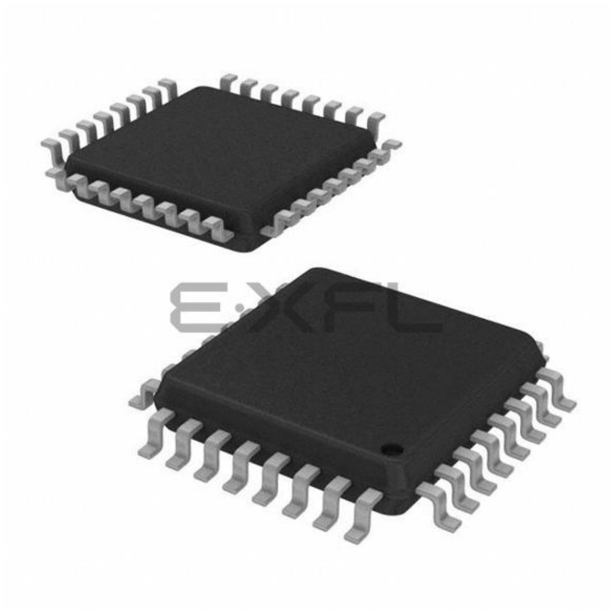


Welcome to E-XFL.COM

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	Not For New Designs
Core Processor	HC08
Core Size	8-Bit
Speed	8MHz
Connectivity	CANbus, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM
Number of I/O	21
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 8x10b
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/mc68908gz16cfje

18.4 Interrupts 251
18.5 TBM Interrupt Rate 252
18.6 Low-Power Modes 253
18.6.1 Wait Mode 253
18.6.2 Stop Mode 253
18.7 Timebase Control Register 254

Chapter 19
Timer Interface Module (TIM)

19.1 Introduction 255
19.2 Features 257
19.3 Pin Name Conventions 257
19.4 Functional Description 257
19.4.1 TIM Counter Prescaler 259
19.4.2 Input Capture 259
19.4.3 Output Compare 260
19.4.3.1 Unbuffered Output Compare 260
19.4.3.2 Buffered Output Compare 260
19.4.4 Pulse Width Modulation (PWM) 261
19.4.4.1 Unbuffered PWM Signal Generation 261
19.4.4.2 Buffered PWM Signal Generation 262
19.4.4.3 PWM Initialization 262
19.5 Interrupts 263
19.6 Low-Power Modes 263
19.6.1 Wait Mode 263
19.6.2 Stop Mode 264
19.7 TIM During Break Interrupts 264
19.8 I/O Signals 264
19.9 I/O Registers 264
19.9.1 TIM Status and Control Register 265
19.9.2 TIM Counter Registers 266
19.9.3 TIM Counter Modulo Registers 267
19.9.4 TIM Channel Status and Control Registers 267
19.9.5 TIM Channel Registers 270

Chapter 20
Development Support

20.1 Introduction 271
20.2 Break Module (BRK) 271
20.2.1 Functional Description 271
20.2.1.1 Flag Protection During Break Interrupts 273
20.2.1.2 TIM During Break Interrupts 273
20.2.1.3 COP During Break Interrupts 273
20.2.2 Break Module Registers 273
20.2.2.1 Break Status and Control Register 274
20.2.2.2 Break Address Registers 274
20.2.2.3 Break Status Register 275
20.2.2.4 Break Flag Control Register 275

20.2.3 Low-Power Modes 275
20.3 Monitor ROM (MON) 275
20.3.1 Functional Description 276
20.3.1.1 Normal Monitor Mode 280
20.3.1.2 Forced Monitor Mode 280
20.3.1.3 Monitor Vectors 280
20.3.1.4 Data Format 281
20.3.1.5 Break Signal 281
20.3.1.6 Baud Rate 281
20.3.1.7 Commands 281
20.3.2 Security 285

Chapter 21
Electrical Specifications

21.1 Introduction 287
21.2 Absolute Maximum Ratings 287
21.3 Functional Operating Range 288
21.4 Thermal Characteristics 288
21.5 5-Vdc Electrical Characteristics 289
21.6 3.3-Vdc Electrical Characteristics 291
21.7 5.0-Volt Control Timing 293
21.8 3.3-Volt Control Timing 293
21.9 Clock Generation Module Characteristics 294
21.9.1 CGM Component Specifications 294
21.9.2 CGM Electrical Specifications 294
21.10 5.0-Volt ADC Characteristics 295
21.11 3.3-Volt ADC Characteristics 296
21.12 5.0-Volt SPI Characteristics 297
21.13 3.3-Volt SPI Characteristics 298
21.14 Timer Interface Module Characteristics 301
21.15 Memory Characteristics 302

Chapter 22
Ordering Information and Mechanical Specifications

22.1 Introduction 303
22.2 MC Order Numbers 303
22.3 Package Dimensions 303

Appendix A
MC68HC908GZ8

A.1 Introduction 311
A.2 Block Diagram 311
A.3 Memory 311
A.4 Ordering Information 314

Chapter 1
General Description

1.1 Introduction

The MC68HC908GZ16 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	Memory Size
MC68HC908QZ16	16 Kbytes user FLASH
MC68HC908GZ8	8 Kbytes user FLASH

The information contained in this document pertains to both the MC68HC908GZ16 and the MC68HC908GZ8 with the exceptions shown Appendix A MC68HC908GZ8

1.2 Features

For convenience, features have been organized to reflect:
- Standard features
- Features of the CPU08

