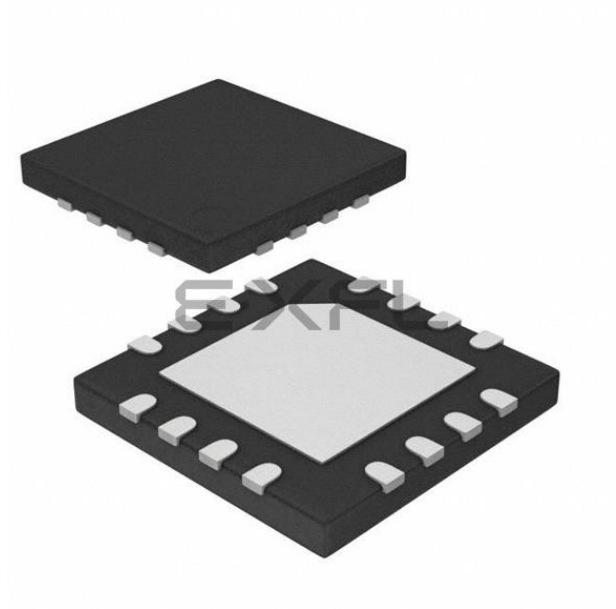


**Welcome to E-XFL.COM**

## 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 | PIC |
| Core Size | 8-Bit |
| Speed | 32MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 12 |
| Program Memory Size | 3.5KB (2K x 14) |
| Program Memory Type | FLASH |
| EEPROM Size | 224 x 8 |
| RAM Size | 256 x 8 |
| Voltage - Supply (Vcc/Vdd) | 1.8V ~ 3.6V |
| Data Converters | A/D 11x10b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 16-UQFN Exposed Pad |
| Supplier Device Package | 16-UQFN (4x4) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic16lf15323t-i-jq |

**TABLE 4: 14/16-PIN ALLOCATION TABLE (PIC16(L)F15323)**

| I/O(2) | 14-Pin PDIP/SOIC/TSSOP | 16-Pin QFN/UQFN | ADC | Reference | Comparator | NCO | DAC | Timers | CCP | PWM | CWG | MSSP | ZCD | EUSART | CLC | CLKR | Interrupt | Pull-up | Basic |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RA0 | 13 | 12 | ANA0 | — | C1IN0+ | — | DAC1OUT | — | — | — | — | — | — | — | — | — | IOCA0 | Y | ICSPDAT |
| RA1 | 12 | 11 | ANA1 | VREF+ | C1IN0-<br>C2IN0- | — | DAC1REF+ | T0CKI(1) | — | — | — | — | — | — | — | — | IOCA1 | Y | ICSPCLK |
| RA2 | 11 | 10 | ANA2 | — | — | — | — | — | — | — | CWG1IN(1) | — | ZCD1 | — | — | — | INT(1)<br>IOCA2 | Y | — |
| RA3 | 4 | 3 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IOCA3 | Y | $\overline{MCLR}$<br>VPP |
| RA4 | 3 | 2 | ANA4 | — | — | — | — | T1G(1) | — | — | — | — | — | — | — | — | IOCA4 | Y | CLKOUT<br>OSC2 |
| RA5 | 2 | 1 | ANA5 | — | — | — | — | T1CKI(1)<br>T2IN | | — | — | — | — | — | CLCIN3(1) | — | IOCA5 | Y | CLKIN<br>OSC1<br>EIN |
| RC0 | 10 | 9 | ANC0 | — | C2IN0+ | — | — | — | — | — | — | SCK1(1)<br>SCL1(1,4) | — | TX2(1)<br>CK2(1) | — | — | IOCC0 | Y | — |
| RC1 | 9 | 8 | ANC1 | — | C1IN1-<br>C2IN1- | — | — | — | — | — | — | SDA1(1,4)<br>SDI1(1) | — | RX2(1)<br>DT2(1) | CLCIN2(1) | — | IOCC1 | Y | — |
| RC2 | 8 | 7 | ANC2 | — | C1IN2-<br>C2IN2- | — | — | — | — | — | — | — | — | — | — | — | IOCC2 | Y | — |
| RC3 | 7 | 6 | ANC3 | — | C1IN3-<br>C2IN3- | — | — | — | CCP2(1) | — | — | $\overline{SS1}$(1) | — | — | CLCIN0(1) | — | IOCC3 | Y | — |
| RC4 | 6 | 5 | ANC4 | — | — | — | — | — | — | — | — | — | — | TX1(1)<br>CK1(1) | CLCIN1(1) | — | IOCC4 | Y | — |
| RC5 | 5 | 4 | ANC5 | — | — | — | — | — | CCP1(1) | — | — | — | — | RX1(1)<br>DT1(1) | — | — | IOCC5 | Y | — |
| VDD | 1 | 16 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VDD |
| VSS | 14 | 13 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VSS |

**Note 1:** This is a PPS re-mappable input signal. The input function may be moved from the default location shown to one of several other PORTx pins.
**2:** All digital output signals shown in this row are PPS re-mappable. These signals may be mapped to output onto one of several PORTx pin options.
**3:** This is a bidirectional signal. For normal module operation, the firmware should map this signal to the same pin in both the PPS input and PPS output registers.
**4:** These pins are configured for $I^2C$ logic levels. PPS assignments to the other pins will operate, but input logic levels will be standard TTL/ST as selected by the INLVL register, instead of the $I^2C$ specific or SMBUS input buffer thresholds.

