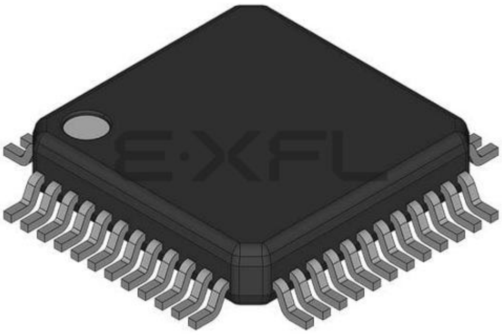

Welcome to E-XFL.COM

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	37
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	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908gr48acfae

Table of Contents

Chapter 1
General Description

Chapter 2
Memory

Chapter 1
General Description

1.1 Introduction

The MC68HC908GR60A, MC68HC908GR48A, and MC68HC908GR32A are members 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.

The information contained in this document pertains to all three devices with the exceptions noted in Appendix A MC68HC908GR48A and Appendix B MC68HC908GR32A.

1.2 Features

For convenience, features have been organized to reflect:

- Standard features
- Features of the CPU08

1.2.1 Standard Features

Features of the MC68HC908GR60A 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
- 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)
- System protection features:
  – Optional computer operating properly (COP) reset
  – Low-voltage detection with optional reset and selectable trip points for 3.3-V and 5.0-V operation
  – Illegal opcode detection with reset
  – Illegal address detection with reset
- Low-power design; fully static with stop and wait modes
- Standard low-power modes of operation:
  – Wait mode
  – Stop mode

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

Decreasing the value in FL1BPR by one increases the protected range by one page (128 bytes). However, programming the block protect register with $FE protects a range twice that size, 256 bytes, in the corresponding array. $FE means that locations $FF00–$FFFF are protected in FLASH-1.

The FLASH memory does not exist at some locations. The block protection range configuration is unaffected if FLASH memory does not exist in that range. Refer to Figure 2-1 and make sure that the desired locations are protected.

2.6.3 FLASH-1 Block Protection

Due to the ability of the on-board charge pump to erase and program the FLASH memory in the target application, provision is made for protecting blocks of memory from unintentional erase or program operations due to system malfunction. This protection is done by using the FLASH-1 block protection register (FL1BPR). FL1BPR determines the range of the FLASH-1 memory which is to be protected. The range of the protected area starts from a location defined by FL1BPR and ends at the bottom of the FLASH-1 memory ($FFFF). When the memory is protected, the HVEN bit can not be set in either ERASE or PROGRAM operations.

NOTE

In performing a program or erase operation, the FLASH-1 block protect register must be read after setting the PGM or ERASE bit and before asserting the HVEN bit.

When the FLASH-1 block protect register is programmed with all 0’s, the entire memory is protected from being programmed and erased. When all the bits are erased (all 1’s), the entire memory is accessible for program and erase.

When bits within FL1BPR are programmed (0), they lock a block of memory address ranges as shown in Figure 2-4. If FL1BPR is programmed with any value other than $FF, the protected block of FLASH memory can not be erased or programmed.

NOTE

The vector locations and the FLASH block protect registers are located in the same page. FL1BPR and FL2BPR are not protected with special hardware or software. Therefore, if this page is not protected by FL1BPR and the vector locations are erased by either a page or a mass erase operation, then both FL1BPR and FL2BPR will also get erased.

E. *The time between each FLASH address change (step 7 to step 7), or the time between the last FLASH address programmed to clearing the PGM bit (step 7 to step 10) must not exceed the maximum programming time, t_{PROG} maximum.*

F. *Be cautious when programming the FLASH-2 array to ensure that non-FLASH locations are not used as the address that is written to when selecting either the desired row address range in step 3 of the algorithm or the byte to be programmed in step 7 of the algorithm.*

2.7.7 Low-Power Modes

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

2.7.7.1 Wait Mode

Putting the MCU into wait mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The WAIT instruction should not be executed while performing a program or erase operation on the FLASH. Wait mode will suspend any FLASH program/erase operations and leave the memory in a standby mode.

2.7.7.2 Stop Mode

Putting the MCU into stop mode while the FLASH is in read mode does not affect the operation of the FLASH memory directly; however, no memory activity will take place since the CPU is inactive.

The STOP instruction should not be executed while performing a program or erase operation on the FLASH. Stop mode will suspend any FLASH program/erase operations and leave the memory in a standby mode.

NOTE

Standby mode is the power saving mode of the FLASH module, in which all internal control signals to the FLASH are inactive and the current consumption of the FLASH is minimum.

Chapter 3
Analog-to-Digital Converter (ADC)

3.1 Introduction

This section describes the 10-bit analog-to-digital converter (ADC).

