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	Active
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 ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	64-QFP
Supplier Device Package	64-QFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908mr32vfue

1.4.6 External Filter Capacitor Pin (CGMXFC)

CGMXFC is an external filter capacitor connection for the CGM. See Chapter 4 Clock Generator Module (CGM).

1.4.7 Analog Power Supply Pins (V_{DDAD} and V_{SSAD})

V_{DDAD} and V_{SSAD} are the power supply pins for the analog-to-digital converter. Decoupling of these pins should be per the digital supply. See Chapter 3 Analog-to-Digital Converter (ADC).

1.4.8 ADC Voltage Decoupling Capacitor Pin (V_{REFH})

V_{REFH} is the power supply for setting the reference voltage. Connect the V_{REFH} pin to the same voltage potential as V_{DDAD}. See Chapter 3 Analog-to-Digital Converter (ADC).

1.4.9 ADC Voltage Reference Low Pin (V_{REFL})

V_{REFL} is the lower reference supply for the ADC. Connect the V_{REFL} pin to the same voltage potential as V_{SSAD}. See Chapter 3 Analog-to-Digital Converter (ADC).

1.4.10 Port A Input/Output (I/O) Pins (PTA7–PTA0)

PTA7–PTA0 are general-purpose bidirectional input/output (I/O) port pins. See Chapter 10 Input/Output (I/O) Ports (PORTS).

1.4.11 Port B I/O Pins (PTB7/ATD7–PTB0/ATD0)

Port B is an 8-bit special function port that shares all eight pins with the analog-to-digital converter (ADC). See Chapter 3 Analog-to-Digital Converter (ADC) and Chapter 10 Input/Output (I/O) Ports (PORTS).

1.4.12 Port C I/O Pins (PTC6–PTC2 and PTC1/ATD9–PTC0/ATD8)

PTC6–PTC2 are general-purpose bidirectional I/O port pins Chapter 10 Input/Output (I/O) Ports (PORTS). PTC1/ATD9–PTC0/ATD8 are special function port pins that are shared with the analog-to-digital converter (ADC). See Chapter 3 Analog-to-Digital Converter (ADC) and Chapter 10 Input/Output (I/O) Ports (PORTS).

1.4.13 Port D Input-Only Pins (PTD6/$\overline{\text{IS3}}$–PTD4/$\overline{\text{IS1}}$ and PTD3/FAULT4–PTD0/FAULT1)

PTD6/$\overline{\text{IS3}}$–PTD4/$\overline{\text{IS1}}$ are special function input-only port pins that also serve as current sensing pins for the pulse-width modulator module (PWMMC). PTD3/FAULT4–PTD0/FAULT1 are special function port pins that also serve as fault pins for the PWMMC. See Chapter 12 Pulse-Width Modulator for Motor Control (PWMMC) and Chapter 10 Input/Output (I/O) Ports (PORTS).

1.4.14 PWM Pins (PWM6–PWM1)

PWM6–PWM1 are dedicated pins used for the outputs of the pulse-width modulator module (PWMMC). These are high-current sink pins. See Chapter 12 Pulse-Width Modulator for Motor Control (PWMMC) and Chapter 19 Electrical Specifications.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0032	PWM 5 Value Register High (PMVAL5H) See page 145.	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
$0033	PWM 5 Value Register Low (PVAL5L) See page 145.	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
$0034	PWM 6 Value Register High (PVAL6H) See page 145.	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
$0035	PWM 6 Value Register Low (PMVAL6L) See page 145.	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
$0036	Dead-Time Write-Once Register (DEADTM) See page 150.	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
$0037	PWM Disable Mapping Write-Once Register (DISMAP) See page 137.	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
$0038	SCI Control Register 1 (SCC1) See page 169.	Read:	LOOPS	ENSCI	TXINV	M	WAKE	ILTY	PEN	PTY
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0039	SCI Control Register 2 (SCC2) See page 171.	Read:	SCTIE	TCIE	SCRIE	ILIE	TE	RE	RWU	SBK
		Write:								
		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)

