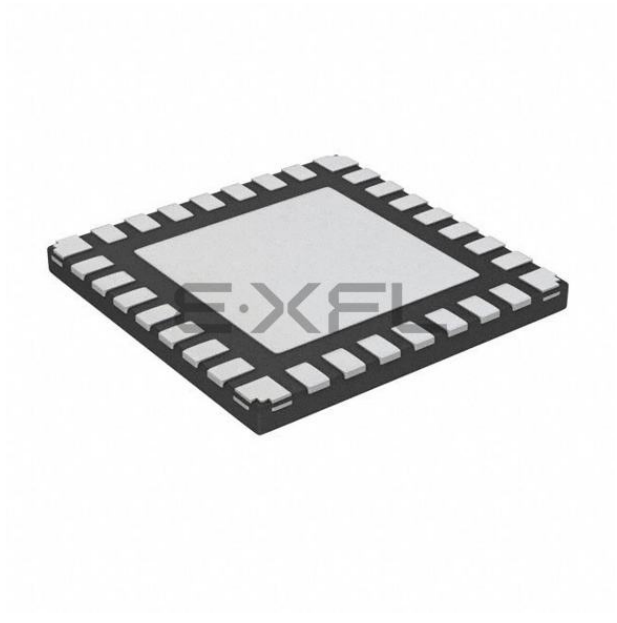


**Welcome to E-XFL.COM**

## 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"

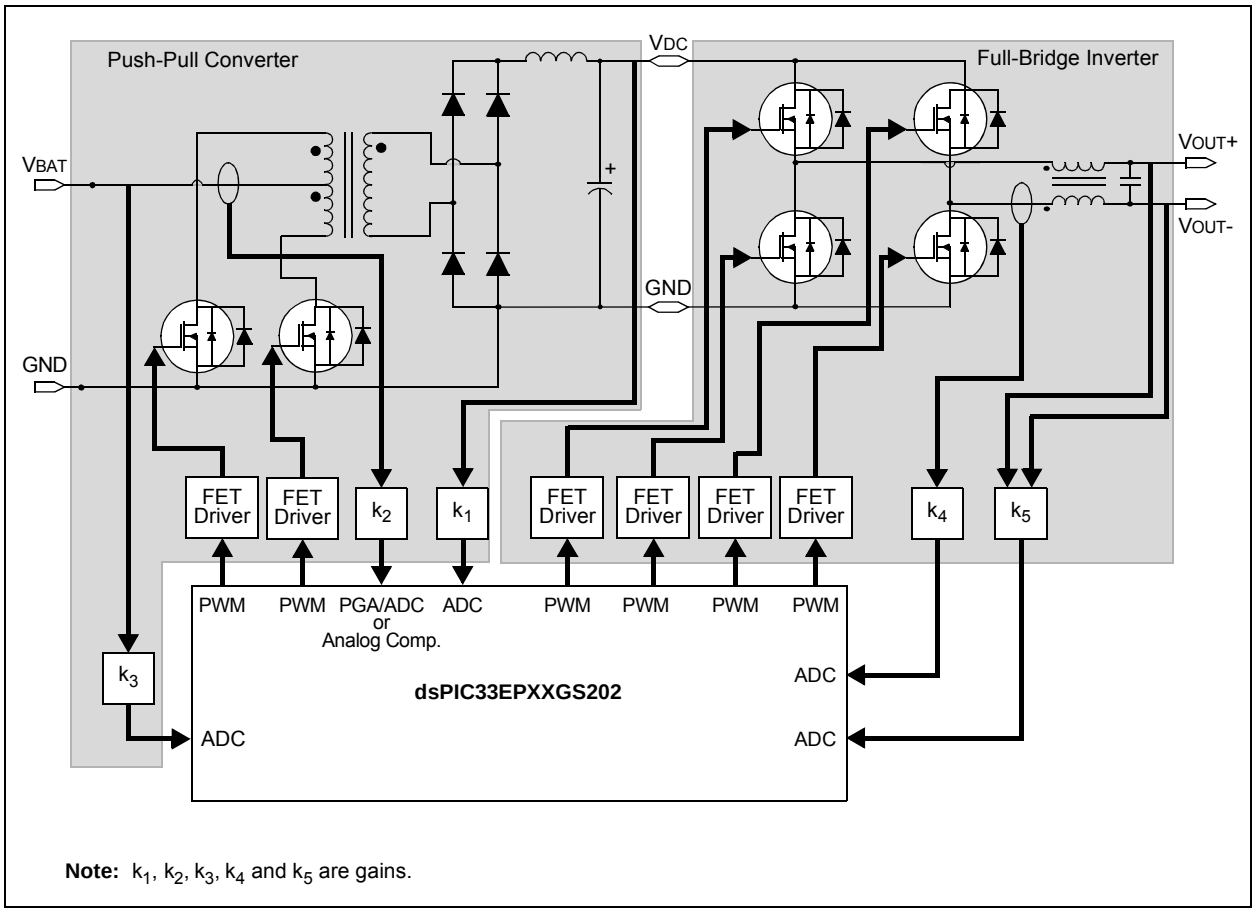
| Details | |
|---|---|
| Product Status | Active |
| Core Processor | dsPIC |
| Core Size | 16-Bit |
| Speed | 70 MIPs |
| Connectivity | I²C, IrDA, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 21 |
| Program Memory Size | 16KB (16K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 12x12b, 2x12b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-UQFN Exposed Pad |
| Supplier Device Package | 28-UQFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic33ep16gs202t-i-mx |

**TABLE 1-1: PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name | Pin Type | Buffer Type | PPS | Description |
|---|---|---|---|---|
| PGA1P1-PGA1P3 | I | Analog | No | PGA1 Positive Inputs 1 through 3. |
| PGA1N2 | I | Analog | No | PGA1 Negative Input 2. |
| PGA2P1-PGA2P3 | I | Analog | No | PGA2 Positive Inputs 1 through 3. |
| PGA2N2 | I | Analog | No | PGA2 Negative Input 2. |
| ADTRG31 | I | ST | No | External ADC trigger source. |
| PGED1 | I/O | ST | No | Data I/O pin for Programming/Debugging Communication Channel 1. |
| PGEC1 | I | ST | No | Clock input pin for Programming/Debugging Communication Channel 1. |
| PGED2 | I/O | ST | No | Data I/O pin for Programming/Debugging Communication Channel 2. |
| PGEC2 | I | ST | No | Clock input pin for Programming/Debugging Communication Channel 2. |
| PGED3 | I/O | ST | No | Data I/O pin for Programming/Debugging Communication Channel 3. |
| PGEC3 | I | ST | No | Clock input pin for Programming/Debugging Communication Channel 3. |
| $\overline{MCLR}$ | I/P | ST | No | Master Clear (Reset) input. This pin is an active-low Reset to the device. |
| $AV_{DD}$ | P | P | No | Positive supply for analog modules. This pin must be connected at all times. |
| $AV_{SS}$ | P | P | No | Ground reference for analog modules. This pin must be connected at all times. |
| $V_{DD}$ | P | — | No | Positive supply for peripheral logic and I/O pins. |
| $V_{CAP}$ | P | — | No | CPU logic filter capacitor connection. |
| $V_{SS}$ | P | — | No | Ground reference for logic and I/O pins. |

