



Welcome to [E-FL.COM](https://www.e-fl.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	17
Program Memory Size	7.5KB (7.5K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 6x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-SOIC (0.295", 7.50mm Width)
Supplier Device Package	28-SOIC
Purchase URL	https://www.e-fl.com/product-detail/nxp-semiconductors/mc908gr8amdwe

Memory

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE05	Interrupt Status Register 2 (INT2) See page 168.	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 168.	Read:	0	0	0	0	0	0	IF16	IF15
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE07	Reserved	Read:	R	R	R	R	R	R	R	R
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
\$FE08	FLASH Control Register (FLCR) See page 39.	Read:	0	0	0	0	HVEN	MASS	ERASE	PGM
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE09	Break Address Register High (BRKH) See page 219.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:	Bit 15	14	13	12	11	10	9	Bit 8
		Reset:	0	0	0	0	0	0	0	0
\$FE0A	Break Address Register Low (BRKL) See page 219.	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:	Bit 7	6	5	4	3	2	1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
\$FE0B	Break Status and Control Register (BRKSCR) See page 219.	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:	BRKE	BRKA						
		Reset:	0	0	0	0	0	0	0	0
\$FE0C	LVI Status Register (LVISR) See page 113.	Read:	LVIOUT	0	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FF7E	FLASH Block Protect Register (FLBPR) ⁽¹⁾ See page 44.	Read:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Write:	BPR7	BPR6	BPR5	BPR4	BPR3	BPR2	BPR1	BPR0
		Reset:	Unaffected by reset							

1. Non-volatile FLASH register

\$FFFF	COP Control Register (COPCTL) See page 81.	Read:	Low byte of reset vector							
		Write:	Writing clears COP counter (any value)							
		Reset:	Unaffected by reset							

= Unimplemented R = Reserved U = Unaffected

Figure 2-2. Control, Status, and Data Registers (Sheet 7 of 7)

3.7.2 ADC Data Register

One 8-bit result register, ADC data register (ADR), is provided. This register is updated each time an ADC conversion completes.

Address: \$003D

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

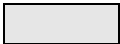
 = Unimplemented

Figure 3-4. ADC Data Register (ADR)

3.7.3 ADC Clock Register

The ADC clock register (ADCLK) selects the clock frequency for the ADC.

Address: \$003E

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


 = Unimplemented

Figure 3-5. ADC Clock Register (ADCLK)

ADIV2–ADIV0 — ADC Clock Prescaler Bits

ADIV2–ADIV0 form a 3-bit field which selects the divide ratio used by the ADC to generate the internal ADC clock. Table 3-2 shows the available clock configurations. The ADC clock should be set to approximately 1 MHz.

Table 3-2. ADC Clock Divide Ratio

ADIV2	ADIV1	ADIV0	ADC Clock Rate
0	0	0	ADC input clock ÷ 1
0	0	1	ADC input clock ÷ 2
0	1	0	ADC input clock ÷ 4
0	1	1	ADC input clock ÷ 8
1	X ⁽¹⁾	X ⁽¹⁾	ADC input clock ÷ 16

1. X = Don't care

ADICLK — ADC Input Clock Select Bit

ADICLK selects either the bus clock or the oscillator output clock (CGMXCLK) as the input clock source to generate the internal ADC clock. Reset selects CGMXCLK as the ADC clock source.

1 = Internal bus clock

0 = Oscillator output clock (CGMXCLK)

sensitive, interrupts should be disabled to prevent PLL interrupt service routines from impeding software performance or from exceeding stack limitations.

NOTE

Software can select the CGMVCLK divided by two as the CGMOUT source even if the PLL is not locked (LOCK = 0). Therefore, software should make sure the PLL is locked before setting the BCS bit.

4.7 Special Modes

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

4.7.1 Wait Mode

The WAIT instruction does not affect the CGM. Before entering wait mode, software can disengage and turn off the PLL by clearing the BCS and PLLON bits in the PLL control register (PCTL) to save power. Less power-sensitive applications can disengage the PLL without turning it off, so that the PLL clock is immediately available at WAIT exit. This would be the case also when the PLL is to wake the MCU from wait mode, such as when the PLL is first enabled and waiting for LOCK or LOCK is lost.

4.7.2 Stop Mode

If the OSCSTOPENB bit in the CONFIG register is cleared (default), then the STOP instruction disables the CGM (oscillator and phase locked loop) and holds low all CGM outputs (CGMXCLK, CGMOUT, and CGMINT).

If the STOP instruction is executed with the VCO clock, CGMVCLK, divided by two driving CGMOUT, the PLL automatically clears the BCS bit in the PLL control register (PCTL), thereby selecting the crystal clock, CGMXCLK, divided by two as the source of CGMOUT. When the MCU recovers from STOP, the crystal clock divided by two drives CGMOUT and BCS remains clear.

If the OSCSTOPENB bit in the CONFIG register is set, then the phase locked loop is shut off but the oscillator will continue to operate in stop mode.

4.7.3 CGM During Break Interrupts

The system integration module (SIM) controls whether status bits in other modules can be cleared during the break state. The BCFE bit in the SIM break flag control register (SBFCR) enables software to clear status bits during the break state. (See 14.7.1 SIM Break Status Register.)

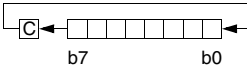
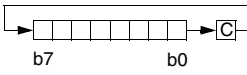
To allow software to clear status bits during a break interrupt, write a logic 1 to the BCFE bit. If a status bit is cleared during the break state, it remains cleared when the MCU exits the break state.

