

Welcome to [E-XFL.COM](http://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	S08
Core Size	8-Bit
Speed	40MHz
Connectivity	CANbus, I <sup>2</sup> C, LINbus, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	39
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	External
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz60f2mlf">https://www.e-xfl.com/product-detail/nxp-semiconductors/s9s08dz60f2mlf</a>

# Contents

Section Number	Title	Page
<b>Chapter 1</b>		
<b>Device Overview</b>		
1.1	Devices in the MC9S08DZ60 Series.....	21
1.2	MCU Block Diagram .....	22
1.3	System Clock Distribution .....	24
<b>Chapter 2</b>		
<b>Pins and Connections</b>		
2.1	Device Pin Assignment .....	27
2.2	Recommended System Connections .....	30
2.2.1	Power .....	31
2.2.2	Oscillator .....	31
2.2.3	RESET .....	31
2.2.4	Background / Mode Select (BKGD/MS) .....	32
2.2.5	ADC Reference Pins ( $V_{REFH}$ , $V_{REFL}$ ) .....	32
2.2.6	General-Purpose I/O and Peripheral Ports .....	32
<b>Chapter 3</b>		
<b>Modes of Operation</b>		
3.1	Introduction .....	35
3.2	Features .....	35
3.3	Run Mode.....	35
3.4	Active Background Mode.....	35
3.5	Wait Mode .....	36
3.6	Stop Modes.....	37
3.6.1	Stop3 Mode .....	37
3.6.2	Stop2 Mode .....	38
3.6.3	On-Chip Peripheral Modules in Stop Modes .....	39
<b>Chapter 4</b>		
<b>Memory</b>		
4.1	MC9S08DZ60 Series Memory Map .....	41
4.2	Reset and Interrupt Vector Assignments .....	42
4.3	Register Addresses and Bit Assignments.....	44
4.4	RAM.....	52
4.5	Flash and EEPROM .....	52
4.5.1	Features .....	52

which can be performed at the same time the Flash memory is programmed. The 1:0 state disengages security; the other three combinations engage security. Notice the erased state (1:1) makes the MCU secure. During development, whenever the Flash is erased, it is good practice to immediately program the SEC0 bit to 0 in NVOPT so SEC = 1:0. This would allow the MCU to remain unsecured after a subsequent reset.

The on-chip debug module cannot be enabled while the MCU is secure. The separate background debug controller can be used for background memory access commands, but the MCU cannot enter active background mode except by holding BKGD low at the rising edge of reset.

A user can choose to allow or disallow a security unlocking mechanism through an 8-byte backdoor security key. If the nonvolatile KEYEN bit in NVOPT/FOPT is 0, the backdoor key is disabled and there is no way to disengage security without completely erasing all Flash locations. If KEYEN is 1, a secure user program can temporarily disengage security by:

1. Writing 1 to KEYACC in the FCNFG register. This makes the Flash module interpret writes to the backdoor comparison key locations (NVBACKKEY through NVBACKKEY+7) as values to be compared against the key rather than as the first step in a Flash program or erase command.
2. Writing the user-entered key values to the NVBACKKEY through NVBACKKEY+7 locations. These writes must be performed in order starting with the value for NVBACKKEY and ending with NVBACKKEY+7. STHX must not be used for these writes because these writes cannot be performed on adjacent bus cycles. User software normally would get the key codes from outside the MCU system through a communication interface such as a serial I/O.
3. Writing 0 to KEYACC in the FCNFG register. If the 8-byte key that was written matches the key stored in the Flash locations, SEC bits are automatically changed to 1:0 and security will be disengaged until the next reset.

The security key can be written only from secure memory (either RAM, EEPROM, or Flash), so it cannot be entered through background commands without the cooperation of a secure user program.

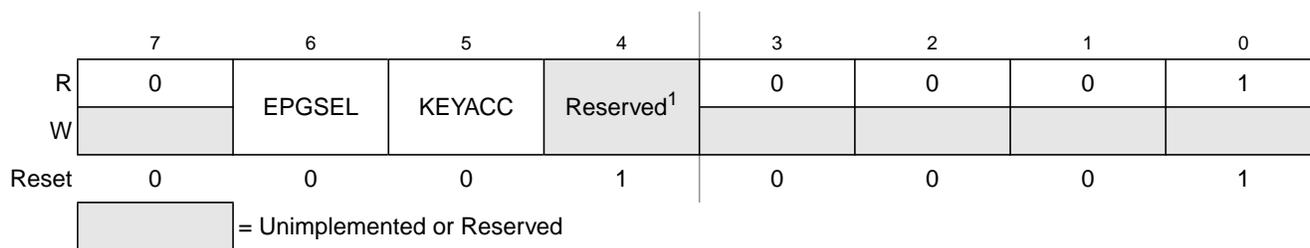
The backdoor comparison key (NVBACKKEY through NVBACKKEY+7) is located in Flash memory locations in the nonvolatile register space so users can program these locations exactly as they would program any other Flash memory location. The nonvolatile registers are in the same 768-byte block of Flash as the reset and interrupt vectors, so block protecting that space also block protects the backdoor comparison key. Block protects cannot be changed from user application programs, so if the vector space is block protected, the backdoor security key mechanism cannot permanently change the block protect, security settings, or the backdoor key.

Security can always be disengaged through the background debug interface by taking these steps:

1. Disable any block protections by writing FPROT. FPROT can be written only with background debug commands, not from application software.
2. Mass erase Flash if necessary.
3. Blank check Flash. Provided Flash is completely erased, security is disengaged until the next reset.

To avoid returning to secure mode after the next reset, program NVOPT so SEC = 1:0.

### 4.5.11.3 Flash and EEPROM Configuration Register (FCNFG)



**Figure 4-7. Flash Configuration Register (FCNFG)**

<sup>1</sup> User must write a 1 to this bit. Failing to do so may result in unexpected behavior.

**Table 4-11. FCNFG Register Field Descriptions**

Field	Description
6 EPGSEL	<b>EEPROM Page Select</b> — This bit selects which EEPROM page is accessed in the memory map. 0 Page 0 is in foreground of memory map. Page 1 is in background and can not be accessed. 1 Page 1 is in foreground of memory map. Page 0 is in background and can not be accessed.
5 KEYACC	<b>Enable Writing of Access Key</b> — This bit enables writing of the backdoor comparison key. For more detailed information about the backdoor key mechanism, refer to <a href="#">Section 4.5.9, “Security.”</a> 0 Writes to 0xFFB0–0xFFB7 are interpreted as the start of a Flash programming or erase command. 1 Writes to NVBACKKEY (0xFFB0–0xFFB7) are interpreted as comparison key writes.

