



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	SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	48KB (48K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 24x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc908gr48acfje

List of Chapters

Chapter 1 General Description.....19

Chapter 2 Memory.....29

Chapter 3 Analog-to-Digital Converter (ADC).....59

Chapter 4 Clock Generator Module (CGM).....71

Chapter 5 Configuration Register (CONFIG).....89

Chapter 6 Computer Operating Properly (COP) Module.....93

Chapter 7 Central Processor Unit (CPU).....97

Chapter 8 External Interrupt (IRQ).....109

Chapter 9 Keyboard Interrupt Module (KBI).....113

Chapter 10 Low-Power Modes.....121

Chapter 11 Low-Voltage Inhibit (LVI).....127

Chapter 12 Input/Output (I/O) Ports.....131

Chapter 13 Enhanced Serial Communications Interface (ESCI) Module.....151

Chapter 14 System Integration Module (SIM).....183

Chapter 15 Serial Peripheral Interface (SPI) Module.....201

Chapter 16 Timebase Module (TBM).....221

Chapter 17 Timer Interface Module (TIM1).....225

Chapter 18 Timer Interface Module (TIM2).....241

Chapter 19 Development Support.....261

Chapter 20 Electrical Specifications.....277

Chapter 21 Ordering Information and Mechanical Specifications.....295

Appendix A MC68HC908GR48A.....305

Appendix B MC68HC908GR32A.....309

Chapter 9

Keyboard Interrupt Module (KBI)

9.1	Introduction	113
9.2	Features	113
9.3	Functional Description	113
9.4	Keyboard Initialization	116
9.5	Low-Power Modes	117
9.5.1	Wait Mode	117
9.5.2	Stop Mode	117
9.6	Keyboard Module During Break Interrupts.	117
9.7	I/O Registers	117
9.7.1	Keyboard Status and Control Register.	118
9.7.2	Keyboard Interrupt Enable Register.	119
9.7.3	Keyboard Interrupt Polarity Register	119

Chapter 10

Low-Power Modes

10.1	Introduction	121
10.1.1	Wait Mode	121
10.1.2	Stop Mode	121
10.2	Analog-to-Digital Converter (ADC).	121
10.2.1	Wait Mode	121
10.2.2	Stop Mode	121
10.3	Break Module (BRK)	121
10.3.1	Wait Mode	121
10.3.2	Stop Mode	121
10.4	Central Processor Unit (CPU)	122
10.4.1	Wait Mode	122
10.4.2	Stop Mode	122
10.5	Clock Generator Module (CGM)	122
10.5.1	Wait Mode	122
10.5.2	Stop Mode	122
10.6	Computer Operating Properly Module (COP)	122
10.6.1	Wait Mode	122
10.6.2	Stop Mode	122
10.7	External Interrupt Module (IRQ)	123
10.7.1	Wait Mode	123
10.7.2	Stop Mode	123
10.8	Keyboard Interrupt Module (KBI)	123
10.8.1	Wait Mode	123
10.8.2	Stop Mode	123
10.9	Low-Voltage Inhibit Module (LVI)	123
10.9.1	Wait Mode	123
10.9.2	Stop Mode	123

Table of Contents

13.8.3	ESCI Control Register 3	169
13.8.4	ESCI Status Register 1	170
13.8.5	ESCI Status Register 2	173
13.8.6	ESCI Data Register	173
13.8.7	ESCI Baud Rate Register	174
13.8.8	ESCI Prescaler Register	175
13.9	ESCI Arbiter	177
13.9.1	ESCI Arbiter Control Register	179
13.9.2	ESCI Arbiter Data Register	180
13.9.3	Bit Time Measurement	180
13.9.4	Arbitration Mode	180

Chapter 14 System Integration Module (SIM)

14.1	Introduction	183
14.2	SIM Bus Clock Control and Generation	186
14.2.1	Bus Timing	186
14.2.2	Clock Startup from POR or LVI Reset	186
14.2.3	Clocks in Stop Mode and Wait Mode	186
14.3	Reset and System Initialization	187
14.3.1	External Pin Reset	187
14.3.2	Active Resets from Internal Sources	187
14.3.2.1	Power-On Reset	188
14.3.2.2	Computer Operating Properly (COP) Reset	188
14.3.2.3	Illegal Opcode Reset	189
14.3.2.4	Illegal Address Reset	189
14.3.2.5	Low-Voltage Inhibit (LVI) Reset	189
14.3.2.6	Monitor Mode Entry Module Reset (MODRST)	189
14.4	SIM Counter	190
14.4.1	SIM Counter During Power-On Reset	190
14.4.2	SIM Counter During Stop Mode Recovery	190
14.4.3	SIM Counter and Reset States	190
14.5	Exception Control	190
14.5.1	Interrupts	190
14.5.1.1	Hardware Interrupts	191
14.5.1.2	SWI Instruction	193
14.5.1.3	Interrupt Status Registers	193
14.5.2	Reset	196
14.5.3	Break Interrupts	196
14.5.4	Status Flag Protection in Break Mode	196
14.6	Low-Power Modes	196
14.6.1	Wait Mode	197
14.6.2	Stop Mode	198
14.7	SIM Registers	199
14.7.1	Break Status Register	199
14.7.2	SIM Reset Status Register	199
14.7.3	Break Flag Control Register	200

