

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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

| Product Status | Obsolete |
|----------------------------|--|
| Core Processor | HC08 |
| Core Size | 8-Bit |
| Speed | 8MHz |
| Connectivity | CANbus, SCI, SPI |
| Peripherals | LVD, POR, PWM |
| Number of I/O | 21 |
| Program Memory Size | 32KB (32K 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 ~ 105°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 32-LQFP |
| Supplier Device Package | 32-LQFP (7x7) |
| Purchase URL | https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc908gz32vfje |

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



Revision History



General Description

- Standard low-power modes of operation:
 - Wait mode
 - Stop mode
- Master reset pin and power-on reset (POR)
- On-chip FLASH memory:
 - MC68HC908GZ60 60 Kbytes
 - MC68HC908GZ48 48 Kbytes
 - MC68HC908GZ32 32 Kbytes
- Random-access memory (RAM):
 - MC68HC908GZ60 2048 bytes
 - MC68HC908GZ48 1536 bytes
 - MC68HC908GZ32 1536 bytes
- Serial peripheral interface (SPI) module
- Enhanced serial communications interface (ESCI) module
- One 16-bit, 2-channel timer interface module (TIM1) with selectable input capture, output compare, and pulse-width modulation (PWM) capability on each channel
- One 16-bit, 6-channel timer interface module (TIM2) with selectable input capture, output compare, and pulse-width modulation (PWM) capability on each channel
- Timebase module with clock prescaler circuitry for eight user selectable periodic real-time interrupts with optional active clock source during stop mode for periodic wakeup from stop using an external crystal
- 24-channel, 10-bit successive approximation analog-to-digital converter (ADC)
- 8-bit keyboard wakeup port with software selectable rising or falling edge detect, as well as high or low level detection
- Up to 53 general-purpose input/output (I/O) pins, including:
 - 40 shared-function I/O pins, depending on package choice
 - Up to 13 dedicated I/O pins, depending on package choice
- Selectable pullups on inputs only on ports A, C, and D. Selection is on an individual port bit basis. During output mode, pullups are disengaged.
- Internal pullups on IRQ and RST to reduce customer system cost
- · High current 10-mA sink/source capability on all port pins
- Higher current 20-mA sink/source capability on PTC0–PTC4 and PTF0–PTF3
- User selectable clockout feature with divide by 1, 2, and 4 of the bus or crystal frequency
- User selection of having the oscillator enabled or disabled during stop mode
- BREAK module (BRK) to allow single breakpoint setting during in-circuit debugging
- Available packages:
 - 32-pin low-profile quad flat pack (LQFP)
 - 48-pin low-profile quad flat pack (LQFP)
 - 64-pin quad flat pack (QFP)
- Specific features in 32-pin LQFP are:
 - Port A is only 4 bits: PTA0–PTA3; shared with ADC and KBI modules
 - Port B is only 6 bits: PTB0-PTB5; shared with ADC module
 - Port C is only 2 bits: PTC0-PTC1; shared with MSCAN module
 - Port D is only 7 bits: PTD0–PTD6; shared with SPI, TIM1 and TIM2 modules
 - Port E is only 2 bits: PTE0–PTE1; shared with ESCI module



Memory

2.6.7 Low-Power Modes

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

2.6.7.1 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.

2.6.7.2 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.

2.7 FLASH-2 Memory (FLASH-2)

This subsection describes the operation of the embedded FLASH-2 memory. This memory can be read, programmed, and erased from a single external supply. The program and erase operations are enabled through the use of an internal charge pump.

2.7.1 Functional Description

The FLASH-2 memory is a non-continuous array consisting of a total of 29,822 bytes. An erased bit reads as a 1 and a programmed bit reads as a 0.

Memory in the FLASH-2 array is organized into rows within pages. There are two rows of memory per page with 64 bytes per row. The minimum erase block size is a single page,128 bytes. Programming is performed on a per-row basis, 64 bytes at a time. Program and erase operations are facilitated through control bits in the FLASH-2 control register (FL2CR). Details for these operations appear later in this subsection.