To protect the PLLF bit during the break state, write a logic 0 to the BCFE bit. With BCFE at logic 0 (its default state), software can read and write the PLL control register during the break state without affecting the PLLF bit.

4.8 Acquisition/Lock Time Specifications

The acquisition and lock times of the PLL are, in many applications, the most critical PLL design parameters. Proper design and use of the PLL ensures the highest stability and lowest acquisition/lock times.

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				
PULA	Pull A from Stack	$SP \leftarrow (SP + 1); \text{Pull (A)}$	–	–	–	–	–	INH	86		2
PULH	Pull H from Stack	$SP \leftarrow (SP + 1); \text{Pull (H)}$	–	–	–	–	–	INH	8A		2
PULX	Pull X from Stack	$SP \leftarrow (SP + 1); \text{Pull (X)}$	–	–	–	–	–	INH	88		2
ROL <i>opr</i> ROLA ROLX ROL <i>opr</i> ,X ROL ,X ROL <i>opr</i> ,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 <i>opr</i> RORA RORX ROR <i>opr</i> ,X ROR ,X ROR <i>opr</i> ,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	$SP \leftarrow \$FF$	–	–	–	–	–	INH	9C		1
RTI	Return from Interrupt	$SP \leftarrow (SP + 1); \text{Pull (CCR)}$ $SP \leftarrow (SP + 1); \text{Pull (A)}$ $SP \leftarrow (SP + 1); \text{Pull (X)}$ $SP \leftarrow (SP + 1); \text{Pull (PCH)}$ $SP \leftarrow (SP + 1); \text{Pull (PCL)}$	↑	↑	↑	↑	↑	INH	80		7
RTS	Return from Subroutine	$SP \leftarrow SP + 1; \text{Pull (PCH)}$ $SP \leftarrow SP + 1; \text{Pull (PCL)}$	–	–	–	–	–	INH	81		4
SBC # <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> SBC <i>opr</i> ,X SBC <i>opr</i> ,X SBC ,X SBC <i>opr</i> ,SP SBC <i>opr</i> ,SP	Subtract with Carry	$A \leftarrow (A) - (M) - (C)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A2 B2 C2 D2 E2 F2 9EE2 9ED2	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5
SEC	Set Carry Bit	$C \leftarrow 1$	–	–	–	–	1	INH	99		1
SEI	Set Interrupt Mask	$I \leftarrow 1$	–	–	1	–	–	INH	9B		2
STA <i>opr</i> STA <i>opr</i> STA <i>opr</i> ,X STA <i>opr</i> ,X STA ,X STA <i>opr</i> ,SP STA <i>opr</i> ,SP	Store A in M	$M \leftarrow (A)$	0	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	B7 C7 D7 E7 F7 9EE7 9ED7	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
STHX <i>opr</i>	Store H:X in M	$(M:M + 1) \leftarrow (H:X)$	0	–	–	–	↑	DIR	35	dd	4
STOP	Enable Interrupts, Stop Processing, Refer to MCU Documentation	$I \leftarrow 0$; Stop Processing	–	–	0	–	–	INH	8E		1
STX <i>opr</i> STX <i>opr</i> STX <i>opr</i> ,X STX <i>opr</i> ,X STX ,X STX <i>opr</i> ,SP STX <i>opr</i> ,SP	Store X in M	$M \leftarrow (X)$	0	–	–	–	↑	DIR EXT IX2 IX1 IX SP1 SP2	BF CF DF EF FF 9EEF 9EDF	dd hh ll ee ff ff ff ff ee ff	3 4 4 3 2 4 5
SUB # <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> SUB <i>opr</i> ,X SUB <i>opr</i> ,X SUB ,X SUB <i>opr</i> ,SP SUB <i>opr</i> ,SP	Subtract	$A \leftarrow (A) - (M)$	↑	–	–	–	↑	IMM DIR EXT IX2 IX1 IX SP1 SP2	A0 B0 C0 D0 E0 F0 9EE0 9ED0	ii dd hh ll ee ff ff ff ff ee ff	2 3 4 4 3 2 4 5

Chapter 8

External Interrupt (IRQ)

8.1 Introduction

The IRQ (external interrupt) module provides a maskable interrupt input.

8.2 Features

Features of the IRQ module include:

- A dedicated external interrupt pin ($\overline{\text{IRQ}}$)
- IRQ interrupt control bits
- Hysteresis buffer
- Programmable edge-only or edge and level interrupt sensitivity
- Automatic interrupt acknowledge
- Internal pullup resistor

8.3 Functional Description

A falling edge applied to the external interrupt pin can latch a central processor unit (CPU) interrupt request. Figure 8-1 shows the structure of the IRQ module.

Interrupt signals on the $\overline{\text{IRQ}}$ pin are latched into the IRQ latch. An interrupt latch remains set until one of the following actions occurs:

- Vector fetch — A vector fetch automatically generates an interrupt acknowledge signal that clears the latch that caused the vector fetch.
- Software clear — Software can clear an interrupt latch by writing to the appropriate acknowledge bit in the interrupt status and control register (INTSCR). Writing a 1 to the ACK bit clears the IRQ latch.
- Reset — A reset automatically clears the interrupt latch.

The external interrupt pin is falling-edge triggered out of reset and is software-configurable to be either falling-edge or falling-edge and low-level triggered. The MODE bit in the INTSCR controls the triggering sensitivity of the $\overline{\text{IRQ}}$ pin.

When an interrupt pin is edge-triggered only (MODE = 0), the interrupt remains set until a vector fetch, software clear, or reset occurs.