3.2 Features

Features of the ADC module include:

- 24 channels with multiplexed input
- Linear successive approximation with monotonicity
- 10-bit resolution
- Single or continuous conversion
- Conversion complete flag or conversion complete interrupt
- Selectable ADC clock
- Left or right justified result
- Left justified sign data mode

3.3 Functional Description

The ADC provides 24 pins for sampling external sources at pins PTG7/AD23–PTG0/AD16, PTA7/KBD7/AD15–PTA0/KBD0/AD8, and PTB7/AD7–PTB0/AD0. An analog multiplexer allows the single ADC converter to select one of 24 ADC channels as ADC voltage in (V_{ADIN}). V_{ADIN} is converted by the successive approximation register-based analog-to-digital converter. When the conversion is completed, ADC places the result in the ADC data register and sets a flag or generates an interrupt. See Figure 3-2.

3.3.1 ADC Port I/O Pins

PTG7/AD23–PTG0/AD16, PTA7/KBD7/AD15–PTA0/KBD0/AD8, and PTB7/AD7–PTB0/AD0 are general-purpose I/O (input/output) pins that share with the ADC channels. The channel select bits define which ADC channel/port pin will be used as the input signal. The ADC overrides the port I/O logic by forcing that pin as input to the ADC. The remaining ADC channels/port pins are controlled by the port I/O logic and can be used as general-purpose I/O. Writes to the port register or data direction register (DDR) will not have any affect on the port pin that is selected by the ADC. A read of a port pin in use by the ADC will return a 0.

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							ADRH
	Bit 7	6	5	4	3	2	1	Bit 0
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)

3.8.3 ADC Clock Register

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

Address:	$003F							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	ADIV2	ADIV1	ADIV0	ADICLK	MODE1	MODE0	R	0
Write:								▨
Reset:	0	0	0	0	0	1	0	0

▨ = Unimplemented R = Reserved

Figure 3-9. 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

The following conditions apply when in manual mode:

- $\overline{\text{ACQ}}$ is a writable control bit that controls the mode of the filter. Before turning on the PLL in manual mode, the $\overline{\text{ACQ}}$ bit must be clear.
- Before entering tracking mode ($\overline{\text{ACQ}}$ = 1), software must wait a given time, t_{ACQ} (See 4.8 Acquisition/Lock Time Specifications.), after turning on the PLL by setting PLLON in the PLL control register (PCTL).
- Software must wait a given time, t_{AL}, after entering tracking mode before selecting the PLL as the clock source to CGMOUT (BCS = 1).
- The LOCK bit is disabled.
- CPU interrupts from the CGM are disabled.

4.3.6 Programming the PLL

Use the following procedure to program the PLL. For reference, the variables used and their meaning are shown in Table 4-1.

Table 4-1. Variable Definitions

Variable	Definition
f_{BUSDES}	Desired bus clock frequency
$f_{VCLKDES}$	Desired VCO clock frequency
f_{RCLK}	Chosen reference crystal frequency
f_{VCLK}	Calculated VCO clock frequency
f_{BUS}	Calculated bus clock frequency
f_{NOM}	Nominal VCO center frequency
f_{VRS}	Programmed VCO center frequency

NOTE

The round function in the following equations means that the real number should be rounded to the nearest integer number.

1. Choose the desired bus frequency, f_{BUSDES}.
2. Calculate the desired VCO frequency (four times the desired bus frequency).

$$f_{VCLKDES} = 4 \text{ x } f_{BUSDES}$$

3. Choose a practical PLL (crystal) reference frequency, f_{RCLK}. Typically, the reference crystal is 1–8 MHz.

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

$$f_{VCLK} = (N)\ (f_{RCLK})$$

   N, the range multiplier, must be an integer.

An interrupt signal on an edge-triggered pin can be acknowledged immediately after enabling the pin. An interrupt signal on an edge- and level-triggered interrupt pin must be acknowledged after a delay that depends on the external load.

Another way to avoid a false interrupt:

1. Configure the keyboard pins as outputs by setting the appropriate DDRA bits in data direction register A.
2. Write 1s (or 0s) to the appropriate port A data register bits.
3. Enable the KBI pins and polarity by setting the appropriate KBIEx bits in the keyboard interrupt enable register and the KBIPx bits in the keyboard interrupt polarity register.

9.5 Low-Power Modes

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

9.5.1 Wait Mode

The keyboard module remains active in wait mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of wait mode.

9.5.2 Stop Mode

The keyboard module remains active in stop mode. Clearing the IMASKK bit in the keyboard status and control register enables keyboard interrupt requests to bring the MCU out of stop mode.

9.6 Keyboard Module During Break Interrupts

The system integration module (SIM) controls whether the keyboard interrupt latch can be cleared during the break state. The BCFE bit in the break flag control register (BFCR) enables software to clear status bits during the break state.

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

To protect the latch during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), writing to the keyboard acknowledge bit (ACKK) in the keyboard status and control register during the break state has no effect. See 9.7.1 Keyboard Status and Control Register.