Memory

\$0000 ↓ \$003F	I/O REGISTERS 64 BYTES	\$FE00	SIM BREAK STATUS REGISTER (BSR)
\$0040 ↓ \$043F	RAM-1 1024 BYTES	\$FE01	SIM RESET STATUS REGISTER (SRSR)
\$0440 ↓ \$0461	I/O REGISTERS 34 BYTES	\$FE02	RESERVED
\$0462 ↓ \$04FF	FLASH-2 158 BYTES	\$FE03	SIM BREAK FLAG CONTROL REGISTER (BFCR)
\$0500 ↓ \$057F	RESERVED 128 BYTES	\$FE04	INTERRUPT STATUS REGISTER 1 (INT1)
\$0580 ↓ \$097F	RAM-2 1024 BYTES	\$FE05	INTERRUPT STATUS REGISTER 2 (INT2)
\$0980 ↓ \$1B7F	FLASH-2 4608 BYTES	\$FE06	INTERRUPT STATUS REGISTER 3 (INT3)
\$1B80 ↓ \$1DFF	RESERVED 640 BYTES	\$FE07	INTERRUPT STATUS REGISTER 4 (INT4)
\$1E00 ↓ \$1E0F	MONITOR ROM 16 BYTES	\$FE08	FLASH-2 CONTROL REGISTER (FL2CR)
\$1E10 ↓ \$1E1F	RESERVED 16 BYTES	\$FE09	BREAK ADDRESS REGISTER HIGH (BRKH)
\$1E20 ↓ \$7FFF	FLASH-2 25,056 BYTES	\$FE0A	BREAK ADDRESS REGISTER LOW (BRKL)
\$8000 ↓ \$FDFF	FLASH-1 32,256 BYTES	\$FE0B	BREAK STATUS AND CONTROL REGISTER (BRKSCR)
		\$FE0C	LVI STATUS REGISTER (LVISR)
		\$FE0D	FLASH-2 TEST CONTROL REGISTER (FLTCR2)
		\$FE0E	FLASH-1 TEST CONTROL REGISTER (FLTCR1)
		\$FE0F	UNIMPLEMENTED
		\$FE10 ↓ \$FE1F	UNIMPLEMENTED 16 BYTES RESERVED FOR COMPATIBILITY WITH MONITOR CODE FOR A-FAMILY PART
		\$FE20 ↓ \$FF7F	MONITOR ROM 352 BYTES
		\$FF80	FLASH-1 BLOCK PROTECT REGISTER (FL1BPR)
		\$FF81	FLASH-2 BLOCK PROTECT REGISTER (FL2BPR)
		\$FF82 ↓ \$FF87	RESERVED 6 BYTES
		\$FF88	FLASH-1 CONTROL REGISTER (FL1CR)
		\$FF89 ↓ \$FFCB	RESERVED 67 BYTES
		\$FFCC ↓ \$FFFF ⁽¹⁾	FLASH-1 VECTORS 52 BYTES

1. \$FFF6–\$FFFD used for eight security bytes

Figure 2-1. MC68HC908GR60A Memory Map

2.6.4 FLASH-1 Mass Erase Operation

Use this step-by-step procedure to erase the entire FLASH-1 memory:

1. Set both the ERASE bit and the MASS bit in the FLASH-1 control register (FL1CR).
2. Read the FLASH-1 block protect register (FL1BPR).

NOTE

Mass erase is disabled whenever any block is protected (FL1BPR does not equal \$FF).

3. Write to any FLASH-1 address within the FLASH-1 array with any data.
4. Wait for a time, t_{NVS} (minimum 10 μ s).
5. Set the HVEN bit.
6. Wait for a time, t_{MERASE} (minimum 4 ms).
7. Clear the ERASE and MASS bits.
8. Wait for a time, t_{NVHL} (minimum 100 μ s).
9. Clear the HVEN bit.
10. Wait for a time, t_{RCV} , (typically 1 μ s) after which the memory can be accessed in normal read mode.

NOTE

- A. *Programming and erasing of FLASH locations can not be performed by code being executed from the same FLASH array.*
- B. *While these operations must be performed in the order shown, other unrelated operations may occur between the steps. However, care must be taken to ensure that these operations do not access any address within the FLASH array memory space such as the COP control register (COPCTL) at \$FFFF.*
- C. *It is highly recommended that interrupts be disabled during program/erase operations.*

HVEN — High-Voltage Enable Bit

This read/write bit enables the charge pump to drive high voltages for program and erase operations in the array. HVEN can only be set if either PGM = 1 or ERASE = 1 and the proper sequence for program or erase is followed.

- 1 = High voltage enabled to array and charge pump on
- 0 = High voltage disabled to array and charge pump off

MASS — Mass Erase Control Bit

Setting this read/write bit configures the FLASH-2 array for mass or page erase operation.

- 1 = Mass erase operation selected
- 0 = Page erase operation selected

ERASE — Erase Control Bit

This read/write bit configures the memory for erase operation. ERASE is interlocked with the PGM bit such that both bits cannot be set at the same time.

- 1 = Erase operation selected
- 0 = Erase operation unselected

PGM — Program Control Bit

This read/write bit configures the memory for program operation. PGM is interlocked with the ERASE bit such that both bits cannot be equal to 1 or set to 1 at the same time.