**Legend:** CMOS = CMOS compatible input or output
Analog = Analog input
P = Power
ST = Schmitt Trigger input with CMOS levels
O = Output
I = Input
PPS = Peripheral Pin Select
TTL = TTL input buffer

**FIGURE 2-6: OFF-LINE UPS**




**Note:** $k_1$, $k_2$, $k_3$, $k_4$ and $k_5$ are gains.

**TABLE 4-4: TIMER1 THROUGH TIMER3 REGISTER MAP**

| File Name | Addr. | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMR1 | 0100 | Timer1 Register | | | | | | | | | | | | | | | | xxxx |
| PR1 | 0102 | Period Register 1 | | | | | | | | | | | | | | | | FFFF |
| T1CON | 0104 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS1 | TCKPS0 | — | TSYNC | TCS | — | 0000 |
| TMR2 | 0106 | Timer2 Register | | | | | | | | | | | | | | | | xxxx |
| TMR3HLD | 0108 | Timer3 Holding Register (for 32-bit timer operations only) | | | | | | | | | | | | | | | | xxxx |
| TMR3 | 010A | Timer3 Register | | | | | | | | | | | | | | | | xxxx |
| PR2 | 010C | Period Register 2 | | | | | | | | | | | | | | | | FFFF |
| PR3 | 010E | Period Register 3 | | | | | | | | | | | | | | | | FFFF |
| T2CON | 0110 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS1 | TCKPS0 | T32 | — | TCS | — | 0000 |
| T3CON | 0112 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS1 | TCKPS0 | — | — | TCS | — | 0000 |

**Legend:** x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-5: INPUT CAPTURE 1 REGISTER MAP**

| File Name | Addr. | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IC1CON1 | 0140 | — | — | ICSIDL | ICTSEL2 | ICTSEL1 | ICTSEL0 | — | — | — | ICI1 | ICI0 | ICOV | ICBNE | ICM2 | ICM1 | ICM0 | 0000 |
| IC1CON2 | 0142 | — | — | — | — | — | — | — | — | ICTRIG | TRIGSTAT | — | SYNCSEL4 | SYNCSEL3 | SYNCSEL2 | SYNCSEL1 | SYNCSEL0 | 000D |
| IC1BUF | 0144 | Input Capture 1 Buffer Register | | | | | | | | | | | | | | | | xxxx |
| IC1TMR | 0146 | Input Capture 1 Timer Register | | | | | | | | | | | | | | | | 0000 |

**Legend:** x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

**TABLE 4-6: OUTPUT COMPARE 1 REGISTER MAP**

| File Name | Addr. | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OC1CON1 | 0900 | — | — | OCSIDL | OCTSEL2 | OCTSEL1 | OCTSEL0 | — | — | ENFLTA | — | — | OCFLTA | TRIGMODE | OCM2 | OCM1 | OCM0 | 0000 |
| OC1CON2 | 0902 | FLTMD | FLTOUT | FLTTRIEN | OCINV | — | — | — | — | OCTRIG | TRIGSTAT | OCTRIS | SYNCSEL4 | SYNCSEL3 | SYNCSEL2 | SYNCSEL1 | SYNCSEL0 | 000C |
| OC1RS | 0904 | Output Compare 1 Secondary Register | | | | | | | | | | | | | | | | xxxx |
| OC1R | 0906 | Output Compare 1 Register | | | | | | | | | | | | | | | | xxxx |
| OC1TMR | 0908 | Timer Value 1 Register | | | | | | | | | | | | | | | | xxxx |

**Legend:** x = unknown value on Reset; — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

### 4.5.2 EXTENDED X DATA SPACE

The lower portion of the base address space range, between 0x0000 and 0x7FFF, is always accessible regardless of the contents of the Data Space Read Page register. It is indirectly addressable through the register indirect instructions. It can be regarded as being located in the default EDS Page 0 (i.e., EDS address range of 0x000000 to 0x007FFF with the base address bit, EA<15> = 0, for this address range). However, Page 0 cannot be accessed through the upper 32 Kbytes, 0x8000 to 0xFFFF, of base Data Space in combination with DSRPAG = 0x000. Consequently, DSRPAG is initialized to 0x001 at Reset.

| | |
|---|---|
| **Note:** | DSRPAG should not be used to access Page 0. An EDS access with DSRPAG set to 0x000 will generate an address error trap. |

The remaining PSV pages are only accessible using the DSRPAG register in combination with the upper 32 Kbytes, 0x8000 to 0xFFFF, of the base address, where base address bit, EA<15> = 1.

### 4.5.3 SOFTWARE STACK

The W15 register serves as a dedicated Software Stack Pointer (SSP) and is automatically modified by exception processing, subroutine calls and returns; however, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies reading, writing and manipulating the Stack Pointer (for example, creating stack frames).

| | |
|---|---|
| **Note:** | To protect against misaligned stack accesses, W15<0> is fixed to '0' by the hardware. |

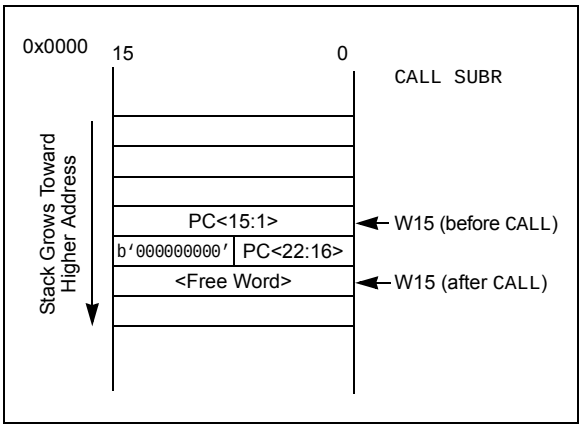
W15 is initialized to 0x1000 during all Resets. This address ensures that the SSP points to valid RAM in all dsPIC33EPXXGS202 devices and permits stack availability for non-maskable trap exceptions. These can occur before the SSP is initialized by the user software. You can reprogram the SSP during initialization to any location within Data Space.