The FLASH-2 memory map consists of:

- \$0462–\$04FF: user memory (158 bytes)
- \$0980-\$1B7F: user memory (4608 bytes)
- \$1E20-\$7FFF: user memory (25056 bytes)
- \$FF81: FLASH-2 block protect register (FL2BPR)

NOTE

FL2BPR physically resides within FLASH-1 memory addressing space

\$FE08: FLASH-2 control register (FL2CR)



Memory

2.7.2.2 FLASH-2 Block Protect Register

The FLASH-2 block protect register (FL2BPR) is implemented as a byte within the FLASH-1 memory; therefore, can only be written during a FLASH-1 programming sequence. The value in this register determines the starting location of the protected range within the FLASH-2 memory.



Figure 2-8. FLASH-2 Block Protect Register (FL2BPR)

NOTE The FLASH-2 block protect register (FL2BPR) controls the block protection for the FLASH-2 array. However, FL2BPR is implemented within the FLASH-1 memory array and therefore, the FLASH-1 control register (FL1CR) must be used to program/erase FL2BPR.

FL2BPR[7:0] — Block Protect Register Bits 7 to 0

These eight bits represent bits [14:7] of a 16-bit memory address. Bit 15 is a 0 and bits [6:0] are 0s.

The resultant 16-bit address is used for specifying the start address of the FLASH-2 memory for block protection. FLASH-2 is protected from this start address to the end of FLASH-2 memory at \$7FFF. With this mechanism, the protect start address can be \$XX00 and \$XX80 (128 byte page boundaries) within the FLASH-2 array.



Figure 2-9. FLASH-2 Block Protect Start Address

| FL2BPR[7:0] | Protected Range |
|-------------|--------------------|
| \$FF | No Protection |
| \$FE | \$7F00-\$7FFF |
| \$FD ↓ | \$7E80–\$7FFF ↓ |
| \$0B | \$0580-\$7FFF |
| \$0A | \$0500-\$7FFF |
| \$09 | \$0480-\$7FFF |
| \$08 ↓ | \$0462–\$7FFF ↓ |
| \$04 | \$0462\$7FFF |
| \$03 | \$0462\$7FFF |
| \$02 | \$0462\$7FFF |
| \$01 | \$0462\$7FFF |
| \$00 | \$0462\$7FFF |

Table 2-3. FLASH-2 Protected Ranges



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 the 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. Bit Truncation Mode Error

3.4 Monotonicity

The conversion process is monotonic and has no missing codes.

3.5 Interrupts

When the AIEN bit is set, the ADC module is capable of generating CPU interrupts after each ADC conversion. A CPU interrupt is generated if the COCO bit is a 0. The COCO bit is not used as a conversion complete flag when interrupts are enabled.



Analog-to-Digital Converter (ADC)

3.8.3 ADC Clock Register

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



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.

| 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 |

Table 3-2. ADC Clock Divide Ratio

1. X = Don't care

ADICLK — ADC Input Clock Select Bit

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

1 = Internal bus clock

0 = Oscillator output clock (CGMXCLK)

The ADC requires a clock rate of approximately 1 MHz for correct operation. If the selected clock source is not fast enough, the ADC will generate incorrect conversions. See 21.10 5.0-Volt ADC Characteristics.

$$f_{ADIC} = \frac{f_{CGMXCLK} \text{ or bus frequency}}{ADIV[2:0]} \cong 1 \text{ MHz}$$

MODE1 and MODE0 — Modes of Result Justification Bits

MODE1 and MODE0 select 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 signed data mode



Clock Generator Module (CGM)

frequency, f_{RCLK}. The circuit determines the mode of the PLL and the lock condition based on this comparison.

4.3.4 Acquisition and Tracking Modes

The PLL filter is manually or automatically configurable into one of two operating modes:

- Acquisition mode In acquisition mode, the filter can make large frequency corrections to the VCO. This mode is used at PLL start up or when the PLL has suffered a severe noise hit and the VCO frequency is far off the desired frequency. When in acquisition mode, the ACQ bit is clear in the PLL bandwidth control register. (See 4.5.2 PLL Bandwidth Control Register.)
- Tracking mode In tracking mode, the filter makes only small corrections to the frequency of the VCO. PLL jitter is much lower in tracking mode, but the response to noise is also slower. The PLL enters tracking mode when the VCO frequency is nearly correct, such as when the PLL is selected as the base clock source. (See 4.3.8 Base Clock Selector Circuit.) The PLL is automatically in tracking mode when not in acquisition mode or when the ACQ bit is set.

4.3.5 Manual and Automatic PLL Bandwidth Modes

The PLL can change the bandwidth or operational mode of the loop filter manually or automatically. Automatic mode is recommended for most users.