9.7 I/O Registers

These registers control and monitor operation of the keyboard module:

- Keyboard status and control register (INTKBSCR)
- Keyboard interrupt enable register (INTKBIER)
- Keyboard interrupt polarity register (INTKBIPR)

12.5 Port C

Port C is a 7-bit, general-purpose bidirectional I/O port. Port C also has software configurable pullup devices if configured as an input port.

12.5.1 Port C Data Register

The port C data register (PTC) contains a data latch for each of the seven port C pins.

NOTE

Bit 6 through bit 2 of PTC are not available in the 32-pin LQFP package.

Address: $0002

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	1	PTC6	PTC5	PTC4	PTC3	PTC2	PTC1	PTC0
Write:	▨							
Reset:	Unaffected by reset							

▨ = Unimplemented

Figure 12-9. Port C Data Register (PTC)

PTC6–PTC0 — Port C Data Bits

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

12.5.2 Data Direction Register C

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

Address: $0006

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
Write:	▨							
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

Figure 12-10. Data Direction Register C (DDRC)

DDRC6–DDRC0 — Data Direction Register C Bits

These read/write bits control port C data direction. Reset clears DDRC6–DDRC0, configuring all port C pins as inputs.

1 = Corresponding port C pin configured as output
0 = Corresponding port C pin configured as input

NOTE

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

Figure 12-11 shows the port C I/O logic.

When bit DDRCx is a 1, reading address \$0002 reads the PTCx data latch. When bit DDRCx is a 0, reading address \$0002 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-4 summarizes the operation of the port C pins.

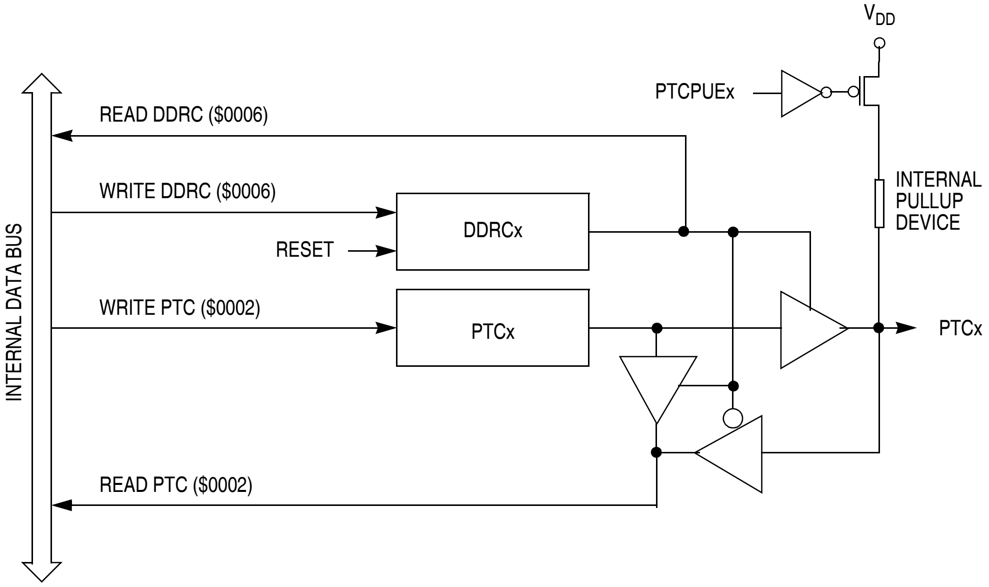

Figure 12-11. Port C I/O Circuit

Table 12-4. Port C Pin Functions

PTCPUE Bit	DDRC Bit	PTC Bit	I/O Pin Mode	Accesses to DDRC	Accesses to PTC	
				Read/Write	**Read**	**Write**
1	0	X[(1)]	Input, V_{DD}[(2)]	DDRC6–DDRC0	Pin	PTC6–PTC0[(3)]
0	0	X	Input, Hi-Z[(4)]	DDRC6–DDRC0	Pin	PTC6–PTC0[(3)]
X	1	X	Output	DDRC6–DDRC0	PTC6–PTC0	PTC6–PTC0

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

When bit DDRDx is a 1, reading address $0003 reads the PTDx data latch. When bit DDRDx is a 0, reading address $0003 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-5 summarizes the operation of the port D pins.