**TABLE 4-9: SPECIAL FUNCTION REGISTER SUMMARY BANKS 0-63 (ALL BANKS)**

| Bank Offset Bank 0-Bank 63 | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on: MCLR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **All Banks** | | | | | | | | | | | |
| x00h or x80h | INDF0 | Addressing this location uses contents of FSR0H/FSR0L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| x01h or x81h | INDF1 | Addressing this location uses contents of FSR1H/FSR1L to address data memory (not a physical register) | | | | | | | | xxxx xxxx | xxxx xxxx |
| x02h or x82h | PCL | PCL | | | | | | | | 0000 0000 | 0000 0000 |
| x03h or x83h | STATUS | — | — | — | $\overline{TO}$ | $\overline{PD}$ | Z | DC | C | ---1 1000 | ---q quuu |
| x04h or x84h | FSR0L | FSR0L | Indirect Data Memory Address 0 Low Pointer | | | | | | | 0000 0000 | uuuu uuuu |
| x05h or x85h | FSR0H | FSR0H | Indirect Data Memory Address 0 High Pointer | | | | | | | 0000 0000 | 0000 0000 |
| x06h or x86h | FSR1L | FSR1L | Indirect Data Memory Address 1 Low Pointer | | | | | | | 0000 0000 | uuuu uuuu |
| x07h or x87h | FSR1H | FSR1H | Indirect Data Memory Address 1 High Pointer | | | | | | | 0000 0000 | 0000 0000 |
| x08h or x88h | BSR | — | — | BSR<5:0> | | | | | | --00 0000 | --00 0000 |
| x09h or x89h | WREG | Working Register | | | | | | | | 0000 0000 | uuuu uuuu |
| x0Ah or x8Ah | PCLATH | — | Write Buffer for the upper 7 bits of the Program Counter | | | | | | | -000 0000 | -000 0000 |
| x0Bh or x8Bh | INTCON | GIE | PEIE | — | — | — | — | — | INTEDG | 00-- ---1 | 00-- ---1 |

**Legend:** x = unknown, u = unchanged, q = depends on condition, - = unimplemented, read as '0', r = reserved. Shaded locations unimplemented, read as '0'.

**Note 1:** These Registers can be accessed from any bank.

## 9.5 Register Definitions: Oscillator Control

### REGISTER 9-1: OSCCON1: OSCILLATOR CONTROL REGISTER1

| U-0 | R/W-f/f(1) | R/W-f/f(1) | R/W-f/f(1) | R/W-q/q | R/W-q/q | R/W-q/q | R/W-q/q |
|---|---|---|---|---|---|---|---|
| — | NOSC<2:0>(2,3) | | | NDIV<3:0>(2,3,4) | | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | f = determined by fuse setting |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **NOSC<2:0>:** New Oscillator Source Request bits
The setting requests a source oscillator and PLL combination per Table 9-1.
POR value = RSTOSC (Register 5-1).

bit 3-0 **NDIV<3:0>:** New Divider Selection Request bits
The setting determines the new postscaler division ratio per Table 9-1.

**Note 1:** The default value (f/f) is set equal to the RSTOSC Configuration bits.
**2:** If NOSC is written with a reserved value (Table 9-1), the operation is ignored and neither NOSC nor NDIV is written.
**3:** When CSWEN = 0, this register is read-only and cannot be changed from the POR value.
**4:** When NOSC = 110 (HFINTOSC 4 MHz), the NDIV bits will default to ‘0010’ upon Reset; for all other NOSC settings the NDIV bits will default to ‘0000’ upon Reset.

### REGISTER 9-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

| U-0 | R-n/n(2) | R-n/n(2) | R-n/n(2) | R-n/n(2) | R-n/n(2) | R-n/n(2) | R-n/n(2) |
|---|---|---|---|---|---|---|---|
| — | COSC<2:0> | | | CDIV<3:0> | | | |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7 **Unimplemented:** Read as ‘0’

bit 6-4 **COSC<2:0>:** Current Oscillator Source Select bits (read-only)
Indicates the current source oscillator and PLL combination per Table 9-1.

bit 3-0 **CDIV<3:0>:** Current Divider Select bits (read-only)
Indicates the current postscaler division ratio per Table 9-1.

**Note 1:** The POR value is the value present when user code execution begins.
**2:** The Reset value (n/n) is the same as the NOSC/NDIV bits.

**REGISTER 9-4: OSCSTAT: OSCILLATOR STATUS REGISTER 1**

| R-q/q | R-q/q | R-q/q | R-q/q | U-0 | R-q/q | U-0 | R-q/q |
|---|---|---|---|---|---|---|---|
| EXTOR | HFOR | MFOR | LFOR | — | ADOR | — | PLLR |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7 **EXTOR:** EXTOSC (external) Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 6 **HFOR:** HFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 5 **MFOR:** MFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 4 **LFOR:** LFINTOSC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 3 **Unimplemented:** Read as ‘0’

bit 2 **ADOR:** CRC Oscillator Ready bit
1 = The oscillator is ready to be used
0 = The oscillator is not enabled, or is not yet ready to be used.

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **PLLR:** PLL is Ready bit
1 = The PLL is ready to be used
0 = The PLL is not enabled, the required input source is not ready, or the PLL is not locked.

**REGISTER 10-3: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

| R/W-0/0 | R/W-0/0 | U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| OSFIE | CSWIE | — | — | — | — | — | ADIE |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7 **OSFIE:** Oscillator Fail Interrupt Enable bit
1 = Enables the Oscillator Fail Interrupt
0 = Disables the Oscillator Fail Interrupt

bit 6 **CSWIE:** Clock Switch Complete Interrupt Enable bit
1 = The clock switch module interrupt is enabled
0 = The clock switch module interrupt is disabled

bit 5-1 **Unimplemented:** Read as ‘0’

bit 0 **ADIE:** Analog-to-Digital Converter (ADC) Interrupt Enable bit
1 = Enables the ADC interrupt
0 = Disables the ADC interrupt

**Note:** Bit PEIE of the INTCON register must be set to enable any peripheral interrupt controlled by registers PIE1-PIE7

## 15.8 Register Definitions: PPS Input Selection

**REGISTER 15-1: xxxPPS: PERIPHERAL xxx INPUT SELECTION(1)**

| U-0 | U-0 | R/W-q/u | R/W-q/u | R/W/q/u | R/W-q/u | R/W-q/u | R/W-q/u |
|---|---|---|---|---|---|---|---|
| — | — | xxxPPS<5:0> | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = value depends on peripheral |

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-0 **xxxPPS<5:0>:** Peripheral xxx Input Selection bits
See Table 15-1 and Table 15-2.

