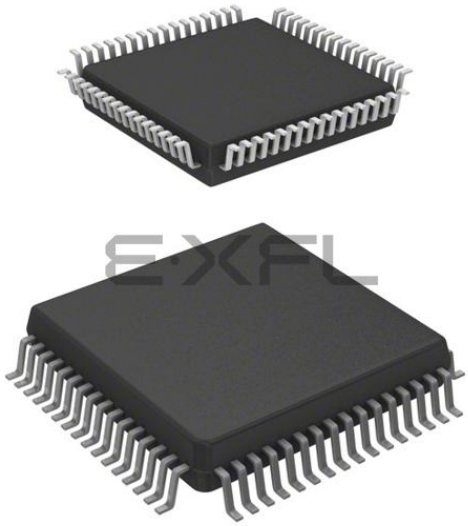


Welcome to E-XFL.COM

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	44
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 10x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908mr32cfu

MC68HC908MR32
MC68HC908MR16

Data Sheet

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify you have the latest information available, refer to:

http://freescale.com

Chapter 1
General Description

1.1 Introduction

The MC68HC908MR32 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.

The information contained in this document pertains to the MC68HC908MR16 with the exceptions shown in Appendix A MC68HC908MR16.

1.2 Features

Features include:

- High-performance M68HC08 architecture
- Fully upward-compatible object code with M6805, M146805, and M68HC05 Families
- 8-MHz internal bus frequency
- On-chip FLASH memory with in-circuit programming capabilities of FLASH program memory:
  MC68HC908MR32 — 32 Kbytes
  MC68HC908MR16 — 16 Kbytes
- On-chip programming firmware for use with host personal computer
- FLASH data security[1]
- 768 bytes of on-chip random-access memory (RAM)
- 12-bit, 6-channel center-aligned or edge-aligned pulse-width modulator (PWMMC)
- Serial peripheral interface module (SPI)
- Serial communications interface module (SCI)
- 16-bit, 4-channel timer interface module (TIMA)
- 16-bit, 2-channel timer interface module (TIMB)
- Clock generator module (CGM)
- Low-voltage inhibit (LVI) module with software selectable trip points
- 10-bit, 10-channel analog-to-digital converter (ADC)
- System protection features:
  – Optional computer operating properly (COP) reset
  – Low-voltage detection with optional reset
  – Illegal opcode or address detection with optional reset
  – Fault detection with optional PWM disabling

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
$000E	TIMA Status/Control Register (TASC) See page 226.	Read:	TOF	TOIE	TSTOP	0	0	PS2	PS1	PS0
		Write:	0			TRST	R			
		Reset:	0	0	1	0	0	0	0	0
$000F	TIMA Counter Register High (TACNTH) See page 227.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$0010	TIMA Counter Register Low (TACNTL) See page 227.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$0011	TIMA Counter Modulo Register High (TAMODH) See page 228.	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0012	TIMA Counter Modulo Register Low (TAMODL) See page 228.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	1	1	1	1	1	1	1	1
$0013	TIMA Channel 0 Status/Control Register (TASC0) See page 229.	Read:	CH0F	CH0IE	MS0B	MS0A	ELS0B	ELS0A	TOV0	CH0MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0
$0014	TIMA Channel 0 Register High (TACH0H) See page 232.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0015	TIMA Channel 0 Register Low (TACH0L) See page 232.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0016	TIMA Channel 1 Status/Control Register (TASC1) See page 232.	Read:	CH1F	CH1IE	0	MS1A	ELS1B	ELS1A	TOV1	CH1MAX
		Write:	0		R					
		Reset:	0	0	0	0	0	0	0	0
$0017	TIMA Channel 1 Register High (TACH1H) See page 232.	Read:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Write:								
		Reset:	Indeterminate after reset							
$0018	TIMA Channel 1 Register Low (TACH1L) See page 232.	Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Write:								
		Reset:	Indeterminate after reset							
$0019	TIMA Channel 2 Status/Control Register (TASC2) See page 229.	Read:	CH2F	CH2IE	MS2B	MS2A	ELS2B	ELS2A	TOV2	CH2MAX
		Write:	0							
		Reset:	0	0	0	0	0	0	0	0

