clear the SCRF bit by reading IRSCS1 with SCRF set and then reading the IRSCDR. Reset clears SCRF.

1 = Received data available in IRSCDR
0 = Data not available in IRSCDR

IDLE — Receiver Idle Bit

This clearable, read-only bit is set when 10 or 11 consecutive logic 1s appear on the receiver input. IDLE generates an SCI receiver CPU interrupt request if the ILIE bit in IRSCC2 is also set. Clear the IDLE bit by reading IRSCS1 with IDLE set and then reading the IRSCDR. After the receiver is enabled,

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0010	SPI Control Register (SPCR)	Read: Write:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Reset:	0	0	1	0	1	0	0	0
$0011	SPI Status and Control Register (SPSCR)	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:								
		Reset:	0	0	0	0	1	0	0	0
$0012	SPI Data Register (SPDR)	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved

Figure 13-1. SPI I/O Register Summary

13.4 Functional Description

Figure 13-2 shows the structure of the SPI module.

The SPI module allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs. Software can poll the SPI status flags or SPI operation can be interrupt-driven.

The following paragraphs describe the operation of the SPI module.

13.4.1 Master Mode

The SPI operates in master mode when the SPI master bit, SPMSTR, is set.

NOTE

Configure the SPI modules as master or slave before enabling them. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI. (See 13.13.1 SPI Control Register.)

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the transmit data register. If the shift register is empty, the byte immediately transfers to the shift register, setting the SPI transmitter empty bit, SPTE. The byte begins shifting out on the MOSI pin under the control of the serial clock. (See Figure 13-3.)

The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. (See 13.13.2 SPI Status and Control Register.) Through the SPSCK pin, the baud rate generator of the master also controls the shift register of the slave peripheral.

As the byte shifts out on the MOSI pin of the master, another byte shifts in from the slave on the master's MISO pin. The transmission ends when the receiver full bit, SPRF, becomes set. At the same time that SPRF becomes set, the byte from the slave transfers to the receive data register. In normal operation, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Writing to the SPI data register clears the SPTE bit.