1.2.1 Standard Features

Features include:
- High-performance M68HC08 architecture optimized for C-compilers
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- 8-MHz internal bus frequency
- Clock generation module supporting 1-MHz to 8-MHz crystals
- MSCAN08 (implementing 2.0b protocol as defined in BOSCH specification dated September 1991)
- FLASH program memory security[1]
- On-chip programming firmware for use with host personal computer which does not require high voltage for entry
- In-system programming (ISP)

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0000	Port A Data Register (PTA) See page 158.	Read:	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		Write:								
		Reset:	Unaffected by reset							
$0001	Port B Data Register (PTB) See page 160.	Read:	PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
		Write:								
		Reset:	Unaffected by reset							
$0002	Port C Data Register (PTC) See page 162.	Read:	1	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
		Write:								
		Reset:	Unaffected by reset							
$0003	Port D Data Register (PTD) See page 164.	Read:	PTD7	PTD6	PTD5	PTD4	PTD3	PTD2	PTD1	PTD0
		Write:								
		Reset:	Unaffected by reset							
$0004	Data Direction Register A (DDRA) See page 158.	Read:	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0005	Data Direction Register B (DDRB) See page 161.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC) See page 162.	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0007	Data Direction Register D (DDRD) See page 165.	Read:	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0008	Port E Data Register (PTE) See page 167.	Read:	0	0	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$0009	ESCI Prescaler Register (SCPSC) See page 206.	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 209.	Read:	AM1	ALOST	AM0	ACLK	AFIN	ARUN	AOVFL	ARD8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000B	ESCI Arbiter Data Register (SCIADAT) See page 210.	Read:	ARD7	ARD6	ARD5	ARD4	ARD3	ARD2	ARD1	ARD0
		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 8)

3.8.2.3 Left Justified Signed Data Mode

In left justified signed data mode, the ADRH register holds the eight MSBs of the 10-bit result. The only difference from left justified mode is that the AD9 is complemented. The ADRL register holds the two LSBs of the 10-bit result. All other bits read as 0. ADRH and ADRL are updated each time an ADC single channel conversion completes. Reading ADRH latches the contents of ADRL until ADRL is read. All subsequent results will be lost until the ADRH and ADRL reads are completed.

Address: $003D	Bit 7	6	5	4	3	2	1	Bit 0
Read:	$\overline{AD9}$	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							
Address: $003E								
Read:	AD1	AD0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-7. ADC Data Register High (ADRH) and Low (ADRL)

3.8.2.4 Eight Bit Truncation Mode

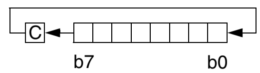
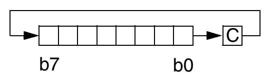
In 8-bit truncation mode, the ADRL register holds the eight MSBs of the 10-bit result. The ADRH register is unused and reads as 0. The ADRL register is updated each time an ADC single channel conversion completes. In 8-bit mode, the ADRL register contains no interlocking with ADRH.

Address: $003D	Bit 7	6	5	4	3	2	1	Bit 0 (ADRH)
Read:	0	0	0	0	0	0	0	0
Write:								
Reset:	Unaffected by reset							
Address: $003E								ADRL
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:								
Reset:	Unaffected by reset							

= Unimplemented

Figure 3-8. ADC Data Register High (ADRH) and Low (ADRL)

Table 7-1. Instruction Set Summary (Sheet 5 of 6)

Source Form	Operation	Description	Effect on CCR						Address Mode	Opcode	Operand	Cycles
			V	H	I	N	Z	C				
PULA	Pull A from Stack	SP ← (SP + 1); Pull (A)	–	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	SP ← (SP + 1); Pull (H)	–	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	SP ← (SP + 1); Pull (X)	–	–	–	–	–	–	INH	88		2
ROL *opr* ROLA ROLX ROL *opr*,X ROL ,X ROL *opr*,SP	Rotate Left through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	39 49 59 69 79 9E69	dd ff ff	4 1 1 4 3 5
ROR *opr* RORA RORX ROR *opr*,X ROR ,X ROR *opr*,SP	Rotate Right through Carry		↕	–	–	↕	↕	↕	DIR INH INH IX1 IX SP1	36 46 56 66 76 9E66	dd ff ff	4 1 1 4 3 5
RSP	Reset Stack Pointer	SP ← \$FF	–	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	SP ← (SP) + 1; Pull (CCR) SP ← (SP) + 1; Pull (A) SP ← (SP) + 1; Pull (X) SP ← (SP) + 1; Pull (PCH) SP ← (SP) + 1; Pull (PCL)	↕	↕	↕	↕	↕	↕	INH	80		7
RTS	Return from Subroutine	SP ← SP + 1; Pull (PCH) SP ← SP + 1; Pull (PCL)	–	–	–	–	–	–	INH	81		4
SBC #*opr* SBC *opr* SBC *opr* SBC *opr*,X SBC *opr*,X SBC ,X SBC *opr*,SP SBC *opr*,SP	Subtract with Carry	A ← (A) – (M) – (C)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	C ← 1	–	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	I ← 1	–	–	1	–	–	–	INH	9B		2
STA *opr* STA *opr* STA *opr*,X STA *opr*,X STA ,X STA *opr*,SP STA *opr*,SP	Store A in M	M ← (A)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
STHX *opr*	Store H:X in M	(M:M + 1) ← (H:X)	0	–	–	↕	↕	–	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	I ← 0; Stop Processing	–	–	0	–	–	–	INH	8E		1
STX *opr* STX *opr* STX *opr*,X STX *opr*,X STX ,X STX *opr*,SP STX *opr*,SP	Store X in M	M ← (X)	0	–	–	↕	↕	–	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ee ff	3 4 4 3 2 4 5
SUB #*opr* SUB *opr* SUB *opr* SUB *opr*,X SUB *opr*,X SUB ,X SUB *opr*,SP SUB *opr*,SP	Subtract	A ← (A) – (M)	↕	–	–	↕	↕	↕	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ee ff	2 3 4 4 3 2 4 5