U = Unaffected X = Indeterminate R = Reserved Bold = Buffered = Unimplemented

Figure 2-2. Control, Status, and Data Registers Summary (Sheet 2 of 8)

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0032	PWM 5 Value Register High (PMVAL5H) See page 145.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	0	0	0	0	0	0	0	0
$0033	PWM 5 Value Register Low (PVAL5L) See page 145.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$0034	PWM 6 Value Register High (PVAL6H) See page 145.	Read: Write:	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		Reset:	0	0	0	0	0	0	0	0
$0035	PWM 6 Value Register Low (PMVAL6L) See page 145.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	0	0	0	0	0	0	0	0
$0036	Dead-Time Write-Once Register (DEADTM) See page 150.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0037	PWM Disable Mapping Write-Once Register (DISMAP) See page 137.	Read: Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		Reset:	1	1	1	1	1	1	1	1
$0038	SCI Control Register 1 (SCC1) See page 169.	Read: Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Reset:	0	0	0	0	0	0	0	0
$0039	SCI Control Register 2 (SCC2) See page 171.	Read: Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0
$003A	SCI Control Register 3 (SCC3) See page 173.	Read:	R8	T8	0	0	ORIE	NEIE	FEIE	PEIE
		Write:	R		R	R				
		Reset:	U	U	0	0	0	0	0	0
$003B	SCI Status Register 1 (SCS1) See page 174.	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:	R	R	R	R	R	R	R	R
		Reset:	1	1	0	0	0	0	0	0
$003C	SCI Status Register 2 (SCS2) See page 176.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$003D	SCI Data Register (SCDR) See page 177.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							

U = Unaffected X = Indeterminate R = Reserved Bold = Buffered = Unimplemented

Figure 2-2. Control, Status, and Data Registers Summary (Sheet 5 of 8)

Address: $0042

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

R = Reserved

Figure 3-8. ADC Data Register Low (ADRL) Right Justified Mode

In 8-bit mode, this 8-bit result register holds the eight MSBs of the 10-bit result. This register is updated each time an ADC conversion completes. In 8-bit mode, this register contains no interlocking with ADRH.

Address: $0042

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2
Write:	R	R	R	R	R	R	R	R
Reset:	Unaffected by reset							

R = Reserved

Figure 3-9. ADC Data Register Low (ADRL) 8-Bit Mode

3.7.4 ADC Clock Register

This register selects the clock frequency for the ADC, selecting between modes of operation.

Address: $0043

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

R = Reserved

Figure 3-10. ADC Clock Register (ADCLK)

ADIV2:ADIV0 — ADC Clock Prescaler Bits

ADIV2, ADIV1, and 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.

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	X	ADC input clock ÷ 16

X = don’t care

ADICLK — ADC Input Clock Select Bit

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

If the external clock (CGMXCLK) is equal to or greater than 1 MHz, CGMXCLK can be used as the clock source for the ADC. If CGMXCLK is less than 1 MHz, use the PLL-generated bus clock as the clock source. As long as the internal ADC clock is at f_{ADIC}, correct operation can be guaranteed. See 19.13 Analog-to-Digital Converter (ADC) Characteristics.

1 = Internal bus clock

0 = External clock, CGMXCLK

$$f_{ADIC} = \frac{\text{CGMXCLK or bus frequency}}{\text{ADIV[2:0]}}$$

MODE1:MODE0 — Modes of Result Justification Bits

MODE1:MODE0 selects among four modes of operation. The manner in which the ADC conversion results will be placed in the ADC data registers is controlled by these modes of operation. Reset returns right-justified mode.

00 = 8-bit truncation mode

01 = Right justified mode

10 = Left justified mode

11 = Left justified sign data mode

6.3.6 COPD (COP Disable)

The COPD signal reflects the state of the COP disable bit (COPD) in the configuration register (CONFIG). See Chapter 5 Configuration Register (CONFIG).

