

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	16KB (16K 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	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908mr16cfu">https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc908mr16cfu</a>

4.4	I/O Signals	64
4.4.1	Crystal Amplifier Input Pin (OSC1)	64
4.4.2	Crystal Amplifier Output Pin (OSC2)	64
4.4.3	External Filter Capacitor Pin (CGMXFC)	64
4.4.4	PLL Analog Power Pin ( $V_{DDA}$ )	64
4.4.5	Oscillator Enable Signal (SIMOSCEN)	64
4.4.6	Crystal Output Frequency Signal (CGMXCLK)	65
4.4.7	CGM Base Clock Output (CGMOUT)	65
4.4.8	CGM CPU Interrupt (CGMINT)	65
4.5	CGM Registers	65
4.5.1	PLL Control Register	66
4.5.2	PLL Bandwidth Control Register	67
4.5.3	PLL Programming Register	68
4.6	Interrupts	69
4.7	Wait Mode	69
4.8	Acquisition/Lock Time Specifications	70
4.8.1	Acquisition/Lock Time Definitions	70
4.8.2	Parametric Influences on Reaction Time	70
4.8.3	Choosing a Filter Capacitor	71
4.8.4	Reaction Time Calculation	71

## Chapter 5

### Configuration Register (CONFIG)

5.1	Introduction	73
5.2	Functional Description	73
5.3	Configuration Register	74

## Chapter 6

### Computer Operating Properly (COP)

6.1	Introduction	75
6.2	Functional Description	75
6.3	I/O Signals	76
6.3.1	CGMXCLK	76
6.3.2	COPCTL Write	76
6.3.3	Power-On Reset	76
6.3.4	Internal Reset	76
6.3.5	Reset Vector Fetch	76
6.3.6	COPD (COP Disable)	77
6.4	COP Control Register	77
6.5	Interrupts	77
6.6	Monitor Mode	77
6.7	Wait Mode	77
6.8	Stop Mode	77

## Table of Contents

12.6.2	Software Output Disable	141
12.6.3	Output Port Control	141
12.7	Initialization and the PWMEN Bit	142
12.8	PWM Operation in Wait Mode	143
12.9	Control Logic Block	143
12.9.1	PWM Counter Registers	143
12.9.2	PWM Counter Modulo Registers	144
12.9.3	PWMx Value Registers	145
12.9.4	PWM Control Register 1	146
12.9.5	PWM Control Register 2	148
12.9.6	Dead-Time Write-Once Register	150
12.9.7	PWM Disable Mapping Write-Once Register	150
12.9.8	Fault Control Register	150
12.9.9	Fault Status Register	152
12.9.10	Fault Acknowledge Register	153
12.9.11	PWM Output Control Register	154
12.10	PWM Glossary	155

## Chapter 13 Serial Communications Interface Module (SCI)

13.1	Introduction	157
13.2	Features	157
13.3	Functional Description	159
13.3.1	Data Format	160
13.3.2	Transmitter	161
13.3.2.1	Character Length	162
13.3.2.2	Character Transmission	162
13.3.2.3	Break Characters	162
13.3.2.4	Idle Characters	163
13.3.2.5	Inversion of Transmitted Output	163
13.3.2.6	Transmitter Interrupts	163
13.3.3	Receiver	163
13.3.3.1	Character Length	164
13.3.3.2	Character Reception	165
13.3.3.3	Data Sampling	165
13.3.3.4	Framing Errors	167
13.3.3.5	Receiver Wakeup	167
13.3.3.6	Receiver Interrupts	167
13.3.3.7	Error Interrupts	167
13.4	Wait Mode	168
13.5	SCI During Break Module Interrupts	168
13.6	I/O Signals	168
13.6.1	PTF5/TxD (Transmit Data)	168
13.6.2	PTF4/RxD (Receive Data)	169
13.7	I/O Registers	169
13.7.1	SCI Control Register 1	169
13.7.2	SCI Control Register 2	171

significant bits (LSB), located in the ADC data register low, ADRL, can be ignored. However, ADRL must be read after ADRH or else the interlocking will prevent all new conversions from being stored.

Right justification will place only the two MSBs in the corresponding ADC data register high, ADRH, and the eight LSBs in ADC data register low, ADRL. This mode of operation typically is used when a 10-bit unsigned result is desired.