The Software Stack Pointer always points to the first available free word and fills the software stack, working from lower toward higher addresses. Figure 4-7 illustrates how it pre-decrements for a stack pop (read) and post-increments for a stack push (writes).

When the PC is pushed onto the stack, PC<15:0> are pushed onto the first available stack word, then PC<22:16> are pushed into the second available stack location. For a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, as shown in Figure 4-7. During exception processing, the MSB of the PC is concatenated with the lower 8 bits of the CPU STATUS Register, SR. This allows the contents of SRL to be preserved automatically during interrupt processing.

| | |
|---|---|
| **Note 1:** | To maintain the Software Stack Pointer (W15) coherency, W15 is never subject to (EDS) paging, and is therefore, restricted to an address range of 0x0000 to 0xFFFF. The same applies to the W14 when used as a Stack Frame Pointer (SFA = 1). |
| **2:** | As the stack can be placed in, and can access X and Y spaces, care must be taken regarding its use, particularly with regard to local automatic variables in a C development environment |

**FIGURE 4-7: CALL STACK FRAME**



## 8.1 CPU Clocking System

The dsPIC33EPXXGS202 family of devices provides six system clock options:

- Fast RC (FRC) Oscillator
- FRC Oscillator with Phase-Locked Loop (PLL)
- FRC Oscillator with Postscaler
- Primary (XT, HS or EC) Oscillator
- Primary Oscillator with PLL
- Low-Power RC (LPRC) Oscillator

Instruction execution speed or device operating frequency, FCY, is given by Equation 8-1.

**EQUATION 8-1: DEVICE OPERATING FREQUENCY**

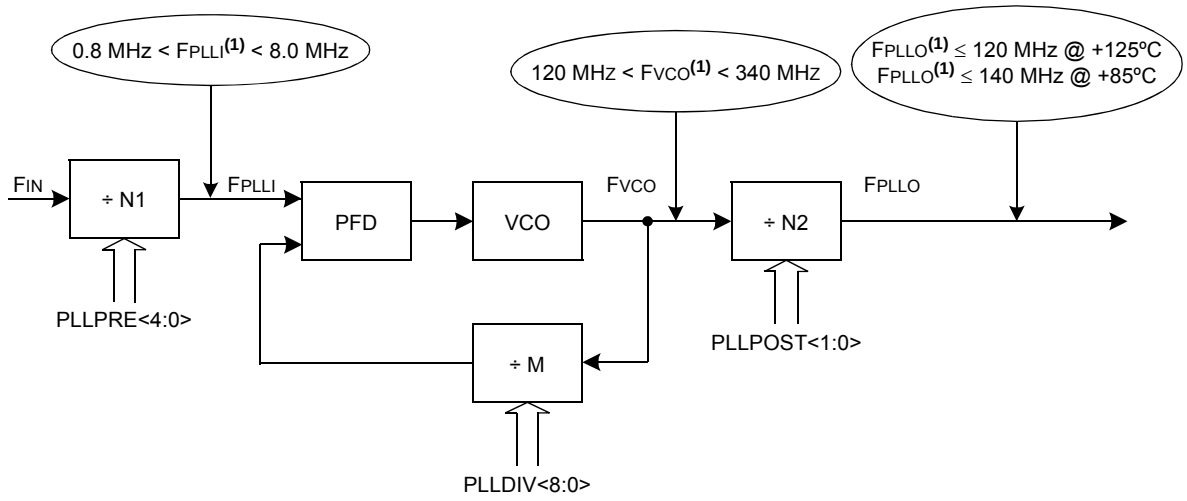
$$FCY = FOSC/2$$

Figure 8-2 is a block diagram of the PLL module.

Equation 8-2 provides the relationship between input frequency (FIN) and output frequency (FPLLO).

Equation 8-3 provides the relationship between input frequency (FIN) and VCO frequency (FVCO).

**FIGURE 8-2: PLL BLOCK DIAGRAM**




**Note 1:** This frequency range must be met at all times.

**EQUATION 8-2: FPLLO CALCULATION**

$$FPLLO = FIN \times \left(\frac{M}{N1 \times N2}\right) = FIN \times \left(\frac{(PLLDIV<8:0> + 2)}{(PLLPRE<4:0> + 2) \times 2(PLLPOST<1:0> + 1)}\right)$$

Where:

$N1 = PLLPRE<4:0> + 2$

$N2 = 2 \text{ x } (PLLPOST<1:0> + 1)$

$M = PLLDIV<8:0> + 2$

**EQUATION 8-3: FVCO CALCULATION**

$$FVCO = FIN \times \left(\frac{M}{N1}\right) = FIN \times \left(\frac{(PLLDIV<8:0> + 2)}{(PLLPRE<4:0> + 2)}\right)$$

**REGISTER 9-1: PMD1: PERIPHERAL MODULE DISABLE CONTROL REGISTER 1**

| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | — | T3MD | T2MD | T1MD | — | PWMMD | — |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | U-0 | R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| I2C1MD | — | U1MD | — | SPI1MD | — | — | ADCMD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 15-14 **Unimplemented:** Read as '0'

bit 13 **T3MD:** Timer3 Module Disable bit
1 = Timer3 module is disabled
0 = Timer3 module is enabled

bit 12 **T2MD:** Timer2 Module Disable bit
1 = Timer2 module is disabled
0 = Timer2 module is enabled

bit 11 **T1MD:** Timer1 Module Disable bit
1 = Timer1 module is disabled
0 = Timer1 module is enabled

bit 10 **Unimplemented:** Read as '0'

bit 9 **PWMMD:** PWM Module Disable bit
1 = PWM module is disabled
0 = PWM module is enabled

bit 8 **Unimplemented:** Read as '0'

bit 7 **I2C1MD:** I2C1 Module Disable bit
1 = I2C1 module is disabled
0 = I2C1 module is enabled

bit 6 **Unimplemented:** Read as '0'

bit 5 **U1MD:** UART1 Module Disable bit
1 = UART1 module is disabled
0 = UART1 module is enabled

bit 4 **Unimplemented:** Read as '0'

bit 3 **SPI1MD:** SPI1 Module Disable bit
1 = SPI1 module is disabled
0 = SPI1 module is enabled

bit 2-1 **Unimplemented:** Read as '0'

bit 0 **ADCMD:** ADC Module Disable bit
1 = ADC module is disabled
0 = ADC module is enabled