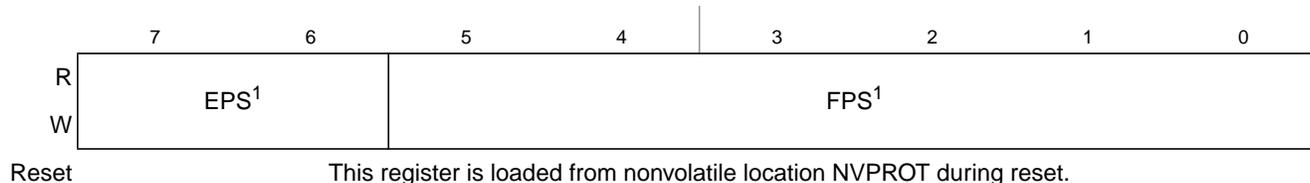
### 4.5.11.4 Flash and EEPROM Protection Register (FPROT and NVPROT)

The FPROT register defines which Flash and EEPROM sectors are protected against program and erase operations.

During the reset sequence, the FPROT register is loaded from the nonvolatile location NVPROT. To change the protection that will be loaded during the reset sequence, the sector containing NVPROT must be unprotected and erased, then NVPROT can be reprogrammed.

FPROT bits are readable at any time and writable as long as the size of the protected region is being increased. Any write to FPROT that attempts to decrease the size of the protected memory will be ignored.

Trying to alter data in any protected area will result in a protection violation error and the FPVIOL flag will be set in the FSTAT register. Mass erase is not possible if any one of the sectors is protected.



<sup>1</sup> Background commands can be used to change the contents of these bits in FPROT.

**Figure 4-8. Flash and EEPROM Protection Register (FPROT)**

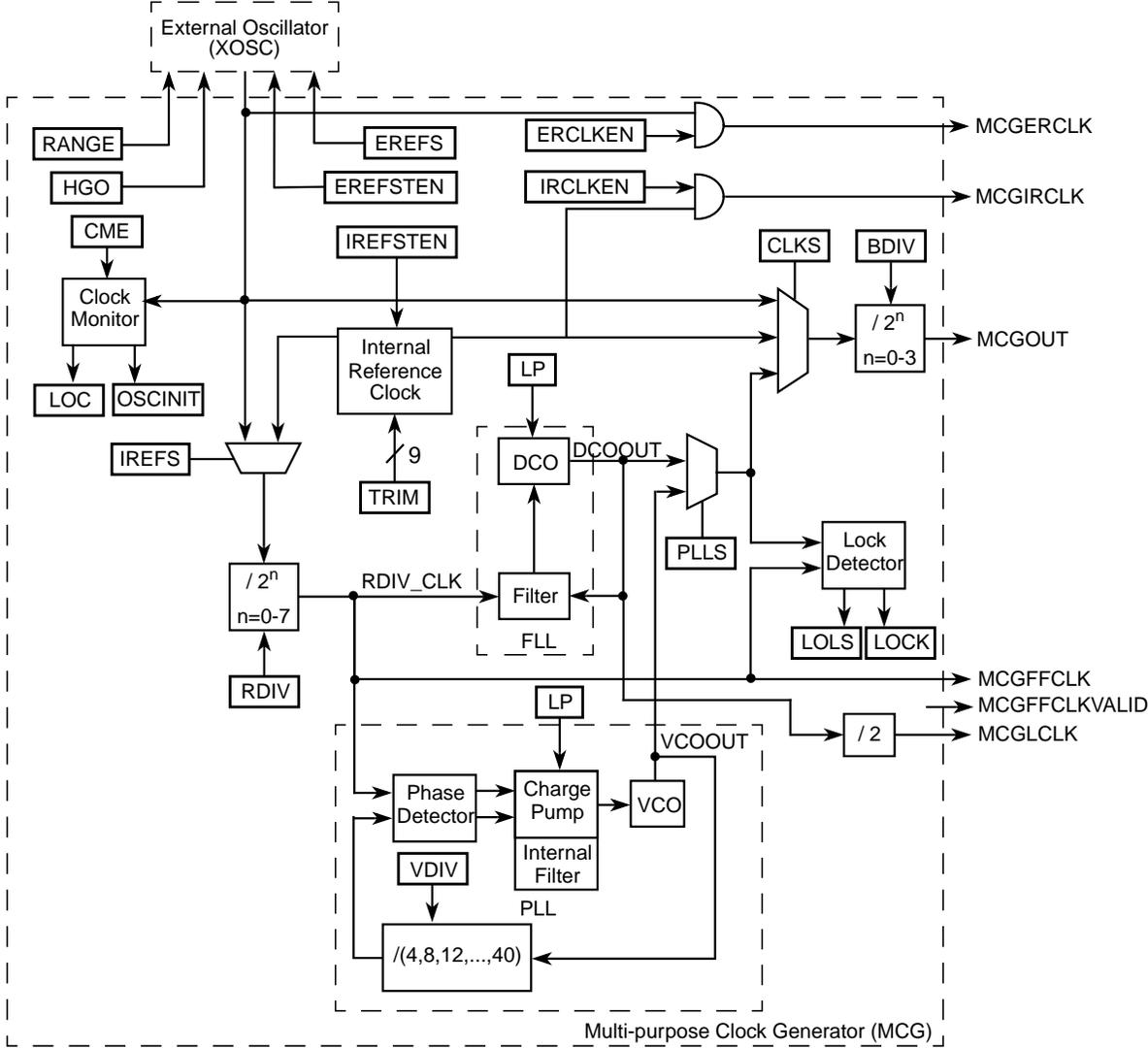


Figure 8-2. Multi-Purpose Clock Generator (MCG) Block Diagram

The nine states of the MCG are shown as a state diagram and are described below. The arrows indicate the allowed movements between the states.

#### 8.4.1.1 FLL Engaged Internal (FEI)

FLL engaged internal (FEI) is the default mode of operation and is entered when all the following conditions occur:

- CLKS bits are written to 00
- IREFS bit is written to 1
- PLLS bit is written to 0
- RDIV bits are written to 000. Since the internal reference clock frequency should already be in the range of 31.25 kHz to 39.0625 kHz after it is trimmed, no further frequency divide is necessary.