Left justified sign data mode is similar to left justified mode with one exception. The MSB of the 10-bit result, AD9 located in ADRH, is complemented. This mode of operation is useful when a result, represented as a signed magnitude from mid-scale, is needed. Finally, 8-bit truncation mode will place the eight MSBs in ADC data register low, ADRL. The two LSBs are dropped. This mode of operation is used when compatibility with 8-bit ADC designs are required. No interlocking between ADRH and ADRL is present.

#### NOTE

Quantization error is affected when only the most significant eight bits are used as a result. See [Figure 3-3](#).

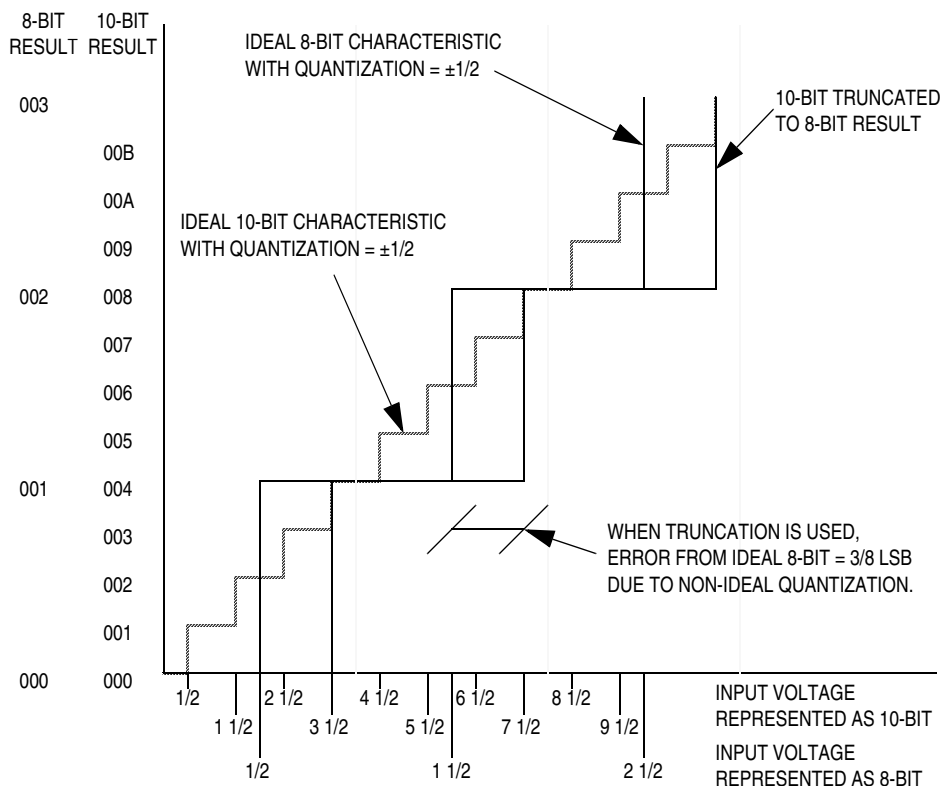


Figure 3-3. 8-Bit Truncation Mode Error

### 3.3.6 Monotonicity

The conversion process is monotonic and has no missing codes.

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

### 4.3.1 Crystal Oscillator Circuit

The crystal oscillator circuit consists of an inverting amplifier and an external crystal. The OSC1 pin is the input to the amplifier and the OSC2 pin is the output. The SIMOSCEN signal from the system integration module (SIM) enables the crystal oscillator circuit.

The CGMXCLK signal is the output of the crystal oscillator circuit and runs at a rate equal to the crystal frequency. CGMXCLK is then buffered to produce CGMRCLK, the PLL reference clock.

CGMXCLK can be used by other modules which require precise timing for operation. The duty cycle of CGMXCLK is not guaranteed to be 50 percent and depends on external factors, including the crystal and related external components.

An externally generated clock also can feed the OSC1 pin of the crystal oscillator circuit. Connect the external clock to the OSC1 pin and let the OSC2 pin float.

### 4.3.2 Phase-Locked Loop Circuit (PLL)

The PLL is a frequency generator that can operate in either acquisition mode or tracking mode, depending on the accuracy of the output frequency. The PLL can change between acquisition and tracking modes either automatically or manually.

