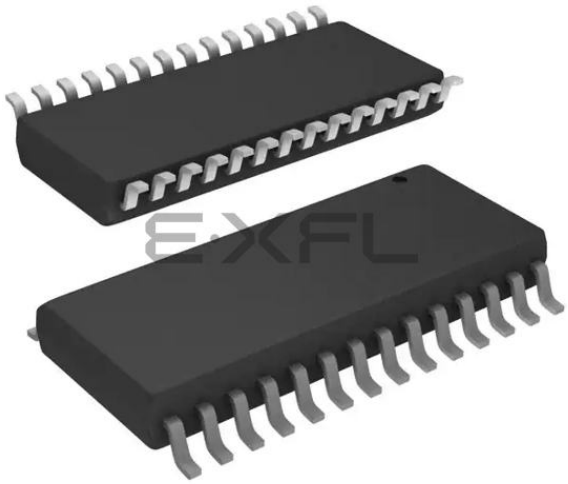


**Welcome to E-XFL.COM**

## 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 | 20 MIPS |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, Motor Control PWM, QEI, POR, PWM, WDT |
| Number of I/O | 20 |
| Program Memory Size | 12KB (4K x 24) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 512 x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.5V ~ 5.5V |
| Data Converters | A/D 6x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SOIC (0.295", 7.50mm Width) |
| Supplier Device Package | 28-SOIC |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic30f2010-20e-so |

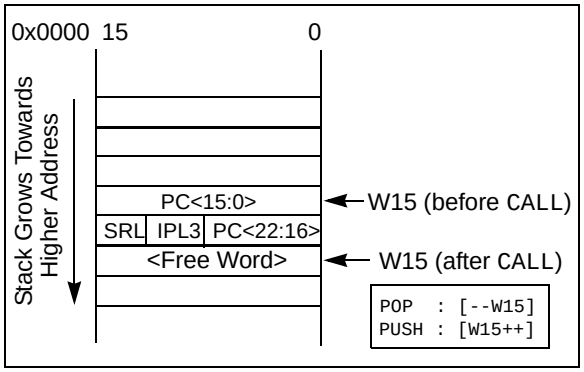
## 5.4 Interrupt Sequence

All interrupt event flags are sampled in the beginning of each instruction cycle by the IFSx registers. A pending interrupt request (IRQ) is indicated by the flag bit being equal to a '1' in an IFSx register. The IRQ will cause an interrupt to occur if the corresponding bit in the interrupt enable (IECx) register is set. For the remainder of the instruction cycle, the priorities of all pending interrupt requests are evaluated.

If there is a pending IRQ with a priority level greater than the current processor priority level in the IPL bits, the processor will be interrupted.

The processor then stacks the current program counter and the low byte of the processor STATUS register (SRL), as shown in Figure 5-2. The low byte of the status register contains the processor priority level at the time, prior to the beginning of the interrupt cycle. The processor then loads the priority level for this interrupt into the STATUS register. This action will disable all lower priority interrupts until the completion of the Interrupt Service Routine (ISR).

**FIGURE 5-2: INTERRUPT STACK FRAME**




**Note 1:** The user can always lower the priority level by writing a new value into SR. The Interrupt Service Routine must clear the interrupt flag bits in the IFSx register before lowering the processor interrupt priority, in order to avoid recursive interrupts.

**2:** The IPL3 bit (CORCON<3>) is always clear when interrupts are being processed. It is set only during execution of traps.

The RETFIE (Return from Interrupt) instruction will unstack the program counter and status registers to return the processor to its state prior to the interrupt sequence.

## 5.5 Alternate Vector Table

In Program Memory, the Interrupt Vector Table (IVT) is followed by the Alternate Interrupt Vector Table (AIVT), as shown in Figure 5-1. Access to the Alternate Vector Table is provided by the ALTIVT bit in the INTCON2 register. If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors. The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment, without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time.

If the AIVT is not required, the program memory allocated to the AIVT may be used for other purposes. AIVT is not a protected section and may be freely programmed by the user.

## 5.6 Fast Context Saving

A context saving option is available using shadow registers. Shadow registers are provided for the DC, N, OV, Z and C bits in SR, and the registers W0 through W3. The shadows are only one level deep. The shadow registers are accessible using the PUSH.S and POP.S instructions only.