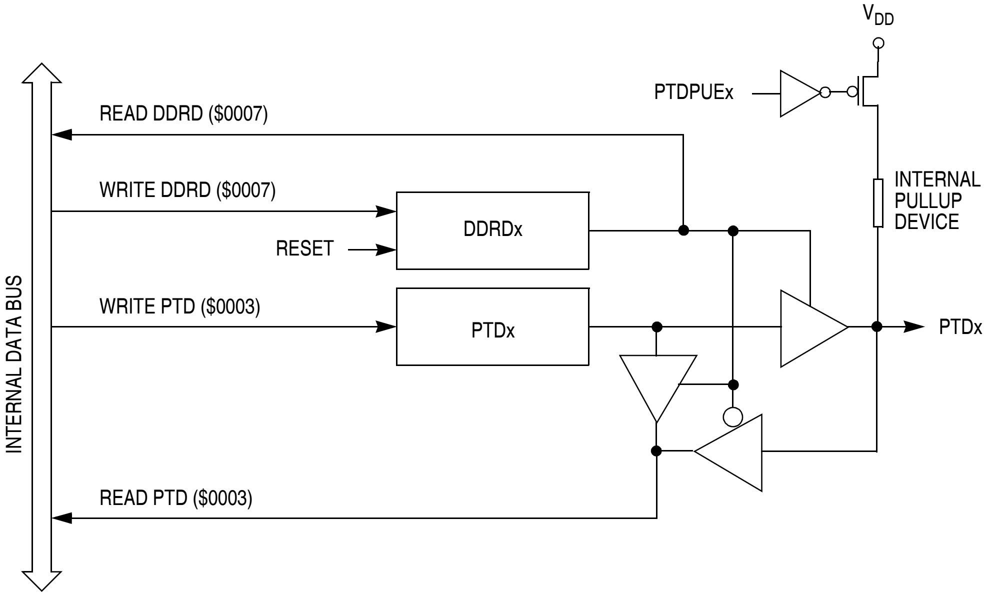

Figure 12-15. Port D I/O Circuit

Table 12-5. Port D Pin Functions

PTDPUE Bit	DDRD Bit	PTD Bit	I/O Pin Mode	Accesses to DDRD	Accesses to PTD	
				Read/Write	Read	Write
1	0	X[1]	Input, V_{DD}[2]	DDRD7–DDRD0	Pin	PTD7–PTD0[3]
0	0	X	Input, Hi-Z[4]	DDRD7–DDRD0	Pin	PTD7–PTD0[3]
X	1	X	Output	DDRD7–DDRD0	PTD7–PTD0	PTD7–PTD0

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 imp[edance

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 Enhanced Serial Communications Interface (ESCI) 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 Enhanced Serial Communications Interface (ESCI) Module.

12.7.2 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	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Write:	▨	▨						
Reset:	0	0	0	0	0	0	0	0

▨ = Unimplemented

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

DDRE5–DDRE0 — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE5–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.

When bit DDREx is a 1, reading address $0008 reads the PTEx data latch. When bit DDREx is a 0, reading address $0008 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 12-6 summarizes the operation of the port E pins.

Table 13-11. ESCI Baud Rate Selection Examples

PDS[2:1:0]	PSSB[4:3:2:1:0]	SCP[1:0]	Prescaler Divisor (BPD)	SCR[2:1:0]	Baud Rate Divisor (BD)	Baud Rate (f_{Bus}= 4.9152 MHz)
0 0 0	X X X X X	0 0	1	0 0 0	1	76,800
1 1 1	0 0 0 0 0	0 0	1	0 0 0	1	9600
1 1 1	0 0 0 0 1	0 0	1	0 0 0	1	9562.65
1 1 1	0 0 0 1 0	0 0	1	0 0 0	1	9525.58
1 1 1	1 1 1 1 1	0 0	1	0 0 0	1	8563.07
0 0 0	X X X X X	0 0	1	0 0 1	2	38,400
0 0 0	X X X X X	0 0	1	0 1 0	4	19,200
0 0 0	X X X X X	0 0	1	0 1 1	8	9600
0 0 0	X X X X X	0 0	1	1 0 0	16	4800
0 0 0	X X X X X	0 0	1	1 0 1	32	2400
0 0 0	X X X X X	0 0	1	1 1 0	64	1200
0 0 0	X X X X X	0 0	1	1 1 1	128	600
0 0 0	X X X X X	0 1	3	0 0 0	1	25,600
0 0 0	X X X X X	0 1	3	0 0 1	2	12,800
0 0 0	X X X X X	0 1	3	0 1 0	4	6400
0 0 0	X X X X X	0 1	3	0 1 1	8	3200
0 0 0	X X X X X	0 1	3	1 0 0	16	1600
0 0 0	X X X X X	0 1	3	1 0 1	32	800
0 0 0	X X X X X	0 1	3	1 1 0	64	400
0 0 0	X X X X X	0 1	3	1 1 1	128	200
0 0 0	X X X X X	1 0	4	0 0 0	1	19,200
0 0 0	X X X X X	1 0	4	0 0 1	2	9600
0 0 0	X X X X X	1 0	4	0 1 0	4	4800
0 0 0	X X X X X	1 0	4	0 1 1	8	2400
0 0 0	X X X X X	1 0	4	1 0 0	16	1200
0 0 0	X X X X X	1 0	4	1 0 1	32	600
0 0 0	X X X X X	1 0	4	1 1 0	64	300
0 0 0	X X X X X	1 0	4	1 1 1	128	150
0 0 0	X X X X X	1 1	13	0 0 0	1	5908
0 0 0	X X X X X	1 1	13	0 0 1	2	2954
0 0 0	X X X X X	1 1	13	0 1 0	4	1477
0 0 0	X X X X X	1 1	13	0 1 1	8	739
0 0 0	X X X X X	1 1	13	1 0 0	16	369
0 0 0	X X X X X	1 1	13	1 0 1	32	185
0 0 0	X X X X X	1 1	13	1 1 0	64	92
0 0 0	X X X X X	1 1	13	1 1 1	128	46