PTA3–PTA0 — Port A Data Bits

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

KBD3–KBD0 — Keyboard Inputs

The keyboard interrupt enable bits, KBIE3–KBIE0, in the keyboard interrupt control register (KBICR) enable the port A pins as external interrupt pins. See Chapter 9 Keyboard Interrupt Module (KBI).

12.2.2 Data Direction Register A

Data direction register A (DDRA) determines whether each port A pin is an input or an output. Writing a 1 to a DDRA bit enables the output buffer for the corresponding port A pin; a 0 disables the output buffer.

Address:	\$0004							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	DDRA3	DDRA2	DDRA1	DDRA0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 12-3. Data Direction Register A (DDRA)

DDRA3–DDRA0 — Data Direction Register A Bits

These read/write bits control port A data direction. Reset clears DDRA3–DDRA0, configuring all port A pins as inputs.

1 = Corresponding port A pin configured as output

0 = Corresponding port A pin configured as input

NOTE

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

Figure 12-4 shows the port A I/O logic.

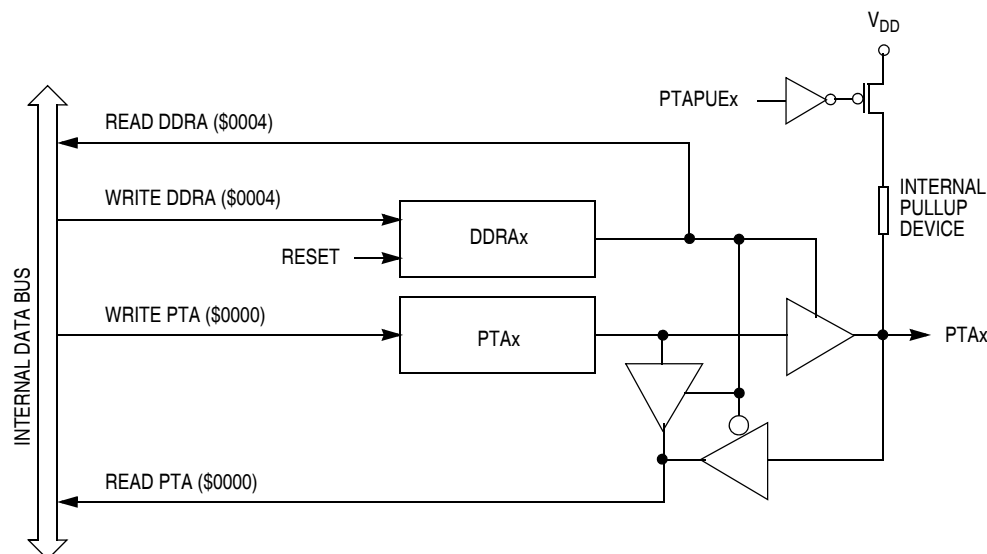


Figure 12-4. Port A I/O Circuit

When bit DDRAx is a 1, reading address \$0000 reads the PTAx data latch. When bit DDRAx is a 0, reading address \$0000 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-2 summarizes the operation of the port A pins.

Table 12-2. Port A Pin Functions

PTAPUE Bit	DDRA Bit	PTA Bit	I/O Pin Mode	Accesses to DDRA	Accesses to PTA	
				Read/Write	Read	Write
1	0	X ⁽¹⁾	Input, V _{DD} ⁽²⁾	DDRA3–DDRA0	Pin	PTA3–PTA0 ⁽³⁾
0	0	X	Input, Hi-Z ⁽⁴⁾	DDRA3–DDRA0	Pin	PTA3–PTA0 ⁽³⁾
X	1	X	Output	DDRA3–DDRA0	PTA3–PTA0	PTA3–PTA0

1. X = Don't care
2. I/O pin pulled up to V_{DD} by internal pullup device
3. Writing affects data register, but does not affect input.
4. Hi-Z = High impedance

12.2.3 Port A Input Pullup Enable Register

The port A input pullup enable register (PTAPUE) contains a software configurable pullup device for each of the four port A pins. Each bit is individually configurable and requires that the data direction register, DDRA, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRA is configured for output mode.

Address: \$000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	PTAPUE3	PTAPUE2	PTAPUE1	PTAPUE0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 12-5. Port A Input Pullup Enable Register (PTAPUE)

PTAPUE3–PTAPUE0 — Port A Input Pullup Enable Bits

These writeable bits are software programmable to enable pullup devices on an input port bit.

- 1 = Corresponding port A pin configured to have internal pullup
- 0 = Corresponding port A pin has internal pullup disconnected

12.5.3 Port D Input Pullup Enable Register

The port D input pullup enable register (PTDPUE) contains a software configurable pullup device for each of the seven port D pins. Each bit is individually configurable and requires that the data direction register, DDRD, bit be configured as an input. Each pullup is automatically and dynamically disabled when a port bit's DDRD is configured for output mode.

Address:	\$000F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	PTDPUE6	PTDPUE5	PTDPUE4	PTDPUE3	PTDPUE2	PTDPUE1	PTDPUE0
Write:								
Reset:	0	0	0	0	0	0	0	0

Figure 12-16. Port D Input Pullup Enable Register (PTDPUE)

PTDPUE6–PTDPUE0 — Port D Input Pullup Enable Bits

These writable bits are software programmable to enable pullup devices on an input port bit.

1 = Corresponding port D pin configured to have internal pullup

0 = Corresponding port D pin has internal pullup disconnected

12.6 Port E

Port E is a 2-bit special-function port that shares two of its pins with the serial communications interface (SCI) module.

12.6.1 Port E Data Register

The port E data register contains a data latch for each of the two port E pins.