#### 4.3.2.1 PLL Circuits

The PLL consists of these circuits:

- Voltage-controlled oscillator (VCO)
- Modulo VCO frequency divider
- Phase detector
- Loop filter
- Lock detector

The operating range of the VCO is programmable for a wide range of frequencies and for maximum immunity to external noise, including supply and CGMXFC noise. The VCO frequency is bound to a range from roughly one-half to twice the center-of-range frequency,  $f_{VRS}$ . Modulating the voltage on the CGMXFC pin changes the frequency within this range. By design,  $f_{VRS}$  is equal to the nominal center-of-range frequency,  $f_{NOM}$ , (4.9152 MHz) times a linear factor, L or (L)  $f_{NOM}$ .

CGMRCLK is the PLL reference clock, a buffered version of CGMXCLK. CGMRCLK runs at a frequency,  $f_{RCLK}$ , and is fed to the PLL through a buffer. The buffer output is the final reference clock, CGMRDV, running at a frequency,  $f_{RDV} = f_{RCLK}$ .

The VCO's output clock, CGMVCLK, running at a frequency,  $f_{VCLK}$ , is fed back through a programmable modulo divider. The modulo divider reduces the VCO clock by a factor, N. The divider's output is the VCO feedback clock, CGMVDV, running at a frequency,  $f_{VDV} = f_{VCLK}/N$ . (See [4.3.2.4 Programming the PLL](#) for more information.)

The phase detector then compares the VCO feedback clock, CGMVDV, with the final reference clock, CGMRDV. A correction pulse is generated based on the phase difference between the two signals. The loop filter then slightly alters the dc voltage on the external capacitor connected to CGMXFC based on the width and direction of the correction pulse. The filter can make fast or slow corrections depending on its mode, described in [4.3.2.2 Acquisition and Tracking Modes](#). The value of the external capacitor and the reference frequency determines the speed of the corrections and the stability of the PLL.

## Clock Generator Module (CGM)

The K factor in the equations is derived from internal PLL parameters.  $K_{ACQ}$  is the K factor when the PLL is configured in acquisition mode, and  $K_{TRK}$  is the K factor when the PLL is configured in tracking mode. See [4.3.2.2 Acquisition and Tracking Modes](#).

$$t_{ACQ} = \left( \frac{V_{DDA}}{f_{RDV}} \right) \left( \frac{8}{K_{ACQ}} \right)$$

$$t_{AL} = \left( \frac{V_{DDA}}{f_{RDV}} \right) \left( \frac{4}{K_{TRK}} \right)$$

$$t_{Lock} = t_{ACQ} + t_{AL}$$

### NOTE

*The inverse proportionality between the lock time and the reference frequency.*

In automatic bandwidth control mode, the acquisition and lock times are quantized into units based on the reference frequency. See [4.3.2.3 Manual and Automatic PLL Bandwidth Modes](#). A certain number of clock cycles,  $n_{ACQ}$ , is required to ascertain that the PLL is within the tracking mode entry tolerance,  $\Delta_{TRK}$ , before exiting acquisition mode. A certain number of clock cycles,  $n_{TRK}$ , is required to ascertain that the PLL is within the lock mode entry tolerance,  $\Delta_{Lock}$ . Therefore, the acquisition time,  $t_{ACQ}$ , is an integer multiple of  $n_{ACQ}/f_{RDV}$ , and the acquisition to lock time,  $t_{AL}$ , is an integer multiple of  $n_{TRK}/f_{RDV}$ . Also, since the average frequency over the entire measurement period must be within the specified tolerance, the total time usually is longer than  $t_{Lock}$  as calculated in the previous example.

In manual mode, it is usually necessary to wait considerably longer than  $t_{Lock}$  before selecting the PLL clock (see [4.3.3 Base Clock Selector Circuit](#)) because the factors described in [4.8.2 Parametric Influences on Reaction Time](#) may slow the lock time considerably.

### 10.3.2 Data Direction Register B

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

Address:	\$0005							
	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Write:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
Reset:	0	0	0	0	0	0	0	0

Figure 10-6. Data Direction Register B (DDRB)

#### DDRB[7:0] — Data Direction Register B Bits

These read/write bits control port B data direction. Reset clears DDRB[7:0], configuring all port B pins as inputs.

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

#### NOTE

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

Figure 10-7 shows the port B I/O logic.