**Note 1:** The “xxx” in the register name “xxxPPS” represents the input signal function name, such as “INT”, “T0CKI”, “RX”, etc. This register summary shown here is only a prototype of the array of actual registers, as each input function has its own dedicated SFR (ex: INTPPS, T0CKIPPS, RXPPS, etc.).

**2:** Each specific input signal may only be mapped to a subset of these I/O pins, as shown in Table 15-1 and Table 15-2. Attempting to map an input signal to a non-supported I/O pin will result in undefined behavior. For example, the “INT” signal map be mapped to any PORTA or PORTB pin. Therefore, the INTPPS register may be written with values from 0x00-0x0F (corresponding to RA0-RB7). Attempting to write 0x10 or higher to the INTPPS register is not supported and will result in undefined behavior.

**REGISTER 16-3: PMD2: PMD CONTROL REGISTER 2**

| U-0 | R/W-0/0 | R/W-0/0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | DAC1MD | ADCMD | — | — | CMP2MD | CMP1MD | ZCDMD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **DAC1MD:** Disable DAC1 bit
1 = DAC module disabled
0 = DAC module enabled

bit 5 **ADCMD:** Disable ADC bit
1 = ADC module disabled
0 = ADC module enabled

bit 4-3 **Unimplemented:** Read as ‘0’

bit 2 **CMP2MD:** Disable Comparator C2 bit(1)
1 = C2 module disabled
0 = C2 module enabled

bit 1 **CMP1MD:** Disable Comparator C1 bit
1 = C1 module disabled
0 = C1 module enabled

bit 0 **ZCDMD:** Disable ZCD bit
1 = ZCD module disabled
0 = ZCD module enabled

**Note 1:** Present only on PIC16(L)F15323.

**REGISTER 16-5: PMD4: PMD CONTROL REGISTER 4**

| U-0 | R/W-0/0 | U-0 | R/W-0/0 | U-0 | U-0 | U-0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | UART1MD | — | MSSP1MD | — | — | — | CWG1MD |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | q = Value depends on condition |

bit 7 **Unimplemented:** Read as ‘0’

bit 6 **UART1MD:** Disable EUSART1 bit
1 = EUSART1 module disabled
0 = EUSART1 module enabled

bit 5 **Unimplemented:** Read as ‘0’

bit 4 **MSSP1MD:** Disable MSSP1 bit
1 = MSSP1 module disabled
0 = MSSP1 module enabled

bit 3-1 **Unimplemented:** Read as ‘0’

bit 0 **CWG1MD:** Disable CWG1 bit
1 = CWG1 module disabled
0 = CWG1 module enabled

**REGISTER 17-3: IOCAF: INTERRUPT-ON-CHANGE PORTA FLAG REGISTER**

| U-0 | U-0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 | R/W/HS-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCAF5 | IOCAF4 | IOCAF3 | IOCAF2 | IOCAF1(1) | IOCAF0(1) |
| bit 7 | | | | | | | bit 0 |

| Legend: | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HS - Bit is set in hardware |

bit 7-6 **Unimplemented:** read as '0'

bit 5-0 **IOCAF<5:0>:** Interrupt-on-Change PORTA Flag bits
1 = An enabled change was detected on the associated pin.
Set when IOCAPx = 1 and a rising edge was detected on RAx, or when IOCANx = 1 and a falling edge was detected on RAx.
0 = No change was detected, or the user cleared the detected change.

**Note 1:** If the debugger is enabled, these bits are not available for use.

**REGISTER 17-6: IOCCF: INTERRUPT-ON-CHANGE PORTC FLAG REGISTER(1)**

| U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | IOCCF5 | IOCCF4 | IOCCF3 | IOCCF2 | IOCCF1 | IOCCF0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HS - Bit is set in hardware |

bit 7-0 **IOCCF<7:0>:** Interrupt-on-Change PORTC Flag bits
1 = An enabled change was detected on the associated pin
Set when IOCCPx = 1 and a rising edge was detected on RCx, or when IOCCNx = 1 and a falling edge was detected on RCx.
0 = No change was detected, or the user cleared the detected change

**Note 1:** Present only on PIC16(L)F15323.

## 20.1 ADC Configuration

When configuring and using the ADC the following functions must be considered:

- Port configuration
- Channel selection
- ADC voltage reference selection
- ADC conversion clock source
- Interrupt control
- Result formatting

### 20.1.1 PORT CONFIGURATION

The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin will be configured for analog by setting the associated TRIS and ANSEL bits. Refer to **Section 14.0 "I/O Ports"** for more information.

| | |
|---|---|
| **Note:** | Analog voltages on any pin that is defined as a digital input may cause the input buffer to conduct excess current. |

### 20.1.2 CHANNEL SELECTION

There are several channel selections available:

- Six Port A channels
- Six Port C channels (PIC16(L)F15323 only)
- Temperature Indicator
- DAC output
- Fixed Voltage Reference (FVR)
- AVSS (Ground)

The CHS<5:0> bits of the ADCON0 register (Register 20-1) determine which channel is connected to the sample and hold circuit.