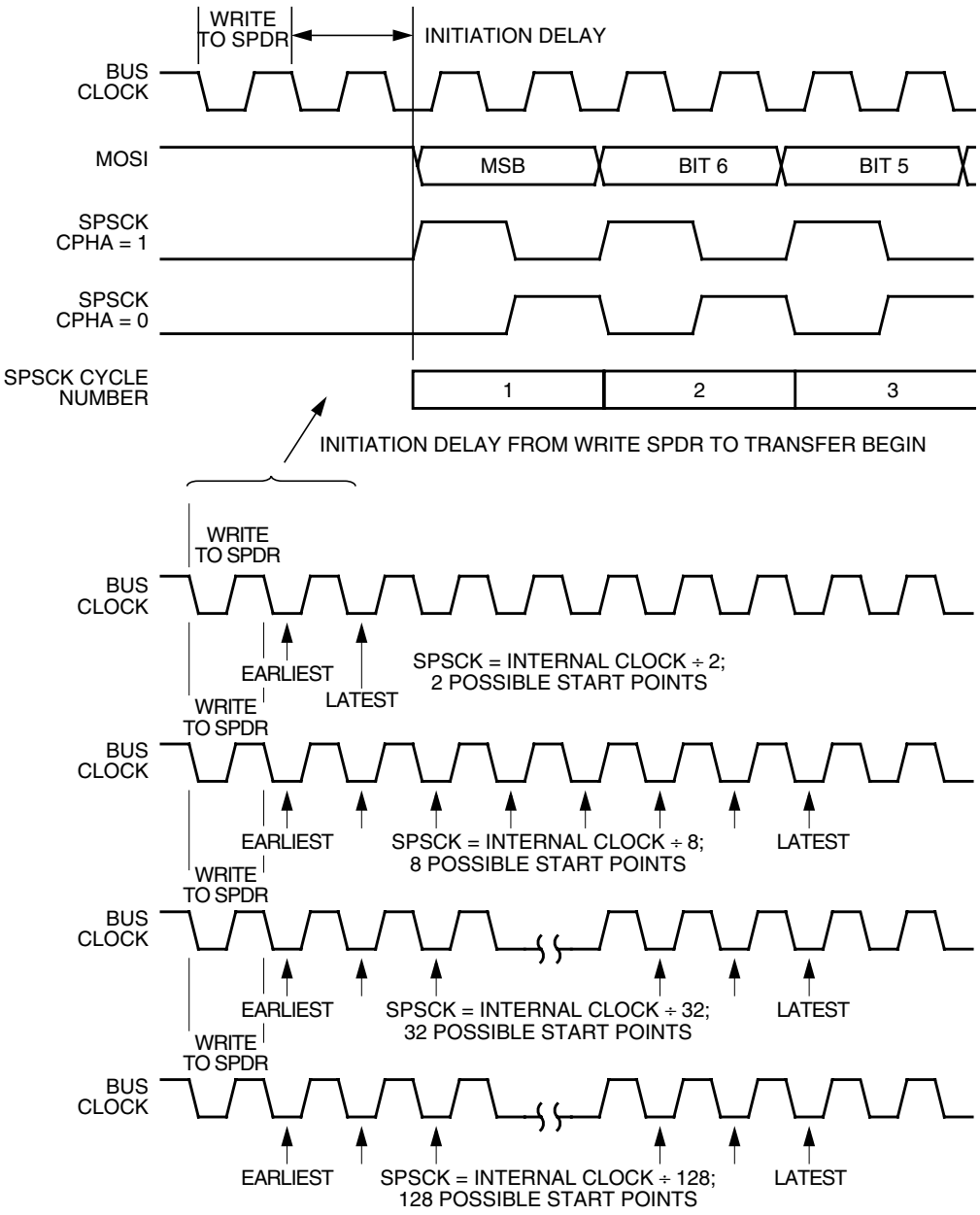


Figure 13-7. Transmission Start Delay (Master)

13.6 Queuing Transmission Data

The double-buffered transmit data register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag (SPTE) indicates when the transmit data buffer is ready to accept new data. Write to the transmit data register only when the SPTE bit is high. Figure 13-8 shows the timing associated with doing back-to-back transmissions with the SPI (SPSCK has CPHA: CPOL = 1:0).

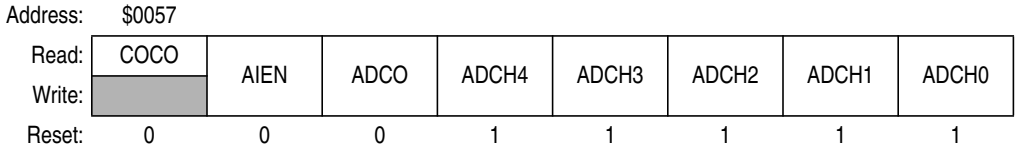
15.7 I/O Registers

These I/O registers control and monitor ADC operation:

- ADC status and control register (ADSCR) — $0057
- ADC clock control register (ADICLK) — $0058
- ADC data register high:low 0 (ADRH0:ADRL0) — $0059:$005A
- ADC data register low 1–3 (ADRL1–ADRL3) — $005B–$005D
- ADC auto-scan control register (ADASCR) — $005E

15.7.1 ADC Status and Control Register

Function of the ADC status and control register is described here.



Address:	$0057							
Read:	COCO	AIEN	ADCO	ADCH4	ADCH3	ADCH2	ADCH1	ADCH0
Write:								
Reset:	0	0	0	1	1	1	1	1

Figure 15-3. ADC Status and Control Register (ADSCR)

COCO — Conversions Complete Bit

When the AIEN bit is a logic 0, the COCO is a read-only bit which is set each time a conversion is completed. This bit is cleared whenever the ADSCR is written, or whenever the ADC clock control register is written, or whenever the ADC data register low, ADRLx, is read.

If the AIEN bit is logic 1, the COCO bit always read as logic 0. ADC interrupt will be generated at the end if an ADC conversion. Reset clears the COCO bit.