In automatic bandwidth control mode (AUTO = 1), the lock detector automatically switches between acquisition and tracking modes. Automatic bandwidth control mode also is used to determine when the VCO clock, CGMVCLK, is safe to use as the source for the base clock, CGMOUT. (See 4.5.2 PLL Bandwidth Control Register.) If PLL interrupts are enabled, the software can wait for a PLL interrupt request and then check the LOCK bit. If interrupts are disabled, software can poll the LOCK bit continuously (for example, during PLL start up) or at periodic intervals. In either case, when the LOCK bit is set, the VCO clock is safe to use as the source for the base clock. (See 4.3.8 Base Clock Selector Circuit.) If the VCO is selected as the source for the base clock and the LOCK bit is clear, the PLL has suffered a severe noise hit and the software must take appropriate action, depending on the application. (See 4.6 Interrupts for information and precautions on using interrupts.)

The following conditions apply when the PLL is in automatic bandwidth control mode:

- The ACQ bit (See 4.5.2 PLL Bandwidth Control Register.) is a read-only indicator of the mode of the filter. (See 4.3.4 Acquisition and Tracking Modes.)
- The ACQ bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See 4.8 Acquisition/Lock Time Specifications for more information.)
- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within a certain tolerance and is cleared when the VCO frequency is out of a certain tolerance. (See 4.8 Acquisition/Lock Time Specifications for more information.)
- CPU interrupts can occur if enabled (PLLIE = 1) when the PLL's lock condition changes, toggling the LOCK bit. (See 4.5.1 PLL Control Register.)

The PLL also may operate in manual mode (AUTO = 0). Manual mode is used by systems that do not require an indicator of the lock condition for proper operation. Such systems typically operate well below f_{BUSMAX} .



CGM Registers

| Addr. | Register Name | | Bit 7 | 6 | 5 | 4 | 3 | 2 | 1 | Bit 0 |
|--------|--|-----------------|-------|-------------|-------|------|------------|------|------|-------|
| \$0039 | PLL Multiplier Select Low Register (PMSL) | Read: Write: | MUL7 | MUL6 | MUL5 | MUL4 | MUL3 | MUL2 | MUL1 | MUL0 |
| | See page 86. | Reset: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| \$003A | PLL VCO Select Range Register (PMRS) | Read: Write: | VRS7 | VRS6 | VRS5 | VRS4 | VRS3 | VRS2 | VRS1 | VRS0 |
| | See page 87. | Reset: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Read: | 0 | 0 | 0 | 0 | Р | Р | Б | Р |
| \$003B | Reserved Register | Write: | | | | | n | n | n | 11 |
| | | Reset: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | [| | = Unimpleme | ented | R | = Reserved | | | |

NOTES:

- 1. When AUTO = 0, PLLIE is forced clear and is read-only.
- 2. When AUTO = 0, PLLF and LOCK read as clear.
- 3. When AUTO = 1, \overline{ACQ} is read-only.

4. When PLLON = 0 or VRS7:VRS0 = \$0. BCS is forced clear and is read-only.

- 5. When PLLON = 1, the PLL programming register is read-only.
- 6. When BCS = 1, PLLON is forced set and is read-only.

Figure 4-3. CGM I/O Register Summary (Continued)

4.5.1 PLL Control Register

The PLL control register (PCTL) contains the interrupt enable and flag bits, the on/off switch, the base clock selector bit, and the VCO power-of-two range selector bits.



Figure 4-4. PLL Control Register (PCTL)

PLLIE — PLL Interrupt Enable Bit

This read/write bit enables the PLL to generate an interrupt request when the LOCK bit toggles, setting the PLL flag, PLLF. When the AUTO bit in the PLL bandwidth control register (PBWC) is clear, PLLIE cannot be written and reads as 0. Reset clears the PLLIE bit.

1 = PLL interrupts enabled

0 = PLL interrupts disabled

PLLF — PLL Interrupt Flag Bit

This read-only bit is set whenever the LOCK bit toggles. PLLF generates an interrupt request if the PLLIE bit also is set. PLLF always reads as 0 when the AUTO bit in the PLL bandwidth control register (PBWC) is clear. Clear the PLLF bit by reading the PLL control register. Reset clears the PLLF bit.