**TABLE 10-1: SELECTABLE INPUT SOURCES (MAPS INPUT TO FUNCTION)**

| Input Name(1) | Function Name | Register | Configuration Bits |
|---|---|---|---|
| External Interrupt 1 | INT1 | RPINR0 | INT1R<7:0> |
| External Interrupt 2 | INT2 | RPINR1 | INT2R<7:0> |
| Timer1 External Clock | T1CK | RPINR2 | T1CKR<7:0> |
| Timer2 External Clock | T2CK | RPINR3 | T2CKR<7:0> |
| Timer3 External Clock | T3CK | RPINR3 | T3CKR<7:0> |
| Input Capture 1 | IC1 | RPINR7 | IC1R<7:0> |
| Output Compare Fault A | OCFA | RPINR11 | OCFAR<7:0> |
| PWM Fault 1 | FLT1 | RPINR12 | FLT1R<7:0> |
| PWM Fault 2 | FLT2 | RPINR12 | FLT2R<7:0> |
| PWM Fault 3 | FLT3 | RPINR13 | FLT3R<7:0> |
| PWM Fault 4 | FLT4 | RPINR13 | FLT4R<7:0> |
| UART1 Receive | U1RX | RPINR18 | U1RXR<7:0> |
| UART1 Clear-to-Send | $\overline{\text{U1CTS}}$ | RPINR18 | U1CTSR<7:0> |
| SPI1 Data Input | SDI1 | RPINR20 | SDI1R<7:0> |
| SPI1 Clock Input | SCK1 | RPINR20 | SCK1R<7:0> |
| SPI1 Slave Select | $\overline{\text{SS1}}$ | RPINR21 | SS1R<7:0> |
| PWM Synchronous Input 1 | SYNCI1 | RPINR37 | SYNCI1R<7:0> |
| PWM Synchronous Input 2 | SYNCI2 | RPINR38 | SYNCI2R<7:0> |
| PWM Fault 5 | FLT5 | RPINR42 | FLT5R<7:0> |
| PWM Fault 6 | FLT6 | RPINR42 | FLT6R<7:0> |
| PWM Fault 7 | FLT7 | RPINR43 | FLT7R<7:0> |
| PWM Fault 8 | FLT8 | RPINR43 | FLT8R<7:0> |

**Note 1:** Unless otherwise noted, all inputs use the Schmitt Trigger input buffers.

## 13.0 INPUT CAPTURE

**Note 1:** This data sheet summarizes the features of the dsPIC33EPXXGS202 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to **"Input Capture with Dedicated Timer"** (DS70000352) in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

**2:** Some registers and associated bits described in this section may not be available on all devices. Refer to **Section 4.0 "Memory Organization"** in this data sheet for device-specific register and bit information.

The input capture module is useful in applications requiring frequency (period) and pulse measurements. The dsPIC33EPXXGS202 family devices support one input capture channel.

Key features of the input capture module include:

- Hardware-Configurable for 32-Bit Operation in all modes by Cascading Two Adjacent Modules
- Synchronous and Trigger modes of Output Compare Operation, with up to 6 User-Selectable Trigger/Sync Sources Available
- A 4-Level FIFO Buffer for Capturing and Holding Timer Values for Several Events
- Configurable Interrupt Generation
- Up to Four Clock Sources Available, Driving a Separate Internal 16-Bit Counter

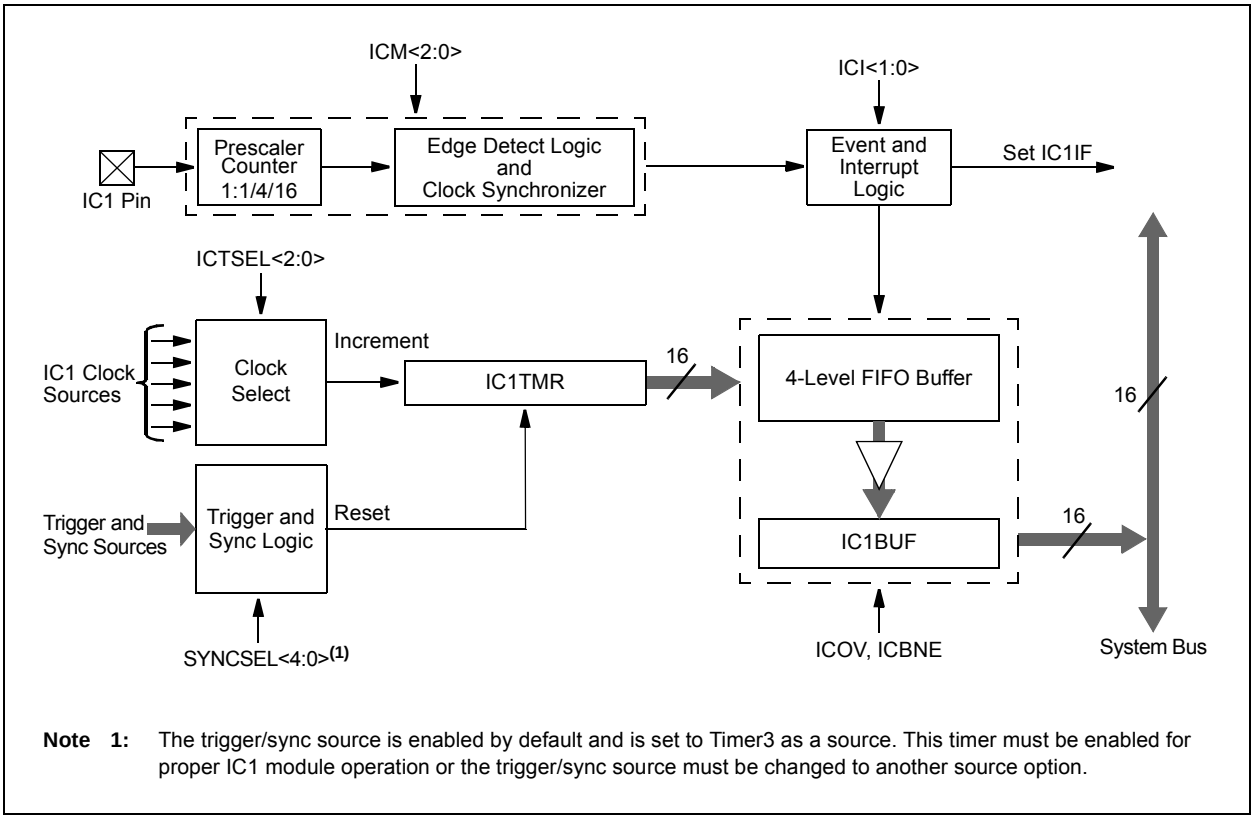
### 13.1 Input Capture Resources

Many useful resources are provided on the main product page of the Microchip web site for the devices listed in this data sheet. This product page contains the latest updates and additional information.