Address:	\$0008							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	PTE1	PTE0
Write:								
Reset:	Unaffected by reset							
Alternative Function:							RxD	TxD

= Unimplemented

Figure 12-17. Port E Data Register (PTE)

PTE1–PTE0 — Port E Data Bits

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

NOTE

Data direction register E (DDRE) does not affect the data direction of port E pins that are being used by the ESCI module. However, the DDRE bits always determine whether reading port E returns the states of the latches or the states of the pins. See Table 12-6.

RxD — SCI Receive Data Input

The PTE1/RxD pin is the receive data input for the ESCI module. When the enable SCI bit, ENSCI, is clear, the ESCI module is disabled, and the PTE1/RxD pin is available for general-purpose I/O. See Chapter 13 Serial Communications Interface (SCI) Module.

TxD — SCI Transmit Data Output

The PTE0/TxD pin is the transmit data output for the ESCI module. When the enable SCI bit, ENSCI, is clear, the ESCI module is disabled, and the PTE0/TxD pin is available for general-purpose I/O. See Chapter 13 Serial Communications Interface (SCI) Module.

12.6.2 Data Direction Register E

Data direction register E (DDRE) determines whether each port E pin is an input or an output. Writing a 1 to a DDRE bit enables the output buffer for the corresponding port E pin; a 0 disables the output buffer.

Address: \$000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	0	0	DDRE1	DDRE0
Write:								
Reset:	0	0	0	0	0	0	0	0


 = Unimplemented

Figure 12-18. Data Direction Register E (DDRE)

DDRE1–DDRE0 — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE1–DDRE0, configuring all port E pins as inputs.

- 1 = Corresponding port E pin configured as output
- 0 = Corresponding port E pin configured as input

NOTE

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

Figure 12-19 shows the port E I/O logic.

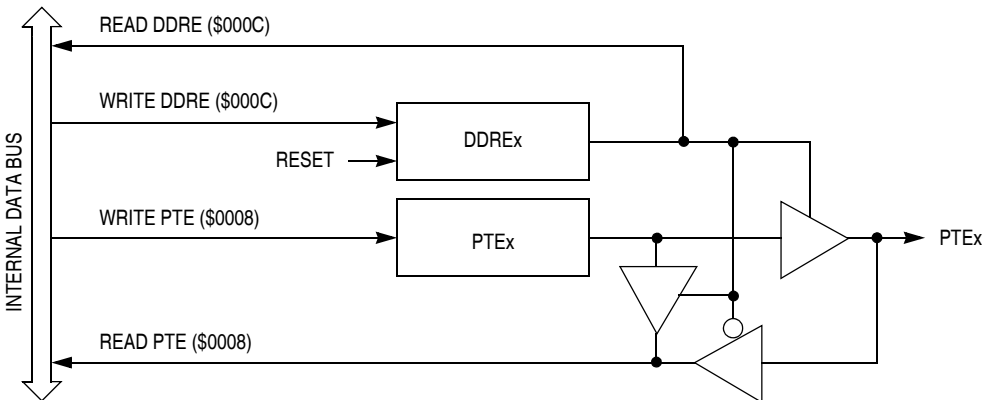


Figure 12-19. Port E I/O Circuit

Fast Data Tolerance

Figure 13-9 shows how much a fast received character can be misaligned without causing a noise error or a framing error. The fast stop bit ends at RT10 instead of RT16 but is still there for the stop bit data samples at RT8, RT9, and RT10.

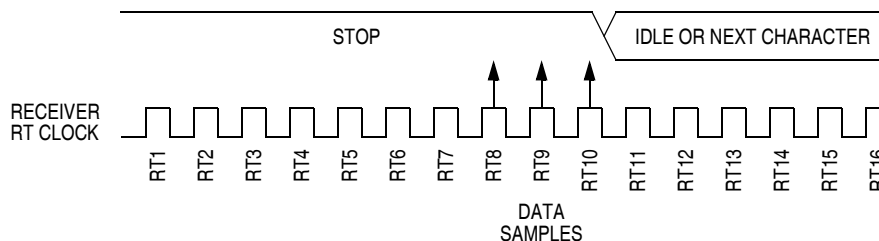


Figure 13-9. Fast Data

For an 8-bit character, data sampling of the stop bit takes the receiver 9 bit times \times 16 RT cycles + 10 RT cycles = 154 RT cycles.

With the misaligned character shown in Figure 13-9, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 10 bit times \times 16 RT cycles = 160 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is

$$\left| \frac{154 - 160}{154} \right| \times 100 = 3.90\%$$

For a 9-bit character, data sampling of the stop bit takes the receiver 10 bit times \times 16 RT cycles + 10 RT cycles = 170 RT cycles.

With the misaligned character shown in Figure 13-9, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 11 bit times \times 16 RT cycles = 176 RT cycles.

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is

$$\left| \frac{170 - 176}{170} \right| \times 100 = 3.53\%$$

13.4.3.6 Receiver Wakeup

So that the MCU can ignore transmissions intended only for other receivers in multiple-receiver systems, the receiver can be put into a standby state. Setting the receiver wakeup bit, RWU, in SCC2 puts the receiver into a standby state during which receiver interrupts are disabled.

Depending on the state of the WAKE bit in SCC1, either of two conditions on the PTE1/RxD pin can bring the receiver out of the standby state:

- **Address mark** — An address mark is a 1 in the most significant bit position of a received character. When the WAKE bit is set, an address mark wakes the receiver from the standby state by clearing the RWU bit. The address mark also sets the SCI receiver full bit, SCRF. Software can then compare the character containing the address mark to the user-defined address of the receiver. If they are the same, the receiver remains awake and processes the characters that follow. If they are not the same, software can set the RWU bit and put the receiver back into the standby state.