1 = Conversion completed (AIEN = 0)

0 = Conversion not completed (AIEN = 0)/CPU interrupt (AIEN=1)

AIEN — ADC Interrupt Enable Bit

When this bit is set, an interrupt is generated at the end of an ADC conversion. The interrupt signal is cleared when the data register, ADR0, is read or the ADSCR is written. Reset clears the AIEN bit.

1 = ADC interrupt enabled

0 = ADC interrupt disabled

ADCO — ADC Continuous Conversion Bit

When set, the ADC will convert samples continuously and update the ADC data register at the end of each conversion. Only one conversion is allowed when this bit is cleared. Reset clears the ADCO bit.

1 = Continuous ADC conversion

0 = One ADC conversion

This bit should not be set when auto-scan mode is enabled; i.e. when ASCAN=1.

ADCH[4:0] — ADC Channel Select Bits

ADCH[4:0] form a 5-bit field which is used to select one of the ADC channels when not in auto-scan mode. The five channel select bits are detailed in Table 15-1.

NOTE

Care should be taken when using a port pin as both an analog and a digital input simultaneously to prevent switching noise from corrupting the analog signal. Recovery from the disabled state requires one conversion cycle to stabilize.

NOTE

Service the COP immediately after reset and before entering or after exiting STOP Mode to guarantee the maximum time before the first COP counter overflow.

A COP reset pulls the $\overline{\text{RST}}$ pin low for 32 ICLK cycles and sets the COP bit in the SIM reset status register (SRSR).

In monitor mode, the COP is disabled if the $\overline{\text{RST}}$ pin or the $\overline{\text{IRQ1}}$ is held at V_{TST}. During the break state, V_{TST} on the $\overline{\text{RST}}$ pin disables the COP.

NOTE

Place COP clearing instructions in the main program and not in an interrupt subroutine. Such an interrupt subroutine could keep the COP from generating a reset even while the main program is not working properly.

19.3 I/O Signals

The following paragraphs describe the signals shown in Figure 19-1.

19.3.1 ICLK

ICLK is the internal oscillator output signal. See Chapter 22 Electrical Specifications for ICLK frequency specification.

19.3.2 STOP Instruction

The STOP instruction clears the COP prescaler.

19.3.3 COPCTL Write

Writing any value to the COP control register (COPCTL) (see 19.4 COP Control Register) clears the COP counter and clears bits 12 through 5 of the prescaler. Reading the COP control register returns the low byte of the reset vector.

19.3.4 Power-On Reset

The power-on reset (POR) circuit clears the COP prescaler 4096 ICLK cycles after power-up.

19.3.5 Internal Reset

An internal reset clears the COP prescaler and the COP counter.

19.3.6 Reset Vector Fetch

A reset vector fetch occurs when the vector address appears on the data bus. A reset vector fetch clears the COP prescaler.

19.3.7 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the CONFIG1 register. (See Figure 19-2 . Configuration Register 1 (CONFIG1).)

21.5.2 Break Address Registers

The break address registers (BRKH and BRKL) 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: Write:	Bit 15	14	13	12	11	10	9	Bit 8
Reset:	0	0	0	0	0	0	0	0

Figure 21-4. Break Address Register High (BRKH)

Address: $FE0D

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

Figure 21-5. Break Address Register Low (BRKL)

21.5.3 SIM Break Status Register

The SIM break status register (SBSR) contains a flag to indicate that a break caused an exit from wait mode. The flag is useful in applications requiring a return to wait mode after exiting from a break interrupt.

Address: $FE00

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

Note: Writing a logic 0 clears SBSW. R = Reserved

Figure 21-6. SIM Break Status Register (SBSR)

SBSW — Break Wait Bit

This status bit is set when a break interrupt causes an exit from wait mode or stop mode. Clear SBSW by writing a logic 0 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 interrupt routine. The user can modify the return address on the stack by subtracting 1 from it. The following code is an example.

```
;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,SBSR, 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
```