15.6.2 Mode Fault Error

Setting SPMSTR selects master mode and configures the SPSCK and MOSI pins as outputs and the MISO pin as an input. Clearing SPMSTR selects slave mode and configures the SPSCK and MOSI pins as inputs and the MISO pin as an output. The mode fault bit, MODF, becomes set any time the state of the slave select pin, $\overline{\text{SS}}$, is inconsistent with the mode selected by SPMSTR.

To prevent SPI pin contention and damage to the MCU, a mode fault error occurs if:

- The $\overline{\text{SS}}$ pin of a slave SPI goes high during a transmission
- The $\overline{\text{SS}}$ pin of a master SPI goes low at any time

For the MODF flag to be set, the mode fault error enable bit (MODFEN) must be set. Clearing the MODFEN bit does not clear the MODF flag but does prevent MODF from being set again after MODF is cleared.

MODF generates a receiver/error CPU interrupt request if the error interrupt enable bit (ERRIE) is also set. The SPRF, MODF, and OVRF interrupts share the same CPU interrupt vector. (See Figure 15-12.) It is not possible to enable MODF or OVRF individually to generate a receiver/error CPU interrupt request. However, leaving MODFEN low prevents MODF from being set.

In a master SPI with the mode fault enable bit (MODFEN) set, the mode fault flag (MODF) is set if $\overline{\text{SS}}$ goes low. A mode fault in a master SPI causes the following events to occur:

- If ERRIE = 1, the SPI generates an SPI receiver/error CPU interrupt request.
- The SPE bit is cleared.
- The SPTE bit is set.
- The SPI state counter is cleared.
- The data direction register of the shared I/O port regains control of port drivers.

NOTE

To prevent bus contention with another master SPI after a mode fault error, clear all SPI bits of the data direction register of the shared I/O port before enabling the SPI.

When configured as a slave (SPMSTR = 0), the MODF flag is set if $\overline{\text{SS}}$ goes high during a transmission. When CPHA = 0, a transmission begins when $\overline{\text{SS}}$ goes low and ends once the incoming SPSCK goes back to its idle level following the shift of the eighth data bit. When CPHA = 1, the transmission begins when the SPSCK leaves its idle level and $\overline{\text{SS}}$ is already low. The transmission continues until the SPSCK returns to its idle level following the shift of the last data bit. See 15.4 Transmission Formats.

NOTE

Setting the MODF flag does not clear the SPMSTR bit. SPMSTR has no function when SPE = 0. Reading SPMSTR when MODF = 1 shows the difference between a MODF occurring when the SPI is a master and when it is a slave.

NOTE

When CPHA = 0, a MODF occurs if a slave is selected ($\overline{\text{SS}}$ is low) and later unselected ($\overline{\text{SS}}$ is high) even if no SPSCK is sent to that slave. This happens because $\overline{\text{SS}}$ low indicates the start of the transmission (MISO driven out with the value of MSB) for CPHA = 0. When CPHA = 1, a slave can be selected and then later unselected with no transmission occurring. Therefore, MODF does not occur since a transmission was never begun.

When enabled, the SPI controls data direction of the MOSI pin regardless of the state of the data direction register of the shared I/O port.

15.11.3 SPSCK (Serial Clock)

The serial clock synchronizes data transmission between master and slave devices. In a master MCU, the SPSCK pin is the clock output. In a slave MCU, the SPSCK pin is the clock input. In full-duplex operation, the master and slave MCUs exchange a byte of data in eight serial clock cycles.

When enabled, the SPI controls data direction of the SPSCK pin regardless of the state of the data direction register of the shared I/O port.

15.11.4 SS (Slave Select)

The SS pin has various functions depending on the current state of the SPI. For an SPI configured as a slave, the SS is used to select a slave. For CPHA = 0, the SS is used to define the start of a transmission. (See 15.4 Transmission Formats.) Since it is used to indicate the start of a transmission, SS must be toggled high and low between each byte transmitted for the CPHA = 0 format. However, it can remain low between transmissions for the CPHA = 1 format. See Figure 15-13.

When an SPI is configured as a slave, the SS pin is always configured as an input. It cannot be used as a general-purpose I/O regardless of the state of the MODFEN control bit. However, the MODFEN bit can still prevent the state of SS from creating a MODF error. See 15.12.2 SPI Status and Control Register.