8.6 IRQ Status and Control Register

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

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

Address: $001D

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

(shaded) = Unimplemented

Figure 8-3. IRQ Status and Control Register (INTSCR)

IRQF — IRQ Flag Bit

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

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

ACK — IRQ Interrupt Request Acknowledge Bit

Writing a logic 1 to this write-only bit clears the IRQ latch. ACK always reads as logic 0. Reset clears ACK.

IMASK — IRQ Interrupt Mask Bit

Writing a logic 1 to this read/write bit disables IRQ interrupt requests. Reset clears IMASK.

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

MODE — IRQ Edge/Level Select Bit

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

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

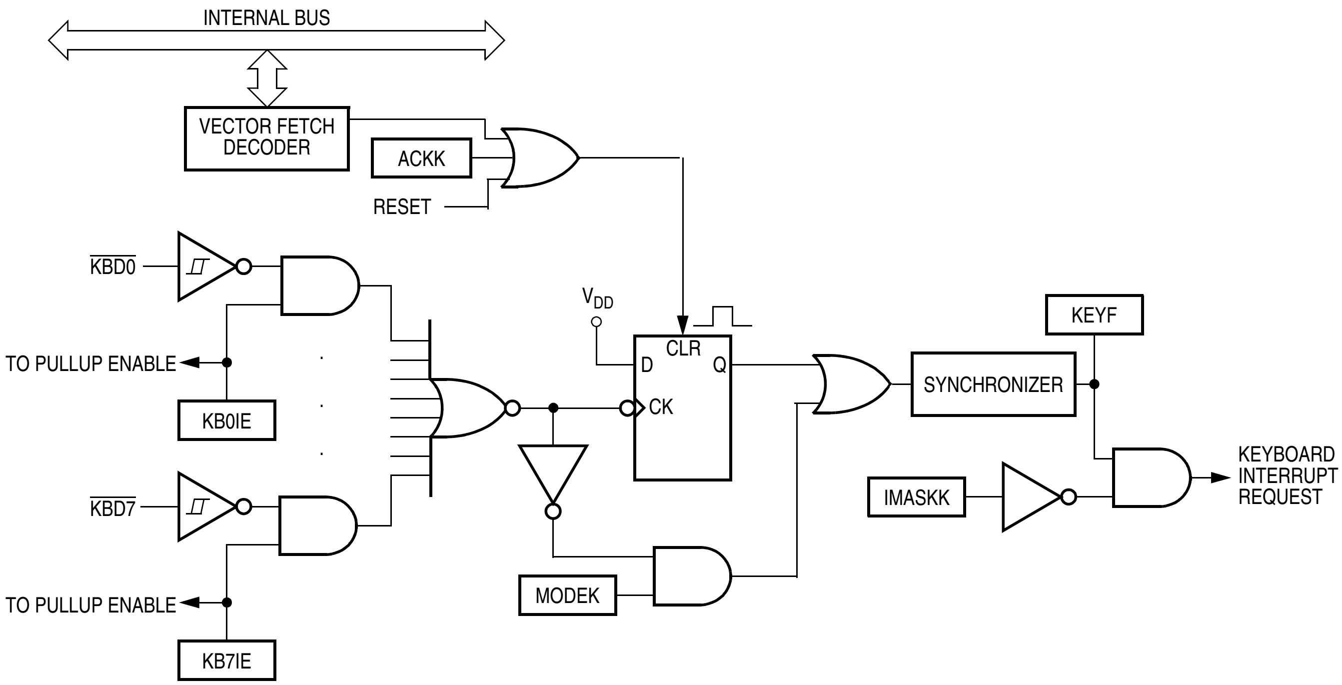


Figure 9-2. Keyboard Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$001A	Keyboard Status and Control Register (INTKBSCR) See page 107.	Read:	0	0	0	0	KEYF	0	IMASKK	MODEK
		Write:						ACKK		
		Reset:	0	0	0	0	0	0	0	0
$001B	Keyboard Interrupt Enable Register (INTKBIER) See page 108.	Read: / Write:	KBIE7	KBIE6	KBIE5	KBIE4	KBIE3	KBIE2	KBIE1	KBIE0
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 9-3. I/O Register Summary

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

- Vector fetch or software clear — A vector fetch generates an interrupt acknowledge signal to clear the interrupt request. Software may generate the interrupt acknowledge signal by writing a logic 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 edge that occurs after writing to the ACKK bit latches another interrupt request. If the keyboard interrupt mask bit, IMASKK, is clear, the CPU loads the program counter with the vector address at locations $FFE0 and $FFE1.
- Return of all enabled keyboard interrupt pins to logic 1 — As long as any enabled keyboard interrupt pin is at logic 0, the keyboard interrupt remains set.

10.9 Low-Voltage Inhibit Module (LVI)

10.9.1 Wait Mode

If enabled, the low-voltage inhibit (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.

10.9.2 Stop Mode

If enabled, 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.