When the processor vectors to an interrupt, the PUSH.S instruction can be used to store the current value of the aforementioned registers into their respective shadow registers.

If an ISR of a certain priority uses the PUSH.S and POP.S instructions for fast context saving, then a higher priority ISR should not include the same instructions. Users must save the key registers in software during a lower priority interrupt, if the higher priority ISR uses fast context saving.

## 5.7 External Interrupt Requests

The interrupt controller supports five external interrupt request signals, INT0-INT4. These inputs are edge sensitive; they require a low-to-high or a high-to-low transition to generate an interrupt request. The INTCON2 register has three bits, INT0EP-INT2EP, that select the polarity of the edge detection circuitry.

## 5.8 Wake-up from Sleep and Idle

The interrupt controller may be used to wake up the processor from either Sleep or Idle modes, if Sleep or Idle mode is active when the interrupt is generated.

If an enabled interrupt request of sufficient priority is received by the interrupt controller, then the standard interrupt request is presented to the processor. At the same time, the processor will wake-up from Sleep or Idle and begin execution of the Interrupt Service Routine needed to process the interrupt request.

**NOTES:**

**TABLE 10-1: TIMER2/3 REGISTER MAP**

| SFR 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 | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TMR2 | 0106 | Timer2 Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| TMR3HLD | 0108 | Timer3 Holding Register (For 32-bit timer operations only) | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| TMR3 | 010A | Timer3 Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| PR2 | 010C | Period Register 2 | | | | | | | | | | | | | | | | 1111 1111 1111 1111 |
| PR3 | 010E | Period Register 3 | | | | | | | | | | | | | | | | 1111 1111 1111 1111 |
| T2CON | 0110 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS1 | TCKPS0 | T32 | — | TCS | — | 0000 0000 0000 0000 |
| T3CON | 0112 | TON | — | TSIDL | — | — | — | — | — | — | TGATE | TCKPS1 | TCKPS0 | — | — | TCS | — | 0000 0000 0000 0000 |

**Legend:** u = uninitialized bit; — = unimplemented bit, read as '0'

**Note:** Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

**NOTES:**

## 11.0 INPUT CAPTURE MODULE

**Note:** This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046).

This section describes the Input Capture module and associated operational modes. The features provided by this module are useful in applications requiring Frequency (Period) and Pulse measurement. Figure 11-1 depicts a block diagram of the Input Capture module. Input capture is useful for such modes as:

- Frequency/Period/Pulse Measurements
- Additional sources of External Interrupts

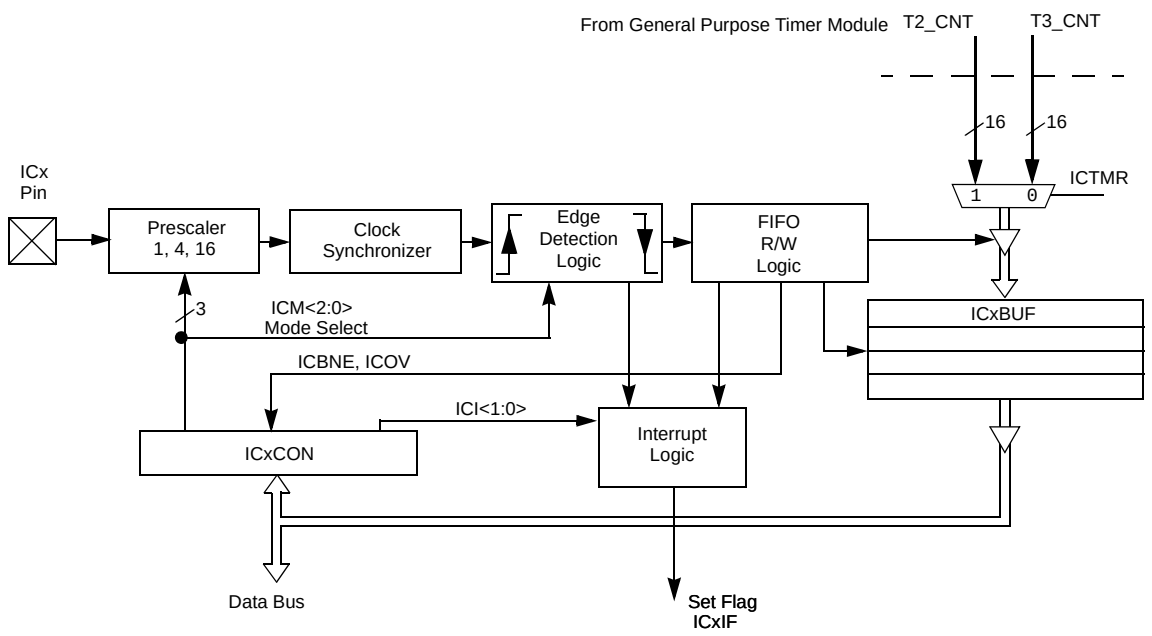
The key operational features of the Input Capture module are:

- Simple Capture Event mode
- Timer2 and Timer3 mode selection
- Interrupt on input capture event

These operating modes are determined by setting the appropriate bits in the ICxCON register (where x = 1,2,...,N). The dsPIC DSC devices contain up to eight capture channels, (i.e., the maximum value of N is 8).

**Note:** The dsPIC30F2010 device has four capture inputs – IC1, IC2, IC7 and IC8. The naming of these four capture channels is intentional and preserves software compatibility with other dsPIC DSC devices.

**FIGURE 11-1: INPUT CAPTURE MODE BLOCK DIAGRAM**




**Note:** Where 'x' is shown, reference is made to the registers or bits associated to the respective input capture channels 1 through N.

**TABLE 11-1: INPUT CAPTURE REGISTER MAP**

| SFR 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 | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IC1BUF | 0140 | Input 1 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC1CON | 0142 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC2BUF | 0144 | Input 2 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC2CON | 0146 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC3BUF | 0148 | Input 3 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC3CON | 014A | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC4BUF | 014C | Input 4 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC4CON | 014E | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC5BUF | 0150 | Input 5 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC5CON | 0152 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC6BUF | 0154 | Input 6 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC6CON | 0156 | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC7BUF | 0158 | Input 7 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC7CON | 015A | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |
| IC8BUF | 015C | Input 8 Capture Register | | | | | | | | | | | | | | | | uuuu uuuu uuuu uuuu |
| IC8CON | 015E | — | — | ICSIDL | — | — | — | — | — | ICTMR | ICI<1:0> | | ICOV | ICBNE | ICM<2:0> | | | 0000 0000 0000 0000 |

**Legend:** u = uninitialized bit; — = unimplemented bit, read as '0'

**Note:** Refer to the "*dsPIC30F Family Reference Manual*" (DS70046) for descriptions of register bit fields.

#### 17.3.4 TRANSMIT INTERRUPT

The transmit interrupt flag (U1TXIF or U2TXIF) is located in the corresponding interrupt flag register.

The transmitter generates an edge to set the UxTXIF bit. The condition for generating the interrupt depends on UTXISEL control bit:

a) If UTXISEL = 0, an interrupt is generated when a word is transferred from the Transmit buffer to the Transmit Shift register (UxTSR). This means that the transmit buffer has at least one empty word.
b) If UTXISEL = 1, an interrupt is generated when a word is transferred from the Transmit buffer to the Transmit Shift register (UxTSR) and the Transmit buffer is empty.

Switching between the two interrupt modes during operation is possible and sometimes offers more flexibility.

#### 17.3.5 TRANSMIT BREAK

Setting the UTXBRK bit (UxSTA<11>) will cause the UxTX line to be driven to logic '0'. The UTXBRK bit overrides all transmission activity. Therefore, the user should generally wait for the transmitter to be Idle before setting UTXBRK.

To send a break character, the UTXBRK bit must be set by software and must remain set for a minimum of 13 baud clock cycles. The UTXBRK bit is then cleared by software to generate Stop bits. The user must wait for a duration of at least one or two baud clock cycles in order to ensure a valid Stop bit(s) before reloading the UxTXB or starting other transmitter activity. Transmission of a break character does not generate a transmit interrupt.