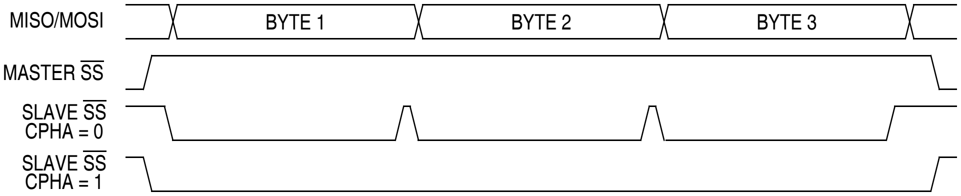

Figure 15-13. CPHA/SS Timing

NOTE

A high on the SS pin of a slave SPI puts the MISO pin in a high-impedance state. The slave SPI ignores all incoming SPSCK clocks, even if it was already in the middle of a transmission.

When an SPI is configured as a master, the SS input can be used in conjunction with the MODF flag to prevent multiple masters from driving MOSI and SPSCK. (See 15.6.2 Mode Fault Error.) For the state of the SS pin to set the MODF flag, the MODFEN bit in the SPSCK register must be set. If MODFEN is 0 for an SPI master, the SS pin can be used as a general-purpose I/O under the control of the data direction register of the shared I/O port. When MODFEN is 1, SS is an input-only pin to the SPI regardless of the state of the data direction register of the shared I/O port.

The CPU can always read the state of the SS pin by configuring the appropriate pin as an input and reading the port data register. See Table 15-2.

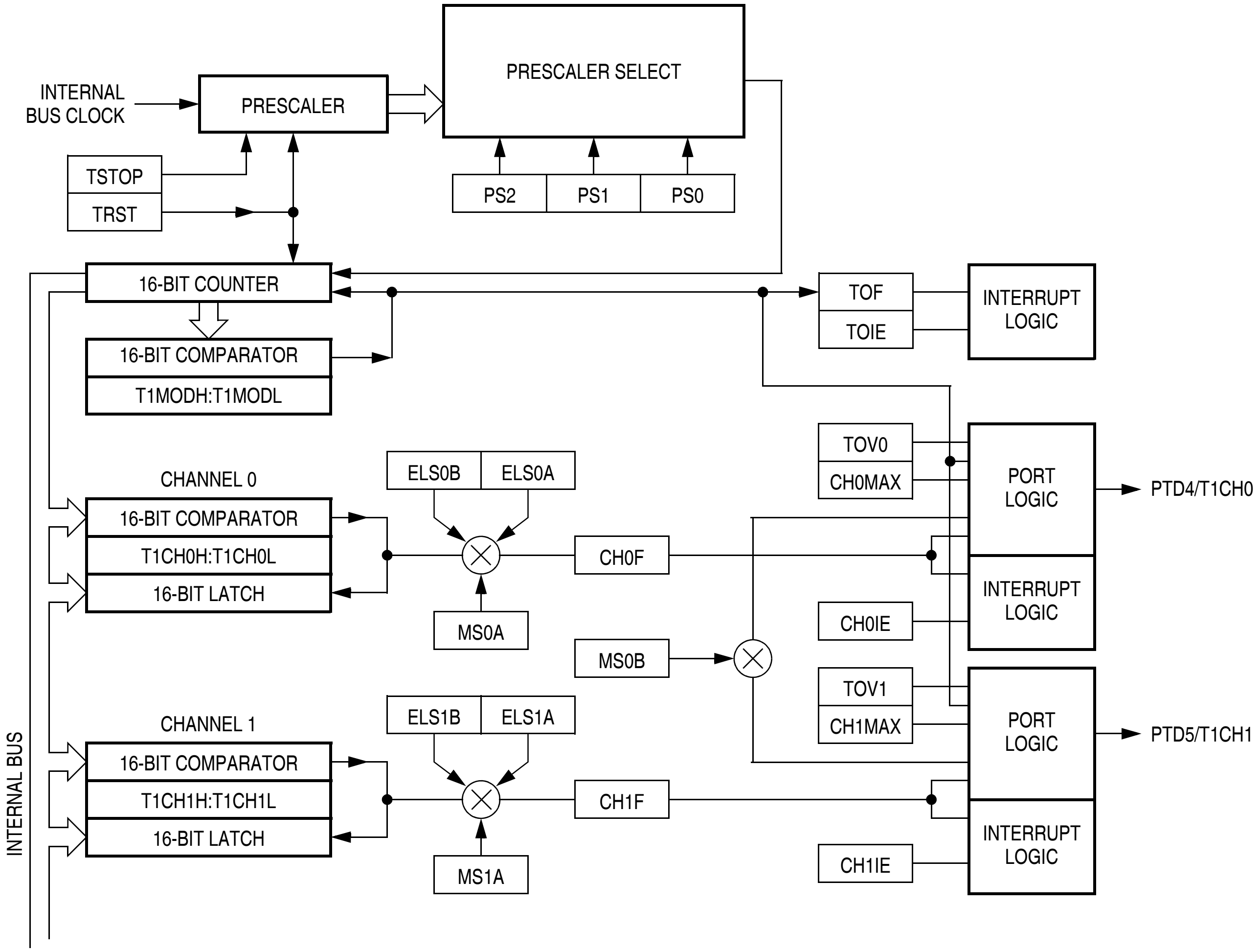

Figure 17-2. TIM1 Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	TIM1 Status and Control Register (T1SC) See page 234.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	▨			
		Reset:	0	0	1	0	0	0	0	0
$0021	TIM1 Counter Register High (T1CNTH) See page 235.	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
$0022	TIM1 Counter Register Low (T1CNTL) See page 235.	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