If the timebase functions are not required during stop mode, reduce power consumption by disabling the timebase module before executing the STOP instruction.

16.6 Timebase Control Register

The timebase has one register, the timebase control register (TBCR), which is used to enable the timebase interrupts and set the rate.

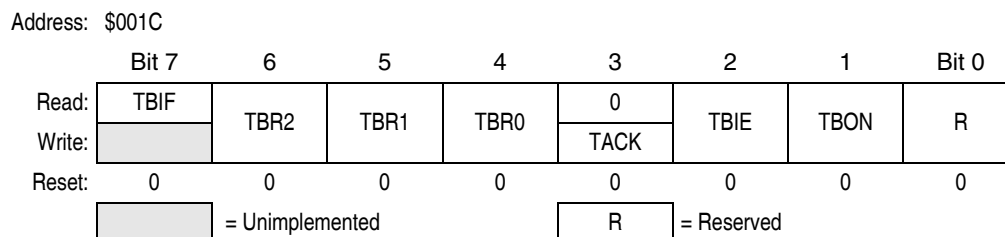


Figure 16-2. Timebase Control Register (TBCR)

TBIF — Timebase Interrupt Flag

This read-only flag bit is set when the timebase counter has rolled over.

1 = Timebase interrupt pending

0 = Timebase interrupt not pending

TBR2–TBR0 — Timebase Divider Selection Bits

These read/write bits select the tap in the counter to be used for timebase interrupts as shown in Table 16-1.

Table 16-1. Timebase Divider Selection

TBR2	TBR1	TBR0	Divider
0	0	0	32,768
0	0	1	8192
0	1	0	2048
0	1	1	128
1	0	0	64
1	0	1	32
1	1	0	16
1	1	1	8

NOTE

Do not change TBR2–TBR0 bits while the timebase is enabled (TBON = 1).

TACK— Timebase Acknowledge Bit

The TACK bit is a write-only bit and always reads as 0. Writing a 1 to this bit clears TBIF, the timebase interrupt flag bit. Writing a 0 to this bit has no effect.

1 = Clear timebase interrupt flag

0 = No effect

Chapter 17

Timer Interface Module (TIM1 and TIM2)

17.1 Introduction

This section describes the timer interface (TIM) module. The timer on this part consists of a 2-channel and a 1-channel timer that provides a timing reference with input capture, output compare, and pulse-width-modulation functions. Figure 17-1 is a block diagram of the TIM.

This particular MCU has two timer interface modules which are denoted as TIM1 and TIM2.