10.10 Enhanced Serial Communications Interface Module (ESCI)

10.10.1 Wait Mode

The enhanced serial communications interface (ESCI), or SCI module for short, module remains active in wait mode. Any enabled CPU interrupt request from the SCI module can bring the MCU out of wait mode.

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

10.10.2 Stop Mode

The SCI module is inactive in stop mode. The STOP instruction does not affect SCI register states. SCI module operation resumes after the MCU exits stop mode.

Because the internal clock is inactive during stop mode, entering stop mode during an SCI transmission or reception results in invalid data.

10.11 Serial Peripheral Interface Module (SPI)

10.11.1 Wait Mode

The serial peripheral interface (SPI) module remains active in wait mode. 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.

10.11.2 Stop Mode

The SPI module is inactive in stop mode. The STOP instruction does not affect SPI register states. SPI operation resumes after an external interrupt. If stop mode is exited by reset, any transfer in progress is aborted, and the SPI is reset.

Table 13-1. Port Control Register Bits Summary

Port	Bit	DDR	Module Control		Pin
A	0	DDRA0	KBD	KBIE0	PTA0/KBD0
	1	DDRA1		KBIE1	PTA1/KBD1
	2	DDRA2		KBIE2	PTA2/KBD2
	3	DDRA3		KBIE3	PTA3/KBD3
	4	DDRA4		KBIE4	PTA4/KBD4
	5	DDRA5		KBIE5	PTA5/KBD5
	6	DDRA6		KBIE6	PTA6/KBD6
	7	DDRA7		KBIE7	PTA7/KBD7
B	0	DDRB0	ADC	ADCH4–ADCH0	PTB0/AD0
	1	DDRB1			PTB1/AD1
	2	DDRB2			PTB2/AD2
	3	DDRB3			PTB3/AD3
	4	DDRB4			PTB4/AD4
	5	DDRB5			PTB5/AD5
	6	DDRB6			PTB6/AD6
	7	DDRB7			PTB7/AD7
C	0	DDRC0	MSCAN08	CANEN	PTC0
	1	DDRC1			PTC1
	2	DDRC2			PTC2
	3	DDRC3			PTC3
	4	DDRC4			PTC4
	5	DDRC5			PTC5
	6	DDRC6			PTC6
D	0	DDRD0	SPI	SPE	PTD0/$\overline{SS}$
	1	DDRD1			PTD1/MISO
	2	DDRD2			PTD2/MOSI
	3	DDRD3			PTD3/SPSCK
	4	DDRD4	TIM1	ELS0B:ELS0A	PTD4/T1CH0
	5	DDRD5		ELS1B:ELS1A	PTD5/T1CH1
	6	DDRD6	TIM2	ELS0B:ELS0A	PTD6/T2CH0
	7	DDRD7		ELS1B:ELS1A	PTD7/T2CH1
E	0	DDRE0	SCI	ENSCI	PTE0/TxD
	1	DDRE1			PTE1/RxD
	2	DDRE2			PTE2
	3	DDRE3			PTE3
	4	DDRE4			PTE4
	5	DDRE5			PTE5

After every instruction, the CPU checks all pending interrupts if the I bit is not set. If more than one interrupt is pending when an instruction is done, the highest priority interrupt is serviced first. In the example shown in Figure 14-4, if an interrupt is pending upon exit from the interrupt service routine, the pending interrupt is serviced before the LDA instruction is executed.

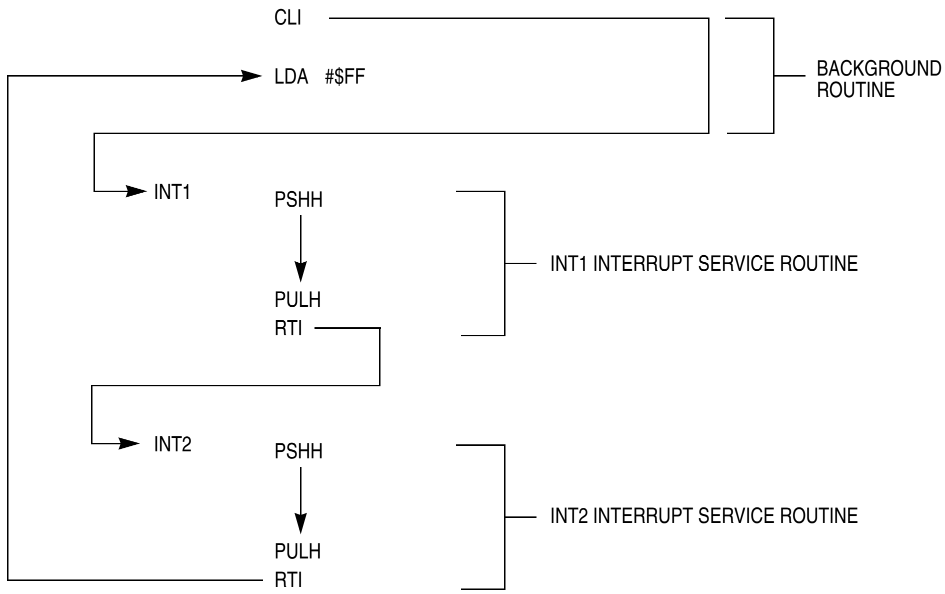


Figure 14-4. Interrupt Recognition Example

The LDA opcode is prefetched by both the INT1 and INT2 RTI instructions. However, in the case of the INT1 RTI prefetch, this is a redundant operation.