#### 13.1.1 KEY RESOURCES

- **"Input Capture with Dedicated Timer"** (DS70000352) in the *"dsPIC33/PIC24 Family Reference Manual"*
- Code Samples
- Application Notes
- Software Libraries
- Webinars
- All Related *"dsPIC33/PIC24 Family Reference Manual"* Sections

**FIGURE 13-1: INPUT CAPTURE MODULE BLOCK DIAGRAM**




**Note 1:** The trigger/sync source is enabled by default and is set to Timer3 as a source. This timer must be enabled for proper IC1 module operation or the trigger/sync source must be changed to another source option.

**REGISTER 15-12: PWMCONx: PWMx CONTROL REGISTER (CONTINUED)**

bit 7-6 **DTC<1:0>:** Dead-Time Control bits
11 = Reserved
10 = Dead-time function is disabled
01 = Negative dead time is actively applied for Complementary Output mode
00 = Positive dead time is actively applied for all Output modes

bit 5-4 **Unimplemented:** Read as '0'

bit 3 **MTBS:** Master Time Base Select bit
1 = PWMx generator uses the secondary master time base for synchronization and the clock source for the PWMx generation logic (if secondary time base is available)
0 = PWMx generator uses the primary master time base for synchronization and the clock source for the PWMx generation logic

bit 2 **CAM:** Center-Aligned Mode Enable bit(2,3,4)
1 = Center-Aligned mode is enabled
0 = Edge-Aligned mode is enabled

bit 1 **XPRES:** External PWMx Reset Control bit(5)
1 = Current-limit source resets the time base for this PWMx generator if it is in Independent Time Base mode
0 = External pins do not affect the PWMx time base

bit 0 **IUE:** Immediate Update Enable bit
1 = Updates to the active Duty Cycle, Phase Offset, Dead-Time and local Time Base Period registers are immediate
0 = Updates to the active Duty Cycle, Phase Offset, Dead-Time and local Time Base Period registers are synchronized to the local PWMx time base

**Note 1:** Software must clear the interrupt status here and in the corresponding IFSx register in the interrupt controller.
**2:** The Independent Time Base mode (ITB = 1) must be enabled to use Center-Aligned mode. If ITB = 0, the CAM bit is ignored.
**3:** These bits should not be changed after the PWM is enabled by setting PTEN (PTCON<15>) = 1.
**4:** Center-Aligned mode ignores the Least Significant 3 bits of the Duty Cycle, Phase and Dead-Time registers. The highest Center-Aligned mode resolution available is 8.32 ns with the clock prescaler set to the fastest clock.
**5:** Configure CLMOD (FCLCONx<8>) = 0 and ITB (PWMCONx<9>) = 1 to operate in External Period Reset mode.

**REGISTER 15-20: IOCONx: PWMx I/O CONTROL REGISTER**

| R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PENH | PENL | POLH | POLL | PMOD1(1) | PMOD0(1) | OVRENH | OVRENL |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| OVRDAT1 | OVRDAT0 | FLTDAT1(2) | FLTDAT0(2) | CLDAT1(2) | CLDAT0(2) | SWAP | OSYNC |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 15 **PENH:** PWMxH Output Pin Ownership bit
1 = PWM module controls the PWMxH pin
0 = GPIO module controls the PWMxH pin

bit 14 **PENL:** PWMxL Output Pin Ownership bit
1 = PWM module controls the PWMxL pin
0 = GPIO module controls the PWMxL pin

bit 13 **POLH:** PWMxH Output Pin Polarity bit
1 = PWMxH pin is active-low
0 = PWMxH pin is active-high

bit 12 **POLL:** PWMxL Output Pin Polarity bit
1 = PWMxL pin is active-low
0 = PWMxL pin is active-high

bit 11-10 **PMOD<1:0>:** PWMx I/O Pin Mode bits(1)
11 = PWMx I/O pin pair is in the True Independent Output mode
10 = PWMx I/O pin pair is in the Push-Pull Output mode
01 = PWMx I/O pin pair is in the Redundant Output mode
00 = PWMx I/O pin pair is in the Complementary Output mode

bit 9 **OVRENH:** Override Enable for PWMxH Pin bit
1 = OVRDAT1 provides data for output on the PWMxH pin
0 = PWMx generator provides data for the PWMxH pin

bit 8 **OVRENL:** Override Enable for PWMxL Pin bit
1 = OVRDAT0 provides data for output on the PWMxL pin
0 = PWMx generator provides data for the PWMxL pin

bit 7-6 **OVRDAT<1:0>:** Data for PWMxH, PWMxL Pins if Override is Enabled bits
If OVERENH = 1, OVRDAT1 provides the data for the PWMxH pin.
If OVERENL = 1, OVRDAT0 provides the data for the PWMxL pin.

bit 5-4 **FLTDAT<1:0>:** State for PWMxH and PWMxL Pins if FLTMOD<1:0> are Enabled bits(2)
IFLTMOD (FCLCONx<15>) = 0: Normal Fault mode:
If Fault is active, then FLTDAT1 provides the state for the PWMxH pin.
If Fault is active, then FLTDAT0 provides the state for the PWMxL pin.
IFLTMOD (FCLCONx<15>) = 1: Independent Fault mode:
If current-limit is active, then FLTDAT1 provides the state for the PWMxH pin.
If Fault is active, then FLTDAT0 provides the state for the PWMxL pin.

**Note 1:** These bits should not be changed after the PWM module is enabled (PTEN = 1).
**2:** State represents the active/inactive state of the PWMx depending on the POLH and POLL bits settings.

**REGISTER 16-1: SPI1STAT: SPI1 STATUS AND CONTROL REGISTER (CONTINUED)**

bit 1 **SPITBF:** SPI1 Transmit Buffer Full Status bit

1 = Transmit has not yet started, SPI1TXB is full
0 = Transmit has started, SPI1TXB is empty

Standard Buffer mode:
Automatically set in hardware when core writes to the SPI1BUF location, loading SPI1TXB. Automatically cleared in hardware when SPI1 module transfers data from SPI1TXB to SPI1SR.

Enhanced Buffer mode:
Automatically set in hardware when the CPU writes to the SPI1BUF location, loading the last available buffer location. Automatically cleared in hardware when a buffer location is available for a CPU write operation.