In FLL engaged internal mode, the MCGOUT clock is derived from the FLL clock, which is controlled by the internal reference clock. The FLL clock frequency locks to 1024 times the reference frequency, as selected by the RDIV bits. The MCGLCLK is derived from the FLL and the PLL is disabled in a low power state.

#### 8.4.1.2 FLL Engaged External (FEE)

The FLL engaged external (FEE) mode is entered when all the following conditions occur:

- CLKS bits are written to 00
- IREFS bit is written to 0
- PLLS bit is written to 0
- RDIV bits are written to divide reference clock to be within the range of 31.25 kHz to 39.0625 kHz

In FLL engaged external mode, the MCGOUT clock is derived from the FLL clock which is controlled by the external reference clock. The external reference clock which is enabled can be an external crystal/resonator or it can be another external clock source. The FLL clock frequency locks to 1024 times the reference frequency, as selected by the RDIV bits. The MCGLCLK is derived from the FLL and the PLL is disabled in a low power state.

#### 8.4.1.3 FLL Bypassed Internal (FBI)

In FLL bypassed internal (FBI) mode, the MCGOUT clock is derived from the internal reference clock and the FLL is operational but its output clock is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUT clock is driven from the internal reference clock.

The FLL bypassed internal mode is entered when all the following conditions occur:

- CLKS bits are written to 01
- IREFS bit is written to 1
- PLLS bit is written to 0
- RDIV bits are written to 000. Since the internal reference clock frequency should already be in the range of 31.25 kHz to 39.0625 kHz after it is trimmed, no further frequency divide is necessary.

The table below shows MCGOUT frequency calculations using RDIV, BDIV, and VDIV settings for each clock mode. The bus frequency is equal to MCGOUT divided by 2.

**Table 8-6. MCGOUT Frequency Calculation Options**

Clock Mode	$f_{MCGOUT}^1$	Note
FEI (FLL engaged internal)	$(f_{int} * 1024) / B$	Typical $f_{MCGOUT} = 16$ MHz immediately after reset. RDIV bits set to %000.
FEE (FLL engaged external)	$(f_{ext} / R * 1024) / B$	$f_{ext} / R$ must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	$f_{ext} / B$	$f_{ext} / R$ must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	$f_{int} / B$	Typical $f_{int} = 32$ kHz
PEE (PLL engaged external)	$[(f_{ext} / R) * M] / B$	$f_{ext} / R$ must be in the range of 1 MHz to 2 MHz
PBE (PLL bypassed external)	$f_{ext} / B$	$f_{ext} / R$ must be in the range of 1 MHz to 2 MHz
BLPI (Bypassed low power internal)	$f_{int} / B$	
BLPE (Bypassed low power external)	$f_{ext} / B$	

<sup>1</sup> R is the reference divider selected by the RDIV bits, B is the bus frequency divider selected by the BDIV bits, and M is the multiplier selected by the VDIV bits.

This section will include 3 mode switching examples using a 4 MHz external crystal. If using an external clock source less than 1 MHz, the MCG should not be configured for any of the PLL modes (PEE and PBE).

### 8.5.2.1 Example # 1: Moving from FEI to PEE Mode: External Crystal = 4 MHz, Bus Frequency = 8 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE mode until the 4 MHz crystal reference frequency is set to achieve a bus frequency of 8 MHz. Because the MCG is in FEI mode out of reset, this example also shows how to initialize the MCG for PEE mode out of reset. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, FEI must transition to FBE mode:
  - a) MCGC2 = 0x36 (%00110110)
    - BDIV (bits 7 and 6) set to %00, or divide-by-1
    - RANGE (bit 5) set to 1 because the frequency of 4 MHz is within the high frequency range
    - HGO (bit 4) set to 1 to configure external oscillator for high gain operation
    - EREFS (bit 2) set to 1, because a crystal is being used
    - ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active
  - b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.