21.5.4 SIM Break Flag Control Register

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

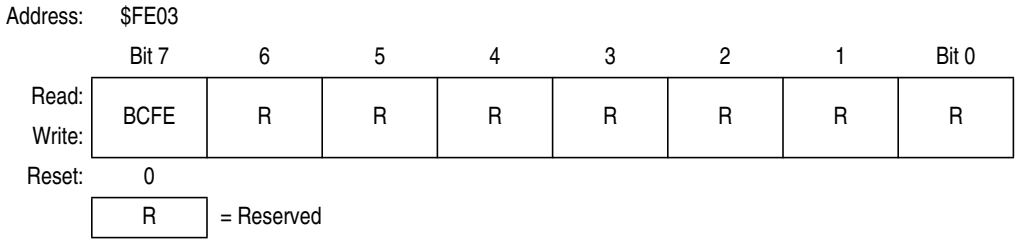


Figure 21-7. SIM Break Flag Control Register (SBFCR)

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.

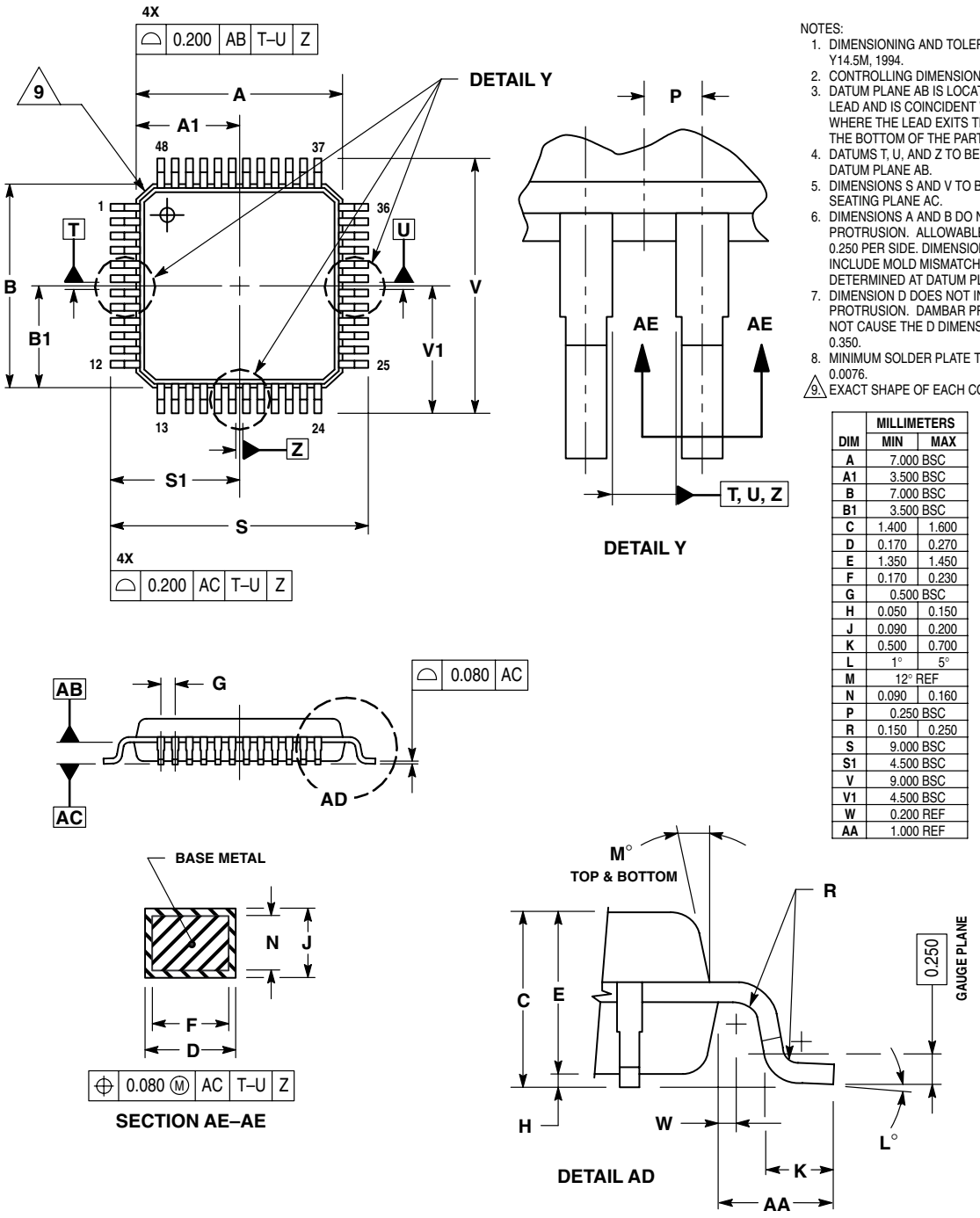
1 = Status bits clearable during break
0 = Status bits not clearable during break