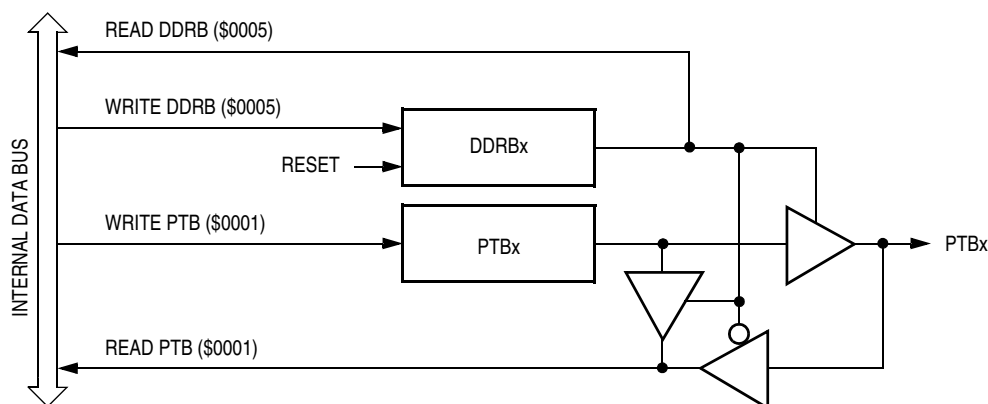


Figure 10-7. Port B I/O Circuit

When bit DDRBx is a logic 1, reading address \$0001 reads the PTBx data latch. When bit DDRBx is a logic 0, reading address \$0001 reads the voltage level on the pin. The data latch can always be written, regardless of the state of its data direction bit. Table 10-2 summarizes the operation of the port B pins.

Table 10-2. Port B Pin Functions

DDRB Bit	PTB Bit	I/O Pin Mode	Accesses to DDRB	Accesses to PTB	
			Read/Write	Read	Write
0	X <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRB[7:0]	Pin	PTB[7:0] <sup>(3)</sup>
1	X	Output	DDRB[7:0]	PTB[7:0]	PTB[7:0]

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



## 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:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
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

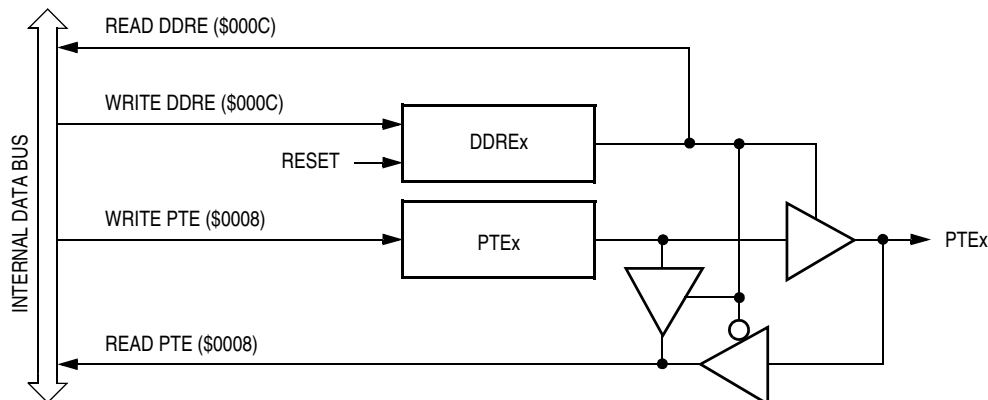
These read/write bits control port E data direction. Reset clears DDRE[7:0], configuring all port E pins as inputs.

- 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 <sup>(1)</sup>	Input, Hi-Z <sup>(2)</sup>	DDRE[7:0]	Pin	PTE[7:0] <sup>(3)</sup>
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.

## Chapter 11

# Power-On Reset (POR)

### 11.1 Introduction

This section describes the power-on reset (POR) module.

### 11.2 Functional Description

The POR module provides a known, stable signal to the microcontroller unit (MCU) at power-on. This signal tracks  $V_{DD}$  until the MCU generates a feedback signal to indicate that it is properly initialized. At this time, the POR drives its output low.

The POR is not a brown-out detector, low-voltage detector, or glitch detector.  $V_{DD}$  at the POR must go completely to 0 to reset the microcontroller unit (MCU). To detect power-loss conditions, use a low-voltage inhibit module (LVI) or other suitable circuit.

## 12.3.2 Prescaler

To permit lower PWM frequencies, a prescaler is provided which will divide the PWM clock frequency by 1, 2, 4, or 8. [Table 12-1](#) shows how setting the prescaler bits in PWM control register 2 affects the PWM clock frequency. This prescaler is buffered and will not be used by the PWM generator until the LDOK bit is set and a new PWM reload cycle begins.