- 1 = Program operation selected
- 0 = Program operation unselected

2.7.2.2 FLASH-2 Block Protect Register

The FLASH-2 block protect register (FL2BPR) is implemented as a byte within the FLASH-1 memory; therefore, can only be written during a FLASH-1 programming sequence. The value in this register determines the starting location of the protected range within the FLASH-2 memory.



Figure 2-8. FLASH-2 Block Protect Register (FL2BPR)

NOTE

The FLASH-2 block protect register (FL2BPR) controls the block protection for the FLASH-2 array. However, FL2BPR is implemented within the FLASH-1 memory array and therefore, the FLASH-1 control register (FL1CR) must be used to program/erase FL2BPR.

FL2BPR[7:0] — Block Protect Register Bits 7 to 0

These eight bits represent bits [14:7] of a 16-bit memory address. Bit 15 is a 0 and bits [6:0] are 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH-2 memory for block protection. FLASH-2 is protected from this start address to the end of FLASH-2 memory at \$7FFF. With this mechanism, the protect start address can be \$XX00 and \$XX80 (128 byte page boundaries) within the FLASH-2 array.

4.5.1 PLL Control Register

The PLL control register (PCTL) contains the interrupt enable and flag bits, the on/off switch, the base clock selector bit, and the VCO power-of-two range selector bits.

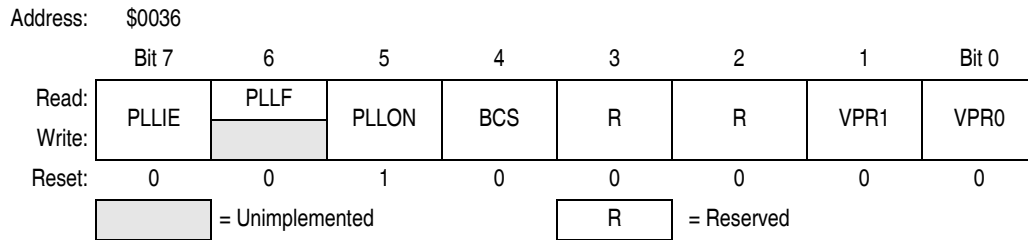


Figure 4-4. PLL Control Register (PCTL)

PLLIE — PLL Interrupt Enable Bit

This read/write bit enables the PLL to generate an interrupt request when the LOCK bit toggles, setting the PLL flag, PLLF. When the AUTO bit in the PLL bandwidth control register (PBWC) is clear, PLLIE cannot be written and reads as 0. Reset clears the PLLIE bit.

- 1 = PLL interrupts enabled
- 0 = PLL interrupts disabled

PLLF — PLL Interrupt Flag Bit

This read-only bit is set whenever the LOCK bit toggles. PLLF generates an interrupt request if the PLLIE bit also is set. PLLF always reads as 0 when the AUTO bit in the PLL bandwidth control register (PBWC) is clear. Clear the PLLF bit by reading the PLL control register. Reset clears the PLLF bit.

- 1 = Change in lock condition
- 0 = No change in lock condition

NOTE

Do not inadvertently clear the PLLF bit. Any read or read-modify-write operation on the PLL control register clears the PLLF bit.

PLLON — PLL On Bit

This read/write bit activates the PLL and enables the VCO clock, CGMVCLK. PLLON cannot be cleared if the VCO clock is driving the base clock, CGMOUT (BCS = 1). (See 4.3.8 Base Clock Selector Circuit.) Reset sets this bit so that the loop can stabilize as the MCU is powering up.

- 1 = PLL on
- 0 = PLL off

BCS — Base Clock Select Bit

This read/write bit selects either the crystal oscillator output, CGMXCLK, or the VCO clock, CGMVCLK, as the source of the CGM output, CGMOUT. CGMOUT frequency is one-half the frequency of the selected clock. BCS cannot be set while the PLLON bit is clear. After toggling BCS, it may take up to three CGMXCLK and three CGMVCLK cycles to complete the transition from one source clock to the other. During the transition, CGMOUT is held in stasis. (See 4.3.8 Base Clock Selector Circuit.) Reset clears the BCS bit.

- 1 = CGMVCLK divided by two drives CGMOUT
- 0 = CGMXCLK divided by two drives CGMOUT

NOTE

PLLON and BCS have built-in protection that prevents the base clock selector circuit from selecting the VCO clock as the source of the base clock

Keyboard Interrupt Module (KBI)

If the MODEK bit is set and depending on the KBIPx bit, the keyboard interrupt pins are both falling (or rising) edge and low (or high) 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 (INTKBSCR). 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 (or rising) 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 1 (or 0) — As long as any enabled keyboard interrupt pin is 0 (or 1), the keyboard interrupt remains set.

The vector fetch or software clear and the return of all enabled keyboard interrupt pins to 1 (or 0) may occur in any order.

If the MODEK bit is clear and depending on the KBIPx bit, the keyboard interrupt pin is falling (or rising) 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 (or 1).

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, use the data direction register to configure the pin as an input and 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.

9.4 Keyboard Initialization

When a keyboard interrupt pin is enabled, it takes time for the internal pullup/pulldown device to reach a 1 (or 0). 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 and polarity by setting the appropriate KBIE_x bits in the keyboard interrupt enable register and the KBIP_x bits in the keyboard interrupt polarity register.
3. Write to the ACKK bit in the keyboard status and control register to clear any false interrupts.
4. Clear the IMASKK bit.