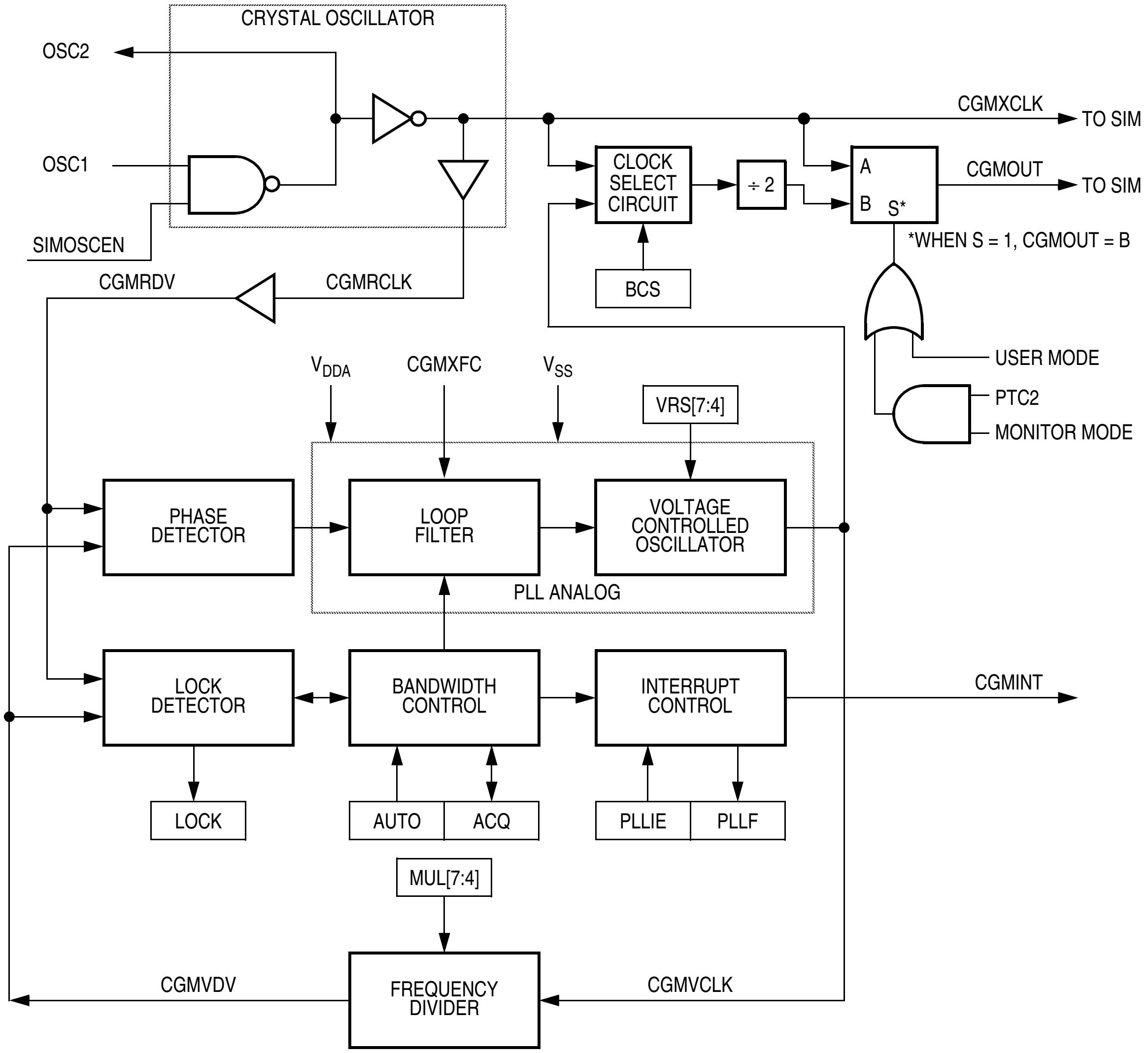


Figure 4-1. CGM Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$005C	PLL Control Register (PCTL) See page 66.	Read:	PLLIE	PLLF	PLLON	BCS	1	1	1	1
		Write:		R			R	R	R	R
		Reset:	0	0	1	0	1	1	1	1
$005D	PLL Bandwidth Control Register (PBWC) See page 67.	Read:	AUTO	LOCK	$\overline{\text{ACQ}}$	XLD	0	0	0	0
		Write:		R			R	R	R	R
		Reset:	0	0	0	0	0	0	0	0
$005E	PLL Programming Register (PPG) See page 68.	Read:	MUL7	MUL6	MUL5	MUL4	VRS7	VRS6	VRS5	VRS4
		Write:								
		Reset:	0	1	1	0	0	1	1	0

R = Reserved

Figure 4-2. CGM I/O Register Summary

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0005	Data Direction Register B (DDRB) See page 105.	Read:	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0006	Data Direction Register C (DDRC) See page 106.	Read:	0	DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
		Write:	R							
		Reset:	0	0	0	0	0	0	0	0
$0007			Unimplemented							
$0008	Port E Data Register (PTE) See page 108.	Read:	PTE7	PTE6	PTE5	PTE4	PTE3	PTE2	PTE1	PTE0
		Write:								
		Reset:	Unaffected by reset							
$0009	Port F Data Register (PTF) See page 110.	Read:	0	0	PTF5	PTF4	PTF3	PTF2	PTF1	PTF0
		Write:	R	R						
		Reset:	Unaffected by reset							
$000A	Unimplemented		▨							
$000B	Unimplemented		▨							
$000C	Data Direction Register E (DDRE) See page 109.	Read:	DDRE7	DDRE6	DDRE5	DDRE4	DDRE3	DDRE2	DDRE1	DDRE0
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$000D	Data Direction Register F (DDRF) See page 110.	Read:	0	0	DDRF5	DDRF4	DDRF3	DDRF2	DDRF1	DDRF0
		Write:	R	R						
		Reset:			0	0	0	0	0	0

R = Reserved ▨ = Unimplemented

Figure 10-1. I/O Port Register Summary (Continued)