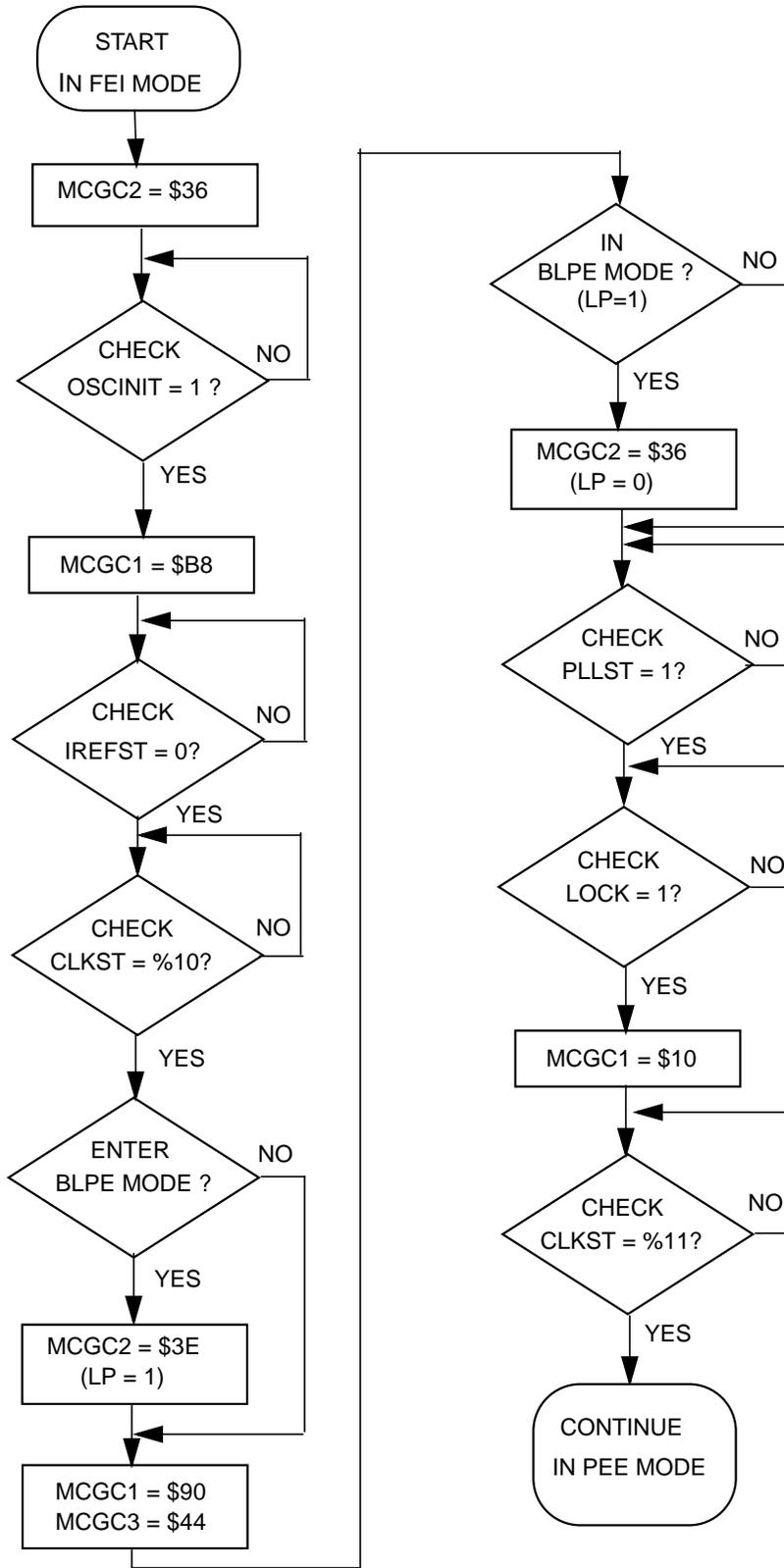


Figure 8-9. Flowchart of FEI to PEE Mode Transition using a 4 MHz crystal

external crystal and a maximum reference divider factor of 128, the resulting frequency of the reference clock for the FLL is 62.5 kHz (greater than the 39.0625 kHz maximum allowed).

Care must be taken in the software to minimize the amount of time spent in this state where the FLL is operating in this condition.

The following code sequence describes how to move from FEI mode to PEE mode until the 8 MHz crystal reference frequency is set to achieve a bus frequency of 8 MHz. Because the MCG is in FEI mode out of reset, this example also shows how to initialize the MCG for PEE mode out of reset. First, the code sequence will be described. Then a flowchart will be included which illustrates the sequence.

1. First, FEI must transition to FBE mode:
  - a)  $MCGC2 = 0x36$  (%00110110)
    - BDIV (bits 7 and 6) set to %00, or divide-by-1
    - RANGE (bit 5) set to 1 because the frequency of 8 MHz is within the high frequency range
    - HGO (bit 4) set to 1 to configure external oscillator for high gain operation
    - EREFS (bit 2) set to 1, because a crystal is being used
    - ERCLKEN (bit 1) set to 1 to ensure the external reference clock is active
  - b) Loop until OSCINIT (bit 1) in MCGSC is 1, indicating the crystal selected by the EREFS bit has been initialized.
  - c) Block Interrupts (If applicable by setting the interrupt bit in the CCR).
  - d)  $MCGC1 = 0xB8$  (%10111000)
    - CLKS (bits 7 and 6) set to %10 in order to select external reference clock as system clock source
    - RDIV (bits 5-3) set to %111, or divide-by-128.

#### NOTE

8 MHz / 128 = 62.5 kHz which is greater than the 31.25 kHz to 39.0625 kHz range required by the FLL. Therefore after the transition to FBE is complete, software must progress through to BLPE mode immediately by setting the LP bit in MCGC2.

- IREFS (bit 2) cleared to 0, selecting the external reference clock
  - e) Loop until IREFST (bit 4) in MCGSC is 0, indicating the external reference is the current source for the reference clock
  - f) Loop until CLKST (bits 3 and 2) in MCGSC are %10, indicating that the external reference clock is selected to feed MCGOUT
2. Then, FBE mode transitions into BLPE mode:
    - a)  $MCGC2 = 0x3E$  (%00111110)
      - LP (bit 3) in MCGC2 to 1 (BLPE mode entered)

#### NOTE

There must be no extra steps (including interrupts) between steps 1d and 2a.

- b) Enable Interrupts (if applicable by clearing the interrupt bit in the CCR).

## 10.1.6 Features

Features of the ADC module include:

- Linear successive approximation algorithm with 12-bit resolution
- Up to 28 analog inputs
- Output formatted in 12-, 10-, or 8-bit right-justified unsigned format
- Single or continuous conversion (automatic return to idle after single conversion)
- Configurable sample time and conversion speed/power
- Conversion complete flag and interrupt
- Input clock selectable from up to four sources
- Operation in wait or stop3 modes for lower noise operation
- Asynchronous clock source for lower noise operation
- Selectable asynchronous hardware conversion trigger
- Automatic compare with interrupt for less-than, or greater-than or equal-to, programmable value
- Temperature sensor

## 10.1.7 ADC Module Block Diagram

Figure 10-2 provides a block diagram of the ADC module.

### 10.3.2 Status and Control Register 2 (ADCSC2)

The ADCSC2 register controls the compare function, conversion trigger, and conversion active of the ADC module.

	7	6	5	4	3	2	1	0
Reset:	0	0	0	0	0	0	0	0

Figure 10-4. Status and Control Register 2 (ADCSC2)

Table 10-5. ADCSC2 Register Field Descriptions

Field	Description
7 ADACT	Conversion Active. Indicates that a conversion is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress 1 Conversion in progress
6 ADTRG	Conversion Trigger Select. Selects the type of trigger used for initiating a conversion. Two types of triggers are selectable: software trigger and hardware trigger. When software trigger is selected, a conversion is initiated following a write to ADCSC1. When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input. 0 Software trigger selected 1 Hardware trigger selected
5 ACFE	Compare Function Enable. Enables the compare function. 0 Compare function disabled 1 Compare function enabled
4 ACFGT	Compare Function Greater Than Enable. Configures the compare function to trigger when the result of the conversion of the input being monitored is greater than or equal to the compare value. The compare function defaults to triggering when the result of the compare of the input being monitored is less than the compare value. 0 Compare triggers when input is less than compare value 1 Compare triggers when input is greater than or equal to compare value

### 10.3.3 Data Result High Register (ADCRH)

In 12-bit operation, ADCRH contains the upper four bits of the result of a 12-bit conversion. In 10-bit mode, ADCRH contains the upper two bits of the result of a 10-bit conversion. When configured for 10-bit mode, ADR[11:10] are cleared. When configured for 8-bit mode, ADR[11:8] are cleared.