Input/Output (I/O) Ports

Addr.	Register Name	Bit 7	6	5	4	3	2	1	Bit 0
\$0003	Port D Data Register (PTD) See page 142.	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1
		Write:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1
		Reset:	Unaffected by reset						
\$0004	Data Direction Register A (DDRA) See page 136.	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1
		Write:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1
		Reset:	0	0	0	0	0	0	0
\$0005	Data Direction Register B (DDRB) See page 138.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1
		Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1
		Reset:	0	0	0	0	0	0	0
\$0006	Data Direction Register C (DDRC) See page 140.	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1
		Write:		DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1
		Reset:	0	0	0	0	0	0	0
\$0007	Data Direction Register D (DDR D) See page 143.	Read:	DDR D7	DDR D6	DDR D5	DDR D4	DDR D3	DDR D2	DDR D1
		Write:	DDR D7	DDR D6	DDR D5	DDR D4	DDR D3	DDR D2	DDR D1
		Reset:	0	0	0	0	0	0	0
\$0008	Port E Data Register (PTE) See page 145.	Read:	0	0	PTE5	PTE4	PTE3	PTE2	PTE1
		Write:			PTE5	PTE4	PTE3	PTE2	PTE1
		Reset:	Unaffected by reset						
\$000C	Data Direction Register E (DDRE) See page 146.	Read:	0	0	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1
		Write:			DDRE5	DDRE4	DDRE3	DDRE2	DDRE1
		Reset:	0	0	0	0	0	0	0
\$000D	Port A Input Pullup Enable Register (PTAPUE) See page 137.	Read:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1
		Write:	PTAPUE7	PTAPUE6	PTAPUE5	PTAPUE4	PTAPUE3	PTAPUE2	PTAPUE1
		Reset:	0	0	0	0	0	0	0
\$000E	Port C Input Pullup Enable Register (PTCPUE) See page 142.	Read:	0	PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1
		Write:		PTCPUE6	PTCPUE5	PTCPUE4	PTCPUE3	PTCPUE2	PTCPUE1
		Reset:	0	0	0	0	0	0	0
\$000F	Port D Input Pullup Enable Register (PTDPUE) See page 145.	Read:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1
		Write:	PTDPUE7	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1
		Reset:	0	0	0	0	0	0	0

= Unimplemented

Figure 12-1. I/O Port Register Summary (Sheet 2 of 3)

NOTE

Avoid glitches on port B pins by writing to the port B data register before changing data direction register B bits from 0 to 1.

Figure 12-8 shows the port B I/O logic.

When bit DDRBx is a 1, reading address \$0001 reads the PTBx data latch. When bit DDRBx is a 0, reading address \$0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-3 summarizes the operation of the port B pins.

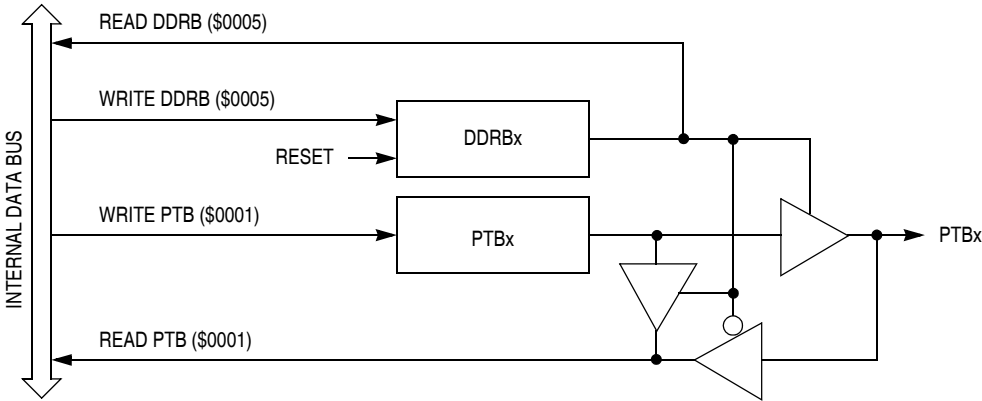


Figure 12-8. Port B I/O Circuit

Table 12-3. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X ⁽¹⁾	Input, Hi-Z ⁽²⁾	DDRB7–DDRB0	Pin	PTB7–PTB0 ⁽³⁾
1	X	Output	DDRB7–DDRB0	PTB7–PTB0	PTB7–PTB0

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$0009	ESCI Prescaler Register (SCPSC) See page 175.	Read:	PDS2	PDS1	PDS0	PSSB4	PSSB3	PSSB2	PSSB1	PSSB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000A	ESCI Arbiter Control Register (SCIACTL) See page 179.	Read:	AM1	Alost	AM0	ACLK	AFIN	ARUN	AROVFL	ARD8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$000B	ESCI Arbiter Data Register (SCIADAT) See page 180.	Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0013	ESCI Control Register 1 (SCC1) See page 166.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0014	ESCI Control Register 2 (SCC2) See page 168.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0015	ESCI Control Register 3 (SCC3) See page 169.	Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
		Write:								
		Reset:	U	0	0	0	0	0	0	0
\$0016	ESCI Status Register 1 (SCS1) See page 170.	Read:	SCTE	TC	SCRf	IDLE	OR	NF	FE	PE
		Write:								
		Reset:	1	1	0	0	0	0	0	0
\$0017	ESCI Status Register 2 (SCS2) See page 173.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$0018	ESCI Data Register (SCDR) See page 173.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
\$0019	ESCI Baud Rate Register (SCBR) See page 174.	Read:	LINT	LINR	SCP1	SCP0	R	SCR2	SCR1	SCR0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented
 = Reserved

Figure 13-4. ESCI I/O Register Summary

RE — Receiver Enable Bit

Setting this read/write bit enables the receiver. Clearing the RE bit disables the receiver but does not affect receiver interrupt flag bits. Reset clears the RE bit.