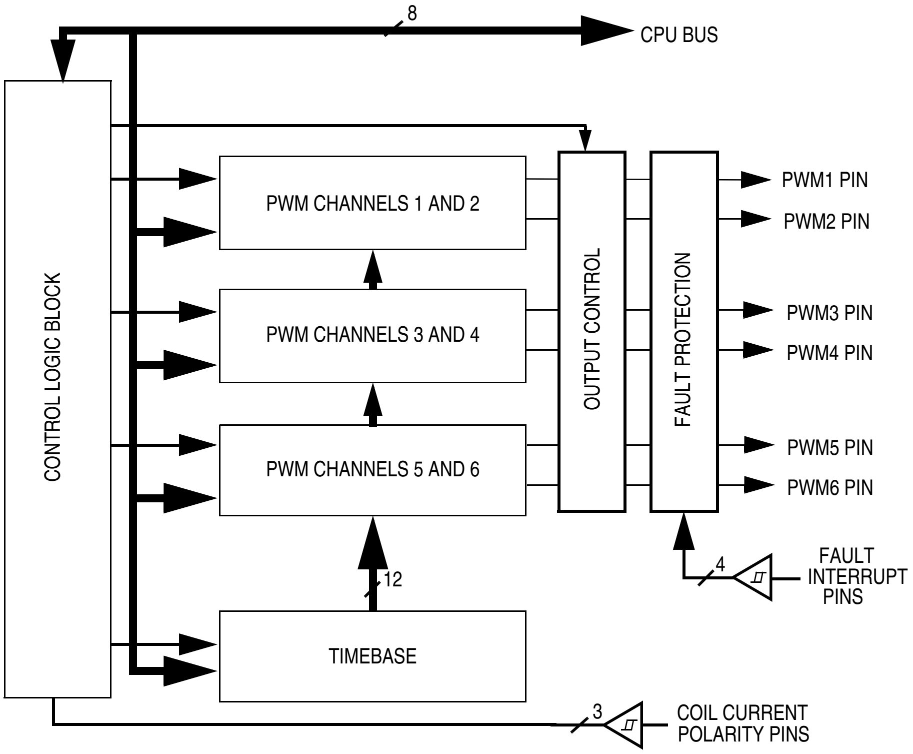


Figure 12-2. PWM Module Block Diagram

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0020	PWM Control Register 1 (PCTL1) See page 146.	Read:	DISX	DISY	PWMINT	PWMF	ISENS1	ISENS0	LDOK	PWMEN
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0021	PWM Control Register 2 (PCTL2) See page 148.	Read:	**LDFQ1**	**LDFQ0**	0	**IPOL1**	**IPOL2**	**IPOL3**	**PRSC1**	**PRSC0**
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0022	Fault Control Register (FCR) See page 150.	Read:	FINT4	FMODE4	FINT3	FMODE3	FINT2	FMODE2	FINT1	FMODE1
		Write:								
		Reset:	0	0	0	0	0	0	0	0
$0023	Fault Status Register (FSR) See page 152.	Read:	FPIN4	FFLAG4	FPIN3	FFLAG3	FPIN2	FFLAG2	FPIN1	FFLAG1
		Write:								
		Reset:	U	0	U	0	U	0	U	0
$0024	Fault Acknowledge Register (FTACK) See page 153.	Read:	0	0	DT6	DT5	DT4	DT3	DT2	DT1
		Write:		FTACK4		FTACK3		FTACK2		FTACK1
		Reset:	0	0	0	0	0	0	0	0

R = Reserved **Bold** = Buffered X = Indeterminate

Figure 12-3. Register Summary (Sheet 1 of 3)

To allow for correction based on different current sensing methods or correction controlled by software, the ISENS1 and ISENS0 bits in PWM control register 1 are provided to choose the correction method. These bits provide correction according to Table 12-5.

Table 12-5. Correction Methods

Current Correction Bits ISENS1 and ISENS0	Correction Method
00 01	Bits IPOL1, IPOL2, and IPOL3 used for correction
10	Current sensing on pins $\overline{\text{IS1}}$, $\overline{\text{IS2}}$, and $\overline{\text{IS3}}$ occurs during the dead-time.
11	Current sensing on pins $\overline{\text{IS1}}$, $\overline{\text{IS2}}$, and $\overline{\text{IS3}}$ occurs at the half cycle in center-aligned mode and at the end of the cycle in edge-aligned mode.

If correction is to be done in software or is not necessary, setting ISENS1:ISENS0 = 00 or = 01 causes the correction to be based on bits IPOL1, IPOL2, and IPOL3 in PWM control register 2. If correction is not required, the user can initialize the IPOLx bits and then only load one PWM value register per PWM pair.

To allow the user to use a current sense scheme based upon sensed phase voltage during dead-time, setting ISENS1:ISENS0 = 10 causes the polarity of the Ix pin to be latched when both the top and bottom PWMs are off (for example, during the dead-time). At the 0 percent and 100 percent duty cycle boundaries, there is no dead-time so no new current value is sensed.

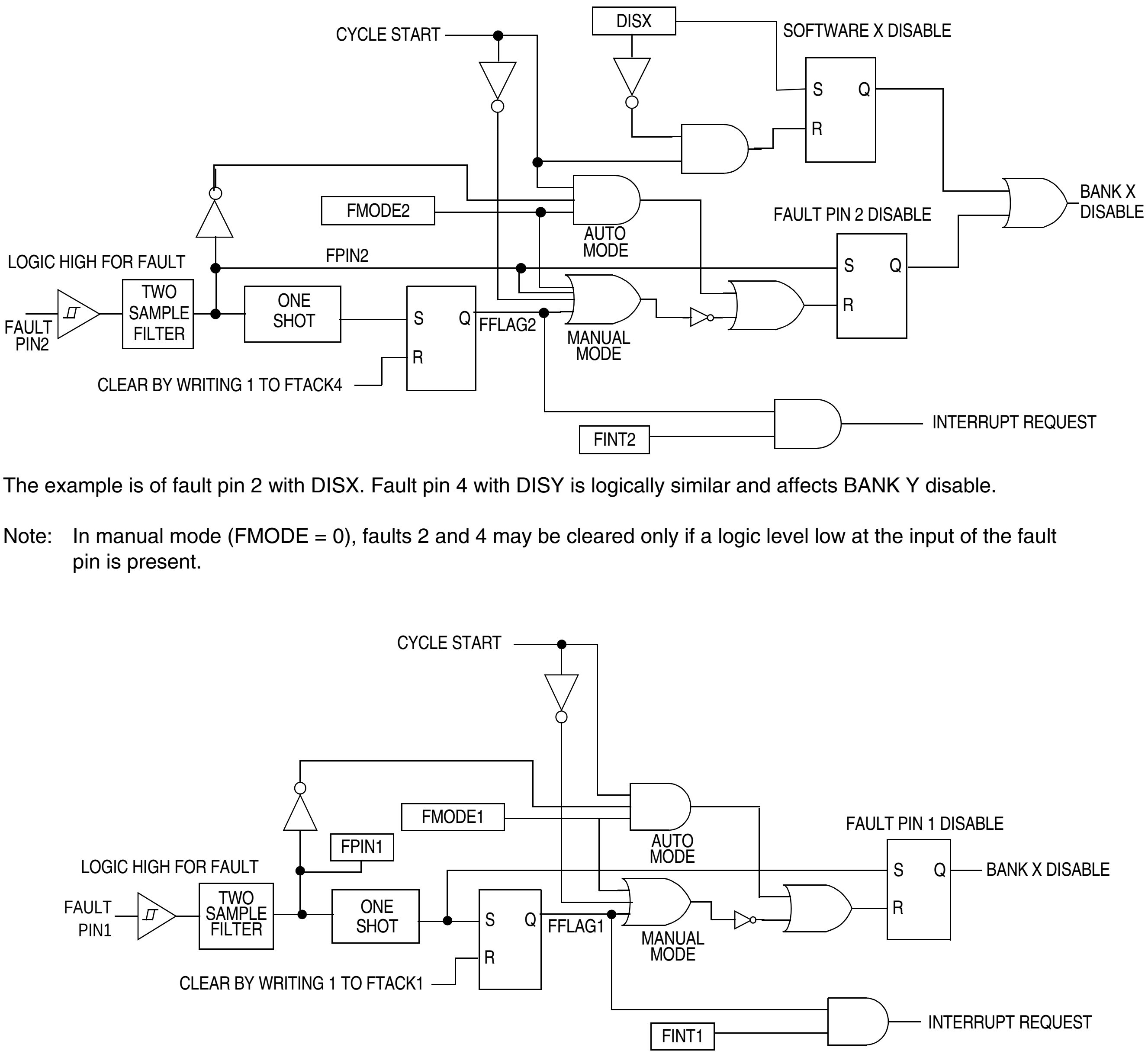
To accommodate other current sensing schemes, setting ISENS1:ISENS0 = 11 causes the polarity of the current sense pin to be latched half-way into the PWM cycle in center-aligned mode and at the end of the cycle in edge-aligned mode. Therefore, even at 0 percent and 100 percent duty cycle, the current is sensed.