22.5 5V DC Electrical Characteristics

Table 22-4. DC Electrical Characteristics (5V)

Characteristic[1]	Symbol	Min	Typ[2]	Max	Unit
Output high voltage (I_{LOAD} = −12mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = −15mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7]	 V_{OH} V_{OH}	 V_{DD}−0.8 V_{DD}−1.0	 — —	 — —	 V V
Output low voltage (I_{LOAD} = 6mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = 12mA) PTB[0:3], PTC[6:7] (I_{LOAD} = 15mA) PTA[0:7], PTB[4:7], PTC[0:5], PTD[0:7] (I_{LOAD} = 15mA) PTB[0:3], PTC[6:7] (I_{LOAD} = 15mA) as TxD, RxD, SCTxD, SCRxD (I_{LOAD} = see Table 22-12) as SDA, SCL	 V_{OL} V_{OL} V_{OL} V_{OL} V_{OLSCI} V_{OLIIC}	 — — — — — —	 — — — — — —	 0.4 0.4 0.8 0.6 0.4 0.4	 V V V V V V
LED sink current (V_{OL} = 3V) PTA[0:7]	I_{OL}	9	15	25	mA
Input high voltage PTA[0:7], PTB[0:7], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$ OSC1	V_{IH}	0.7 × V_{DD} 0.7 × V_{REG}	— —	V_{DD} V_{REG}	V V
Input low voltage PTA[0:7], PTB[0:7], PTC[0:7], PTD[0:7], $\overline{RST}$, $\overline{IRQ1}$ OSC1	V_{IL}	V_{SS} V_{SS}	— —	0.3 × V_{DD} 0.3 × V_{REG}	V V
V_{DD} supply current, f_{OP} = 8 MHz Run[3] Wait[4] Stop (25°C) with OSC, TBM, and LVI modules on[5] with OSC and TBM modules on[5] all modules off[6] Stop (0 to 85°C) with OSC, TBM, and LVI modules on[5] with OSC and TBM modules on[5] all modules off[6]	 I_{DD} 	 — — — — — — — —	 10 2.5 0.8 22 20 1 45 42	 20 10 1.8 150 125 2.5 300 250	 mA mA mA µA µA 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 pF
POR rearm voltage[7]	V_{POR}	0	—	100	mV
POR rise time ramp rate[8]	R_{POR}	0.035	—	—	V/ms
Monitor mode entry voltage	V_{TST}	1.4 × V_{DD}		8.5	V

23.2 48-Pin Low-Profile Quad Flat Pack (LQFP)



NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
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 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 0.350.
8. MINIMUM SOLDER PLATE THICKNESS SHALL BE 0.0076.
9. EXACT SHAPE OF EACH CORNER IS OPTIONAL.

	MILLIMETERS	
DIM	MIN	MAX
A	7.000 BSC	
A1	3.500 BSC	
B	7.000 BSC	
B1	3.500 BSC	
C	1.400	1.600
D	0.170	0.270
E	1.350	1.450
F	0.170	0.230
G	0.500 BSC	
H	0.050	0.150
J	0.090	0.200
K	0.500	0.700
L	1°	5°
M	12° REF	
N	0.090	0.160
P	0.250 BSC	
R	0.150	0.250
S	9.000 BSC	
S1	4.500 BSC	
V	9.000 BSC	
V1	4.500 BSC	
W	0.200 REF	
AA	1.000 REF	

Figure 23-1. 48-Pin LQFP (Case #932)