**Table 12-1. PWM Prescaler**

Prescaler Bits PRSC1 and PRSC0	PWM Clock Frequency
00	$f_{OP}$
01	$f_{OP}/2$
10	$f_{OP}/4$
11	$f_{OP}/8$

## 12.4 PWM Generators

Pulse-width modulator (PWM) generators are discussed in this subsection.

### 12.4.1 Load Operation

To help avoid erroneous pulse widths and PWM periods, the modulus, prescaler, and PWM value registers are buffered. New PWM values, counter modulus values, and prescalers can be loaded from their buffers into the PWM module every one, two, four, or eight PWM cycles. LDFQ1 and LDFQ0 in PWM control register 2 are used to control this reload frequency, as shown in [Table 12-2](#). When a reload cycle arrives, regardless of whether an actual reload occurs (as determined by the LDOK bit), the PWM reload flag bit in PWM control register 1 will be set. If the PWMINT bit in PWM control register 1 is set, a CPU interrupt request will be generated when PWMF is set. Software can use this interrupt to calculate new PWM parameters in real time for the PWM module.

**Table 12-2. PWM Reload Frequency**

Reload Frequency Bits LDFQ1 and LDFQ0	PWM Reload Frequency
00	Every PWM cycle
01	Every 2 PWM cycles
10	Every 4 PWM cycles
11	Every 8 PWM cycles

## 12.5 Output Control

This subsection discusses output control.

### 12.5.1 Selecting Six Independent PWMs or Three Complementary PWM Pairs

The PWM outputs can be configured as six independent PWM channels or three complementary channel pairs. The option INDEP determines which mode is used (see [5.2 Functional Description](#)). If complementary operation is chosen, the PWM pins are paired as shown in [Figure 12-12](#). Operation of one pair is then determined by one PWM value register. This type of operation is meant for use in motor drive circuits such as the one in [Figure 12-13](#).