1 = Change in lock condition

0 = No change in lock condition

Keyboard Interrupt Module (KBI)



1. Pin contains integrated pullup device.

2. Ports are software configurable with pullup device if input port or pullup/pulldown device for keyboard input.

3. Higher current drive port pins

Figure 9-1. Block Diagram Highlighting KBI Block and Pins



Keyboard Interrupt Module (KBI)



Chapter 10 Low-Power Modes

10.1 Introduction

The microcontroller (MCU) may enter two low-power modes: wait mode and stop mode. They are common to all HC08 MCUs and are entered through instruction execution. This section describes how each module acts in the low-power modes.

10.1.1 Wait Mode

The WAIT instruction puts the MCU in a low-power standby mode in which the central processor unit (CPU) clock is disabled but the bus clock continues to run. Power consumption can be further reduced by disabling the low-voltage inhibit (LVI) module through bits in the CONFIG1 register. See Chapter 5 Configuration Register (CONFIG).

10.1.2 Stop Mode

Stop mode is entered when a STOP instruction is executed. The CPU clock is disabled and the bus clock is disabled if the OSCENINSTOP bit in the CONFIG2 register is a 0. See Chapter 5 Configuration Register (CONFIG).

10.2 Analog-to-Digital Converter (ADC)

10.2.1 Wait Mode

The analog-to-digital converter (ADC) continues normal operation during wait mode. Any enabled CPU interrupt request from the ADC can bring the MCU out of wait mode. If the ADC is not required to bring the MCU out of wait mode, power down the ADC by setting ADCH4–ADCH0 bits in the ADC status and control register before executing the WAIT instruction.

10.2.2 Stop Mode

The ADC module is inactive after the execution of a STOP instruction. Any pending conversion is aborted. ADC conversions resume when the MCU exits stop mode after an external interrupt. Allow one conversion cycle to stabilize the analog circuitry.





12.3 External Pins

The MSCAN08 uses two external pins, one input (CAN_{RX}) and one output (CAN_{TX}) . The CAN_{TX} output pin represents the logic level on the CAN: 0 is for a dominant state, and 1 is for a recessive state.

A typical CAN system with MSCAN08 is shown in Figure 12-2.



Figure 12-2. The CAN System

Each CAN station is connected physically to the CAN bus lines through a transceiver chip. The transceiver is capable of driving the large current needed for the CAN and has current protection against defected CAN or defected stations.

12.4 Message Storage

MSCAN08 facilitates a sophisticated message storage system which addresses the requirements of a broad range of network applications.

12.4.1 Background

Modern application layer software is built under two fundamental assumptions:

- 1. Any CAN node is able to send out a stream of scheduled messages without releasing the bus between two messages. Such nodes will arbitrate for the bus right after sending the previous message and will only release the bus in case of lost arbitration.
- 2. The internal message queue within any CAN node is organized as such that the highest priority message will be sent out first if more than one message is ready to be sent.

Above behavior cannot be achieved with a single transmit buffer. That buffer must be reloaded right after the previous message has been sent. This loading process lasts a definite amount of time and has to be



NOTE

The CBTR0 register can be written only if the SFTRES bit in the MSCAN08 module control register is set.

12.13.4 MSCAN08 Bus Timing Register 1



Figure 12-19. Bus Timing Register 1 (CBTR1)

SAMP — Sampling

This bit determines the number of serial bus samples to be taken per bit time. If set, three samples per bit are taken, the regular one (sample point) and two preceding samples, using a majority rule. For higher bit rates, SAMP should be cleared, which means that only one sample will be taken per bit.

- 1 = Three samples per $bit^{(1)}$
- 0 = One sample per bit

TSEG22–TSEG10 — Time Segment

Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point. Time segment 1 (TSEG1) and time segment 2 (TSEG2) are programmable as shown in Table 12-8. The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (T_q) clock cycles per bit as shown in Table 12-4).