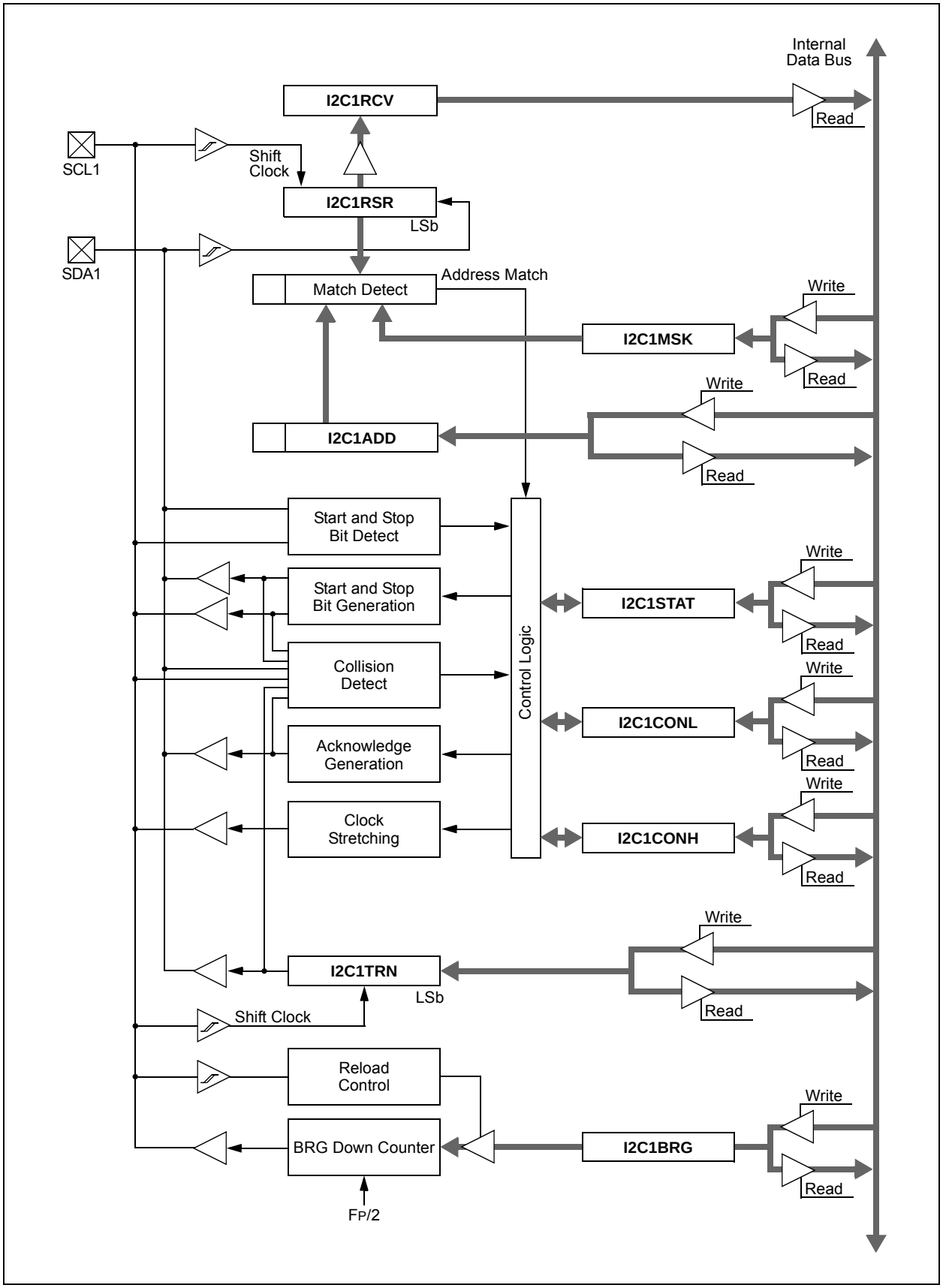
bit 0 **SPIRBF:** SPI1 Receive Buffer Full Status bit

1 = Receive is complete, SPI1RXB is full
0 = Receive is incomplete, SPI1RXB is empty

Standard Buffer mode:
Automatically set in hardware when SPI1 transfers data from SPI1SR to SPI1RXB. Automatically cleared in hardware when the core reads the SPI1BUF location, reading SPI1RXB.

Enhanced Buffer mode:
Automatically set in hardware when SPI1 transfers data from SPI1SR to the buffer, filling the last unread buffer location. Automatically cleared in hardware when a buffer location is available for a transfer from SPI1SR.

**FIGURE 17-1: I2C1 BLOCK DIAGRAM**



**REGISTER 19-26: ADFL0CON: ADC DIGITAL FILTER 0 CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0, HC, HS |
|---|---|---|---|---|---|---|---|
| FLEN | MODE1 | MODE0 | OVRSAM2 | OVRSAM1 | OVRSAM0 | IE | RDY |
| bit 15 | | | | | | | bit 8 |

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | FLCHSEL4 | FLCHSEL3 | FLCHSEL2 | FLCHSEL1 | FLCHSEL0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | HC = Hardware Clearable bit | HS = Hardware Settable bit | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 15 **FLEN:** Filter Enable bit
1 = Filter is enabled
0 = Filter is disabled and the RDY bit is cleared

bit 14-13 **MODE<1:0>:** Filter Mode bits
11 = Averaging mode
10 = Reserved
01 = Reserved
00 = Oversampling mode

bit 12-10 **OVRSAM<2:0>:** Filter Averaging/Oversampling Ratio bits
If MODE<1:0> = 00:
111 = 128x (16-bit result in the ADFL0DAT register is in 12.4 format)
110 = 32x (15-bit result in the ADFL0DAT register is in 12.3 format)
101 = 8x (14-bit result in the ADFL0DAT register is in 12.2 format)
100 = 2x (13-bit result in the ADFL0DAT register is in 12.1 format)
011 = 256x (16-bit result in the ADFL0DAT register is in 12.4 format)
010 = 64x (15-bit result in the ADFL0DAT register is in 12.3 format)
001 = 16x (14-bit result in the ADFL0DAT register is in 12.2 format)
000 = 4x (13-bit result in the ADFL0DAT register is in 12.1 format)
If MODE<1:0> = 11 (12-bit result in the ADFL0DAT register):
111 = 256x
110 = 128x
101 = 64x
100 = 32x
011 = 16x
010 = 8x
001 = 4x
000 = 2x

bit 9 **IE:** Filter Common ADC Interrupt Enable bit
1 = Common ADC interrupt will be generated when the filter result will be ready
0 = Common ADC interrupt will not be generated for the filter

bit 8 **RDY:** Oversampling Filter Data Ready Flag bit
This bit is cleared by hardware when the result is read from the ADFL0DAT register.
1 = Data in the ADFL0DAT register is ready
0 = The ADFL0DAT register has been read and new data in the ADFL0DAT register is not ready

bit 7-5 **Unimplemented:** Read as '0'

## 22.0 SPECIAL FEATURES

**Note:** This data sheet summarizes the features of the dsPIC33EPXXGS202 family of devices. It is not intended to be a comprehensive reference source. To complement the information in this data sheet, refer to the related section in the *"dsPIC33/PIC24 Family Reference Manual"*, which is available from the Microchip web site (www.microchip.com).