NOTE

To maintain compatibility with the M6805 Family, the H register is not pushed on the stack during interrupt entry. If the interrupt service routine modifies the H register or uses the indexed addressing mode, save the H register and then restore it prior to exiting the routine.

See Figure 14-5 for a flowchart depicting interrupt processing.

14.3.2 Sources

The sources in Table 14-1 can generate CPU interrupt requests.

14.3.2.1 Software Interrupt (SWI) Instruction

The software interrupt (SWI) instruction causes a non-maskable interrupt.

NOTE

A software interrupt pushes PC onto the stack. An SWI does ***not*** *push PC – 1, as a hardware interrupt does.*

14.3.2.2 Break Interrupt

The break module causes the CPU to execute an SWI instruction at a software-programmable break point.

14.3.2.11 Timebase Module (TBM)

The timebase module can interrupt the CPU on a regular basis with a rate defined by TBR2–TBR0. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request.

Interrupts must be acknowledged by writing a logic 1 to the TACK bit.

14.3.2.12 MSCAN

MSCAN08 interrupt sources:

- MSCAN08 transmitter empty bits (TXE0–TXE2) — The TXEx bit is set when the corresponding MSCAN08 data buffer is empty. The MSCAN08 transmit interrupt enable bits, TXEIE0–TXEIE2, enables transmitter CPU interrupt requests. TXEx is in MSCAN08 transmitter flag register. TXEIEx is in MSCAN08 transmitter control register.
- MSCAN08 receiver full bit (RXF) — The RXF bit is set when the a MSCAN08 message has been successfully received and loaded into the foreground receive buffer. The MSCAN08 receive interrupt enable bit, RXFIE, enables receiver CPU interrupt requests. RXF is in MSCAN08 receiver flag register. RXFIE is in MSCAN08 receiver interrupt enable register.
- MSCAN08 wakeup bit (WUPIF) — WUPIF is set when activity on the CAN bus occurred during the MSCAN08 internal sleep mode. The wakeup interrupt enable bit, WUPIE, enables MSCAN08 wakeup CPU interrupt requests. WUPIF is in MSCAN08 receiver flag register. WUPIE is in MSCAN08 receiver interrupt enable register.
- Overrun bit (OVRIF) — OVRIF is set when both the foreground and the background receive message buffers are filled with correctly received messages and a further message is being received from the bus. The overrun interrupt enable bit, OVRIE, enables OVRIF to generate MSCAN08 error CPU interrupt requests. OVRIF is in MSCAN08 receiver flag register. OVRIE is in MSCAN08 receiver interrupt enable register.
- Receiver Warning bit (RWRNIF) — RWRNIF is set when the receive error counter has reached the CPU warning limit of 96. The receiver warning interrupt enable bit, RWRNIE, enables RWRNIF to generate MSCAN08 error CPU interrupt requests. RWRNIF is in MSCAN08 receiver flag register. RWRNIE is in MSCAN08 receiver interrupt enable register.
- Transmitter Warning bit (TWRNIF) — TWRNIF is set when the transmit error counter has reached the CPU warning limit of 96. The transmitter warning interrupt enable bit, TWRNIF, enables TWRNIF to generate MSCAN08 error CPU interrupt requests. TWRNIF is in MSCAN08 receiver flag register. TWRNIE is in MSCAN08 receiver interrupt enable register.
- Receiver Error Passive bit (RERRIF) — RERRIF is set when the receive error counter has exceeded the error passive limit of 127 and the MSCAN08 has gone to error passive state. The receiver error passive interrupt enable bit, RERRIE, enables RERRIF to generate MSCAN08 error CPU interrupt requests. RERRIF is in MSCAN08 receiver flag register. RERRIE is in MSCAN08 receiver interrupt enable register.
- Transmitter Error Passive bit (TERRIF) — TERRIF is set when the transmit error counter has exceeded the error passive limit of 127 and the MSCAN08 has gone to error passive state. The transmit error passive interrupt enable bit, TERRIE, enables TERRIF to generate MSCAN08 error CPU interrupt requests. TERRIF is in MSCAN08 receiver flag register. TERRIE is in MSCAN08 receiver interrupt enable register.