1 = Receiver enabled

0 = Receiver disabled

NOTE

Writing to the RE bit is not allowed when the enable ESCI bit (ENSCI) is clear. ENSCI is in ESCI control register 1.

RWU — Receiver Wakeup Bit

This read/write bit puts the receiver in a standby state during which receiver interrupts are disabled. The WAKE bit in SCC1 determines whether an idle input or an address mark brings the receiver out of the standby state and clears the RWU bit. Reset clears the RWU bit.

1 = Standby state

0 = Normal operation

SBK — Send Break Bit

Setting and then clearing this read/write bit transmits a break character followed by a 1. The 1 after the break character guarantees recognition of a valid start bit. If SBK remains set, the transmitter continuously transmits break characters with no 1s between them. Reset clears the SBK bit.

1 = Transmit break characters

0 = No break characters being transmitted

NOTE

Do not toggle the SBK bit immediately after setting the SCTE bit. Toggling SBK before the preamble begins causes the ESCI to send a break character instead of a preamble.

13.8.3 ESCI Control Register 3

ESCI control register 3 (SCC3):

- Stores the ninth ESCI data bit received and the ninth ESCI data bit to be transmitted.
- Enables these interrupts:
 - Receiver overrun
 - Noise error
 - Framing error
 - Parity error

Address:	\$0015							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R8	T8	R	R	ORIE	NEIE	FEIE	PEIE
Write:								
Reset:	U	0	0	0	0	0	0	0
	= Unimplemented			R = Reserved		U = Unaffected		

Figure 13-12. ESCI Control Register 3 (SCC3)

R8 — Received Bit 8

When the ESCI 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 SCDR receives the other 8 bits.

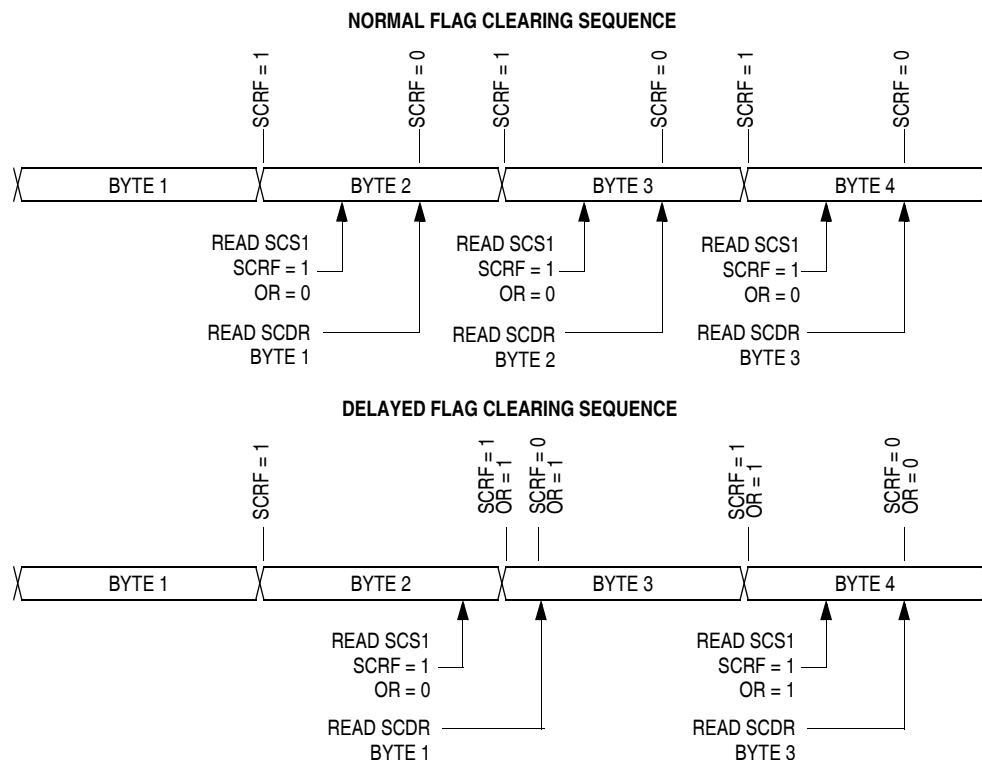


Figure 13-14. Flag Clearing Sequence

In applications that are subject to software latency or in which it is important to know which byte is lost due to an overrun, the flag-clearing routine can check the OR bit in a second read of SCS1 after reading the data register.

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the ESCI detects noise on the Rx pin. NF generates an NF CPU interrupt request if the NEIE bit in SCC3 is also set. Clear the NF bit by reading SCS1 and then reading the SCDR. Reset clears the NF bit.