The dsPIC33EPXXGS202 family devices include several features intended to maximize application flexibility and reliability, and minimize cost through elimination of external components. These are:

- Flexible Configuration
- Watchdog Timer (WDT)
- Code Protection and CodeGuard™ Security
- JTAG Boundary Scan Interface
- In-Circuit Serial Programming™ (ICSP™)
- In-Circuit Emulation
- Brown-out Reset (BOR)

### 22.1 Configuration Bits

In the dsPIC33EPXXGS202 family devices, the Configuration Words are implemented as volatile memory. This means that configuration data must be programmed each time the device is powered up. Configuration data is stored at the end of the on-chip program memory space, known as the Flash Configuration Words. Their specific locations are shown in Table 22-1 with detailed descriptions in Table 22-2. The configuration data is automatically loaded from the Flash Configuration Words to the proper Configuration Shadow registers during device Resets.

**Note:** Configuration data is reloaded on all types of device Resets.

When creating applications for these devices, users should always specifically allocate the location of the Flash Configuration Words for configuration data in their code for the compiler. This is to make certain that program code is not stored in this address when the code is compiled. Program code executing out of configuration space will cause a device Reset.

**Note:** Performing a page erase operation on the last page of program memory clears the Flash Configuration Words.

Most instructions are a single word. Certain double-word instructions are designed to provide all the required information in these 48 bits. In the second word, the 8 MSbs are '0's. If this second word is executed as an instruction (by itself), it executes as a NOP.

The double-word instructions execute in two instruction cycles.

Most single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the Program Counter is changed as a result of the instruction, or a PSV or Table Read is performed. In these cases, the execution takes multiple instruction cycles, with the additional instruction cycle(s) executed as a NOP. Certain instructions that involve skipping over the subsequent instruction require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single-word or two-word instruction. Moreover, double-word moves require two cycles.

**Note:** For more details on the instruction set, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).

**TABLE 23-1: SYMBOLS USED IN OPCODE DESCRIPTIONS**

| Field | Description |
|---|---|
| #text | Means literal defined by "text" |
| (text) | Means "content of text" |
| [text] | Means "the location addressed by text" |
| { } | Optional field or operation |
| a ∈ {b, c, d} | a is selected from the set of values b, c, d |
| <n:m> | Register bit field |
| .b | Byte mode selection |
| .d | Double-Word mode selection |
| .S | Shadow register select |
| .w | Word mode selection (default) |
| Acc | One of two accumulators {A, B} |
| AWB | Accumulator write-back destination address register ∈ {W13, [W13]+ = 2} |
| bit4 | 4-bit bit selection field (used in word addressed instructions) ∈ {0...15} |
| C, DC, N, OV, Z | MCU Status bits: Carry, Digit Carry, Negative, Overflow, Sticky Zero |
| Expr | Absolute address, label or expression (resolved by the linker) |
| f | File register address ∈ {0x0000...0x1FFF} |
| lit1 | 1-bit unsigned literal ∈ {0,1} |
| lit4 | 4-bit unsigned literal ∈ {0...15} |
| lit5 | 5-bit unsigned literal ∈ {0...31} |
| lit8 | 8-bit unsigned literal ∈ {0...255} |
| lit10 | 10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode |
| lit14 | 14-bit unsigned literal ∈ {0...16384} |
| lit16 | 16-bit unsigned literal ∈ {0...65535} |
| lit23 | 23-bit unsigned literal ∈ {0...8388608}; LSb must be '0' |
| None | Field does not require an entry, can be blank |
| OA, OB, SA, SB | DSP Status bits: ACCA Overflow, ACCB Overflow, ACCA Saturate, ACCB Saturate |
| PC | Program Counter |
| Slit10 | 10-bit signed literal ∈ {-512...511} |
| Slit16 | 16-bit signed literal ∈ {-32768...32767} |
| Slit6 | 6-bit signed literal ∈ {-16...16} |
| Wb | Base W register ∈ {W0...W15} |
| Wd | Destination W register ∈ { Wd, [Wd], [Wd++], [Wd--], [++Wd], [--Wd] } |
| Wdo | Destination W register ∈<br>{ Wnd, [Wnd], [Wnd++], [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb] } |
| Wm,Wn | Dividend, Divisor Working register pair (Direct Addressing) |