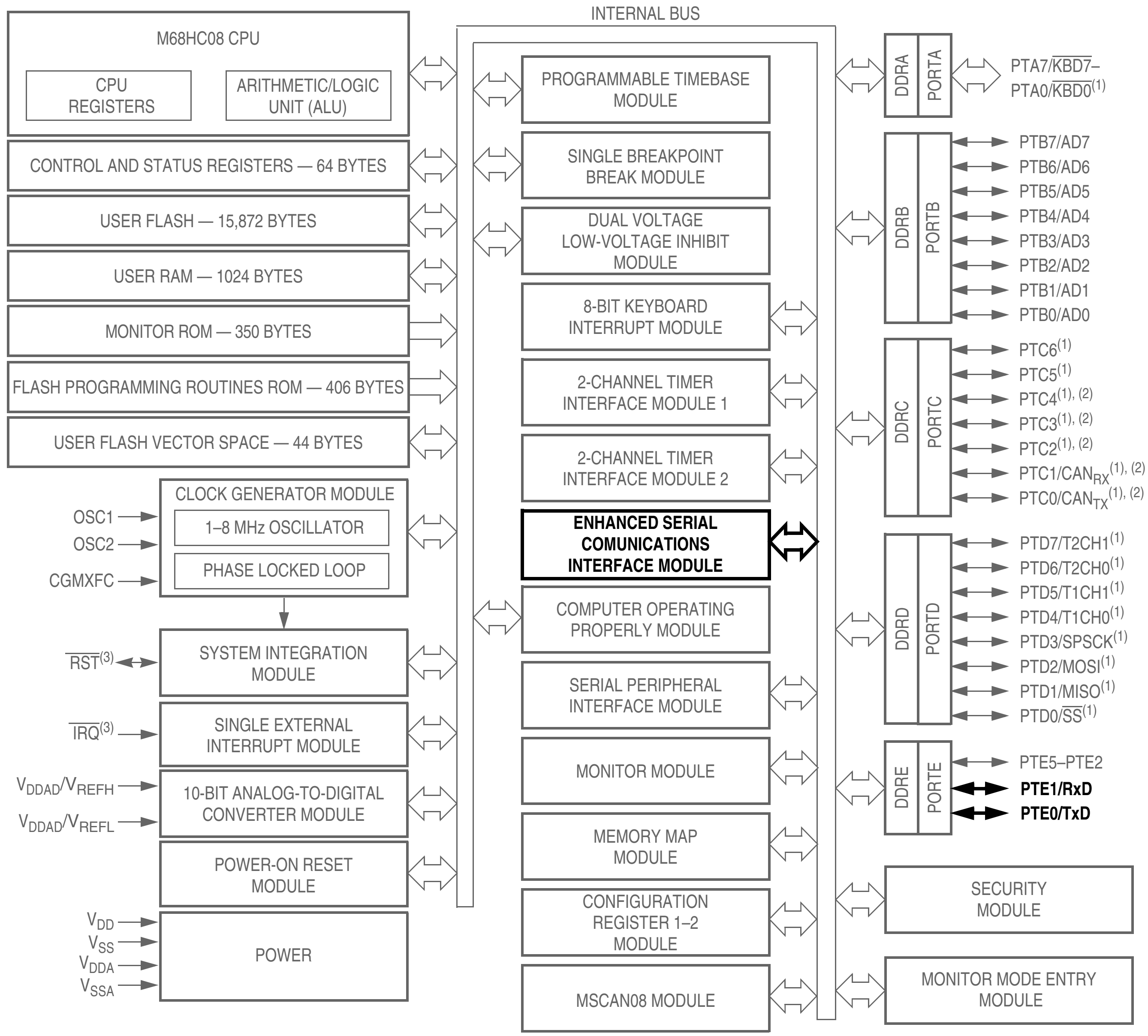
- Bus Off bit (BOFFIF) — BOFFIF is set when the transmit error counter has exceeded 255 and MSCAN08 has gone to bus off state. The bus off interrupt enable bit, BOFFIE, enables BOFFIF to generate MSCAN08 error CPU interrupt requests. BOFFIF is in MSCAN08 receiver flag register. BOFFIE is in MSCAN08 receiver interrupt enable register.

14.3.3 Interrupt Status Registers

The flags in the interrupt status registers identify maskable interrupt sources. Table 14-2 summarizes the interrupt sources and the interrupt status register flags that they set. The interrupt status registers can be useful for debugging.

Table 14-2. Interrupt Source Flags

Interrupt Source	Interrupt Status Register Flag
Reset	—
SWI instruction	—
$\overline{\text{IRQ}}$ pin	IF1
CGM change of lock	IF2
TIM1 channel 0	IF3
TIM1 channel 1	IF4
TIM1 overflow	IF5
TIM2 channel 0	IF6
TIM2 channel 1	IF7
TIM2 overflow	IF8
SPI receive	IF9
SPI transmit	IF10
SCI error	IF11
SCI receive	IF12
SCI transmit	IF13
Keyboard	IF14
ADC conversion complete	IF15
Timebase	IF16
MSCAN08 wakeup	IF17
MSCAN08 error	IF18
MSCAN08 receive	IF19
MSCAN08 transmit	IF20



1. Ports are software configurable with pullup device if input port.
2. Higher current drive port pins
3. Pin contains integrated pullup device

Figure 15-1. Block Diagram Highlighting ESCI Block and Pins

NF — Receiver Noise Flag Bit

This clearable, read-only bit is set when the ESCI detects noise on the RxD 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 logic 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

15.8.5 ESCI Status Register 2

ESCI status register 2 (SCS2) contains flags to signal these conditions:

- Break character detected
- Incoming data

Address: $0017

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

= Unimplemented

Figure 15-15. ESCI Status Register 2 (SCS2)

BKF — Break Flag Bit

This clearable, read-only bit is set when the ESCI detects a break character on the RxD pin. In SCS1, the FE and SCRF bits are also set. In 9-bit character transmissions, the R8 bit in SCC3 is cleared. BKF does not generate a CPU interrupt request. Clear BKF by reading SCS2 with BKF set and then reading the SCDR. Once cleared, BKF can become set again only after logic 1s again appear on the RxD pin followed by another break character. Reset clears the BKF bit.

1 = Break character detected
0 = No break character detected

RPF — Reception in Progress Flag Bit

This read-only bit is set when the receiver detects a logic 0 during the RT1 time period of the start bit search. RPF does not generate an interrupt request. RPF is reset after the receiver detects false start bits (usually from noise or a baud rate mismatch), or when the receiver detects an idle character. Polling RPF before disabling the ESCI module or entering stop mode can show whether a reception is in progress.

1 = Reception in progress
0 = No reception in progress

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
- Modular architecture expandable to 128 interrupt sources

Table 16-1 shows the internal signal names used in this section.