### 17.4 Receiving Data

#### 17.4.1 RECEIVING IN 8-BIT OR 9-BIT DATA MODE

The following steps must be performed while receiving 8-bit or 9-bit data:

1. Set up the UART (see **Section 17.3.1 "Transmitting in 8-bit data mode"**).
2. Enable the UART (see **Section 17.3.1 "Transmitting in 8-bit data mode"**).
3. A receive interrupt will be generated when one or more data words have been received, depending on the receive interrupt settings specified by the URXISEL bits (UxSTA<7:6>).
4. Read the OERR bit to determine if an overrun error has occurred. The OERR bit must be reset in software.
5. Read the received data from UxRXREG. The act of reading UxRXREG will move the next word to the top of the receive FIFO, and the PERR and FERR values will be updated.

#### 17.4.2 RECEIVE BUFFER (UxRXB)

The receive buffer is 4 words deep. Including the Receive Shift register (UxRSR), the user effectively has a 5-word deep FIFO buffer.

URXDA (UxSTA<0>) = 1 indicates that the receive buffer has data available. URXDA = 0 implies that the buffer is empty. If a user attempts to read an empty buffer, the old values in the buffer will be read and no data shift will occur within the FIFO.

The FIFO is reset during any device Reset. It is not affected when the device enters or wakes up from a Power-Saving mode.

#### 17.4.3 RECEIVE INTERRUPT

The receive interrupt flag (U1RXIF) can be read from the corresponding interrupt flag register. The interrupt flag is set by an edge generated by the receiver. The condition for setting the receive interrupt flag depends on the settings specified by the URXISEL<1:0> (UxSTA<7:6>) control bits.

a) If URXISEL<1:0> = 00 or 01, an interrupt is generated every time a data word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer. There may be one or more characters in the receive buffer.
b) If URXISEL<1:0> = 10, an interrupt is generated when a word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer, which, as a result of the transfer, contains 3 characters.
c) If URXISEL<1:0> = 11, an interrupt is set when a word is transferred from the Receive Shift Register (UxRSR) to the Receive Buffer, which, as a result of the transfer, contains 4 characters (i.e., becomes full).

Switching between the Interrupt modes during operation is possible, though generally not advisable during normal operation.

### 17.5 Reception Error Handling

#### 17.5.1 RECEIVE BUFFER OVERRUN ERROR (OERR BIT)

The OERR bit (UxSTA<1>) is set if all of the following conditions occur:

a) The receive buffer is full.
b) The receive shift register is full, but unable to transfer the character to the receive buffer.
c) The Stop bit of the character in the UxRSR is detected, indicating that the UxRSR needs to transfer the character to the buffer.

Once OERR is set, no further data is shifted in UxRSR (until the OERR bit is cleared in software or a Reset occurs). The data held in UxRSR and UxRXREG remains valid.

### 17.5.2 FRAMING ERROR (FERR)

The FERR bit (UxSTA<2>) is set if a '0' is detected instead of a Stop bit. If two Stop bits are selected, both Stop bits must be '1', otherwise FERR will be set. The read-only FERR bit is buffered along with the received data. It is cleared on any Reset.

### 17.5.3 PARITY ERROR (PERR)

The PERR bit (UxSTA<3>) is set if the parity of the received word is incorrect. This error bit is applicable only if a Parity mode (odd or even) is selected. The read-only PERR bit is buffered along with the received data bytes. It is cleared on any Reset.

### 17.5.4 IDLE STATUS

When the receiver is active (i.e., between the initial detection of the Start bit and the completion of the Stop bit), the RIDLE bit (UxSTA<4>) is '0'. Between the completion of the Stop bit and detection of the next Start bit, the RIDLE bit is '1', indicating that the UART is Idle.

### 17.5.5 RECEIVE BREAK

The receiver will count and expect a certain number of bit times based on the values programmed in the PDSEL (UxMODE<2:1>) and STSEL (UxMODE<0>) bits.

If the break is longer than 13 bit times, the reception is considered complete after the number of bit times specified by PDSEL and STSEL. The URXDA bit is set, FERR is set, zeros are loaded into the receive FIFO, interrupts are generated, if appropriate, and the RIDLE bit is set.

When the module receives a long break signal and the receiver has detected the Start bit, the data bits and the invalid Stop bit (which sets the FERR), the receiver must wait for a valid Stop bit before looking for the next Start bit. It cannot assume that the break condition on the line is the next Start bit.