6.4 COP Control Register

The COP control register is located at address $FFFF and overlaps the reset vector. Writing any value to $FFFF clears the COP counter and starts a new timeout period. Reading location $FFFF returns the low byte of the reset vector.

Address:	$FFFF							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Low byte of reset vector							
Write:	Clear COP counter							
Reset:	Unaffected by reset							

Figure 6-3. COP Control Register (COPCTL)

6.5 Interrupts

The COP does not generate CPU interrupt requests.

6.6 Monitor Mode

The COP is disabled in monitor mode when V_{HI} is present on the $\overline{IRQ}$ pin or on the $\overline{RST}$ pin.

6.7 Wait Mode

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

The COP continues to operate during wait mode.

6.8 Stop Mode

Stop mode turns off the CGMXCLK input to the COP and clears the COP prescaler. Service the COP immediately before entering or after exiting stop mode to ensure a full COP timeout period after entering or exiting stop mode.

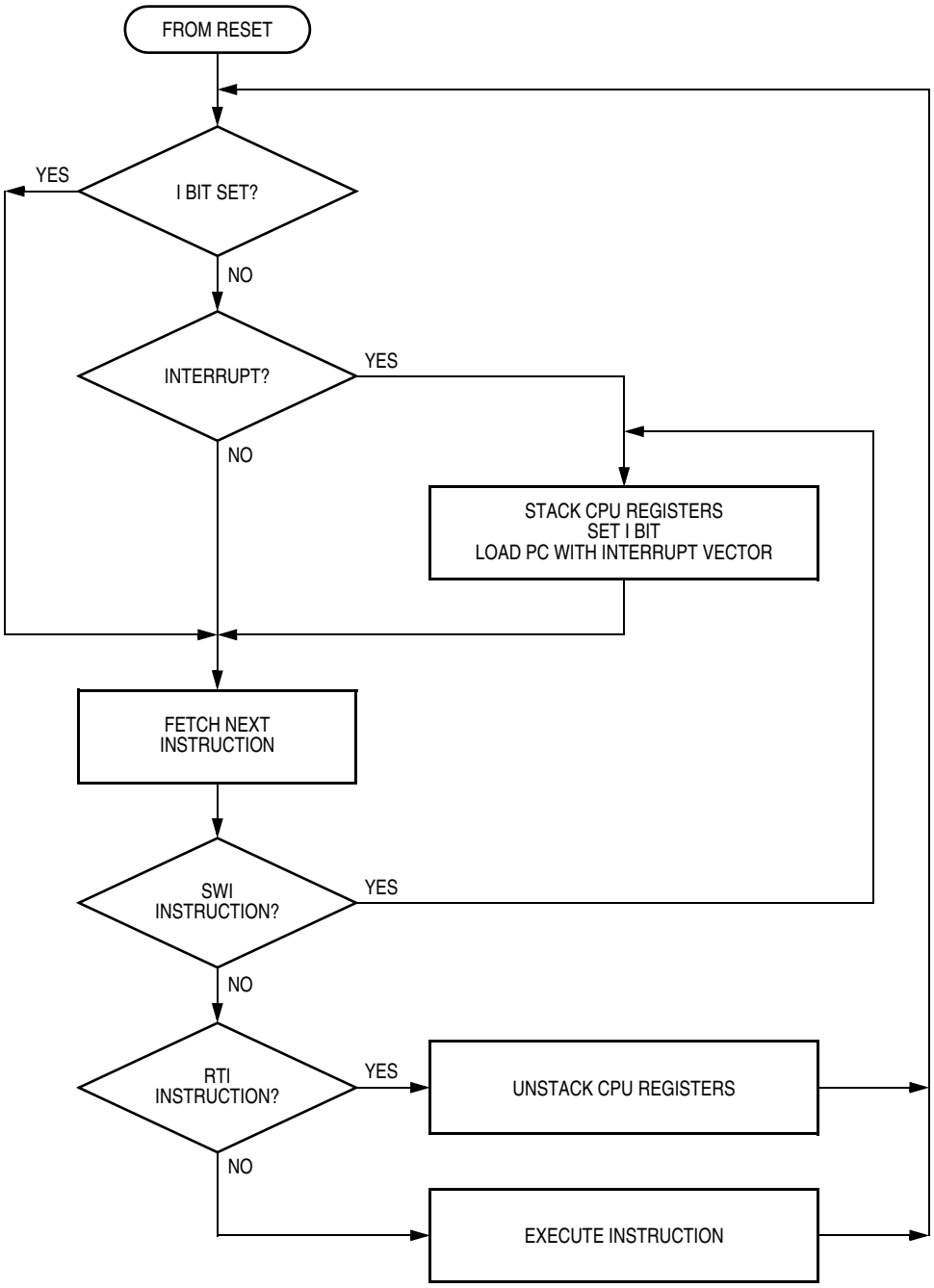


Figure 8-3. IRQ Interrupt Flowchart

10.6.2 Data Direction Register E

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

Address: $000C

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
Reset:	0	0	0	0	0	0	0	0

Figure 10-14. Data Direction Register E (DDRE)

DDRE[7:0] — Data Direction Register E Bits

These read/write bits control port E data direction. Reset clears DDRE[7:0], 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 10-15 shows the port E I/O logic.

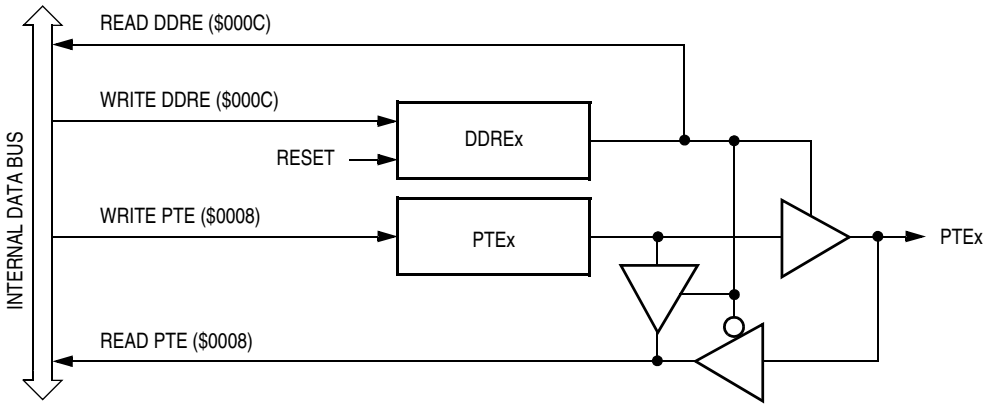


Figure 10-15. Port E I/O Circuit