In 12-bit and 10-bit mode, ADCRH is updated each time a conversion completes except when automatic compare is enabled and the compare condition is not met. When a compare event does occur, the value is the addition of the conversion result and the two's complement of the compare value. In 12-bit and 10-bit mode, reading ADCRH prevents the ADC from transferring subsequent conversion results into the result registers until ADCRL is read. If ADCRL is not read until after the next conversion is completed, the intermediate conversion result is lost. In 8-bit mode, there is no interlocking with ADCRL.

**Table 10-12. APCTL3 Register Field Descriptions**

Field	Description
7 ADPC23	ADC Pin Control 23. ADPC23 controls the pin associated with channel AD23. 0 AD23 pin I/O control enabled 1 AD23 pin I/O control disabled
6 ADPC22	ADC Pin Control 22. ADPC22 controls the pin associated with channel AD22. 0 AD22 pin I/O control enabled 1 AD22 pin I/O control disabled
5 ADPC21	ADC Pin Control 21. ADPC21 controls the pin associated with channel AD21. 0 AD21 pin I/O control enabled 1 AD21 pin I/O control disabled
4 ADPC20	ADC Pin Control 20. ADPC20 controls the pin associated with channel AD20. 0 AD20 pin I/O control enabled 1 AD20 pin I/O control disabled
3 ADPC19	ADC Pin Control 19. ADPC19 controls the pin associated with channel AD19. 0 AD19 pin I/O control enabled 1 AD19 pin I/O control disabled
2 ADPC18	ADC Pin Control 18. ADPC18 controls the pin associated with channel AD18. 0 AD18 pin I/O control enabled 1 AD18 pin I/O control disabled
1 ADPC17	ADC Pin Control 17. ADPC17 controls the pin associated with channel AD17. 0 AD17 pin I/O control enabled 1 AD17 pin I/O control disabled
0 ADPC16	ADC Pin Control 16. ADPC16 controls the pin associated with channel AD16. 0 AD16 pin I/O control enabled 1 AD16 pin I/O control disabled

## 10.4 Functional Description

The ADC module is disabled during reset or when the ADCH bits are all high. The module is idle when a conversion has completed and another conversion has not been initiated. When idle, the module is in its lowest power state.

The ADC can perform an analog-to-digital conversion on any of the software selectable channels. In 12-bit and 10-bit mode, the selected channel voltage is converted by a successive approximation algorithm into a 12-bit digital result. In 8-bit mode, the selected channel voltage is converted by a successive approximation algorithm into a 9-bit digital result.

When the conversion is completed, the result is placed in the data registers (ADCRH and ADCRL). In 10-bit mode, the result is rounded to 10 bits and placed in the data registers (ADCRH and ADCRL). In 8-bit mode, the result is rounded to 8 bits and placed in ADCRL. The conversion complete flag (COCO) is then set and an interrupt is generated if the conversion complete interrupt has been enabled (AIEN = 1).

The ADC module has the capability of automatically comparing the result of a conversion with the contents of its compare registers. The compare function is enabled by setting the ACFE bit and operates with any of the conversion modes and configurations.

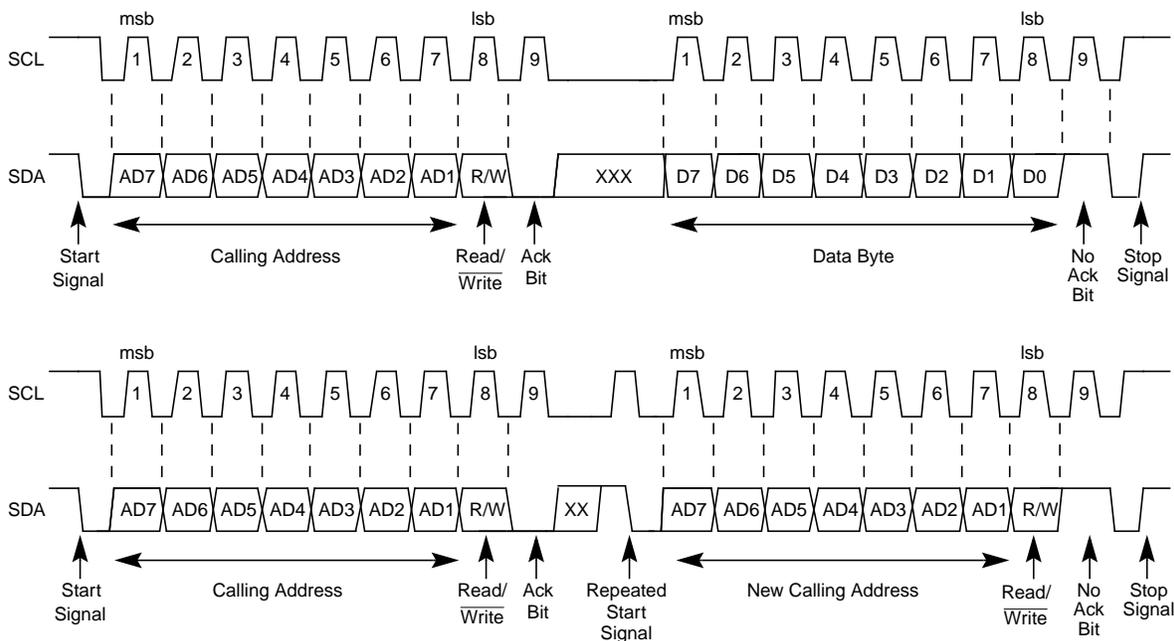


Figure 11-9. IIC Bus Transmission Signals

### 11.4.1.1 Start Signal

When the bus is free, no master device is engaging the bus (SCL and SDA lines are at logical high), a master may initiate communication by sending a start signal. As shown in Figure 11-9, a start signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer (each data transfer may contain several bytes of data) and brings all slaves out of their idle states.

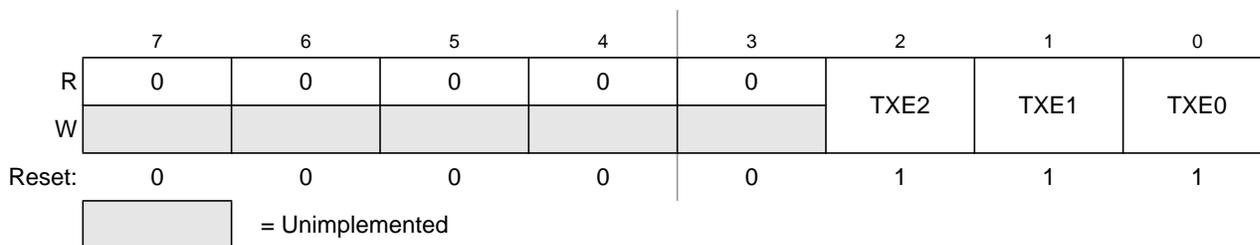
### 11.4.1.2 Slave Address Transmission

The first byte of data transferred immediately after the start signal is the slave address transmitted by the master. This is a seven-bit calling address followed by a  $R/\overline{W}$  bit. The  $R/\overline{W}$  bit tells the slave the desired direction of data transfer.