Table 16-1. Signal Name Conventions

Signal Name	Description
CGMXCLK	Buffered version of OSC1 from clock generator module (CGM)
CGMVCLK	PLL output
CGMOUT	PLL-based or OSC1-based clock output from CGM module (Bus clock = CGMOUT divided by two)
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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$FE00	SIM Break Status Register (SBSR) See page 228.	Read:	R	R	R	R	R	R	SBSW	R
		Write:							Note(1)	
		Reset:	0	0	0	0	0	0	0	0
			1. Writing a logic 0 clears SBSW.							
$FE01	SIM Reset Status Register (SRSR) See page 228.	Read:	POR	PIN	COP	ILOP	ILAD	MODRST	LVI	0
		Write:								
		POR:	1	0	0	0	0	0	0	0
$FE03	SIM Break Flag Control Register (SBFCR) See page 229.	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
$FE04	Interrupt Status Register 1 (INT1) See page 224.	Read:	IF6	IF5	IF4	IF3	IF2	IF1	0	0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE05	Interrupt Status Register 2 (INT2) See page 224.	Read:	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$FE06	Interrupt Status Register 3 (INT3) See page 224.	Read:	0	0	IF20	IF19	IF18	IF17	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0

= Unimplemented   R = Reserved

Figure 16-2. SIM I/O Register Summary

21.3 Functional Operating Range

Characteristic	Symbol	Value	Unit
Operating temperature range	T_A	−40 to +125	°C
Operating voltage range	V_{DD}	5.0 ±10% 3.3 ±10%	V

21.4 Thermal Characteristics

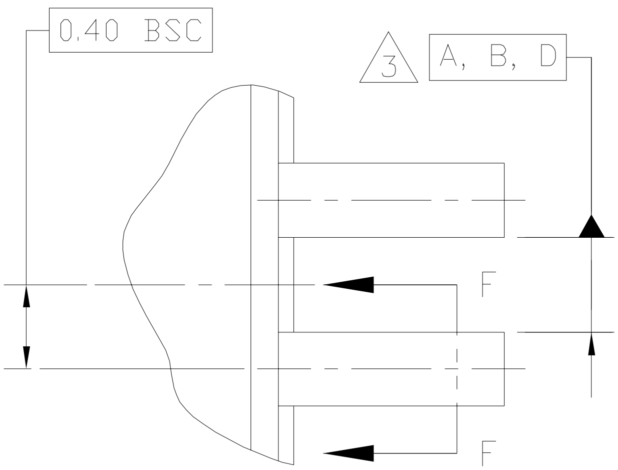
Characteristic	Symbol	Value	Unit
Thermal resistance 32-pin LQFP 48-pin LQFP	θ_{JA}	 95 95	°C/W
I/O pin power dissipation	$P_{I/O}$	User determined	W
Power dissipation(1)	P_D	$P_D = (I_{DD} \times V_{DD}) + P_{I/O} =$ $K/(T_J + 273\ °C)$	W
Constant(2)	K	$P_D \times (T_A + 273\ °C)$ $+ P_D^2 \times \theta_{JA}$	W/°C
Average junction temperature	T_J	$T_A + (P_D \times \theta_{JA})$	°C

1. Power dissipation is a function of temperature.
2. K is a constant unique to the device. K can be determined for a known T_A and measured P_D. With this value of K, P_D and T_J can be determined for any value of T_A.