## 24.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

## 24.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

## 24.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 24.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

**FIGURE 25-15: SPI1 SLAVE MODE (FULL-DUPLEX, CKE = 1, CKP = 0, SMP = 0) TIMING CHARACTERISTICS**

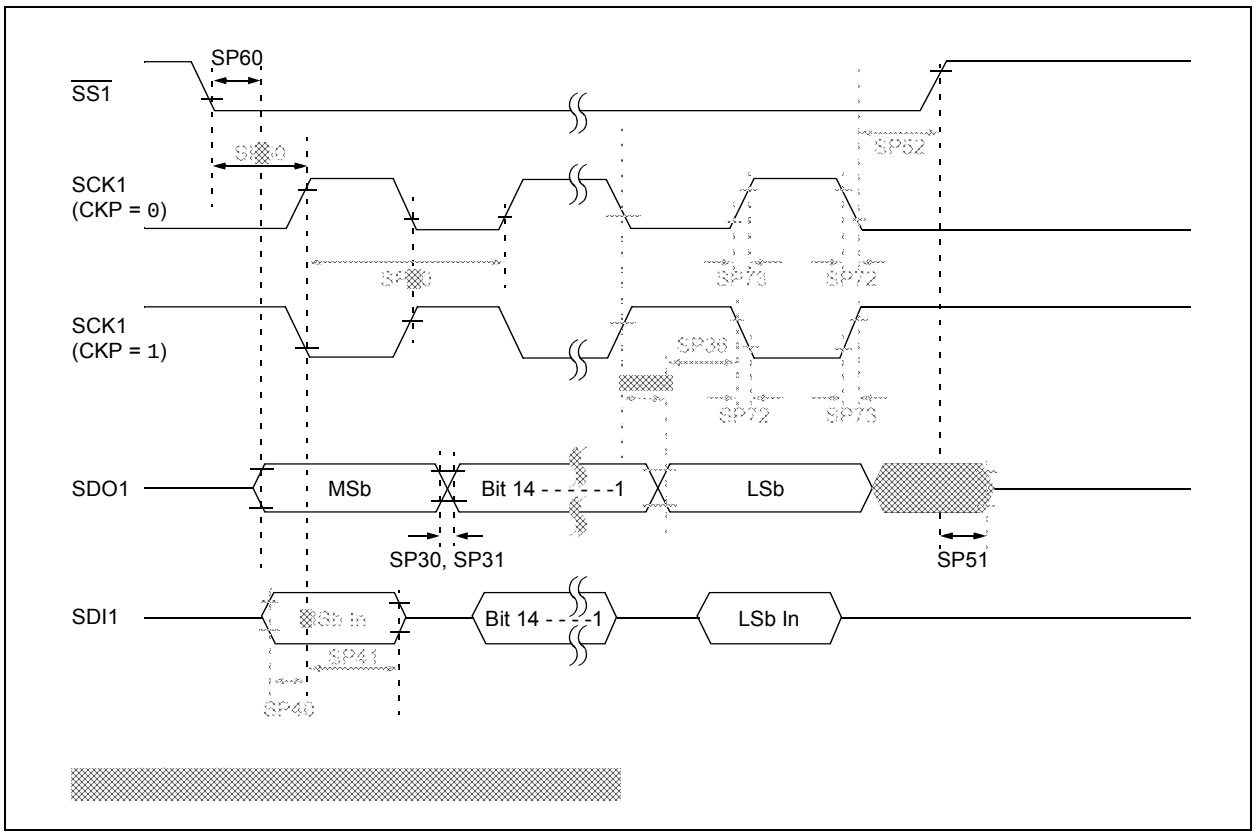


**TABLE 25-42: ADC MODULE SPECIFICATIONS**

| AC CHARACTERISTICS | | | Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)[4]<br>Operating temperature -40°C ≤ TA ≤ +85°C for Industrial<br>-40°C ≤ TA ≤ +125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristics[3]** | **Min.** | **Typical** | **Max.** | **Units** | **Conditions** |
| **Device Supply** | | | | | | | |
| AD01 | AVDD | Module VDD Supply | Greater of: VDD – 0.3 or 3.0 | — | Lesser of: VDD + 0.3 or 3.6 | V | The difference between AVDD supply and VDD supply must not exceed ±300 mV at all times, including device power-up |
| AD02 | AVSS | Module VSS Supply | VSS | — | VSS + 0.3 | V | |
| **Analog Input** | | | | | | | |
| AD12 | VINH-VINL | Full-Scale Input Span | AVSS | — | AVDD | V | |
| AD14 | VIN | Absolute Input Voltage | AVSS – 0.3 | — | AVDD + 0.3 | V | |
| AD15 | VIN+ | Pseudo-Differential Mode | 0 | — | 3.3 | V | VIN- = (VR+ + VR-)/2 ±150 mV |
| AD16 | VIN- | Pseudo-Differential Mode | 0 | — | 3.3 | V | VIN+ = (VR+ + VR-)/2 ±150 mV |
| AD17 | RIN | Recommended Impedance of Analog Voltage Source | — | 100 | — | Ω | For minimum sampling time **(Note 1)** |
| AD66 | VREF1 | Internal Voltage Reference Source | — | 1.2 | — | V | |
| **ADC Accuracy: Pseudo-Differential Input** | | | | | | | |
| AD20a | Nr | Resolution | 12 | | | bits | |
| AD21a | INL | Integral Nonlinearity | > -4 | — | < 4 | LSb | AVSS = 0V, AVDD = 3.3V |
| AD22a | DNL | Pseudo-Differential Nonlinearity | > -1 | — | < 1 | LSb | AVSS = 0V, AVDD = 3.3V **(Note 5)** |
| AD23a | GERR | Gain Error (Dedicated Core) | > -5 | — | < 5 | LSb | AVSS = 0V, AVDD = 3.3V |
| AD24a | EOFF | Offset Error (Dedicated Core) | > -5 | — | < 5 | LSb | AVSS = 0V, AVDD = 3.3V |
| AD25a | — | Monotonicity | — | — | — | — | Guaranteed |

**Note 1:** These parameters are not characterized or tested in manufacturing.

**2:** These parameters are characterized but not tested in manufacturing.

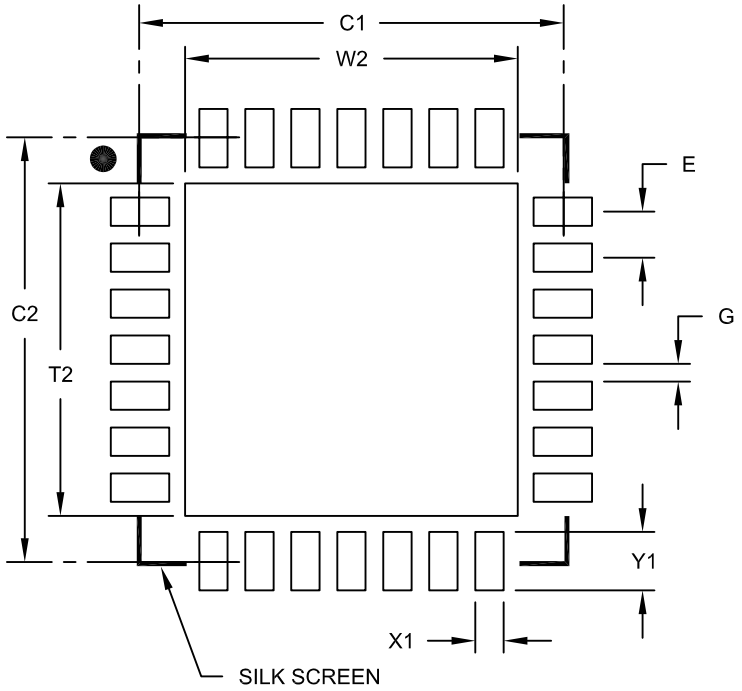
**3:** Characterized with a 1 kHz sine wave.

**4:** The ADC module is functional at VBORMIN < VDD < VDDMIN, but with degraded performance. Unless otherwise stated, module functionality is ensured, but not characterized.

**5:** No missing codes, limits are based on the characterization results.

**28-Lead Plastic Quad Flat, No Lead Package (MM) – 6x6x0.9 mm Body [QFN-S] with 0.40 mm Contact Length**

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.65 BSC | |
| Optional Center Pad Width | W2 | | | 4.70 |
| Optional Center Pad Length | T2 | | | 4.70 |
| Contact Pad Spacing | C1 | | 6.00 | |
| Contact Pad Spacing | C2 | | 6.00 | |
| Contact Pad Width (X28) | X1 | | | 0.40 |
| Contact Pad Length (X28) | Y1 | | | 0.85 |
| Distance Between Pads | G | 0.25 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2124A