- 1 = Read transfer, the slave transmits data to the master.
- 0 = Write transfer, the master transmits data to the slave.

Only the slave with a calling address that matches the one transmitted by the master responds by sending back an acknowledge bit. This is done by pulling the SDA low at the ninth clock (see Figure 11-9).

No two slaves in the system may have the same address. If the IIC module is the master, it must not transmit an address equal to its own slave address. The IIC cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the IIC reverts to slave mode and operates correctly even if it is being addressed by another master.



**Figure 12-10. MSCAN Transmitter Flag Register (CANTFLG)**

**NOTE**

The CANTFLG register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Read: Anytime

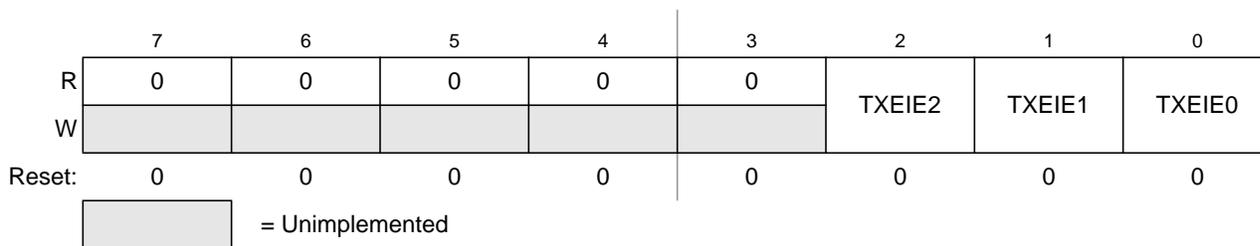
Write: Anytime for TXE<sub>x</sub> flags when not in initialization mode; write of 1 clears flag, write of 0 is ignored

**Table 12-11. CANTFLG Register Field Descriptions**

Field	Description
2:0 TXE[2:0]	<p><b>Transmitter Buffer Empty</b> — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see Section 12.3.8, “MSCAN Transmitter Message Abort Request Register (CANTARQ)”). If not masked, a transmit interrupt is pending while this flag is set.</p> <p>Clearing a TXE<sub>x</sub> flag also clears the corresponding ABTAK<sub>x</sub> (see Section 12.3.9, “MSCAN Transmitter Message Abort Acknowledge Register (CANTAACK)”). When a TXE<sub>x</sub> flag is set, the corresponding ABTRQ<sub>x</sub> bit is cleared (see Section 12.3.8, “MSCAN Transmitter Message Abort Request Register (CANTARQ)”).</p> <p>When listen-mode is active (see Section 12.3.2, “MSCAN Control Register 1 (CANCTL1)”) the TXE<sub>x</sub> flags cannot be cleared and no transmission is started.</p> <p>Read and write accesses to the transmit buffer are blocked, if the corresponding TXE<sub>x</sub> bit is cleared (TXE<sub>x</sub> = 0) and the buffer is scheduled for transmission.</p> <p>0 The associated message buffer is full (loaded with a message due for transmission)            1 The associated message buffer is empty (not scheduled)</p>

### 12.3.7 MSCAN Transmitter Interrupt Enable Register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.



**Figure 12-11. MSCAN Transmitter Interrupt Enable Register (CANTIER)**

- Four identifier acceptance filters, each to be applied to
  - a) the 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages or
  - b) the 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages.
 Figure 12-40 shows how the first 32-bit filter bank (CANIDAR0–CANIDA3, CANIDMR0–3CANIDMR) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.
- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier. Figure 12-41 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

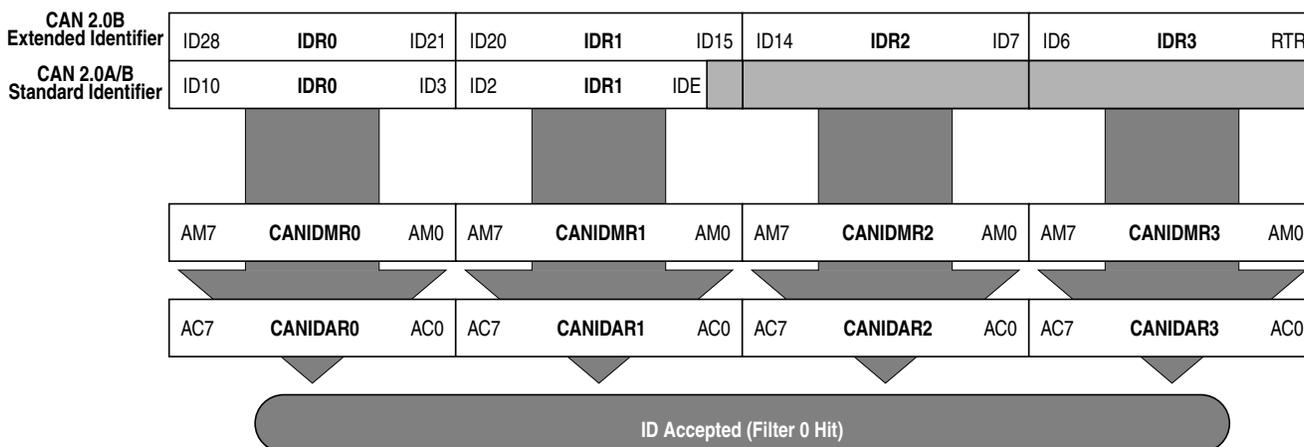


Figure 12-39. 32-bit Maskable Identifier Acceptance Filter

### 12.5.4.3 Emulation Modes

In all emulation modes, the MSCAN module behaves just like normal system operation modes as described within this specification.

### 12.5.4.4 Listen-Only Mode

In an optional CAN bus monitoring mode (listen-only), the CAN node is able to receive valid data frames and valid remote frames, but it sends only “recessive” bits on the CAN bus. In addition, it cannot start a transmission. If the MAC sub-layer is required to send a “dominant” bit (ACK bit, overload flag, or active error flag), the bit is rerouted internally so that the MAC sub-layer monitors this “dominant” bit, although the CAN bus may remain in recessive state externally.