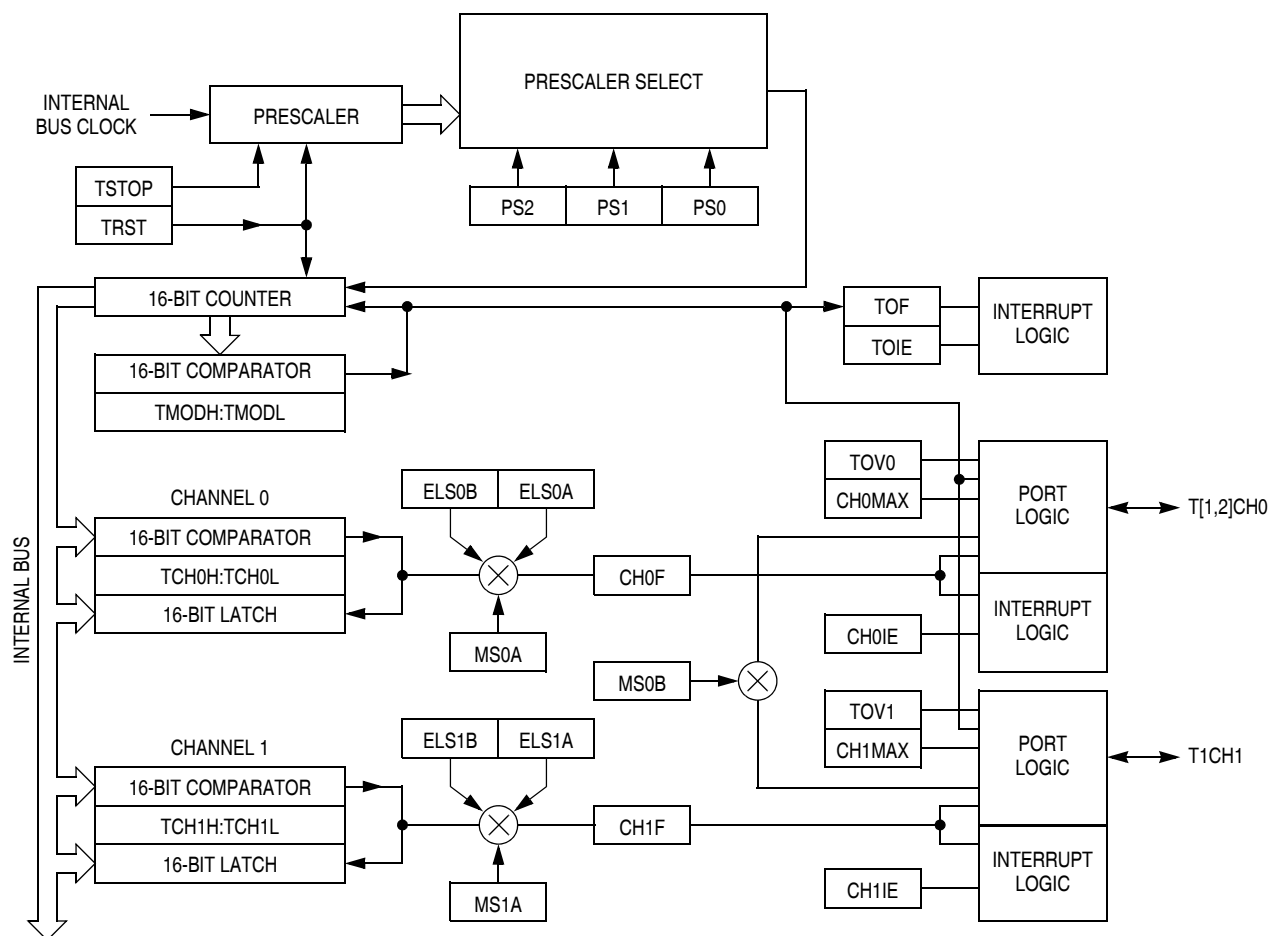


Figure 17-1. TIM Block Diagram

17.4.4 Pulse Width Modulation (PWM)

By using the toggle-on-overflow feature with an output compare channel, the TIM can generate a PWM signal. The value in the TIM counter modulo registers determines the period of the PWM signal. The channel pin toggles when the counter reaches the value in the TIM counter modulo registers. The time between overflows is the period of the PWM signal.

As Figure 17-4 shows, the output compare value in the TIM channel registers determines the pulse width of the PWM signal. The time between overflow and output compare is the pulse width. Program the TIM to clear the channel pin on output compare if the polarity of the PWM pulse is 1 (ELSxA = 0). Program the TIM to set the pin if the polarity of the PWM pulse is 0 (ELSxA = 0).

The value in the TIM counter modulo registers and the selected prescaler output determines the frequency of the PWM output. The frequency of an 8-bit PWM signal is variable in 256 increments. Writing \$00FF (255) to the TIM counter modulo registers produces a PWM period of 256 times the internal bus clock period if the prescaler select value is \$000. See 17.9.1 TIM Status and Control Register.

The value in the TIM channel registers determines the pulse width of the PWM output. The pulse width of an 8-bit PWM signal is variable in 256 increments. Writing \$0080 (128) to the TIM channel registers produces a duty cycle of 128/256 or 50%.

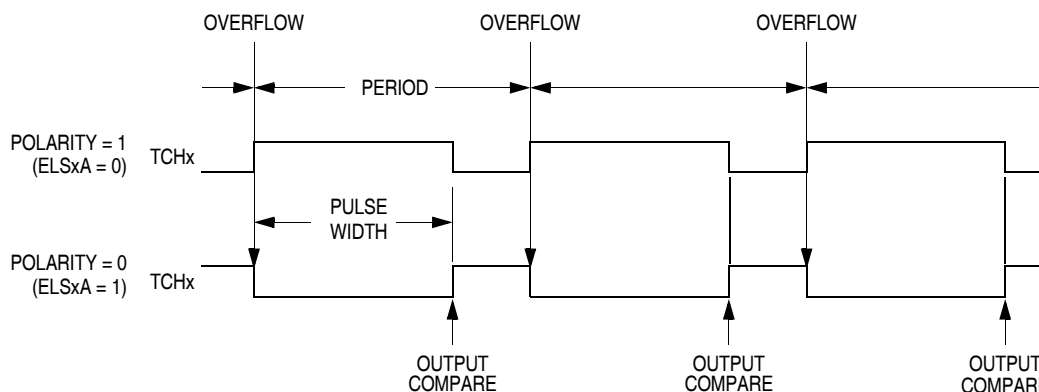


Figure 17-4. PWM Period and Pulse Width

17.4.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 17.4.4 Pulse Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM channel registers.

An unsynchronized write to the TIM channel registers to change a pulse width value could cause incorrect operation for up to two PWM periods. For example, writing a new value before the counter reaches the old value but after the counter reaches the new value prevents any compare during that PWM period. Also, using a TIM overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM may pass the new value before it is written.

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.

- c. Write 1:0 (polarity 1 — to clear output on compare) or 1:1 (polarity 0 — to set output on compare) to the edge/level select bits, ELSxB:ELSxA. The output action on compare must force the output to the complement of the pulse width level. See Table 17-3.

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 can also cause incorrect PWM signal generation when changing the PWM pulse width to a new, much larger value.

5. In the TIM status control register (TSC), clear the TIM stop bit, TSTOP.

Setting MS0B links channels 0 and 1 and configures them for buffered PWM operation. The TIM channel 0 registers (TCH0H:TCH0L) initially control the buffered PWM output. TIM status control register 0 (TSCR0) controls and monitors the PWM signal from the linked channels.

Clearing the toggle-on-overflow bit, TOVx, inhibits output toggles on TIM overflows. Subsequent output compares try to force the output to a state it is already in and have no effect. The result is a 0% duty cycle output.

Setting the channel x maximum duty cycle bit (CHxMAX) and setting the TOVx bit generates a 100% duty cycle output. See 17.9.4 TIM Channel Status and Control Registers.

17.5 Interrupts

The following TIM sources can generate interrupt requests:

- TIM overflow flag (TOF) — The TOF bit is set when the TIM counter reaches the modulo value programmed in the TIM counter modulo registers. The TIM overflow interrupt enable bit, TOIE, enables TIM overflow CPU interrupt requests. TOF and TOIE are in the TIM status and control register.
- TIM channel flags (CH1F:CH0F) — The CHxF bit is set when an input capture or output compare occurs on channel x. Channel x TIM CPU interrupt requests are controlled by the channel x interrupt enable bit, CHxIE. Channel x TIM CPU interrupt requests are enabled when CHxIE = 1. CHxF and CHxIE are in the TIM channel x status and control register.