▨ = Unimplemented

Figure 17-3. TIM1 I/O Register Summary

18.7 I/O Signals

Port D shares two of its pins with the TIM2. Port F shares four of its pins with the TIM2. PTD6/T2CH0 is an external clock input to the TIM2 prescaler. The six TIM2 channel I/O pins are PTD6/T2CH0, PTD7/T2CH1, PTF4/T2CH2, PTF5/T2CH3, PTF6/T2CH4, and PTF7/T2CH5.

18.7.1 TIM2 Clock Pin (T2CH0)

T2CH0 is an external clock input that can be the clock source for the TIM2 counter instead of the prescaled internal bus clock. Select the T2CH0 input by writing 1s to the three prescaler select bits, PS[2:0]. (See 18.8.1 TIM2 Status and Control Register.) The minimum TCLK pulse width is specified in 20.14 Timer Interface Module Characteristics. The maximum TCLK frequency is the least: 4 MHz or bus frequency ÷ 2.

When the PTD6/T2CH0 pin is the TIM2 clock input, it is an input regardless of the state of the DDRD6 bit in data direction register D.

18.7.2 TIM2 Channel I/O Pins (T2CH5:T2CH2 and T2CH1:T2CH0)

Each channel I/O pin is programmable independently as an input capture pin or an output compare pin. T2CH0, T2CH2, and T2CH4 can be configured as buffered output compare or buffered PWM pins.

18.8 I/O Registers

These I/O registers control and monitor TIM2 operation:

- TIM2 status and control register (T2SC)
- TIM2 counter registers (T2CNTH:T2CNTL)
- TIM2 counter modulo registers (T2MODH:T2MODL)
- TIM2 channel status and control registers (T2SC0, T2SC1, T2SC2, T2SC3, T2SC4, and T2SC5)
- TIM2 channel registers (T2CH0H:T2CH0L, T2CH1H:T2CH1L, T2CH2H:T2CH2L, T2CH3H:T2CH3L, T2CH4H:T2CH4L, and T2CH5H:T2CH5L)

18.8.1 TIM2 Status and Control Register

The TIM2 status and control register:

- Enables TIM2 overflow interrupts
- Flags TIM2 overflows
- Stops the TIM2 counter
- Resets the TIM2 counter
- Prescales the TIM2 counter clock

Address: $002B

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
Write:	0			TRST	▨			
Reset:	0	0	1	0	0	0	0	0

▨ = Unimplemented

Figure 18-5. TIM2 Status and Control Register (T2SC)

CHxF — Channel x Flag Bit

When channel x is an input capture channel, this read/write bit is set when an active edge occurs on the channel x pin. When channel x is an output compare channel, CHxF is set when the value in the TIM2 counter registers matches the value in the TIM2 channel x registers.

When CHxIE = 1, clear CHxF by reading TIM2 channel x status and control register with CHxF set, and then writing a 0 to CHxF. If another interrupt request occurs before the clearing sequence is complete, then writing 0 to CHxF has no effect. Therefore, an interrupt request cannot be lost due to inadvertent clearing of CHxF.

Reset clears the CHxF bit. Writing a 1 to CHxF has no effect.

1 = Input capture or output compare on channel x
0 = No input capture or output compare on channel x

CHxIE — Channel x Interrupt Enable Bit

This read/write bit enables TIM2 CPU interrupts on channel x.
Reset clears the CHxIE bit.

1 = Channel x CPU interrupt requests enabled
0 = Channel x CPU interrupt requests disabled

MSxB — Mode Select Bit B

This read/write bit selects buffered output compare/PWM operation. MSxB exists only in the TIM2 channel 0, TIM2 channel 2, and TIM2 channel 4 status and control registers.

Setting MS0B disables the channel 1 status and control register and reverts T2CH1 pin to general-purpose I/O.

Setting MS2B disables the channel 3 status and control register and reverts T2CH3 pin to general-purpose I/O.

Setting MS4B disables the channel 5 status and control register and reverts T2CH5 pin to general-purpose I/O.

Reset clears the MSxB bit.

1 = Buffered output compare/PWM operation enabled
0 = Buffered output compare/PWM operation disabled

MSxA — Mode Select Bit A

When ELSxB:ELSxA ≠ 00, this read/write bit selects either input capture operation or unbuffered output compare/PWM operation. (See Table 18-2.)