When bit DDREx is a logic 1, reading address $0008 reads the PTEx data latch. When bit DDREx is a logic 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 10-5 summarizes the operation of the port E pins.

Table 10-5. Port E Pin Functions

DDRE Bit	PTE Bit	I/O Pin Mode	Accesses to DDRE	Accesses to PTE	
			Read/Write	Read	Write
0	X[1]	Input, Hi-Z[2]	DDRE[7:0]	Pin	PTE[7:0][3]
1	X	Output	DDRE[7:0]	PTE[7:0]	PTE[7:0]

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

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0038	SCI Control Register 1 (SCC1) See page 169.	Read: / Write:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Reset:	0	0	0	0	0	0	0	0
$0039	SCI Control Register 2 (SCC2) See page 171.	Read: / Write:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Reset:	0	0	0	0	0	0	0	0
$003A	SCI Control Register 3 (SCC3) See page 173.	Read:	R8	T8	0	0	ORIE	NEIE	FEIE	PEIE
		Write:	R		R	R				
		Reset:	U	U	0	0	0	0	0	0
$003B	SCI Status Register 1 (SCS1) See page 174.	Read:	SCTE	TC	SCRF	IDLE	OR	NF	FE	PE
		Write:	R	R	R	R	R	R	R	R
		Reset:	1	1	0	0	0	0	0	0
$003C	SCI Status Register 2 (SCS2) See page 176.	Read:	0	0	0	0	0	0	BKF	RPF
		Write:	R	R	R	R	R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$003D	SCI Data Register (SCDR) See page 177.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							
$003E	SCI Baud Rate Register (SCBR) See page 177.	Read:	0	0	SCP1	SCP0	0	SCR2	SCR1	SCR0
		Write:	R	R			R			
		Reset:	0	0	0	0	0	0	0	0

R = Reserved U = Unaffected

Figure 13-3. SCI I/O Register Summary

13.3.1 Data Format

The SCI uses the standard non-return-to-zero mark/space data format illustrated in Figure 13-4.