Break is regarded as a character containing all '0's, with the FERR bit set. The break character is loaded into the buffer. No further reception can occur until a Stop bit is received. Note that RIDLE goes high when the Stop bit has not been received yet.

## 17.6 Address Detect Mode

Setting the ADDEN bit (UxSTA<5>) enables this special mode, in which a 9th bit (URX8) value of '1' identifies the received word as an address rather than data. This mode is only applicable for 9-bit data communication. The URXISEL control bit does not have any impact on interrupt generation in this mode, since an interrupt (if enabled) will be generated every time the received word has the 9th bit set.

## 17.7 Loopback Mode

Setting the LPBACK bit enables this special mode in which the UxTX pin is internally connected to the UxRX pin. When configured for the Loopback mode, the UxRX pin is disconnected from the internal UART receive logic. However, the UxTX pin still functions as in a normal operation.

To select this mode:

a) Configure UART for desired mode of operation.
b) Set LPBACK = 1 to enable Loopback mode.
c) Enable transmission as defined in **Section 17.3 "Transmitting Data"**.

## 17.8 Baud Rate Generator

The UART has a 16-bit Baud Rate Generator to allow maximum flexibility in baud rate generation. The Baud Rate Generator register (UxBRG) is readable and writable. The baud rate is computed as follows:

BRG = 16-bit value held in UxBRG register (0 through 65535)

$F_{CY}$ = Instruction Clock Rate ($1/T_{CY}$)

The Baud Rate is given by Equation 17-1.

**EQUATION 17-1: BAUD RATE**

$$Baud\ Rate = F_{CY}/(16 * (BRG + 1))$$

Therefore, maximum baud rate possible is

$F_{CY}$/16 (if BRG = 0),

and the minimum baud rate possible is

$F_{CY}$/(16 * 65536).

With a full 16-bit Baud Rate Generator, at 30 MIPs operation, the minimum baud rate achievable is 28.5 bps.

## 17.9 Auto Baud Support

To allow the system to determine baud rates of received characters, the input can be optionally linked to a selected capture input. To enable this mode, the user must program the input capture module to detect the falling and rising edges of the Start bit.

## 18.7 A/D Conversion Speeds

The dsPIC30F 10-bit ADC specifications permit a maximum 1 Msps sampling rate. Table 18-1 summarizes the conversion speeds for the dsPIC30F 10-bit ADC and the required operating conditions.

The configuration guidelines give the required setup values for the conversion speeds above 500 ksps, since they require external VREF pins usage and there are some differences in the configuration procedure. Configuration details that are not critical to the conversion speed have been omitted.

Figure 18-2 depicts the recommended circuit for the conversion rates above 500 ksps.

**FIGURE 18-2: ADC VOLTAGE REFERENCE SCHEMATIC**

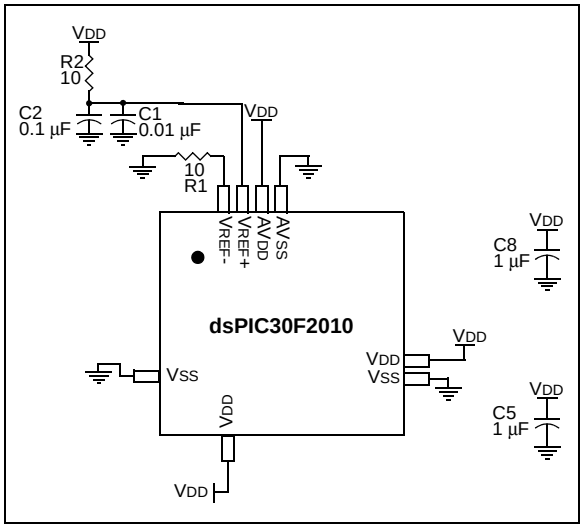
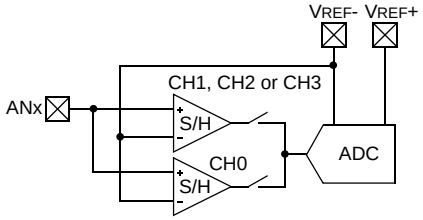
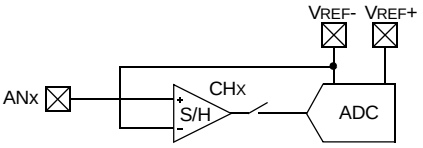
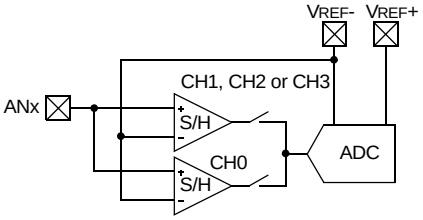
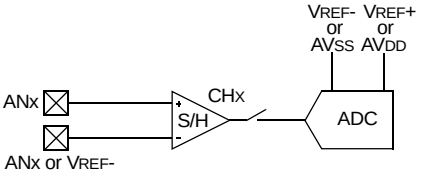
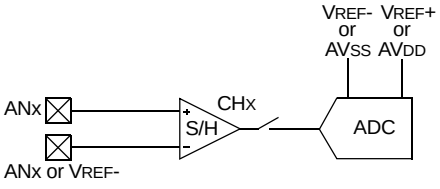