When changing channels, a delay is required before starting the next conversion. Refer to **Section 20.2 "ADC Operation"** for more information.

| | |
|---|---|
| **Note:** | It is recommended that when switching from an ADC channel of a higher voltage to a channel of a lower voltage, that the user selects the VSS channel before connecting to the channel with the lower voltage. If the ADC does not have a dedicated VSS input channel, the VSS selection (DAC1R<4:0> = b'00000') through the DAC output channel can be used. If the DAC is in use, a free input channel can be connected to VSS, and can be used in place of the DAC. |

### 20.1.3 ADC VOLTAGE REFERENCE

The ADPREF<1:0> bits of the ADCON1 register provides control of the positive voltage reference. The positive voltage reference can be:

- VREF+ pin
- VDD
- FVR 2.048V
- FVR 4.096V (Not available on LF devices)

The ADPREF bit of the ADCON1 register provides control of the negative voltage reference. The negative voltage reference can be:

- VREF- pin
- VSS

See **Section 18.0 "Fixed Voltage Reference (FVR)"** for more details on the Fixed Voltage Reference.

### 20.1.4 CONVERSION CLOCK

The source of the conversion clock is software selectable via the ADCS<2:0> bits of the ADCON1 register. There are seven possible clock options:

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- ADCRC (dedicated RC oscillator)

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11.5 TAD periods as shown in Figure 20-2.

For correct conversion, the appropriate TAD specification must be met. Refer to Table 37-13 for more information. Table 20-1 gives examples of appropriate ADC clock selections.

| | |
|---|---|
| **Note:** | Unless using the ADCRC, any changes in the system clock frequency will change the ADC clock frequency, which may adversely affect the ADC result. |

**REGISTER 22-3: NCO1ACCL: NCO1 ACCUMULATOR REGISTER – LOW BYTE**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| NCO1ACC<7:0> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-0 **NCO1ACC<7:0>:** NCO1 Accumulator, Low Byte

**REGISTER 22-4: NCO1ACCH: NCO1 ACCUMULATOR REGISTER – HIGH BYTE**

| R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| NCO1ACC<15:8> | | | | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-0 **NOC1ACC<15:8>:** NCO1 Accumulator, High Byte

**REGISTER 22-5: NCO1ACCU: NCO1 ACCUMULATOR REGISTER – UPPER BYTE(1)**

| U-0 | U-0 | U-0 | U-0 | R/W-0/0 | R/W-0/0 | R/W-0/0 | R/W-0/0 |
|---|---|---|---|---|---|---|---|
| — | — | — | — | NCO1ACC<19:16> | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| ‘1’ = Bit is set | ‘0’ = Bit is cleared | |

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **NCO1ACC<19:16>:** NCO1 Accumulator, Upper Byte

**Note 1:** The accumulator spans registers NCO1ACCU:NCO1ACCH: NCO1ACCL. The 24 bits are reserved but not all are used.This register updates in real-time, asynchronously to the CPU; there is no provision to guarantee atomic access to this 24-bit space using an 8-bit bus. Writing to this register while the module is operating will produce undefined results.

**REGISTER 26-2: T1GCON: TIMER1 GATE CONTROL REGISTER**

| R/W-0/u | R/W-0/u | R/W-0/u | R/W-0/u | R/W/HC-0/u | R-x/x | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| GE | GPOL | GTM | GSPM | GGO/$\overline{\text{DONE}}$ | GVAL | — | — |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | HC = Bit is cleared by hardware |

bit 7 **GE:** Timer1 Gate Enable bit

If ON = 0:
This bit is ignored
If ON = 1:
1 = Timer1 counting is controlled by the Timer1 gate function
0 = Timer1 is always counting

bit 6 **GPOL:** Timer1 Gate Polarity bit

1 = Timer1 gate is active-high (Timer1 counts when gate is high)
0 = Timer1 gate is active-low (Timer1 counts when gate is low)

bit 5 **GTM:** Timer1 Gate Toggle Mode bit

1 = Timer1 Gate Toggle mode is enabled
0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared
Timer1 gate flip-flop toggles on every rising edge.

bit 4 **GSPM:** Timer1 Gate Single-Pulse Mode bit

1 = Timer1 Gate Single-Pulse mode is enabled
0 = Timer1 Gate Single-Pulse mode is disabled

bit 3 **GGO/$\overline{\text{DONE}}$:** Timer1 Gate Single-Pulse Acquisition Status bit

1 = Timer1 gate single-pulse acquisition is ready, waiting for an edge
0 = Timer1 gate single-pulse acquisition has completed or has not been started
This bit is automatically cleared when GSPM is cleared

bit 2 **GVAL:** Timer1 Gate Value Status bit

Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L
Unaffected by Timer1 Gate Enable (GE)

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

### 28.3.4 TIMER2 TIMER RESOURCE

This device has a newer version of the Timer2 module that has many new modes, which allow for greater customization and control of the PWM signals than on older parts. Refer to **Section 27.5 "Operation Examples"** for examples of PWM signal generation using the different modes of Timer2. The CCP operation requires that the timer used as the PWM time base has the FOSC/4 clock source selected

### 28.3.5 PWM PERIOD

The PWM period is specified by the PR2 register of Timer2. The PWM period can be calculated using the formula of Equation 28-1.

**EQUATION 28-1: PWM PERIOD**

$$PWM\ Period = [(PR2) + 1] \bullet 4 \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**Note 1:** TOSC = 1/FOSC

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCPx pin is set. (Exception: If the PWM duty cycle = 0%, the pin will not be set.)
- The PWM duty cycle is transferred from the CCPRxL/H register pair into a 10-bit buffer.

| **Note:** | The Timer postscaler (see **Section 27.4 "Timer2 Interrupt"**) is not used in the determination of the PWM frequency. |
|---|---|

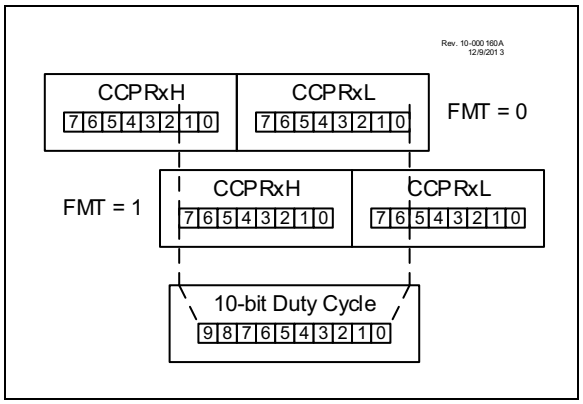
### 28.3.6 PWM DUTY CYCLE

The PWM duty cycle is specified by writing a 10-bit value to the CCPRxH:CCPRxL register pair. The alignment of the 10-bit value is determined by the CCPRxFMT bit of the CCPxCON register (see Figure 28-5). The CCPRxH:CCPRxL register pair can be written to at any time; however the duty cycle value is not latched into the 10-bit buffer until after a match between PR2 and TMR2.