- 1 = Noise detected
- 0 = No noise detected

FE — Receiver Framing Error Bit

This clearable, read-only bit is set when a 0 is accepted as the stop bit. FE generates an ESCI error CPU interrupt request if the FEIE bit in SCC3 also is set. Clear the FE bit by reading SCS1 with FE set and then reading the SCDR. Reset clears the FE bit.

- 1 = Framing error detected
- 0 = No framing error detected

PE — Receiver Parity Error Bit

This clearable, read-only bit is set when the ESCI detects a parity error in incoming data. PE generates a PE CPU interrupt request if the PEIE bit in SCC3 is also set. Clear the PE bit by reading SCS1 with PE set and then reading the SCDR. Reset clears the PE bit.

- 1 = Parity error detected
- 0 = No parity error detected



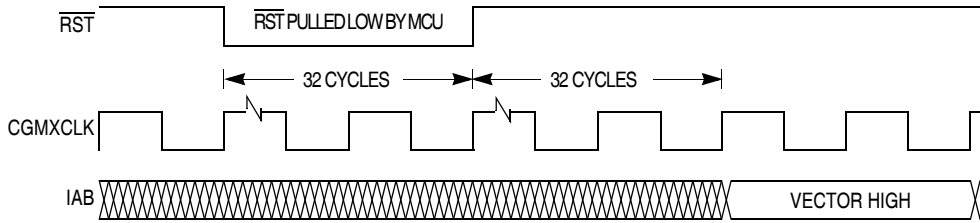


Figure 14-5. Internal Reset Timing

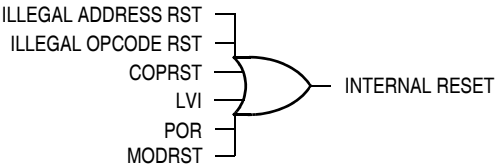


Figure 14-6. Sources of Internal Reset

Table 14-2. Reset Recovery

Reset Recovery Type	Actual Number of Cycles
POR/LVI	4163 (4096 + 64 + 3)
All others	67 (64 + 3)

14.3.2.1 Power-On Reset

When power is first applied to the MCU, the power-on reset module (POR) generates a pulse to indicate that power-on has occurred. The external reset pin (\overline{RST}) is held low while the SIM counter counts out 4096 + 32 CGMXCLK cycles. Thirty-two CGMXCLK cycles later, the CPU and memories are released from reset to allow the reset vector sequence to occur.

At power-on, these events occur:

- A POR pulse is generated.
- The internal reset signal is asserted.
- The SIM enables CGMOUT.
- Internal clocks to the CPU and modules are held inactive for 4096 CGMXCLK cycles to allow stabilization of the oscillator.
- The \overline{RST} pin is driven low during the oscillator stabilization time.
- The POR bit of the SIM reset status register (SRSR) is set.

14.3.2.2 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) if the COPD bit in the CONFIG1 register is cleared. The SIM actively pulls down the \overline{RST} pin for all internal reset sources.

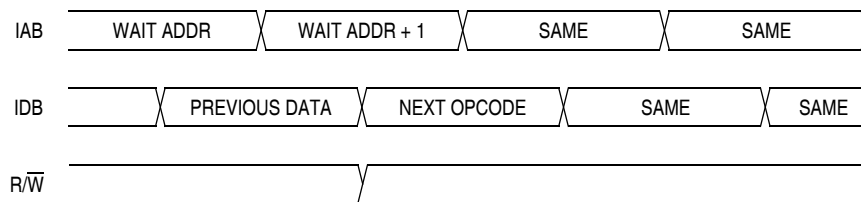
The COP module is disabled if the \overline{RST} pin or the \overline{IRQ} pin is held at V_{TST} while the MCU is in monitor mode. During a break state, V_{TST} on the \overline{RST} pin disables the COP module.

14.6.1 Wait Mode

In wait mode, the CPU clocks are inactive while the peripheral clocks continue to run. Figure 14-16 shows the timing for wait mode entry.

A module that is active during wait mode can wakeup 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.

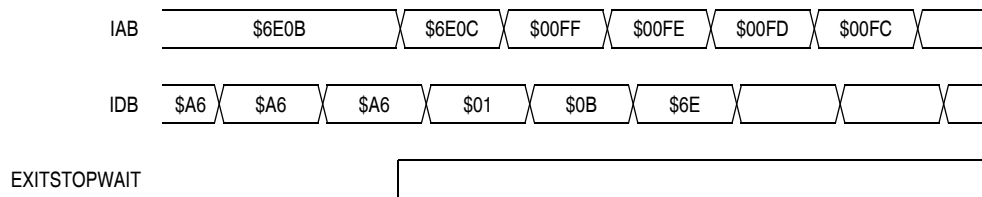
Wait mode also can be exited by a reset or break. A break interrupt during wait mode sets the SIM break stop/wait bit, SBSW, in the SIM break status register (BSR). If the COP disable bit, COPD, in the CONFIG1 register is 0, then the computer operating properly module (COP) is enabled and remains active in wait mode.



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

Figure 14-16. Wait Mode Entry Timing

Figure 14-17 and Figure 14-18 show the timing for WAIT recovery.