Distortion correction is only available in complementary mode. At the beginning of the PWM period, the PWM uses this latched current value or polarity bit to decide whether the top PWM value or bottom PWM value is used. Figure 12-20 shows an example of top/bottom correction for PWMs 1 and 2.

NOTE

The IPOLx bits and the values latched on the ISx pins are buffered so that only one PWM register is used per PWM cycle. If the IPOLx bits or the current sense values change during a PWM period, this new value will not be used until the next PWM period. The ISENSx bits are NOT buffered; therefore, changing the current sensing method could affect the present PWM cycle.

When the PWM is first enabled by setting PWMEN, PWM value registers 1, 3, and 5 will be used if the ISENSx bits are configured for current sensing correction. This is because no current will have previously been sensed.



The example is of fault pin 2 with DISX. Fault pin 4 with DISY is logically similar and affects BANK Y disable.

Note: In manual mode (FMODE = 0), faults 2 and 4 may be cleared only if a logic level low at the input of the fault pin is present.

The example is of fault pin 1. Fault pin 3 is logically similar and affects BANK Y disable.

Note: In manual mode (FMODE = 0), faults 1 and 3 may be cleared regardless of the logic level at the input of the fault pin.

Figure 12-26. PWM Disabling Scheme

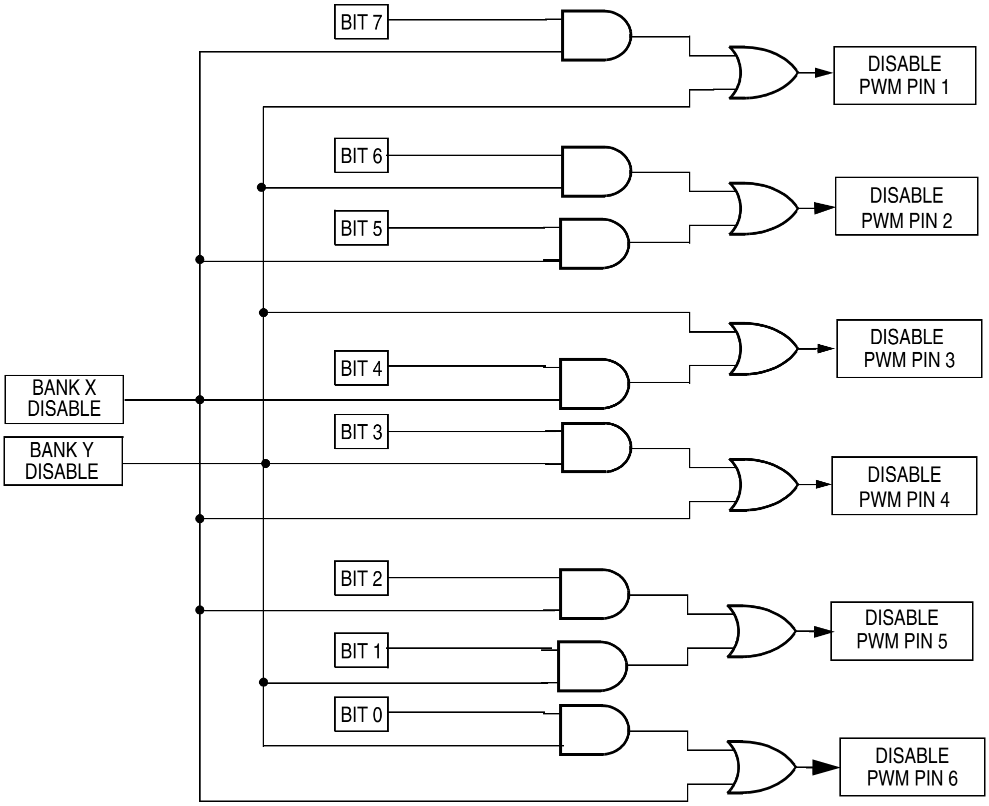


Figure 12-27. PWM Disabling Decode Scheme

12.6.1.1 Fault Pin Filter

Each fault pin incorporates a filter to assist in determining a genuine fault condition. After a fault pin has been logic low for one CPU cycle, a rising edge (logic high) will be synchronously sampled once per CPU cycle for two cycles. If both samples are detected logic high, the corresponding FPIN bit and FFLAG bit will be set. The FPIN bit will remain set until the corresponding fault pin is logic low and synchronously sampled once in the following CPU cycle.