Equation 28-2 is used to calculate the PWM pulse width.

Equation 28-3 is used to calculate the PWM duty cycle ratio.

**FIGURE 28-5: PWM 10-BIT ALIGNMENT**




**EQUATION 28-2: PULSE WIDTH**

$$Pulse\ Width = (CCPRxH{:}CCPRxL\ register\ pair) \bullet T_{OSC} \bullet (TMR2\ Prescale\ Value)$$

**EQUATION 28-3: DUTY CYCLE RATIO**

$$Duty\ Cycle\ Ratio = \frac{(CCPRxH{:}CCPRxL\ register\ pair)}{4(PR2 + 1)}$$

CCPRxH:CCPRxL register pair are used to double buffer the PWM duty cycle. This double buffering provides for glitchless PWM operation.

The 8-bit timer TMR2 register is concatenated with either the 2-bit internal system clock (FOSC), or two bits of the prescaler, to create the 10-bit time base. The system clock is used if the Timer2 prescaler is set to 1:1.

When the 10-bit time base matches the CCPRxH:CCPRxL register pair, then the CCPx pin is cleared (see Figure 28-4).

### 28.3.7 PWM RESOLUTION

The resolution determines the number of available duty cycles for a given period. For example, a 10-bit resolution will result in 1024 discrete duty cycles, whereas an 8-bit resolution will result in 256 discrete duty cycles.

The maximum PWM resolution is ten bits when PR2 is 255. The resolution is a function of the PR2 register value as shown by Equation 28-4.

**EQUATION 28-4: PWM RESOLUTION**

$$Resolution = \frac{log[4(PR2 + 1)]}{log(2)}\ bits$$

| **Note:** | If the pulse width value is greater than the period the assigned PWM pin(s) will remain unchanged. |
|---|---|

## 30.12 Configuring the CWG

The following steps illustrate how to properly configure the CWG.

1. Ensure that the TRIS control bits corresponding to the desired CWG pins for your application are set so that the pins are configured as inputs.
2. Clear the EN bit, if not already cleared.
3. Set desired mode of operation with the MODE bits.
4. Set desired dead-band times, if applicable to mode, with the CWG1DBR and CWG1DBF registers.
5. Setup the following controls in the CWG1AS0 and CWG1AS1 registers.
   a. Select the desired shutdown source.
   b. Select both output overrides to the desired levels (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
   c. Set which pins will be affected by auto-shutdown with the CWG1AS1 register.
   d. Set the SHUTDOWN bit and clear the REN bit.
6. Select the desired input source using the CWG1ISM register.
7. Configure the following controls.
   a. Select desired clock source using the CWG1CLKCON register.
   b. Select the desired output polarities using the CWG1CON1 register.
   c. Set the output enables for the desired outputs.
8. Set the EN bit.
9. Clear TRIS control bits corresponding to the desired output pins to configure these pins as outputs.
10. If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit to start the CWG.

### 30.12.1 PIN OVERRIDE LEVELS

The levels driven to the output pins, while the shutdown input is true, are controlled by the LSBD and LSAC bits of the CWG1AS0 register. LSBD<1:0> controls the CWG1B and D override levels and LSAC<1:0> controls the CWG1A and C override levels. The control bit logic level corresponds to the output logic drive level while in the shutdown state. The polarity control does not affect the override level.

### 30.12.2 AUTO-SHUTDOWN RESTART

After an auto-shutdown event has occurred, there are two ways to resume operation:

- Software controlled
- Auto-restart

The restart method is selected with the REN bit of the CWG1CON2 register. Waveforms of software controlled and automatic restarts are shown in Figure 30-13 and Figure 30-14.

#### 30.12.2.1 Software Controlled Restart

When the REN bit of the CWG1AS0 register is cleared, the CWG must be restarted after an auto-shutdown event by software. Clearing the shutdown state requires all selected shutdown inputs to be low, otherwise the SHUTDOWN bit will remain set. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

#### 30.12.2.2 Auto-Restart

When the REN bit of the CWG1CON2 register is set, the CWG will restart from the auto-shutdown state automatically. The SHUTDOWN bit will clear automatically when all shutdown sources go low. The overrides will remain in effect until the first rising edge event after the SHUTDOWN bit is cleared. The CWG will then resume operation.