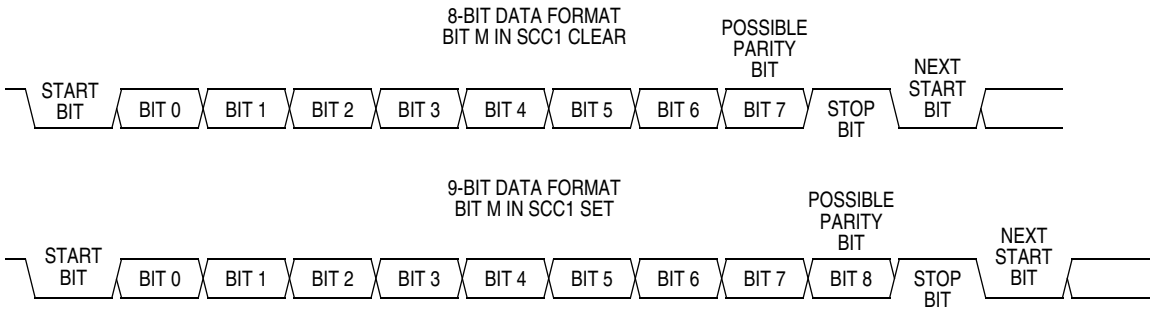


Figure 13-4. SCI Data Formats

- Noise flag (NF) — The NF bit is set when the SCI detects noise on incoming data or break characters, including start, data, and stop bits. The noise error interrupt enable bit, NEIE, in SCC3 enables NF to generate SCI error CPU interrupt requests.
- Framing error (FE) — The FE bit in SCS1 is set when a 0 occurs where the receiver expects a stop bit. The framing error interrupt enable bit, FEIE, in SCC3 enables FE to generate SCI error CPU interrupt requests.
- Parity error (PE) — The PE bit in SCS1 is set when the SCI detects a parity error in incoming data. The parity error interrupt enable bit, PEIE, in SCC3 enables PE to generate SCI error CPU interrupt requests.

13.4 Wait Mode

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

The SCI module remains active after the execution of a WAIT instruction. In wait mode the SCI module registers are not accessible by the CPU. 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.

13.5 SCI During Break Module Interrupts

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

To allow software to clear status bits during a break interrupt, write a 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 status bits during the break state, write a 0 to the BCFE bit. With BCFE at 0 (its default state), software can read and write I/O registers during the break state without affecting status bits. Some status bits have a 2-step read/write clearing procedure. If software does the first step on such a bit before the break, the bit cannot change during the break state as long as BCFE is at 0. After the break, doing the second step clears the status bit.

13.6 I/O Signals

Port F shares two of its pins with the SCI module. The two SCI input/output (I/O) pins are:

- PTF5/TxD — Transmit data
- PTF4/RxD — Receive data

13.6.1 PTF5/TxD (Transmit Data)

The PTF5/TxD pin is the serial data output from the SCI transmitter. The SCI shares the PTF5/TxD pin with port F. When the SCI is enabled, the PTF5/TxD pin is an output regardless of the state of the DDRF5 bit in data direction register F (DDRF).

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 $\overline{SS}$ goes to logic 0. A mode fault in a master SPI causes these 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 $\overline{SS}$ goes high during a transmission. When CPHA = 0, a transmission begins when $\overline{SS}$ 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 $\overline{SS}$ is already low. The transmission continues until the SPSCK returns to its idle level following the shift of the last data bit. See 15.5 Transmission Formats.

NOTE

Setting the MODF flag does not clear the SPMSTR bit. Reading SPMSTR when MODF = 1 will indicate a mode fault error occurred in either master mode or slave mode.