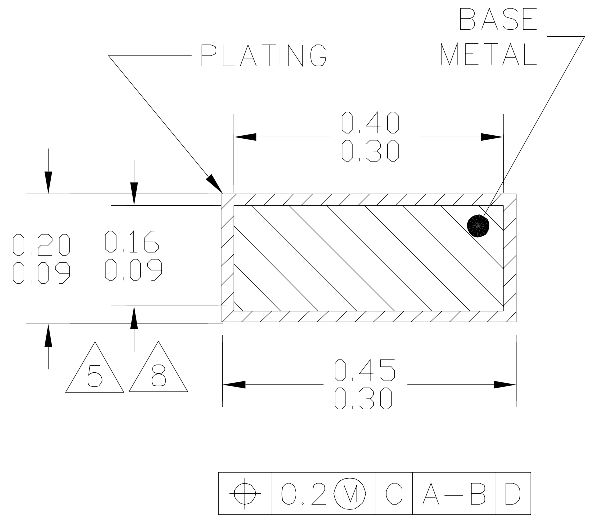
21.5 5-Vdc Electrical Characteristics

Characteristic(1)	Symbol	Min	Typ(2)	Max	Unit
Output high voltage					
(I_{Load} = –2.0 mA) all I/O pins	V_{OH}	$V_{DD} - 0.8$	—	—	V
(I_{Load} = –10.0 mA) all I/O pins	V_{OH}	$V_{DD} - 1.5$	—	—	V
(I_{Load} = –20.0 mA) pins PTC0–PTC4 only	V_{OH}	$V_{DD} - 1.5$	—	—	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0–PTC1, port E, port PTD0–PTD3	I_{OH1}	—	—	50	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2-PTC6, port PTD4–PTD7	I_{OH2}	—	—	50	mA
Maximum total I_{OH} for all port pins	I_{OHT}	—	—	100	mA
Output low voltage					
(I_{Load} = 1.6 mA) all I/O pins	V_{OL}	—	—	0.4	V
(I_{Load} = 10 mA) all I/O pins	V_{OL}	—	—	1.5	V
(I_{Load} = 20mA) pins PTC0–PTC4 only	V_{OL}	—	—	1.5	V
Maximum combined I_{OH} for port PTA7–PTA3, port PTC0-PTC1, port E, port PTD0–PTD3	I_{OL1}	—	—	50	mA
Maximum combined I_{OH} for port PTA2–PTA0, port B, port PTC2–PTC6, port PTD4–PTD7	I_{OL2}	—	—	50	mA
Maximum total I_{OL} for all port pins	I_{OLT}	—	—	100	mA
Input high voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IH}	$0.7 \times V_{DD}$	—	V_{DD}	V
Input low voltage All ports, $\overline{IRQ}$, $\overline{RST}$, OSC1	V_{IL}	V_{SS}	—	$0.2 \times V_{DD}$	V
V_{DD} supply current	I_{DD}				
Run(3)		—	20	30	mA
Wait(4)		—	6	12	mA
Stop(5)		—	0.6	10	µA
Stop with TBM enabled(6)		—	1	1.25	mA
Stop with LVI and TBM enabled(6)		—	1.25	1.6	mA
Stop with LVI		—	250	350	µA
DC injection current(7) (8) (9) (10)	I_{IC}				mA
Single pin limit					
$V_{in} > V_{DD}$		0	—	2	
$V_{in} < V_{SS}$		0	—	–0.2	
Total MCU limit, includes sum of all stressed pins					
$V_{in} > V_{DD}$		0	—	25	
$V_{in} < V_{SS}$		0	—	–5	
I/O ports Hi-Z leakage current(11)	I_{IL}	0	—	±10	µA
Input current	I_{In}	0	—	±1	µA
Pullup resistors (as input only) Ports PTA7/$\overline{KBD7}$–PTA0/$\overline{KBD0}$, PTC6–PTC0/CAN_{TX}, PTD7/T2CH1–PTD0/$\overline{SS}$	R_{PU}	20	45	65	kΩ
Capacitance Ports (as input or output)	C_{Out} C_{In}	— —	— —	12 8	pF

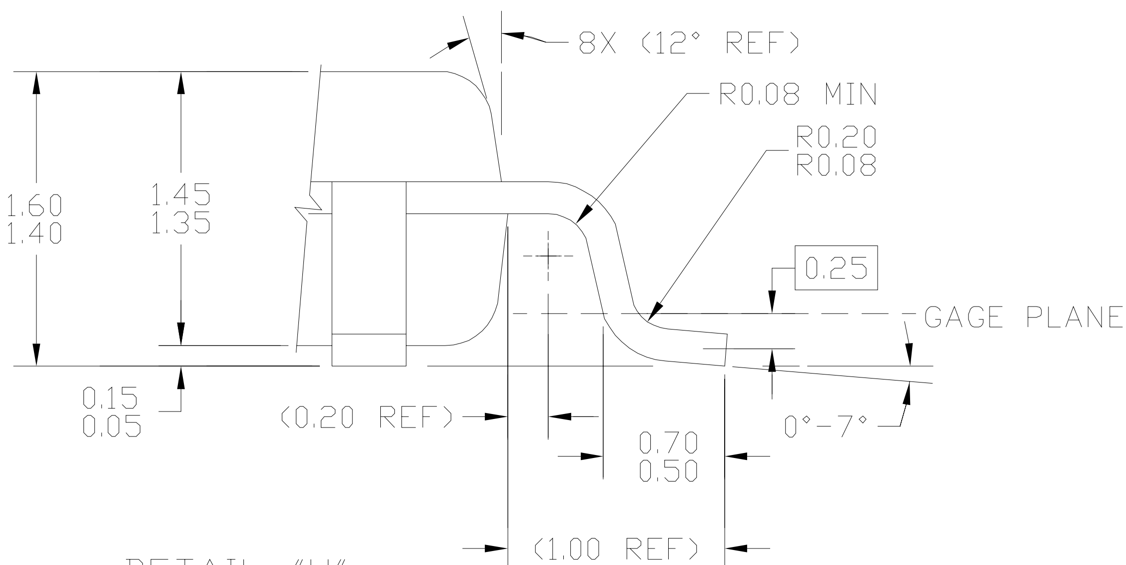
Continued on next page



DETAIL G



SECTION F–F
ROTATED 90°CW
32 PLACES



DETAIL "H"

	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: LOW PROFILE QUAD FLAT PACK (LQFP) 32 LEAD, 0.8 PITCH (7 X 7 X 1.4)	DOCUMENT NO: 98ASH70029A	REV: C
	CASE NUMBER: 873A-04	01 APR 2005
	STANDARD: JEDEC MS-026 BBA	

A.4 Ordering Information

Table A-1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC908GZ8CFJ	–40°C to +85°C	32-pin low-profile quad flat pack (LQFP)
MC908GZ8VFJ	–40°C to +105°C	
MC908GZ8MFJ	–40°C to +125°C	
MC908GZ8CFA	–40°C to +85°C	48-pin low-profile quad flat pack (LQFP)
MC908GZ8VFA	–40°C to +105°C	
MC908GZ8MFA	–40°C to +125°C	

Temperature designators:
C = –40°C to +85°C
V = –40°C to +105°C
M = –40°C to +125°C

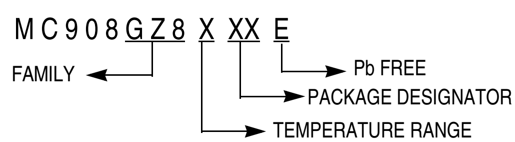


Figure A-3. Device Numbering System