17.6 Low-Power Modes

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

17.6.1 Wait Mode

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

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

Chapter 18

Development Support

18.1 Introduction

This section describes the break module, the monitor module (MON), and the monitor mode entry methods.

18.2 Break Module (BRK)

The break module can generate a break interrupt that stops normal program flow at a defined address to enter a background program.

Features of the break module include:

- Accessible input/output (I/O) registers during the break Interrupt
- Central processor unit (CPU) generated break interrupts
- Software-generated break interrupts
- Computer operating properly (COP) disabling during break interrupts

18.2.1 Functional Description

When the internal address bus matches the value written in the break address registers, the break module issues a breakpoint signal (\overline{BKPT}) to the system integration module (SIM). The SIM then causes the CPU to load the instruction register with a software interrupt instruction (SWI). The program counter vectors to \$FFFC and \$FFFD (\$FEFC and \$FEFD in monitor mode).

The following events can cause a break interrupt to occur:

- A CPU generated address (the address in the program counter) matches the contents of the break address registers.
- Software writes a 1 to the BRKA bit in the break status and control register.

When a CPU generated address matches the contents of the break address registers, the break interrupt is generated. A return-from-interrupt instruction (RTI) in the break routine ends the break interrupt and returns the microcontroller unit (MCU) to normal operation.

Figure 18-2 shows the structure of the break module.

Figure 18-3 provides a summary of the I/O registers.

18.2.2.1 Break Status and Control Register

The break status and control register (BRKSCR) contains break module enable and status bits.

Address: \$FE0B

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


 = Unimplemented

Figure 18-4. Break Status and Control Register (BRKSCR)

BRKE — Break Enable Bit

This read/write bit enables breaks on break address register matches. Clear BRKE by writing a 0 to bit 7. Reset clears the BRKE bit.

- 1 = Breaks enabled on 16-bit address match
- 0 = Breaks disabled

BRKA — Break Active Bit

This read/write status and control bit is set when a break address match occurs. Writing a 1 to BRKA generates a break interrupt. Clear BRKA by writing a 0 to it before exiting the break routine. Reset clears the BRKA bit.

- 1 = Break address match
- 0 = No break address match

18.2.2.2 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: \$FE09

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

Figure 18-5. Break Address Register High (BRKH)

Address: \$FE0A

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

Figure 18-6. Break Address Register Low (BRKL)