12.6.1.2 Automatic Mode

In automatic mode, the PWM(s) are disabled immediately once a filtered fault condition is detected (logic high). The PWM(s) remain disabled until the filtered fault condition is cleared (logic low) and a new PWM cycle begins as shown in Figure 12-28. Clearing the corresponding FFLAGx event bit will not enable the PWMs in automatic mode.

The filtered fault pin's logic state is reflected in the respective FPINx bit. Any write to this bit is overwritten by the pin state. The FFLAGx event bit is set with each rising edge of the respective fault pin after filtering has been applied. To clear the FFLAGx bit, the user must write a 1 to the corresponding FTACKx bit.

f the FINTx bit is set, a fault condition resulting in setting the corresponding FFLAG bit will also latch a CPU interrupt request. The interrupt request latch is not cleared until one of these actions occurs:

- The FFLAGx bit is cleared by writing a 1 to the corresponding FTACKx bit.
- The FINTx bit is cleared. This will not clear the FFLAGx bit.
- A reset automatically clears all four interrupt latches.

FMODE4 —Fault Mode Selection for Fault Pin 4 Bit (automatic versus manual mode)
This read/write bit allows the user to select between automatic and manual mode faults. For further descriptions of each mode, see 12.6 Fault Protection.
1 = Automatic mode
0 = Manual mode

FINT3 — Fault 3 Interrupt Enable Bit
This read/write bit allows the CPU interrupt caused by faults on fault pin 3 to be enabled. The fault protection circuitry is independent of this bit and will always be active. If a fault is detected, the PWM pins will still be disabled according to the disable mapping register.
1 = Fault pin 3 will cause CPU interrupts.
0 = Fault pin 3 will not cause CPU interrupts.

FMODE3 —Fault Mode Selection for Fault Pin 3 Bit (automatic versus manual mode)
This read/write bit allows the user to select between automatic and manual mode faults. For further descriptions of each mode, see 12.6 Fault Protection.
1 = Automatic mode
0 = Manual mode

FINT2 — Fault 2 Interrupt Enable Bit
This read/write bit allows the CPU interrupt caused by faults on fault pin 2 to be enabled. The fault protection circuitry is independent of this bit and will always be active. If a fault is detected, the PWM pins will still be disabled according to the disable mapping register.
1 = Fault pin 2 will cause CPU interrupts.
0 = Fault pin 2 will not cause CPU interrupts.

FMODE2 —Fault Mode Selection for Fault Pin 2 Bit (automatic versus manual mode)
This read/write bit allows the user to select between automatic and manual mode faults. For further descriptions of each mode, see 12.6 Fault Protection.
1 = Automatic mode
0 = Manual mode

FINT1 — Fault 1 Interrupt Enable Bit
This read/write bit allows the CPU interrupt caused by faults on fault pin 1 to be enabled. The fault protection circuitry is independent of this bit and will always be active. If a fault is detected, the PWM pins will still be disabled according to the disable mapping register.
1 = Fault pin 1 will cause CPU interrupts.
0 = Fault pin 1 will not cause CPU interrupts.

FMODE1 —Fault Mode Selection for Fault Pin 1 Bit (automatic versus manual mode)
This read/write bit allows the user to select between automatic and manual mode faults. For further descriptions of each mode, see 12.6 Fault Protection.
1 = Automatic mode
0 = Manual mode

13.3.2 Transmitter

Figure 13-5 shows the structure of the SCI transmitter.