# 32.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP1) MODULE

## 32.1 MSSP Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

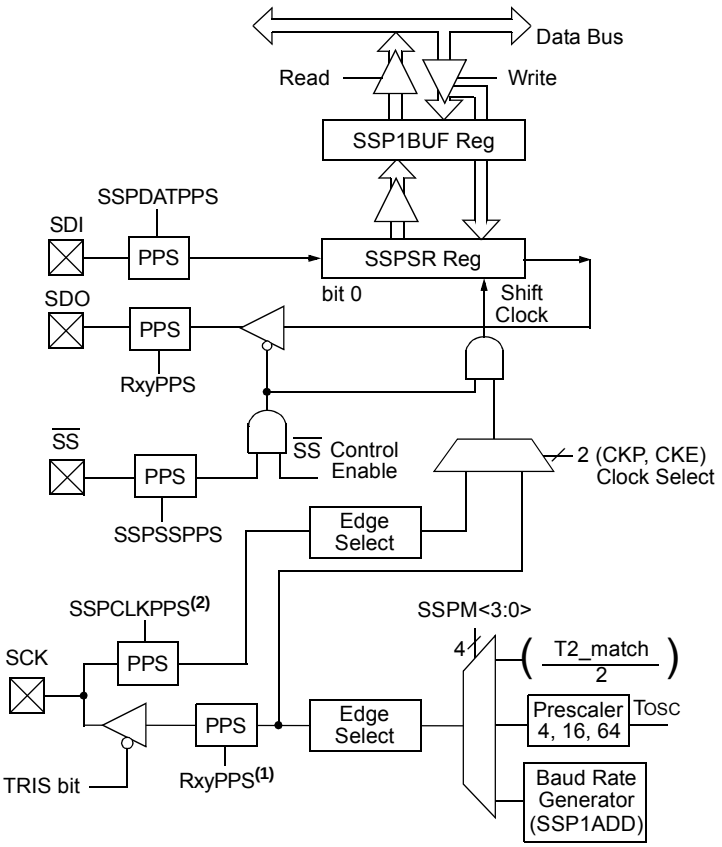
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit ($I^2C$)

The SPI interface supports the following modes and features:

- Master mode
- Slave mode
- Clock Parity
- Slave Select Synchronization (Slave mode only)
- Daisy-chain connection of slave devices

Figure 32-1 is a block diagram of the SPI interface module.

**FIGURE 32-1: MSSP BLOCK DIAGRAM (SPI MODE)**



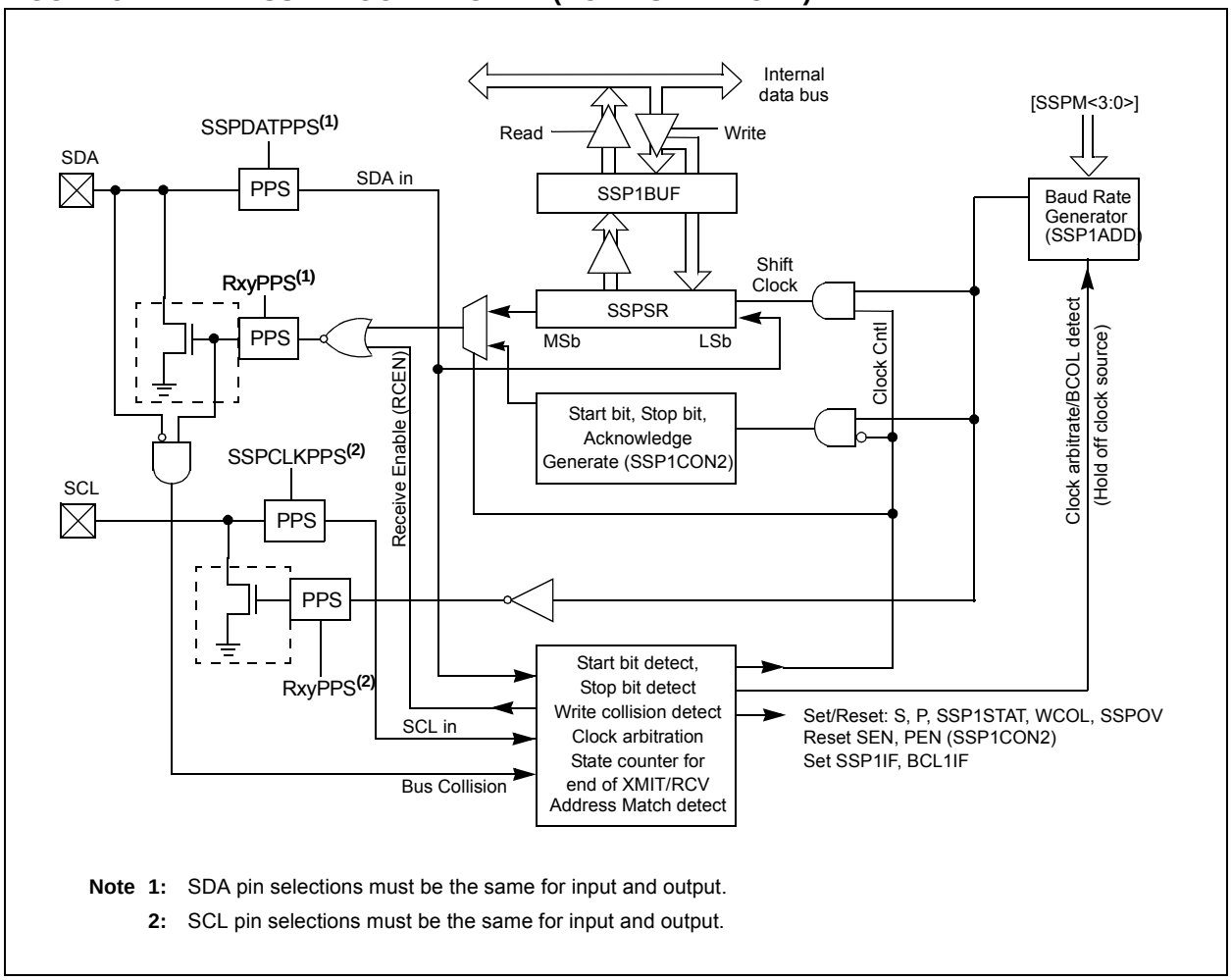

**Note 1:** Output selection for master mode.
**2:** Input selection for slave mode.

The $I^2C$ interface supports the following modes and features:

- Master mode
- Slave mode
- Byte NACKing (Slave mode)
- Limited multi-master support
- 7-bit and 10-bit addressing
- Start and Stop interrupts
- Interrupt masking
- Clock stretching
- Bus collision detection
- General call address matching
- Address masking
- Selectable SDA hold times

Figure 32-2 is a block diagram of the $I^2C$ interface module in Master mode. Figure 32-3 is a diagram of the $I^2C$ interface module in Slave mode.