1 = Unbuffered output compare/PWM operation
0 = Input capture operation

When ELSxB:ELSxA = 00, this read/write bit selects the initial output level of the T2CHx pin once PWM, input capture, or output compare operation is enabled. (See Table 18-2.) Reset clears the MSxA bit.

1 = Initial output level low
0 = Initial output level high

NOTE

Before changing a channel function by writing to the MSxB or MSxA bit, set the TSTOP and TRST bits in the TIM2 status and control register (T2SC).

19.3.1.4 Data Format

Communication with the monitor ROM is in standard non-return-to-zero (NRZ) mark/space data format. Transmit and receive baud rates must be identical.

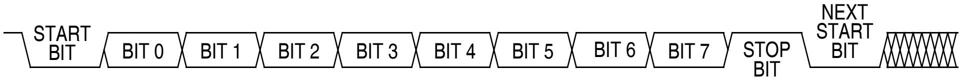

Figure 19-12. Monitor Data Format

19.3.1.5 Break Signal

A start bit (0) followed by nine 0 bits is a break signal. When the monitor receives a break signal, it drives the PTA0 pin high for the duration of approximately two bits and then echoes back the break signal.

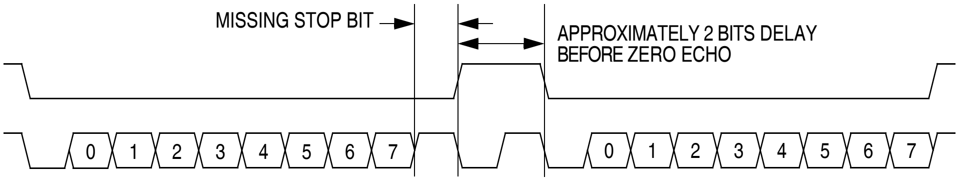

Figure 19-13. Break Transaction

19.3.1.6 Baud Rate

The communication baud rate is controlled by the crystal frequency or external clock and the state of the PTB4 pin (when $\overline{\text{IRQ}}$ is set to V_{TST}) upon entry into monitor mode. If monitor mode was entered with V_{DD} on $\overline{\text{IRQ}}$ and the reset vector blank, then the baud rate is independent of PTB4.

Table 19-1 also lists external frequencies required to achieve a standard baud rate of 7200 bps. The effective baud rate is the bus frequency divided by 278. If using a crystal as the clock source, be aware of the upper frequency limit that the internal clock module can handle. See 20.7 5.0-Volt Control Timing or 20.8 3.3-Volt Control Timing for this limit.

19.3.1.7 Commands

The monitor ROM firmware uses these commands:

- READ (read memory)
- WRITE (write memory)
- IREAD (indexed read)
- IWRITE (indexed write)
- READSP (read stack pointer)
- RUN (run user program)

The monitor ROM firmware echoes each received byte back to the PTA0 pin for error checking. An 11-bit delay at the end of each command allows the host to send a break character to cancel the command. A delay of two bit times occurs before each echo and before READ, IREAD, or READSP data is returned. The data returned by a read command appears after the echo of the last byte of the command.

NOTE

Wait one bit time after each echo before sending the next byte.

20.11 3.3-Volt ADC Characteristics

Characteristic[1]	Symbol	Min	Max	Unit	Comments
Supply voltage	V_{DDAD}	3.0	3.6	V	V_{DDAD} should be tied to the same potential as V_{DD} via separate traces.
Input voltages	V_{ADIN}	0	V_{DDAD}	V	$V_{ADIN} <= V_{DDAD}$
Resolution	B_{AD}	10	10	Bits	
Absolute accuracy	A_{AD}	–6	+6	LSB	Includes quantization
ADC internal clock	f_{ADIC}	500 k	1.048 M	Hz	$t_{AIC} = 1/f_{ADIC}$
Conversion range	R_{AD}	V_{SSAD}	V_{DDAD}	V	
Power-up time	t_{ADPU}	16	—	t_{AIC} cycles	
Conversion time	t_{ADC}	16	17	t_{AIC} cycles	
Sample time	t_{ADS}	5	—	t_{AIC} cycles	
Monotonicity	M_{AD}	Guaranteed			
Zero input reading	Z_{ADI}	000	005	Hex	$V_{ADIN} = V_{SSA}$
Full-scale reading	F_{ADI}	3FA	3FF	Hex	$V_{ADIN} = V_{DDA}$
Input capacitance	C_{ADI}	—	30	pF	Not tested
V_{DDAD}/V_{REFH} current	I_{VREF}	—	1.2	mA	
Absolute accuracy (8-bit truncation mode)	A_{AD}	–1	+1	LSB	Includes quantization
Quantization error (8-bit truncation mode)	—	–1/8	+7/8	LSB	

[1] V_{DD} = 3.3 Vdc ± 10%, V_{SS} = 0 Vdc, V_{DDAD}/V_{REFH} = 3.3 Vdc ± 10%, V_{SSAD}/V_{REFL} = 0 Vdc