Bit time = $\frac{\text{Pres value}}{f_{\text{MSCANCLK}}}$ • number of time quanta

NOTE

The CBTR1 register can only be written if the SFTRES bit in the MSCAN08 module control register is set.

| TSEG13 | TSEG12 | TSEG11 | TSEG10 | Time Segment 1 |
|--------|--------|--------|--------|--|
| 0 | 0 | 0 | 0 | 1 T _q Cycle ⁽¹⁾ |
| 0 | 0 | 0 | 1 | 2 T _q Cycles ⁽¹⁾ |
| 0 | 0 | 1 | 0 | 3T _q Cycles ⁽¹⁾ |
| 0 | 0 | 1 | 1 | 4 T _q Cycles |
| | | | | |
| | | | | |
| 1 | 1 | 1 | 1 | 16 T _q Cycles |

| TSEG22 | TSEG21 | TSEG20 | Time Segment 2 |
|--------|--------|--------|---------------------------------------|
| 0 | 0 | 0 | 1 T _q Cycle ⁽¹⁾ |
| 0 | 0 | 1 | 2 T _q Cycles |
| | | | |
| | | | |
| 1 | 1 | 1 | 8T _q Cycles |

| Table 12 | 2-8. Time | Segment | Va | ues |
|----------|-----------|---------|----|-----|
|----------|-----------|---------|----|-----|

1. This setting is not valid. Please refer to Table 12-4 for valid settings.

1. In this case PHASE_SEG1 must be at least 2 time quanta.



PTDPUE7–PTDPUE0 — Port D Input Pullup Enable Bits

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

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

13.7 Port E

Port E is a 6-bit special-function port that shares two of its pins with the enhanced serial communications interface (ESCI) module.

13.7.1 Port E Data Register

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



Figure 13-17. Port E Data Register (PTE)

PTE5-PTE0 — Port E Data Bits

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

NOTE

Data direction register E (DDRE) does not affect the data direction of port E pins that are being used by the ESCI module. However, the DDRE bits always determine whether reading port E returns the states of the latches or the states of the pins. See Table 13-6.

RxD — SCI Receive Data Input

The PTE1/RxD pin is the receive data input for the ESCI module.

When the enable SCI bit, ENSCI, is clear, the ESCI module is disabled, and the PTE1/RxD pin is available for general-purpose I/O. See Chapter 14 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 14 Enhanced Serial Communications Interface (ESCI) Module.



Serial Peripheral Interface (SPI) Module

16.3.1 Master Mode

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

NOTE

In a multi-SPI system, 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 16.12.1 SPI Control Register.

Only a master SPI module can initiate transmissions. Software begins the transmission from a master SPI module by writing to the transmit 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 16-4.



Figure 16-4. Full-Duplex Master-Slave Connections

The SPR1 and SPR0 bits control the baud rate generator and determine the speed of the shift register. (See 16.12.2 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, SPRF signals the end of a transmission. Software clears SPRF by reading the SPI status and control register with SPRF set and then reading the SPI data register. Writing to the SPI data register (SPDR) clears SPTE.

16.3.2 Slave Mode

The SPI operates in slave mode when SPMSTR is clear. In slave mode, the SPSCK pin is the input for the serial clock from the master MCU. Before a data transmission occurs, the \overline{SS} pin of the slave SPI must be low. \overline{SS} must remain low until the transmission is complete. See 16.6.2 Mode Fault Error.

In a slave SPI module, data enters the shift register under the control of the serial clock from the master SPI module. After a byte enters the shift register of a slave SPI, it transfers to the receive data register, and the SPRF bit is set. To prevent an overflow condition, slave software then must read the receive data register before another full byte enters the shift register.



Serial Peripheral Interface (SPI) Module

input (\overline{SS}) is low, 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 16.6.2 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 16-6.

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 transmit data register. Therefore, the SPI data register of the slave must be loaded with transmit data before the falling edge of \overline{SS} . Any data written after the falling edge is stored in the transmit data register and transferred to the shift register after the current transmission.





16.4.3 Transmission Format When CPHA = 1

Figure 16-7 shows an SPI transmission in which CPHA = 1. 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 low, 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}



Chapter 17 Timebase Module (TBM)

17.1 Introduction

This section describes the timebase module (TBM). The TBM will generate periodic interrupts at user selectable rates using a counter clocked by the external clock source. This TBM version uses 15 divider stages, eight of which are user selectable. A configuration option bit to select an additional 128 divide of the external clock source can be selected. See Chapter 5 Configuration Register (CONFIG)

17.2 Features

Features of the TBM module include:

- External clock or an additional divide-by-128 selected by configuration option bit as clock source
- Software configurable periodic interrupts with divide-by: 8, 16, 32, 64, 128, 2048, 8192, and 32768 taps of the selected clock source
- Configurable for operation during stop mode to allow periodic wakeup from stop

17.3 Functional Description

This module can generate a periodic interrupt by dividing the clock source supplied from the clock generator module, CGMXCLK.