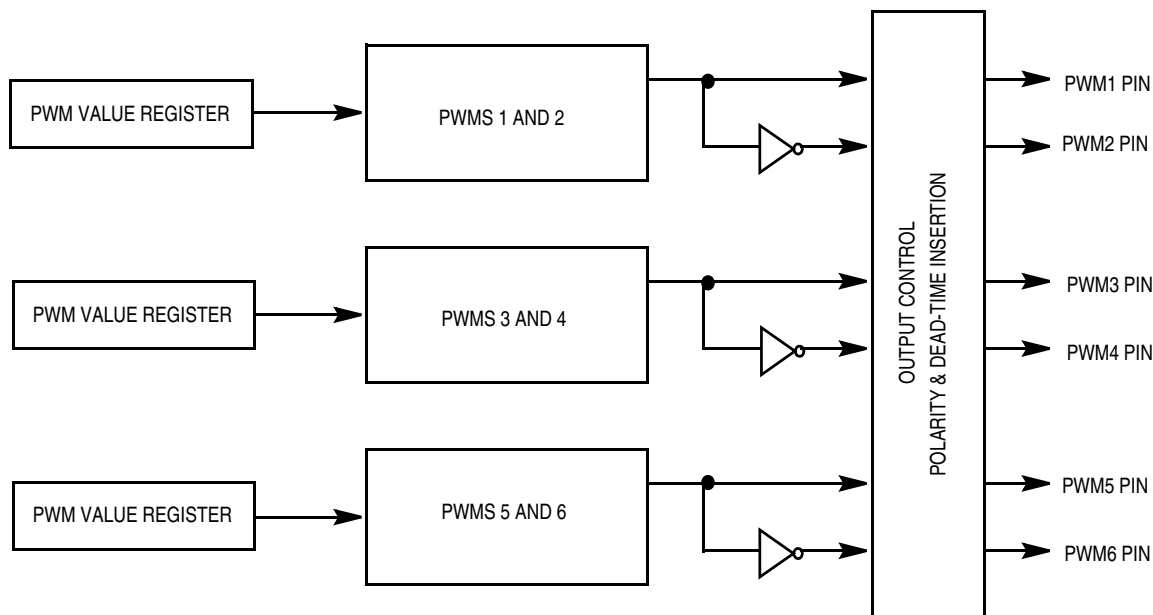


Figure 12-12. Complementary Pairing

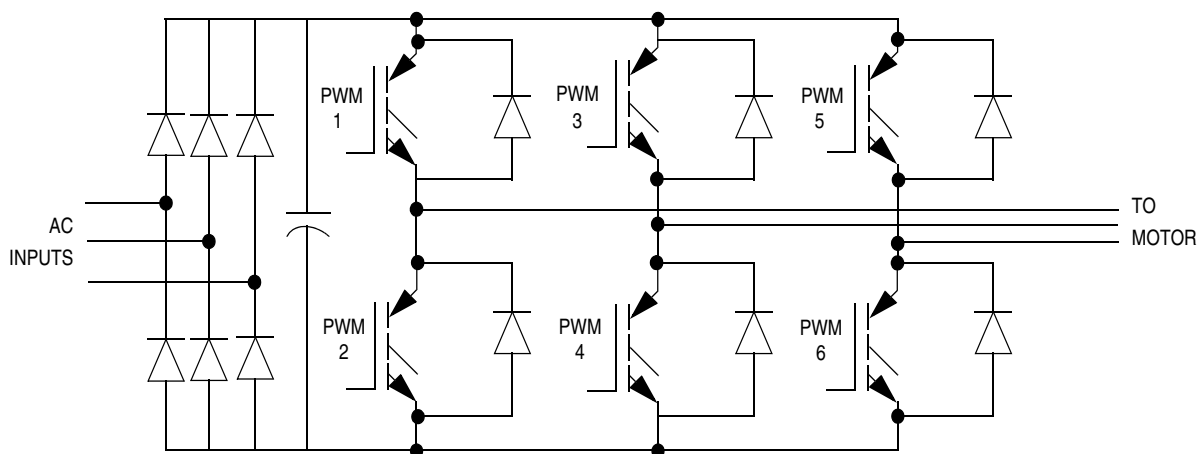


Figure 12-13. Typical AC Motor Drive

# Chapter 13

## Serial Communications Interface Module (SCI)

### 13.1 Introduction

This section describes the serial communications interface module (SCI, version D), which allows high-speed asynchronous communications with peripheral devices and other microcontroller units (MCUs).

### 13.2 Features

Features of the SCI module include:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 32 programmable baud rates
- Programmable 8-bit or 9-bit character length
- Separately enabled transmitter and receiver
- Separate receiver and transmitter CPU interrupt requests
- Separate receiver and transmitter
- Programmable transmitter output polarity
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Interrupt-driven operation with eight interrupt flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- Hardware parity checking
- 1/16 bit-time noise detection





## SPE — SPI Enable Bit

This read/write bit enables the SPI module. Clearing SPE causes a partial reset of the SPI. See [15.8 Resetting the SPI](#). Reset clears the SPE bit.

- 1 = SPI module enabled
- 0 = SPI module disabled

## SPTIE— SPI Transmit Interrupt Enable Bit

This read/write bit enables CPU interrupt requests generated by the SPTE bit. SPTE is set when a byte transfers from the transmit data register to the shift register. Reset clears the SPTIE bit.

- 1 = SPTE CPU interrupt requests enabled
- 0 = SPTE CPU interrupt requests disabled

## 15.12.2 SPI Status and Control Register

The SPI status and control register (SPSCR) contains flags to signal these conditions:

- Receive data register full
- Failure to clear SPRF bit before next byte is received (overflow error)
- Inconsistent logic level on  $\overline{SS}$  pin (mode fault error)
- Transmit data register empty

The SPI status and control register also contains bits that perform these functions:

- Enable error interrupts
- Enable mode fault error detection
- Select master SPI baud rate

Address: \$0045

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
Write:	R		R	R	R			
Reset:	0	0	0	0	1	0	0	0
	R = Reserved							

**Figure 15-15. SPI Status and Control Register (SPSCR)**

## SPRF — SPI Receiver Full Bit

This clearable, read-only flag is set each time a byte transfers from the shift register to the receive data register. SPRF generates a CPU interrupt request if the SPRIE bit in the SPI control register is set also.

During an SPRF CPU interrupt (DMAS = 0), the CPU clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register.

Reset clears the SPRF bit.

- 1 = Receive data register full
- 0 = Receive data register not full

## ERRIE — Error Interrupt Enable Bit

This read/write bit enables the MODF and OVRF bits to generate CPU interrupt requests. Reset clears the ERRIE bit.