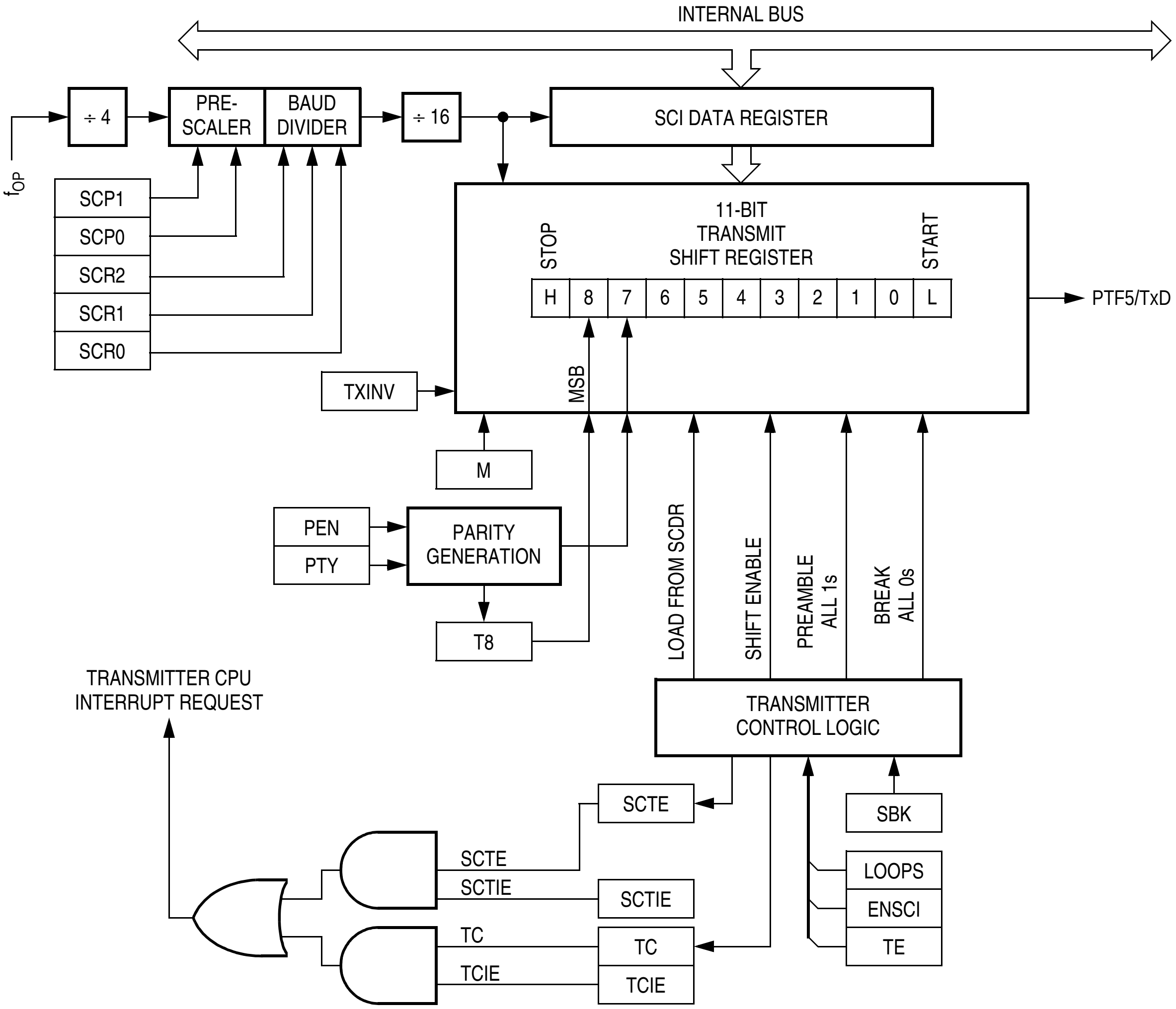


Figure 13-5. SCI Transmitter

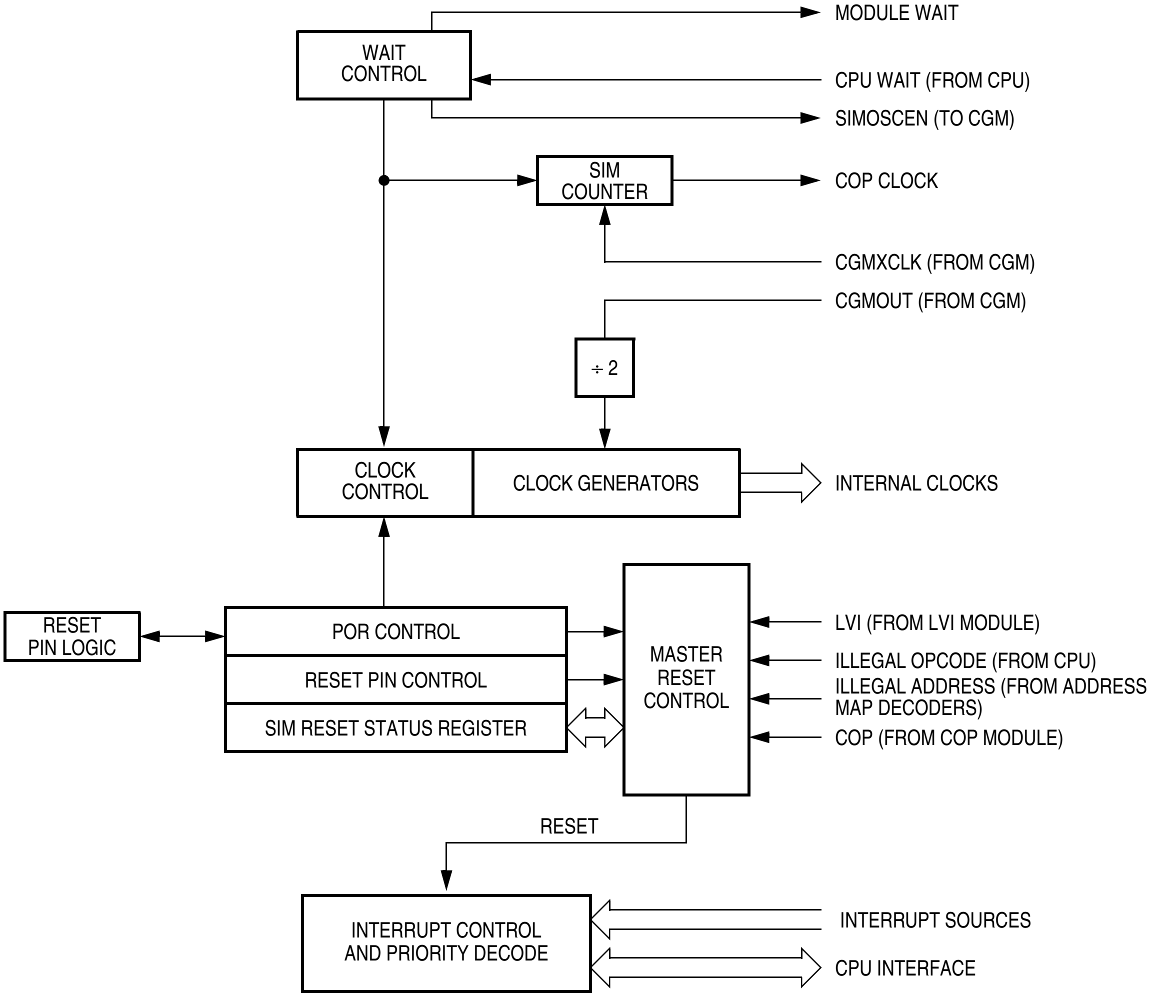


Figure 14-1. SIM Block Diagram

14.2 SIM Bus Clock Control and Generation

The bus clock generator provides system clock signals for the CPU and peripherals on the MCU. The system clocks are generated from an incoming clock, CGMOUT, as shown in Figure 14-2. This clock can come from either an external oscillator or from the on-chip phase-locked loop (PLL) circuit. See Chapter 4 Clock Generator Module (CGM).