Note: EXITSTOPWAIT = $\overline{\text{RST}}$ pin, CPU interrupt, or break interrupt

Figure 14-17. Wait Recovery from Interrupt or Break

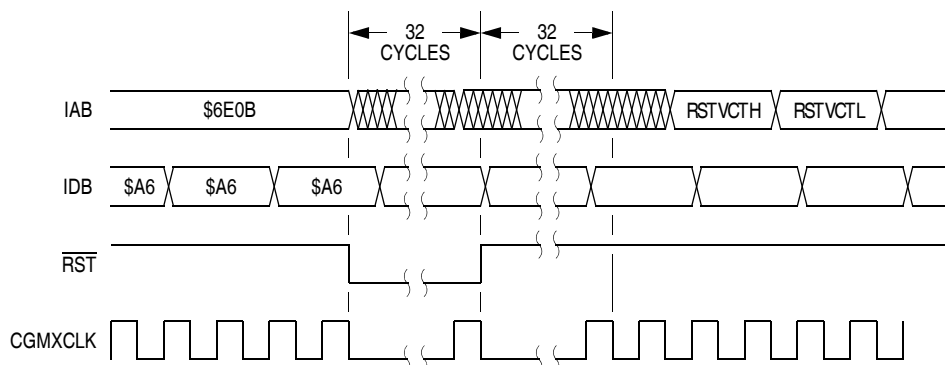


Figure 14-18. Wait Recovery from Internal Reset

The error interrupt enable bit (ERRIE) enables both the MODF and OVRF bits to generate a receiver/error CPU interrupt request.

The mode fault enable bit (MODFEN) can prevent the MODF flag from being set so that only the OVRF bit is enabled by the ERRIE bit to generate receiver/error CPU interrupt requests.

The following sources in the SPI status and control register can generate CPU interrupt requests:

- SPI receiver full bit (SPRF) — SPRF becomes set every time a byte transfers from the shift register to the receive data register. If the SPI receiver interrupt enable bit, SPRIE, is also set, SPRF generates an SPI receiver/error CPU interrupt request.
- SPI transmitter empty (SPTE) — SPTE becomes set every time a byte transfers from the transmit data register to the shift register. If the SPI transmit interrupt enable bit, SPTIE, is also set, SPTE generates an SPTE CPU interrupt request.

15.8 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is 0. Whenever SPE is 0, the following occurs:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.
- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is defaulted back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR register
- All control bits in the SPSCR register (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

15.9 Low-Power Modes

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

15.9.1 Wait Mode

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

If SPI module functions are not required during wait mode, reduce power consumption by disabling the SPI module before executing the WAIT instruction.

To exit wait mode when an overflow condition occurs, enable the OVRF bit to generate CPU interrupt requests by setting the error interrupt enable bit (ERRIE). See 15.7 Interrupts.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM2 overflow interrupts and write the new value in the TIM2 overflow interrupt routine. The TIM2 overflow interrupt occurs at the end of the current PWM period. Writing a larger value in an output compare interrupt routine (at the end of the current pulse) could cause two output compares to occur in the same PWM period.

NOTE

In PWM signal generation, do not program the PWM channel to toggle on output compare. Toggling on output compare prevents reliable 0% duty cycle generation and removes the ability of the channel to self-correct in the event of software error or noise. Toggling on output compare also can cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

18.3.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the T2CH0 pin. The TIM2 channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIM2 channel 0 status and control register (T2SC0) links channel 0 and channel 1. The TIM2 channel 0 registers initially control the pulse width on the T2CH0 pin. Writing to the TIM2 channel 1 registers enables the TIM2 channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM2 channel registers (0 or 1) that control the pulse width are the ones written to last. T2SC0 controls and monitors the buffered PWM function, and TIM2 channel 1 status and control register (T2SC1) is unused. While the MS0B bit is set, the channel 1 pin, T2CH1, is available as a general-purpose I/O pin.

Channels 2 and 3 can be linked to form a buffered PWM channel whose output appears on the T2CH2 pin. The TIM2 channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS2B bit in TIM2 channel 2 status and control register (T2SC2) links channel 2 and channel 3. The TIM2 channel 2 registers initially control the pulse width on the T2CH2 pin. Writing to the TIM2 channel 3 registers enables the TIM2 channel 3 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM2 channel registers (2 or 3) that control the pulse width are the ones written to last. T2SC2 controls and monitors the buffered PWM function, and TIM2 channel 3 status and control register (T2SC3) is unused. While the MS2B bit is set, the channel 3 pin, T2CH3, is available as a general-purpose I/O pin.

Channels 4 and 5 can be linked to form a buffered PWM channel whose output appears on the T2CH4 pin. The TIM2 channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS4B bit in TIM2 channel 4 status and control register (T2SC4) links channel 4 and channel 5. The TIM2 channel 4 registers initially control the pulse width on the T2CH4 pin. Writing to the TIM2 channel 5 registers enables the TIM2 channel 5 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIM2 channel registers (4 or 5) that control the pulse width are the ones written to last. T2SC4 controls and monitors the buffered PWM function, and TIM2 channel 5 status and control register (T2SC5) is unused. While the MS4B bit is set, the channel 5 pin, T2CH5, is available as a general-purpose I/O pin.