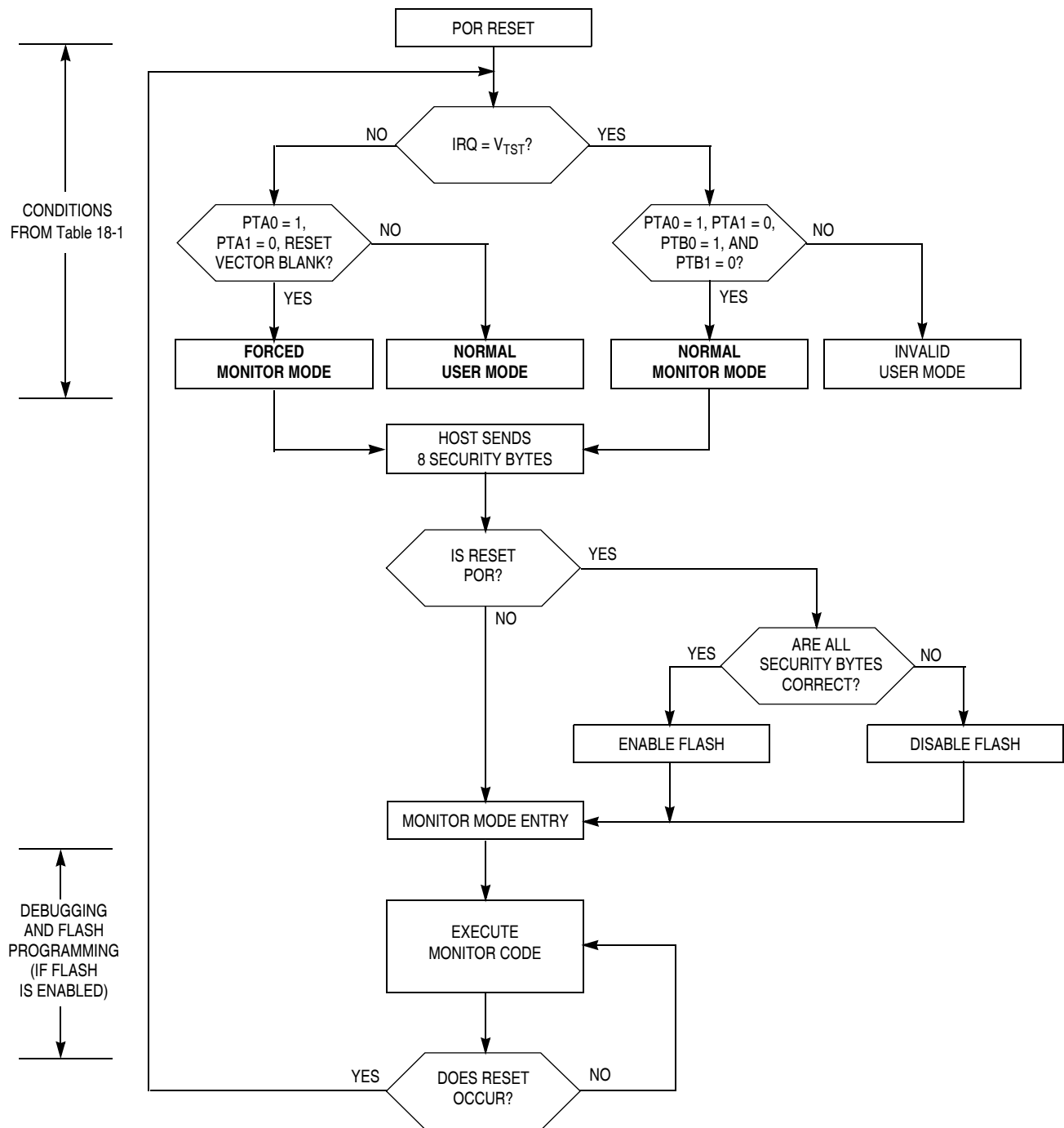


Figure 18-9. Simplified Monitor Mode Entry Flowchart

Table 18-1. Monitor Mode Signal Requirements and Options

Mode	$\overline{\text{IRQ}}$	$\overline{\text{RST}}$	Reset Vector	Serial Communication		Mode Selection		PLL	COP	Communication Speed		
				PTA0	PTA1	PTB0	PTB1			External Clock	Bus Frequency	Baud Rate
Normal Monitor	V_{TST}	V_{DD} or V_{TST}	X	1	0	1	0	OFF	Disabled	9.8304 MHz	2.4576 MHz	9600
Forced Monitor	V_{DD}	V_{DD}	\$FFFF (blank)	1	0	X	X	OFF	Disabled	9.8304 MHz	2.4576 MHz	9600
User	V_{DD} or V_{SS}	V_{DD} or V_{TST}	Not \$FFFF	X	X	X	X	X	Enabled	X	X	X
MON08 Function [Pin No.]	V_{TST} [6]	$\overline{\text{RST}}$ [4]	—	COM [8]	SSEL [10]	MOD0 [12]	MOD1 [14]	—	—	OSC1 [13]	—	—

1. PTA0 must have a pullup resistor to V_{DD} in monitor mode.
2. Communication speed in the table is an example to obtain a baud rate of 9600.
Baud rate using external oscillator is bus frequency / 256.
3. External clock is an 9.8304 MHz crystal on OSC1 and OSC2 or a 9.8304 MHz canned oscillator on OSC1.
4. X = don't care
5. MON08 pin refers to P&E Microcomputer Systems' MON08-Cyclone 2 by 8-pin connector.

NC	1	2	GND
NC	3	4	$\overline{\text{RST}}$
NC	5	6	$\overline{\text{IRQ}}$
NC	7	8	PTA0
NC	9	10	PTA1
NC	11	12	PTB0
OSC1	13	14	PTB1
V_{DD}	15	16	NC

Chapter 20

Ordering Information and Mechanical Specifications

20.1 Introduction

This section provides ordering information for the MC68HC908GR8A and MC68HC908GR4A along with the dimensions for:

- 32-pin low-profile quad flat pack package (case 873A)
- 28-pin dual in-line package (case 710)
- 28-pin small outline package (case 751F)

The following figures show the latest package drawings at the time of this publication. To make sure that you have the latest package specifications, contact your local Freescale Semiconductor Sales Office.

20.2 MC Order Numbers

Table 20-1. MC Order Numbers

MC Order Number	Operating Temperature Range	Package
MC68HC908GR8ACFA	−40°C to +85°C	32-pin low-profile quad flat package (LQFP)
MC68HC908GR4ACFA	−40°C to +85°C	
MC68HC908GR8AVFA	−40°C to +105°C	
MC68HC908GR8AMFA	−40°C to +125°C	
MC68HC908GR8ACP	−40°C to +85°C	28-pin dual in-line package (PDIP)
MC68HC908GR4ACP	−40°C to +85°C	
MC68HC908GR8ACDW	−40°C to +85°C	28-pin small outline package (SOIC)
MC68HC908GR4ACDW	−40°C to +85°C	

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

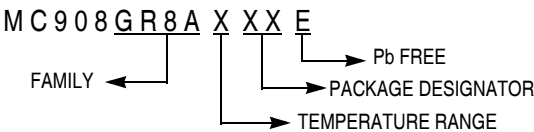


Figure 20-1. Device Numbering System

20.3 Package Dimensions

Refer to the following pages for detailed package dimensions.



NOTES:

1. POSITIONAL TOLERANCE OF LEADS, SHALL BE WITHIN 0.25 MM (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION DOES NOT INCLUDE MOLD FLASH.
4. 710-02 OBSOLETE, NEW STD 710-03.
5. CONTROLLING DIMENSION: INCH

DIM	INCH		MILLIMETER		DIM	INCH		MILLIMETER	
	MIN	MAX	MIN	MAX		MIN	MAX	MIN	MAX
A	1.435	1.465	36.45	37.21					
B	0.540	0.560	13.72	14.22					
C	0.155	0.200	3.94	5.08					
D	0.014	0.022	0.36	0.56					
F	0.040	0.060	1.02	1.52					
G	0.100	BSC	2.54	BSC					
H	0.065	0.085	1.65	2.16					
J	0.008	0.015	0.20	0.38					
K	0.115	0.135	2.92	3.43					
L	0.600	BSC	15.24	BSC					
M	0°	15°	0°	15°					
N	0.020	0.040	0.51	1.02					
© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.			MECHANICAL OUTLINE						
TITLE: 28 LD PDIP					DOCUMENT NO: 98ASB42390B			REV: C	
					CASE NUMBER: 710-03			05 APR 2005	
					STANDARD: NON-JEDEC				