When CPHA = 0, a MODF occurs if a slave is selected ($\overline{SS}$ is at logic 0) and later unselected ($\overline{SS}$ is at logic 1) even if no SPSCK is sent to that slave. This happens because $\overline{SS}$ at logic 0 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.

In a slave SPI (MSTR = 0), the MODF bit generates an SPI receiver/error CPU interrupt request if the ERRIE bit is set. The MODF bit does not clear the SPE bit or reset the SPI in any way. Software can abort the SPI transmission by clearing the SPE bit of the slave.

NOTE

A logic 1 voltage on the $\overline{SS}$ pin of a slave SPI puts the MISO pin in a high impedance state. Also, the slave SPI ignores all incoming SPSCK clocks, even if it was already in the middle of a transmission.

To clear the MODF flag, read the SPSCR with the MODF bit set and then write to the SPCR register. This entire clearing procedure must occur with no MODF condition existing or else the flag is not cleared.

Address: $0044

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
Reset:	0	0	1	0	1	0	0	0

R = Reserved

Figure 15-14. SPI Control Register (SPCR)

SPRIE — SPI Receiver Interrupt Enable Bit

This read/write bit enables CPU interrupt requests generated by the SPRF bit. The SPRF bit is set when a byte transfers from the shift register to the receive data register. Reset clears the SPRIE bit.

1 = SPRF CPU interrupt requests enabled
0 = SPRF CPU interrupt requests disabled

SPMSTR — SPI Master Bit

This read/write bit selects master mode operation or slave mode operation. Reset sets the SPMSTR bit.

1 = Master mode
0 = Slave mode

CPOL — Clock Polarity Bit

This read/write bit determines the logic state of the SPSCK pin between transmissions. See Figure 15-5 and Figure 15-7. To transmit data between SPI modules, the SPI modules must have identical CPOL values. Reset clears the CPOL bit.

CPHA — Clock Phase Bit

This read/write bit controls the timing relationship between the serial clock and SPI data. See Figure 15-5 and Figure 15-7. To transmit data between SPI modules, the SPI modules must have identical CPHA bits. When CPHA = 0, the $\overline{SS}$ pin of the slave SPI module must be set to logic 1 between bytes. See Figure 15-13. Reset sets the CPHA bit.

When CPHA = 0 for a slave, the falling edge of $\overline{SS}$ indicates the beginning of the transmission. This causes the SPI to leave its idle state and begin driving the MISO pin with the MSB of its data, once the transmission begins, no new data is allowed into the shift register from the data register. Therefore, the slave data register must be loaded with the desired transmit data before the falling edge of $\overline{SS}$. Any data written after the falling edge is stored in the data register and transferred to the shift register at the current transmission.

When CPHA = 1 for a slave, the first edge of the SPSCK indicates the beginning of the transmission. The same applies when $\overline{SS}$ is high for a slave. The MISO pin is held in a high-impedance state, and the incoming SPSCK is ignored. In certain cases, it may also cause the MODF flag to be set. See 15.6.2 Mode Fault Error. A logic 1 on the $\overline{SS}$ pin does not in any way affect the state of the SPI state machine.

SPWOM — SPI Wired-OR Mode Bit

This read/write bit disables the pullup devices on pins SPSCK, MOSI, and MISO so that those pins become open-drain outputs.

1 = Wired-OR SPSCK, MOSI, and MISO pins
0 = Normal push-pull SPSCK, MOSI, and MISO pins

16.7.5 TIMA Channel Registers

These read/write registers contain the captured TIMA counter value of the input capture function or the output compare value of the output compare function. The state of the TIMA channel registers after reset is unknown.

In input capture mode (MSxB:MSxA = 0:0), reading the high byte of the TIMA channel x registers (TACHxH) inhibits input captures until the low byte (TACHxL) is read.

In output compare mode (MSxB:MSxA ≠ 0:0), writing to the high byte of the TIMA channel x registers (TACHxH) inhibits output compares until the low byte (TACHxL) is written.