**FIGURE 32-2: MSSP BLOCK DIAGRAM ($I^2C$ MASTER MODE)**




**Note 1:** SDA pin selections must be the same for input and output.

**2:** SCL pin selections must be the same for input and output.

**BCF** **Bit Clear f**

| | |
|---|---|
| Syntax: | [ *label* ] BCF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 0 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is cleared. |

**BRA** **Relative Branch**

| | |
|---|---|
| Syntax: | [ *label* ] BRA label<br>[ *label* ] BRA $+k |
| Operands: | -256 ≤ label - PC + 1 ≤ 255<br>-256 ≤ k ≤ 255 |
| Operation: | (PC) + 1 + k → PC |
| Status Affected: | None |
| Description: | Add the signed 9-bit literal 'k' to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + k. This instruction is a 2-cycle instruction. This branch has a limited range. |

**BRW** **Relative Branch with W**

| | |
|---|---|
| Syntax: | [ *label* ] BRW |
| Operands: | None |
| Operation: | (PC) + (W) → PC |
| Status Affected: | None |
| Description: | Add the contents of W (unsigned) to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 1 + (W). This instruction is a 2-cycle instruction. |

**BSF** **Bit Set f**

| | |
|---|---|
| Syntax: | [ *label* ] BSF f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | 1 → (f<b>) |
| Status Affected: | None |
| Description: | Bit 'b' in register 'f' is set. |

**BTFSC** **Bit Test f, Skip if Clear**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSC f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b ≤ 7 |
| Operation: | skip if (f<b>) = 0 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '1', the next instruction is executed.<br>If bit 'b', in register 'f', is '0', the next instruction is discarded, and a NOP is executed instead, making this a 2-cycle instruction. |

**BTFSS** **Bit Test f, Skip if Set**

| | |
|---|---|
| Syntax: | [ *label* ] BTFSS f,b |
| Operands: | 0 ≤ f ≤ 127<br>0 ≤ b < 7 |
| Operation: | skip if (f<b>) = 1 |
| Status Affected: | None |
| Description: | If bit 'b' in register 'f' is '0', the next instruction is executed.<br>If bit 'b' is '1', then the next instruction is discarded and a NOP is executed instead, making this a 2-cycle instruction. |

**TABLE 37-13: ANALOG-TO-DIGITAL CONVERTER (ADC) CONVERSION TIMING SPECIFICATIONS**

| Standard Operating Conditions (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Sym.** | **Characteristic** | **Min.** | **Typ†** | **Max.** | **Units** | **Conditions** |
| AD20 | TAD | ADC Clock Period | 1 | — | 9 | µs | The requirement is to set ADCCS correctly to produce this period/frequency. |
| AD21 | | | 1 | 2 | 6 | µs | Using FRC as the ADC clock source ADOSC = 1 |
| AD22 | TCNV | Conversion Time | — | 11 | — | TAD | Set of GO/DONE bit to Clear of GO/DONE bit |
| AD23 | TACQ | Acquisition Time | — | 2 | — | µs | |
| AD24 | THCD | Sample and Hold Capacitor Disconnect Time | — | — | — | µs | FOSC-based clock source<br>FRC-based clock source |

* These parameters are characterized but not tested.

† Data in "Typ" column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**FIGURE 37-10: ADC CONVERSION TIMING (ADC CLOCK FOSC-BASED)**