The counter is initialized to all 0s when TBON bit is cleared. The counter, shown in Figure 17-1, starts counting when the TBON bit is set. When the counter overflows at the tap selected by TBR2–TBR0, the TBIF bit gets set. If the TBIE bit is set, an interrupt request is sent to the CPU. The TBIF flag is cleared by writing a 1 to the TACK bit. The first time the TBIF flag is set after enabling the timebase module, the interrupt is generated at approximately half of the overflow period. Subsequent events occur at the exact period.

The timebase module may remain active after execution of the STOP instruction if the crystal oscillator has been enabled to operate during stop mode through the OSCENINSTOP bit in the configuration register. The timebase module can be used in this mode to generate a periodic wakeup from stop mode.

17.4 Interrupts

The timebase module can periodically interrupt the CPU with a rate defined by the selected TBMCLK and the select bits TBR2–TBR0. When the timebase counter chain rolls over, the TBIF flag is set. If the TBIE bit is set, enabling the timebase interrupt, the counter chain overflow will generate a CPU interrupt request.

NOTE

Interrupts must be acknowledged by writing a 1 to the TACK bit.



Timer Interface Module (TIM1)



Figure 18-4. PWM Period and Pulse Width

18.3.4.1 Unbuffered PWM Signal Generation

Any output compare channel can generate unbuffered PWM pulses as described in 18.3.4 Pulse Width Modulation (PWM). The pulses are unbuffered because changing the pulse width requires writing the new pulse width value over the old value currently in the TIM1 channel registers.

An unsynchronized write to the TIM1 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 TIM1 overflow interrupt routine to write a new, smaller pulse width value may cause the compare to be missed. The TIM1 may pass the new value before it is written to the timer channel (T1CHxH:T1CHxL) registers.

Use the following methods to synchronize unbuffered changes in the PWM pulse width on channel x:

- When changing to a shorter pulse width, enable channel x output compare interrupts and write the new value in the output compare interrupt routine. The output compare interrupt occurs at the end of the current pulse. The interrupt routine has until the end of the PWM period to write the new value.
- When changing to a longer pulse width, enable TIM1 overflow interrupts and write the new value in the TIM1 overflow interrupt routine. The TIM1 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% 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.

Timer Interface Module (TIM1)

TSTOP — TIM1 Stop Bit

This read/write bit stops the TIM1 counter. Counting resumes when TSTOP is cleared. Reset sets the TSTOP bit, stopping the TIM1 counter until software clears the TSTOP bit.

1 = TIM1 counter stopped

0 = TIM1 counter active

NOTE

Do not set the TSTOP bit before entering wait mode if the TIM1 is required to exit wait mode. Also, when the TSTOP bit is set and the timer is configured for input capture operation, input captures are inhibited until the TSTOP bit is cleared.

TRST — TIM1 Reset Bit

Setting this write-only bit resets the TIM1 counter and the TIM1 prescaler. Setting TRST has no effect on any other registers. Counting resumes from \$0000. TRST is cleared automatically after the TIM1 counter is reset and always reads as 0. Reset clears the TRST bit.

1 = Prescaler and TIM1 counter cleared

0 = No effect

NOTE

Setting the TSTOP and TRST bits simultaneously stops the TIM1 counter at a value of \$0000.

PS[2:0] — Prescaler Select Bits

These read/write bits select one of the seven prescaler outputs as the input to the TIM1 counter as Table 18-1 shows. Reset clears the PS[2:0] bits.

| PS2 | PS1 | PS0 | TIM1 Clock Source |
|-----|-----|-----|-------------------------|
| 0 | 0 | 0 | Internal bus clock ÷ 1 |
| 0 | 0 | 1 | Internal bus clock ÷ 2 |
| 0 | 1 | 0 | Internal bus clock ÷ 4 |
| 0 | 1 | 1 | Internal bus clock ÷ 8 |
| 1 | 0 | 0 | Internal bus clock ÷ 16 |
| 1 | 0 | 1 | Internal bus clock ÷ 32 |
| 1 | 1 | 0 | Internal bus clock ÷ 64 |
| 1 | 1 | 1 | Not available |

Table 18-1. Prescaler Selection