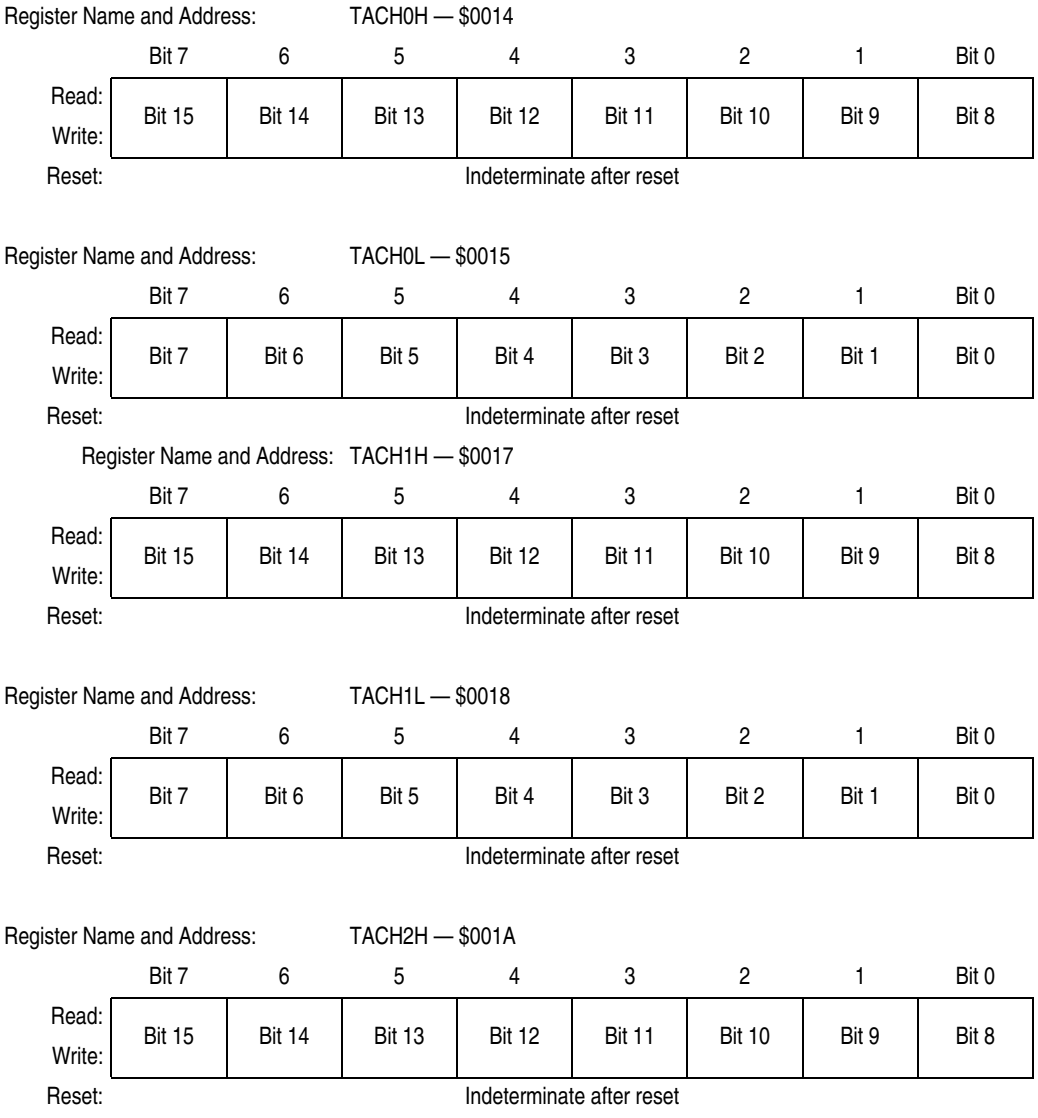


Figure 16-10. TIMA Channel Registers (TACH0H/L–TACH3H/L)

17.7.2 TIMB Counter Registers

The two read-only TIMB counter registers contain the high and low bytes of the value in the TIMB counter. Reading the high byte (TBCNTH) latches the contents of the low byte (TBCNTL) into a buffer. Subsequent reads of TBCNTH do not affect the latched TBCNTL value until TBCNTL is read. Reset clears the TIMB counter registers. Setting the TIMB reset bit (TRST) also clears the TIMB counter registers.