- 1 = MODF and OVRF can generate CPU interrupt requests.
- 0 = MODF and OVRF cannot generate CPU interrupt requests.



The value in the TIMB 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 TIMB channel registers produces a duty cycle of 128/256 or 50 percent.

#### 17.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in [17.3.4 Pulse-Width Modulation \(PWM\)](#). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the value currently in the TIMB channel registers.

An unsynchronized write to the TIMB 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 TIMB overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIMB may pass the new value before it is written to the TIMB channel registers.

Use this method to synchronize unbuffered changes in the PWM pulse width on channel x:

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

#### NOTE

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

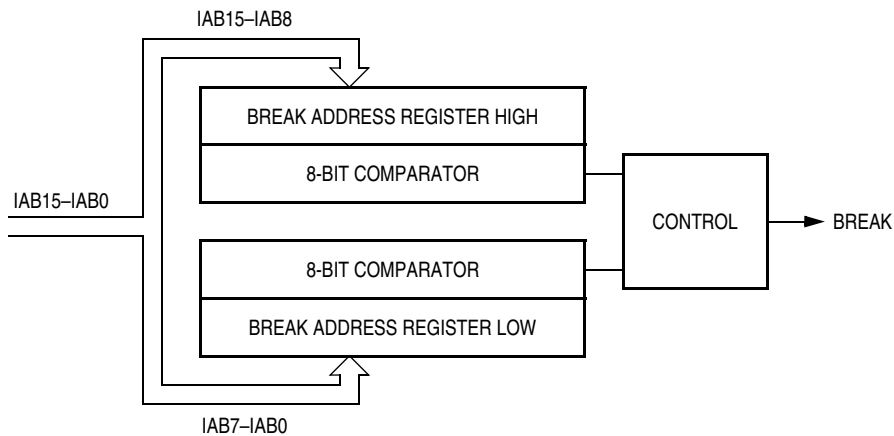
#### 17.3.4.2 Buffered PWM Signal Generation

Channels 0 and 1 can be linked to form a buffered PWM channel whose output appears on the PTE1/TCH0B pin. The TIMB channel registers of the linked pair alternately control the pulse width of the output.

Setting the MS0B bit in TIMB channel 0 status and control register (TBSC0) links channel 0 and channel 1. The TIMB channel 0 registers initially control the pulse width on the PTE1/TCH0B pin. Writing to the TIMB channel 1 registers enables the TIMB channel 1 registers to synchronously control the pulse width at the beginning of the next PWM period. At each subsequent overflow, the TIMB channel registers (0 or 1) that control the pulse width are the ones written to last. TBSC0 controls and monitors the buffered PWM function, and TIMB channel 1 status and control register (TBSC1) is unused. While the MS0B bit is set, the channel 1 pin, PTE2/TCH1B, is available as a general-purpose I/O pin.

#### NOTE

*In buffered PWM signal generation, do not write new pulse width values to the currently active channel registers. User software should track the*



**Figure 18-1. Break Module Block Diagram**

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
\$FE00	SIM Break Status Register (SBSR) <a href="#">See page 255.</a>	Read:	R	R	R	R	R	R	BW	R
		Write:								
		Reset:	0							
\$FE03	SIM Break Flag Control Register (SBFCR) <a href="#">See page 255.</a>	Read:	BCFE	R	R	R	R	R	R	R
		Write:								
		Reset:	0							
\$FE0C	Break Address Register High (BRKH) <a href="#">See page 254.</a>	Read:	Bit 15	14	13	12	11	10	9	Bit 8
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0D	Break Address Register Low (BRKL) <a href="#">See page 254.</a>	Read:	Bit 7	6	5	4	3	2	1	Bit 0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
\$FE0E	Break Status and Control Register (BRKSCR) <a href="#">See page 254.</a>	Read:	BRKE	BRKA	0	0	0	0	0	0
		Write:								
		Reset:	0	0	0	0	0	0	0	0

Note: Writing a 0 clears BW.