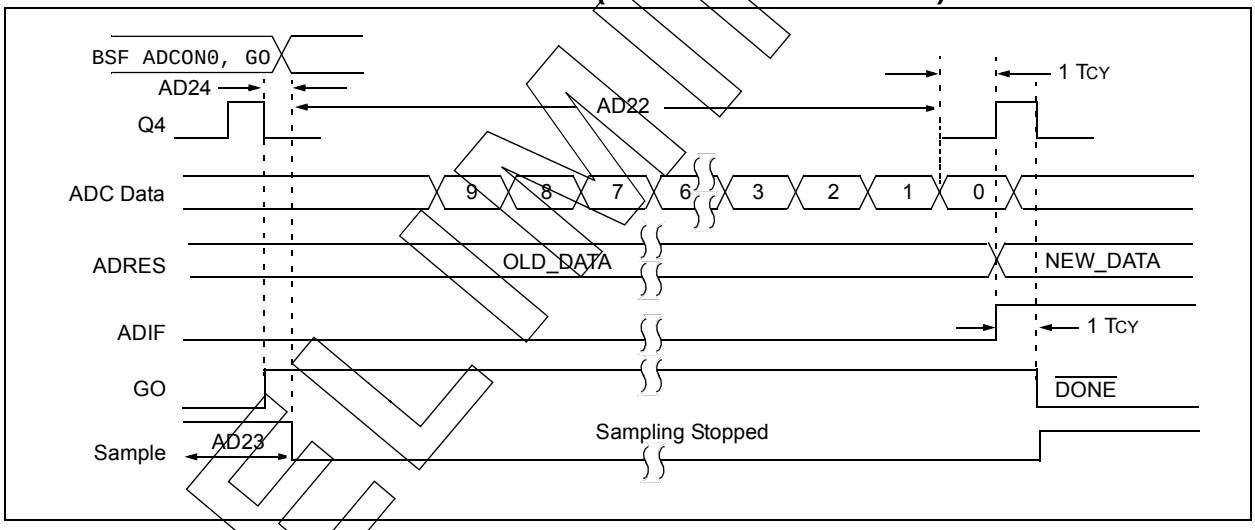
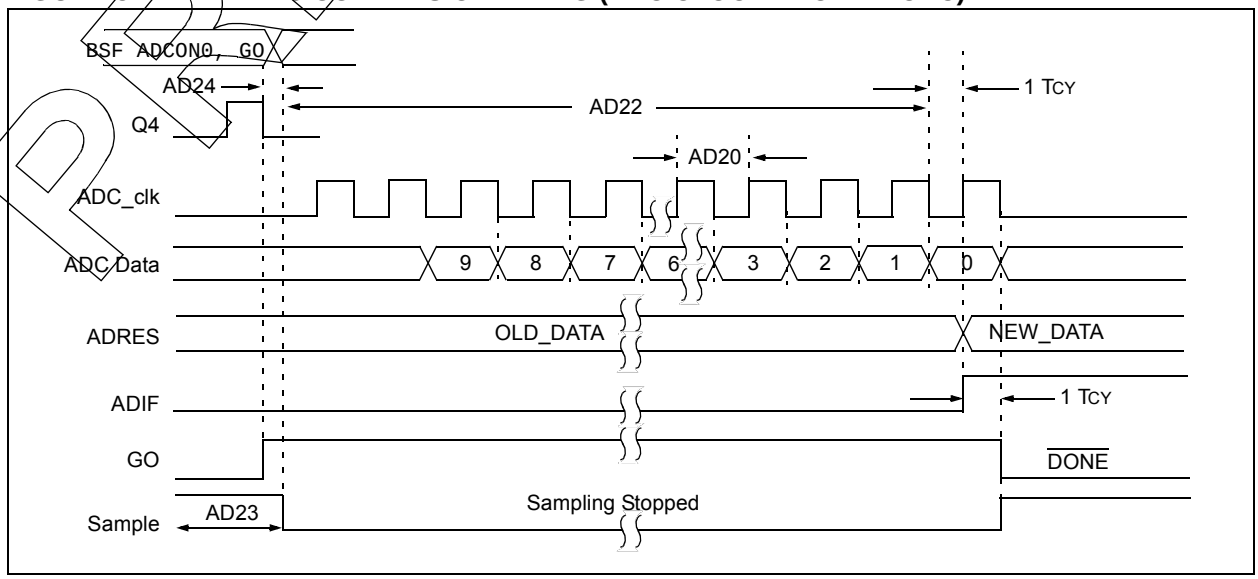
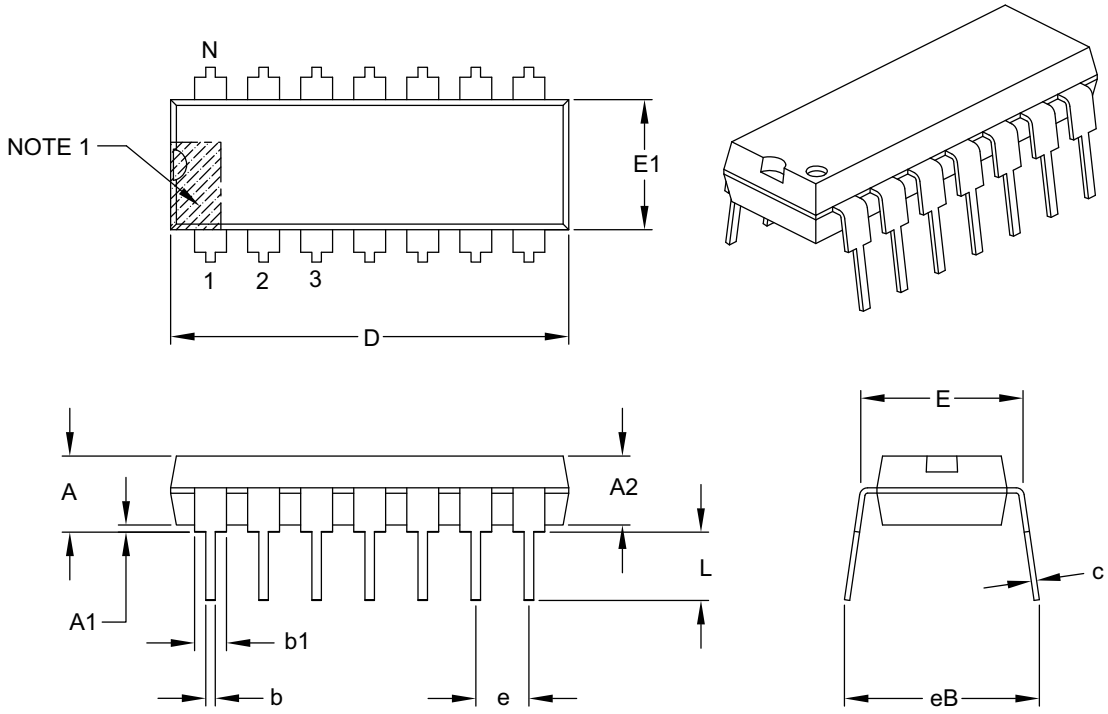



**FIGURE 37-11: ADC CONVERSION TIMING (ADC CLOCK FROM ADCRC)**



### 14-Lead Plastic Dual In-Line (P) – 300 mil Body [PDIP]

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




| Units | | INCHES | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 14 | |
| Pitch | e | | .100 BSC | |
| Top to Seating Plane | A | – | – | .210 |
| Molded Package Thickness | A2 | .115 | .130 | .195 |
| Base to Seating Plane | A1 | .015 | – | – |
| Shoulder to Shoulder Width | E | .290 | .310 | .325 |
| Molded Package Width | E1 | .240 | .250 | .280 |
| Overall Length | D | .735 | .750 | .775 |
| Tip to Seating Plane | L | .115 | .130 | .150 |
| Lead Thickness | c | .008 | .010 | .015 |
| Upper Lead Width | b1 | .045 | .060 | .070 |
| Lower Lead Width | b | .014 | .018 | .022 |
| Overall Row Spacing § | eB | – | – | .430 |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located with the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-005B