Table 19-4. WRITE (Write Memory) Command

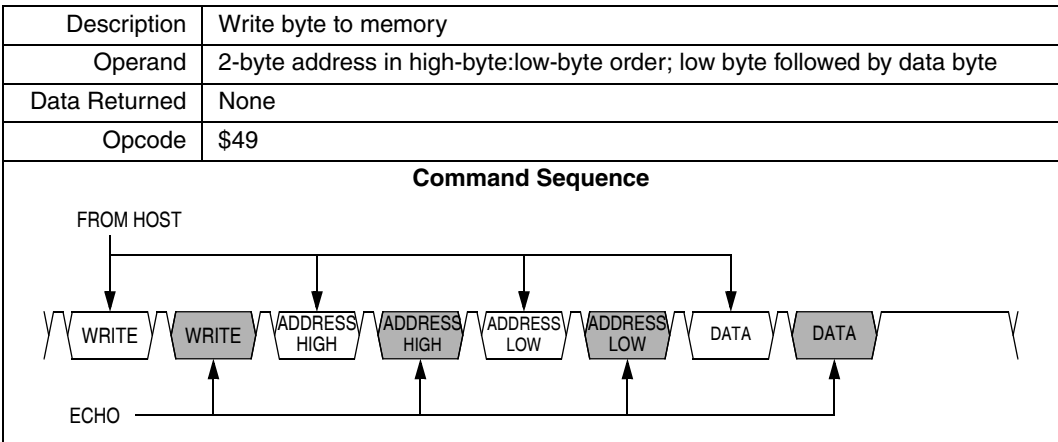


Table 19-5. IREAD (Indexed Read) Command

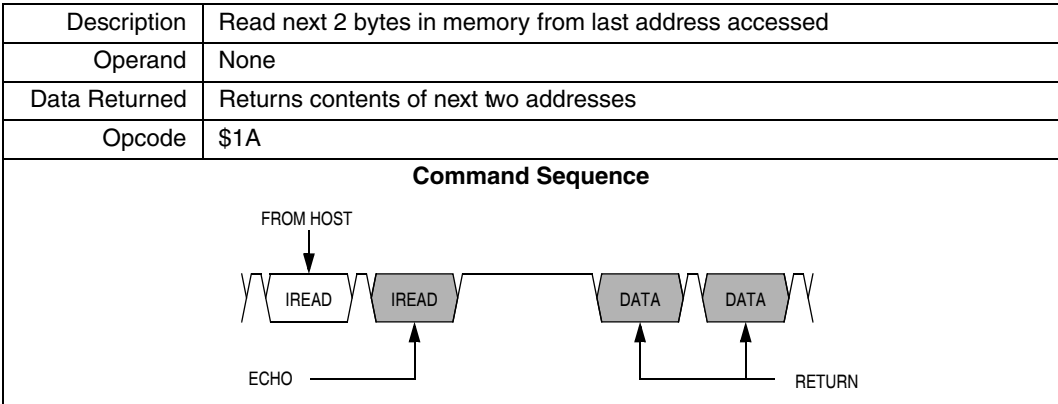
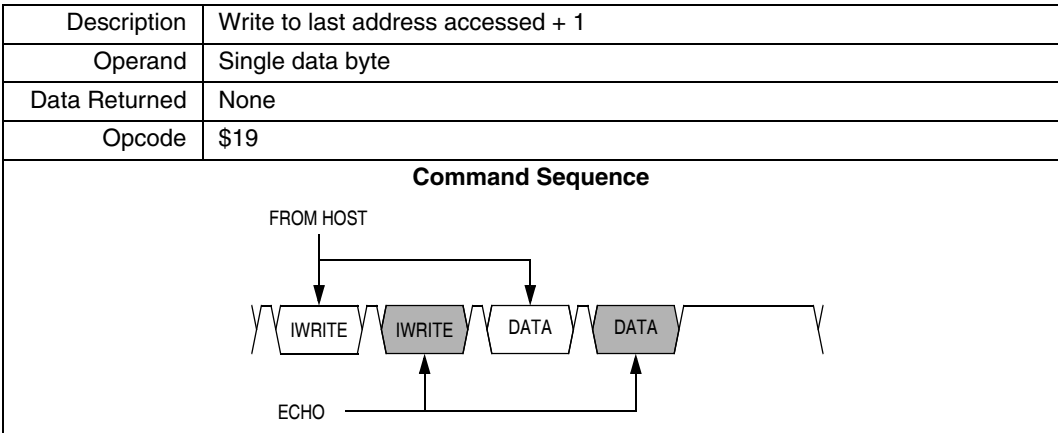


Table 19-6. IWRITE (Indexed Write) Command



A sequence of IREAD or IWRITE commands can access a block of memory sequentially over the full 64-Kbyte memory map.



NOTES:

- 1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
- 2. CONTROLLING DIMENSION: MILLIMETER.
- 3. DATUM PLANE –H– 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 A–B AND –D– TO BE DETERMINED AT DATUM PLANE –H–.

- 5. DIMENSIONS TO BE DETERMINED AT SEATING PLANE –C–.
- 6. DIMENSIONS DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. DIMENSIONS DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE –H–.
- 7. DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL BE 0.08mm TOTAL IN EXCESS OF THE DIMENSION AT MAXIMUM MATERIAL CONDICTION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.		MECHANICAL OUTLINE		PRINT VERSION NOT TO SCALE	
TITLE: 64LD QFP (14 X 14)			DOCUMENT NO: 98ASB42844B		REV: A
			CASE NUMBER: 840B-02		06 APR 2005
			STANDARD: NON-JEDEC		