14.2.1 Bus Timing

In user mode, the internal bus frequency is either the crystal oscillator output (CGMXCLK) divided by four or the PLL output (CGMVCLK) divided by four. See Chapter 4 Clock Generator Module (CGM).

14.2.2 Clock Startup from POR or LVI Reset

When the power-on reset (POR) module or the low-voltage inhibit (LVI) module generates a reset, the clocks to the CPU and peripherals are inactive and held in an inactive phase until after the 4096 CGMXCLK cycle POR timeout has completed. The $\overline{\text{RST}}$ pin is driven low by the SIM during this entire period. The internal bus (IBUS) clocks start upon completion of the timeout.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$0044	SPI Control Register (SPCR) See page 211.	Read:	SPRIE	R	SPMSTR	CPOL	CPHA	SPWOM	SPE	SPTIE
		Write:								
		Reset:	0	0	1	0	1	0	0	0
$0045	SPI Status and Control Register (SPSCR) See page 212.	Read:	SPRF	ERRIE	OVRF	MODF	SPTE	MODFEN	SPR1	SPR0
		Write:	R		R	R	R			
		Reset:	0	0	0	0	1	0	0	0
$0046	SPI Data Register (SPDR) See page 214.	Read:	R7	R6	R5	R4	R3	R2	R1	R0
		Write:	T7	T6	T5	T4	T3	T2	T1	T0
		Reset:	Unaffected by reset							

R = Reserved

Figure 15-3. SPI I/O Register Summary

15.4.1 Master Mode

The SPI operates in master mode when the SPI master bit, SPMSTR, is set.

NOTE

Configure the SPI modules as master or slave before enabling them. Enable the master SPI before enabling the slave SPI. Disable the slave SPI before disabling the master SPI. See 15.12.1 SPI Control Register.

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the SPI data register. If the shift register is empty, the byte immediately transfers to the shift register, setting the SPI transmitter empty bit, SPTE. The byte begins shifting out on the MOSI pin under the control of the serial clock. See Figure 15-4.

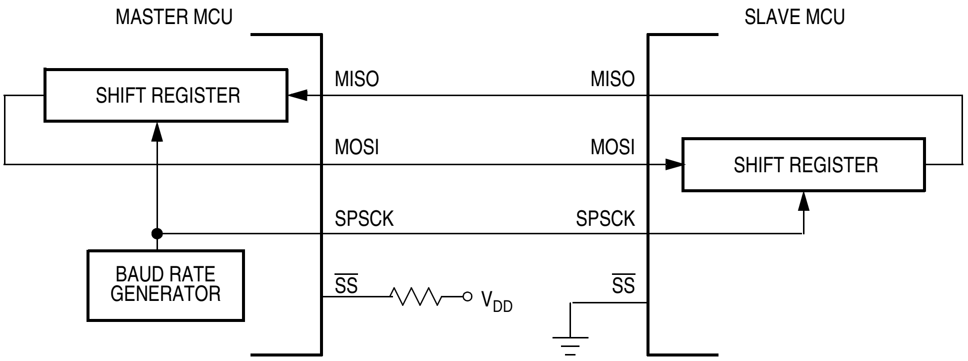


Figure 15-4. Full-Duplex Master-Slave Connections

The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. See 15.12.2 SPI Status and Control Register. Through the SPSCK pin, the baud-rate generator of the master also controls the shift register of the slave peripheral.

As the byte shifts out on the MOSI pin of the master, another byte shifts in from the slave on the master's MISO pin. The transmission ends when the receiver full bit, SPRF, becomes set. At the same time that SPRF becomes set, the byte from the slave transfers to the receive data register. In normal operation, The clock phase (CPHA) control bit selects one of two fundamentally different transmission formats. The clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

NOTE

Before writing to the CPOL bit or the CPHA bit, disable the SPI by clearing the SPI enable bit (SPE).

15.5.2 Transmission Format When CPHA = 0

Figure 15-5 shows an SPI transmission in which CPHA is logic 0. The figure should not be used as a replacement for data sheet parametric information.Two waveforms are shown for SPSCK: one for CPOL = 0 and another for CPOL = 1. The diagram may be interpreted as a master or slave timing diagram since the serial clock (SPSCK), master in/slave out (MISO), and master out/slave in (MOSI) pins are directly connected between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The slave SPI drives its MISO output only when its slave select input ($\overline{SS}$) is at logic 0, so that only the selected slave drives to the master. The $\overline{SS}$ pin of the master is not shown but is assumed to be inactive. The $\overline{SS}$ pin of the master must be high or must be reconfigured as general-purpose I/O not affecting the SPI. (See 15.6.2 Mode Fault Error.) When CPHA = 0, the first SPSCK edge is the MSB capture strobe. Therefore, the slave must begin driving its data before the first SPSCK edge, and a falling edge on the $\overline{SS}$ pin is used to start the slave data transmission. The slave's $\overline{SS}$ pin must be toggled back to high and then low again between each byte transmitted as shown in Figure 15-6.

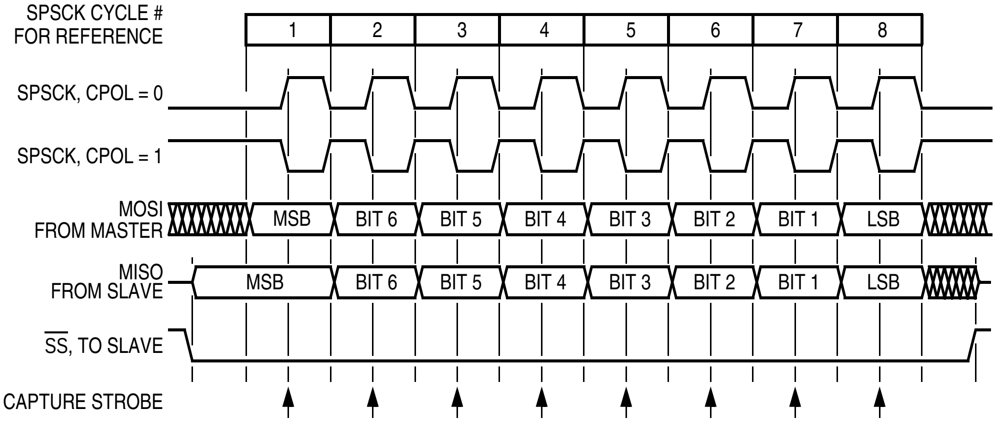


Figure 15-5. Transmission Format (CPHA = 0)