  = Unimplemented      R = Reserved

**Figure 18-2. I/O Register Summary**

Table 18-2. Monitor Mode Signal Requirements and Options

$\overline{\text{IRQ}}$	RESET (S1)	\$FFFE /\$FFFF	PLL	PTC3	PTC4	PTC2 (S2)	External Clock <sup>(1)</sup>	CGMOUT	Bus Frequency	COP	For Serial Communication <sup>(2)</sup>			Comment
											PTA0	PTA7 (S3)	Baud Rate <sup>(3) (4)</sup>	
X	GND	X	X	X	X	X	X	0	0	Disabled	X	X	0	No operation until reset goes high
$V_{\text{TST}}$	$V_{\text{DD}}$ or $V_{\text{TST}}$	X	OFF	1	0	0	4.9152 MHz	4.9152 MHz	2.4576 MHz	Disabled	1	0	9600	PTC3 and PTC2 voltages only required if $\overline{\text{IRQ}} = V_{\text{TST}}$ ; PTC2 determines frequency divider
											X	1	DNA	
$V_{\text{TST}}$	$V_{\text{DD}}$ or $V_{\text{TST}}$	X	OFF	1	0	1	9.8304 MHz	4.9152 MHz	2.4576 MHz	Disabled	1	0	9600	PTC3 and PTC2 voltages only required if $\overline{\text{IRQ}} = V_{\text{TST}}$ ; PTC2 determines frequency divider
											X	1	DNA	
$V_{\text{DD}}$	$V_{\text{DD}}$	\$FFFF Blank	OFF	X	X	X	9.8304 MHz	4.9152 MHz	2.4576 MHz	Disabled	1	0	9600	External frequency always divided by 4
											X	1	DNA	
$V_{\text{DD}}$ or GND	$V_{\text{TST}}$	\$FFFF Blank	OFF	X	X	X	X	—	—	Enabled	X	X	—	Enters user mode — will encounter an illegal address reset
$V_{\text{DD}}$ or GND	$V_{\text{DD}}$ or $V_{\text{TST}}$	Non-\$FF Programmed	OFF	X	X	X	X	—	—	Enabled	X	X	—	Enters user mode

1. External clock is derived by a 32.768 kHz crystal or a 4.9152/9.8304 MHz off-chip oscillator.

2. DNA = does not apply, X = don't care

3. PTA0 = 1 if serial communication; PTA0 = X if parallel communication

4. PTA7 = 0 → serial, PTA7 = 1 → parallel communication for security code entry



## 19.12 CGM Acquisition/Lock Time Specifications

Description	Symbol	Min	Typ	Max	Notes
Filter capacitor multiply factor	$C_{FACT}$	—	0.0154	—	F/sV
Acquisition mode time factor	$K_{ACQ}$	—	0.1135	—	V
Tracking mode time factor	$K_{TRK}$	—	0.0174	—	V
Manual mode time to stable	$t_{ACQ}$	—	$(8 \cdot V_{DDA}) / (f_{XCLK} \cdot K_{ACQ})$	—	If $C_F$ chosen correctly
Manual stable to lock time	$t_{AL}$	—	$(4 \cdot V_{DDA}) / (f_{XCLK} \cdot K_{TRK})$	—	If $C_F$ chosen correctly
Manual acquisition time	$t_{Lock}$	—	$t_{ACQ} + t_{AL}$	—	
Tracking mode entry frequency tolerance	$\Delta_{TRK}$	0	—	$\pm 3.6\%$	
Acquisition mode entry frequency tolerance	$\Delta_{ACQ}$	$\pm 6.3\%$	—	$\pm 7.2\%$	
Lock entry frequency tolerance	$\Delta_{Lock}$	0	—	$\pm 0.9\%$	
Lock exit frequency tolerance	$\Delta_{UNL}$	$\pm 0.9\%$	—	$\pm 1.8\%$	
Reference cycles per acquisition mode measurement	$n_{ACQ}$	—	32	—	
Reference cycles per tracking mode measurement	$n_{TRK}$	—	128	—	
Automatic mode time to stable	$t_{ACQ}$	$n_{ACQ} / f_{XCLK}$	$(8 \cdot V_{DDA}) / (f_{XCLK} \cdot K_{ACQ})$	—	If $C_F$ chosen correctly
Automatic stable to lock time	$t_{AL}$	$n_{TRK} / f_{XCLK}$	$(4 \cdot V_{DDA}) / (f_{XCLK} \cdot K_{TRK})$	—	If $C_F$ chosen correctly
Automatic lock time	$t_{Lock}$	—	$t_{ACQ} + t_{AL}$	—	
PLL jitter (deviation of average bus frequency over 2 ms)	$f_J$	0	—	$\pm (f_{XCLK}) \cdot (0.025\%) \cdot (N/4)$	N = VCO freq. mult.