### 12.5.4.5 Security Modes

The MSCAN module has no security features.

### 12.5.4.6 Loopback Self Test Mode

Loopback self test mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input. The RXCAN input pin is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

## 12.5.5 Low-Power Options

If the MSCAN is disabled ( $CANE = 0$ ), the MSCAN clocks are stopped for power saving.

If the MSCAN is enabled ( $CANE = 1$ ), the MSCAN has two additional modes with reduced power consumption, compared to normal mode: sleep and power down mode. In sleep mode, power consumption is reduced by stopping all clocks except those to access the registers from the CPU side. In power down mode, all clocks are stopped and no power is consumed.

Table 12-36 summarizes the combinations of MSCAN and CPU modes. A particular combination of modes is entered by the given settings on the CSWAI and SLPRQ/SLPAK bits.

For all modes, an MSCAN wake-up interrupt can occur only if the MSCAN is in sleep mode ( $SLPRQ = 1$  and  $SLPAK = 1$ ), wake-up functionality is enabled ( $WUPE = 1$ ), and the wake-up interrupt is enabled ( $WUPIE = 1$ ).

The MSCAN is able to leave sleep mode (wake up) only when:

- CAN bus activity occurs and  $WUPE = 1$   
or
- the CPU clears the SLPRQ bit

**NOTE**

The CPU cannot clear the SLPRQ bit before sleep mode ( $SLPRQ = 1$  and  $SLPAK = 1$ ) is active.

After wake-up, the MSCAN waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, if the MSCAN is woken-up by a CAN frame, this frame is not received.

The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions will be executed upon wake-up; copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN remains in bus-off state after sleep mode was exited, it continues counting the 128 occurrences of 11 consecutive recessive bits.

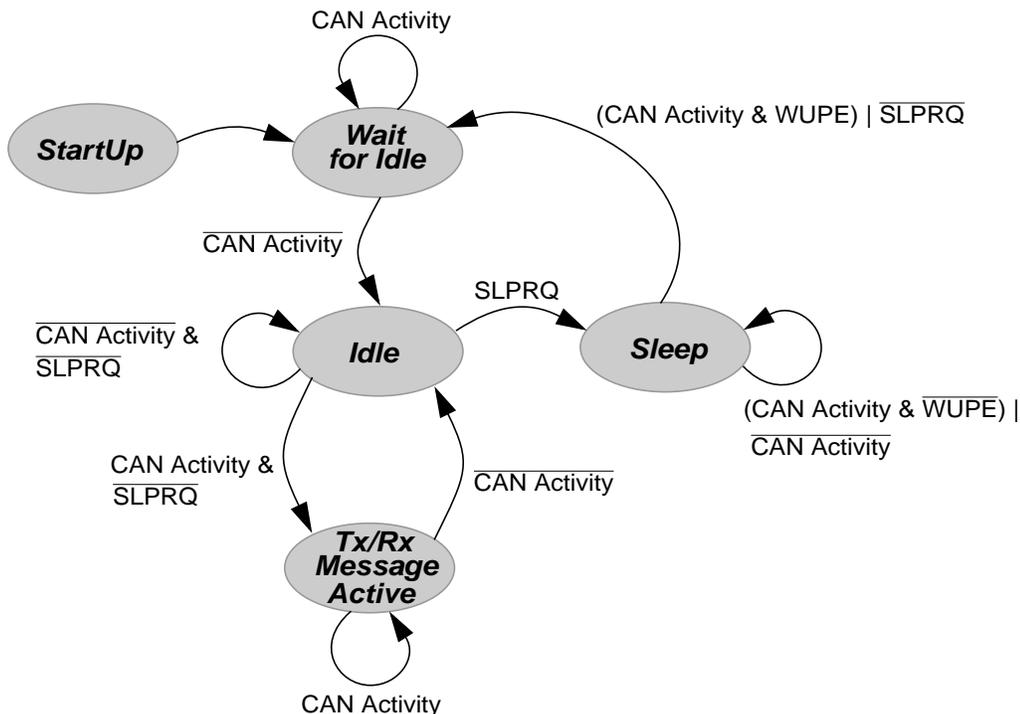


Figure 12-45. Simplified State Transitions for Entering/Leaving Sleep Mode

**Table 13-5. SPI Baud Rate Prescaler Divisor**

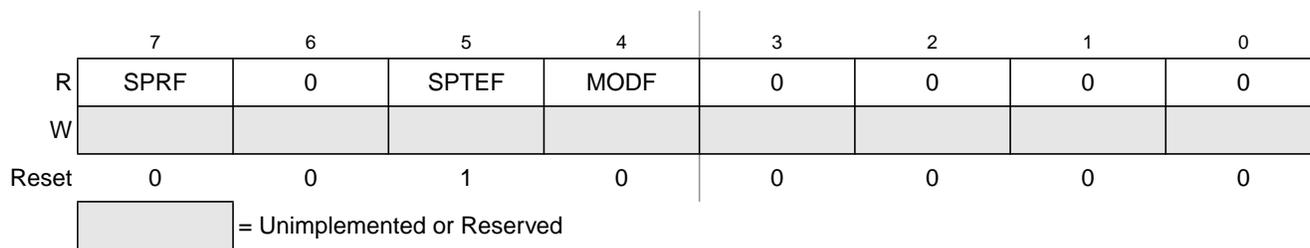
SPPR2:SPPR1:SPPR0	Prescaler Divisor
0:0:0	1
0:0:1	2
0:1:0	3
0:1:1	4
1:0:0	5
1:0:1	6
1:1:0	7
1:1:1	8

**Table 13-6. SPI Baud Rate Divisor**

SPR2:SPR1:SPR0	Rate Divisor
0:0:0	2
0:0:1	4
0:1:0	8
0:1:1	16
1:0:0	32
1:0:1	64
1:1:0	128
1:1:1	256

### 13.4.4 SPI Status Register (SPIS)

This register has three read-only status bits. Bits 6, 3, 2, 1, and 0 are not implemented and always read 0. Writes have no meaning or effect.



**Figure 13-8. SPI Status Register (SPIS)**

### 17.4.3.5 Debug FIFO High Register (DBGFH)

This register provides read-only access to the high-order eight bits of the FIFO. Writes to this register have no meaning or effect. In the event-only trigger modes, the FIFO only stores data into the low-order byte of each FIFO word, so this register is not used and will read 0x00.