NOTE

If TBCNTH is read during a break interrupt, be sure to unlatch TBCNTL by reading TBCNTL before exiting the break interrupt. Otherwise, TBCNTL retains the value latched during the break.

Register Name and Address: TBCNTH — $0052

	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:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

Register Name and Address: TBCNTL — $0053

	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:	R	R	R	R	R	R	R	R
Reset:	0	0	0	0	0	0	0	0

R = Reserved

Figure 17-6. TIMB Counter Registers (TBCNTH and TBCNTL)

17.7.3 TIMB Counter Modulo Registers

The read/write TIMB modulo registers contain the modulo value for the TIMB counter. When the TIMB counter reaches the modulo value, the overflow flag (TOF) becomes set, and the TIMB counter resumes counting from $0000 at the next timer clock. Writing to the high byte (TBMODH) inhibits the TOF bit and overflow interrupts until the low byte (TBMODL) is written. Reset sets the TIMB counter modulo registers.

Register Name and Address: TBMODH — $0054

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

Register Name and Address: TBMODL — $0055

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

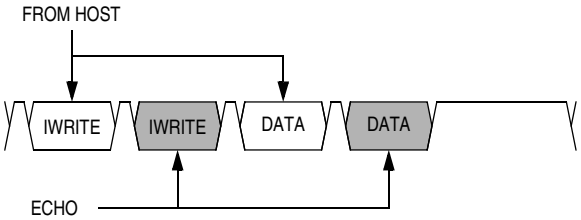
Figure 17-7. TIMB Counter Modulo Registers (TBMODH and TBMODL)

NOTE

Reset the TIMB counter before writing to the TIMB counter modulo registers.

Table 18-6. IWRITE (Indexed Write) Command

Description	Write to last address accessed + 1
Operand	Single data byte
Data Returned	None
Opcode	$19
Command Sequence	



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

Table 18-7. READSP (Read Stack Pointer) Command

Description	Reads stack pointer
Operand	None
Data Returned	Returns incremented stack pointer value (SP + 1) in high-byte:low-byte order
Opcode	$0C
Command Sequence	

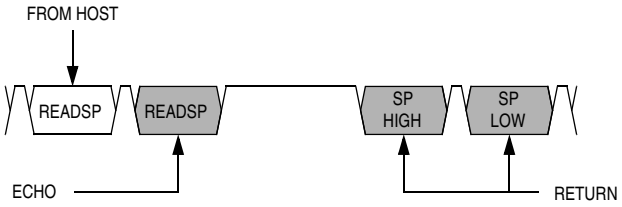
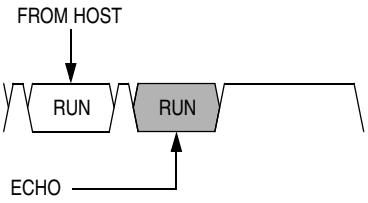


Table 18-8. RUN (Run User Program) Command

Description	Executes PULH and RTI instructions
Operand	None
Data Returned	None
Opcode	$28
Command Sequence	



19.13 Analog-to-Digital Converter (ADC) Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Supply voltage	V_{DDAD}	4.5	—	5.5	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}	—	—	± 4	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	—	003	Hex	$V_{ADIN} = V_{SSAD}$
Full-scale reading	F_{ADI}	3FC	—	3FF	Hex	$V_{ADIN} = V_{DDAD}$
Input capacitance	C_{ADI}	—	—	30	pF	Not tested
V_{REFH}/V_{REFL} current	I_{VREF}	—	1.6	—	mA	
Absolute accuracy (8-bit truncation mode)	A_{AD}	—	—	± 1	LSB	Includes quantization
Quantization error (8-bit truncation mode)	—	—	—	+ 7/8 – 1/8	LSB	

20.4 56-Pin Shrink Dual In-Line Package (SDIP)