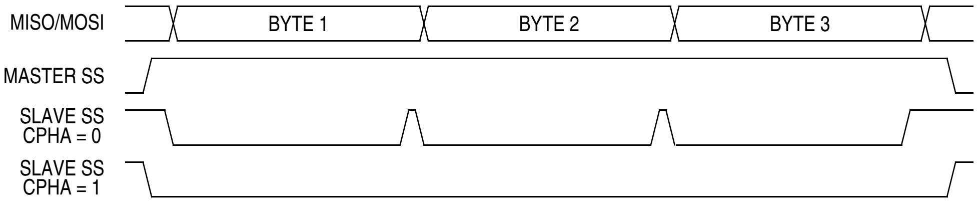


Figure 15-6. CPHA/$\overline{SS}$ Timing

15.8 Resetting the SPI

Any system reset completely resets the SPI. Partial resets occur whenever the SPI enable bit (SPE) is low. Whenever SPE is low:

- The SPTE flag is set.
- Any transmission currently in progress is aborted.
- The shift register is cleared.
- The SPI state counter is cleared, making it ready for a new complete transmission.
- All the SPI port logic is defaulted back to being general-purpose I/O.

These items are reset only by a system reset:

- All control bits in the SPCR
- All control bits in the SPSCR (MODFEN, ERRIE, SPR1, and SPR0)
- The status flags SPRF, OVRF, and MODF

By not resetting the control bits when SPE is low, the user can clear SPE between transmissions without having to set all control bits again when SPE is set back high for the next transmission.

By not resetting the SPRF, OVRF, and MODF flags, the user can still service these interrupts after the SPI has been disabled. The user can disable the SPI by writing 0 to the SPE bit. The SPI can also be disabled by a mode fault occurring in an SPI that was configured as a master with the MODFEN bit set.

15.9 Queuing Transmission Data

The double-buffered transmit data register allows a data byte to be queued and transmitted. For an SPI configured as a master, a queued data byte is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag (SPTE) indicates when the transmit data buffer is ready to accept new data. Write to the transmit data register only when the SPTE bit is high. Figure 15-12 shows the timing associated with doing back-to-back transmissions with the SPI (SPSCK has CPHA:CPOL = 1:0).

For a slave, the transmit data buffer allows back-to-back transmissions without the slave precisely timing its writes between transmissions as in a system with a single data buffer. Also, if no new data is written to the data buffer, the last value contained in the shift register is the next data word to be transmitted.

For an idle master or idle slave that has no data loaded into its transmit buffer, the SPTE is set again no more than two bus cycles after the transmit buffer empties into the shift register. This allows the user to queue up a 16-bit value to send. For an already active slave, the load of the shift register cannot occur until the transmission is completed. This implies that a back-to-back write to the transmit data register is not possible. The SPTE indicates when the next write can occur.

Table 15-4. SPI Master Baud Rate Selection

SPR1:SPR0	Baud Rate Divisor (BD)
00	2
01	8
10	32
11	128

Use this formula to calculate the SPI baud rate:

$$\text{Baud rate} = \frac{\text{CGMOUT}}{2 \times \text{BD}}$$

where:
CGMOUT = base clock output of the clock generator module (CGM)
BD = baud rate divisor

15.12.3 SPI Data Register

The SPI data register consists of the read-only receive data register and the write-only transmit data register. Writing to the SPI data register writes data into the transmit data register. Reading the SPI data register reads data from the receive data register. The transmit data and receive data registers are separate registers that can contain different values. See Figure 15-2.

Address: $0046

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	R7	R6	R5	R4	R3	R2	R1	R0
Write:	T7	T6	T5	T4	T3	T2	T1	T0
Reset:	Indeterminate after reset							

Figure 15-16. SPI Data Register (SPDR)

R7:R0/T7:T0 — Receive/Transmit Data Bits

NOTE

Do not use read-modify-write instructions on the SPI data register since the register read is not the same as the register written.

Register Name and Address: TACH2L — $001B

	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:	Indeterminate after reset							

Register Name and Address: TACH3H — $001D

	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:	Indeterminate after reset							

Register Name and Address: TACH3L — $001E

	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:	Indeterminate after reset							

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

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

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.

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}	—	$(8*V_{DDA})/$ $(f_{XCLK}*K_{ACQ})$	—	If C_F chosen correctly
Manual stable to lock time	t_{AL}	—	$(4*V_{DDA})/$ $(f_{XCLK}*K_{TRK})$	—	If C_F chosen correctly
Manual acquisition time	t_{Lock}	—	$t_{ACQ}+t_{AL}$	—	
Tracking mode entry frequency tolerance	Δ_{TRK}	0	—	± 3.6%	
Acquisition mode entry frequency tolerance	Δ_{ACQ}	±6.3%	—	± 7.2%	
Lock entry frequency tolerance	Δ_{Lock}	0	—	± 0.9%	
Lock exit frequency tolerance	Δ_{UNL}	±0.9%	—	± 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}	$(8*V_{DDA})/$ $(f_{XCLK}*K_{ACQ})$	—	If C_F chosen correctly
Automatic stable to lock time	t_{AL}	n_{TRK}/f_{XCLK}	$(4*V_{DDA})/$ $(f_{XCLK}*K_{TRK})$	—	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	—	$\pm (f_{XCLK})$ *(0.025%) *(N/4)	N = VCO freq. mult.

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	