Reading DBGFH does not cause the FIFO to shift to the next word. When reading 16-bit words out of the FIFO, read DBGFH before reading DBGFL because reading DBGFL causes the FIFO to advance to the next word of information.

### 17.4.3.6 Debug FIFO Low Register (DBGFL)

This register provides read-only access to the low-order eight bits of the FIFO. Writes to this register have no meaning or effect.

Reading DBGFL causes the FIFO to shift to the next available word of information. When the debug module is operating in event-only modes, only 8-bit data is stored into the FIFO (high-order half of each FIFO word is unused). When reading 8-bit words out of the FIFO, simply read DBGFL repeatedly to get successive bytes of data from the FIFO. It isn't necessary to read DBGFH in this case.

Do not attempt to read data from the FIFO while it is still armed (after arming but before the FIFO is filled or ARMF is cleared) because the FIFO is prevented from advancing during reads of DBGFL. This can interfere with normal sequencing of reads from the FIFO.

Reading DBGFL while the debugger is not armed causes the address of the most-recently fetched opcode to be stored to the last location in the FIFO. By reading DBGFH then DBGFL periodically, external host software can develop a profile of program execution. After eight reads from the FIFO, the ninth read will return the information that was stored as a result of the first read. To use the profiling feature, read the FIFO eight times without using the data to prime the sequence and then begin using the data to get a delayed picture of what addresses were being executed. The information stored into the FIFO on reads of DBGFL (while the FIFO is not armed) is the address of the most-recently fetched opcode.

Table A-9. 12-bit ADC Operating Conditions (continued)

Characteristic	Conditions	Symb	Min	Typ <sup>1</sup>	Max	Unit	Comment
Ref Voltage High		$V_{REFH}$	2.7	$V_{DDAD}$	$V_{DDAD}$	V	Applicable in only 64-pin packages { $V_{REFH} < V_{DDAD}$ characterized but not production test}
Ref Voltage Low		$V_{REFL}$	$V_{SSAD}$	$V_{SSAD}$	$V_{SSAD}$	V	Not Applicable in 64-pin packages (only 32- and 48-pin packages)
Input Voltage		$V_{ADIN}$	$V_{REFL}$	—	$V_{REFH}$	V	
Input Capacitance		$C_{ADIN}$	—	4.5	5.5	pF	
Input Resistance		$R_{ADIN}$	—	3	5	k $\Omega$	
Analog Source Resistance	12 bit mode $f_{ADCK} > 4\text{MHz}$ $f_{ADCK} < 4\text{MHz}$	$R_{AS}$	—	—	2	k $\Omega$	External to MCU
			—	—	5		
	10 bit mode $f_{ADCK} > 4\text{MHz}$ $f_{ADCK} < 4\text{MHz}$		—	—	5		
	8 bit mode (all valid $f_{ADCK}$ )		—	—	10		
ADC Conversion Clock Freq.	High Speed (ADLPC=0)	$f_{ADCK}$	0.4	—	8.0	MHz	
	Low Power (ADLPC=1)		0.4	—	4.0		

<sup>1</sup> Typical values assume  $V_{DDAD} = 5.0\text{V}$ , Temp = 25°C,  $f_{ADCK}=1.0\text{MHz}$  unless otherwise stated. Typical values are for reference only and are not tested in production.

<sup>2</sup> DC potential difference.

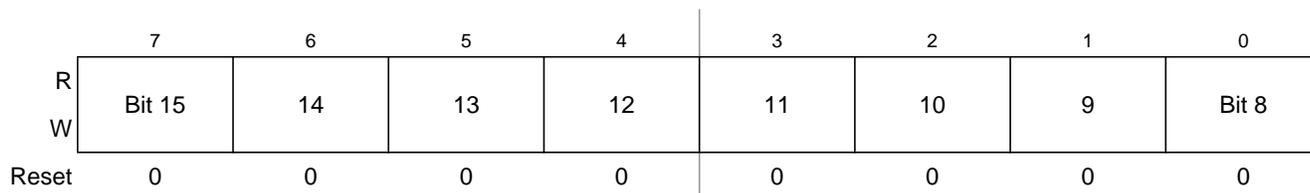
**Table B-5. Mode, Edge, and Level Selection**

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	XX	00	Pin not used for TPM channel; use as an external clock for the TPM or revert to general-purpose I/O	
0	00	01	Input capture	Capture on rising edge only
		10		Capture on falling edge only
		11		Capture on rising or falling edge
	01	00	Output compare	Software compare only
		01		Toggle output on compare
		10		Clear output on compare
11		Set output on compare		
1X	10	Edge-aligned PWM	High-true pulses (clear output on compare)	
	X1		Low-true pulses (set output on compare)	
1	XX	10	Center-aligned PWM	High-true pulses (clear output on compare-up)
		X1		Low-true pulses (set output on compare-up)

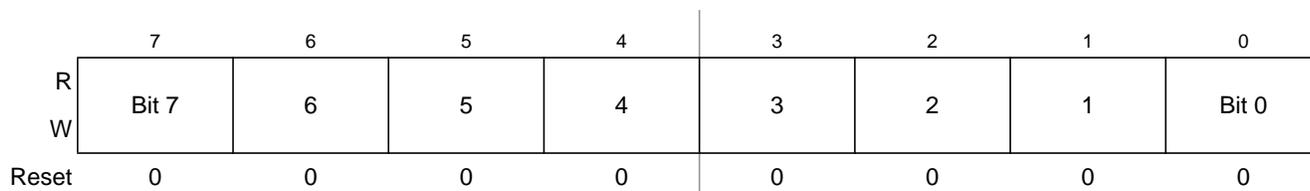
If the associated port pin is not stable for at least two bus clock cycles before changing to input capture mode, it is possible to get an unexpected indication of an edge trigger. Typically, a program would clear status flags after changing channel configuration bits and before enabling channel interrupts or using the status flags to avoid any unexpected behavior.

### B.2.5 Timer Channel Value Registers (TPMxCnVH:TPMxCnVL)

These read/write registers contain the captured TPM counter value of the input capture function or the output compare value for the output compare or PWM functions. The channel value registers are cleared by reset.



**Figure B-8. Timer Channel Value Register High (TPMxCnVH)**



**Figure B-9. Timer Channel Value Register Low (TPMxCnVL)**