**TABLE 18-1: 10-BIT A/D CONVERSION RATE PARAMETERS**

| dsPIC30F 10-bit A/D Converter Conversion Rates | | | | | | |
|---|---|---|---|---|---|---|
| **A/D Speed** | **TAD Minimum** | **Sampling Time Min** | **RS Max** | **VDD** | **Temperature** | **A/D Channels Configuration** |
| Up to 1 Msps(1) | 83.33 ns | 12 TAD | 500Ω | 4.5V to 5.5V | -40°C to +85°C |  |
| Up to 750 ksps(1) | 95.24 ns | 2 TAD | 500Ω | 4.5V to 5.5V | -40°C to +85°C |  |
| Up to 600 ksps(1) | 138.89 ns | 12 TAD | 500Ω | 3.0V to 5.5V | -40°C to +125°C |  |
| Up to 500 ksps | 153.85 ns | 1 TAD | 5.0 kΩ | 4.5V to 5.5V | -40°C to +125°C |  |
| Up to 300 ksps | 256.41 ns | 1 TAD | 5.0 kΩ | 3.0V to 5.5V | -40°C to +125°C |  |

**Note 1:** External VREF- and VREF+ pins must be used for correct operation. See Figure 18-2 for recommended circuit.

Table 19-5 shows the Reset conditions for the RCON Register. Since the control bits within the RCON register are R/W, the information in the table implies that all the bits are negated prior to the action specified in the condition column.

**TABLE 19-5: INITIALIZATION CONDITION FOR RCON REGISTER CASE 1**

| Condition | Program Counter | TRAPR | IOPUWR | EXTR | SWR | WDTO | IDLE | SLEEP | POR | BOR |
|---|---|---|---|---|---|---|---|---|---|---|
| Power-on Reset | 0x000000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Brown-out Reset | 0x000000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\overline{\text{MCLR}}$ Reset during normal operation | 0x000000 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Software Reset during normal operation | 0x000000 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0x000000 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| $\overline{\text{MCLR}}$ Reset during Idle | 0x000000 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| WDT Time-out Reset | 0x000000 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| WDT Wake-up | PC + 2 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Interrupt Wake-up from Sleep | PC + 2(1) | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Clock Failure Trap | 0x000004 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Trap Reset | 0x000000 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Illegal Operation Trap | 0x000000 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Note 1:** When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

Table 19-6 shows a second example of the bit conditions for the RCON Register. In this case, it is not assumed the user has set/cleared specific bits prior to action specified in the condition column.

**TABLE 19-6: INITIALIZATION CONDITION FOR RCON REGISTER CASE 2**

| Condition | Program Counter | TRAPR | IOPUWR | EXTR | SWR | WDTO | IDLE | SLEEP | POR | BOR |
|---|---|---|---|---|---|---|---|---|---|---|
| Power-on Reset | 0x000000 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Brown-out Reset | 0x000000 | u | u | u | u | u | u | u | 0 | 1 |
| $\overline{\text{MCLR}}$ Reset during normal operation | 0x000000 | u | u | 1 | 0 | 0 | 0 | 0 | u | u |
| Software Reset during normal operation | 0x000000 | u | u | 0 | 1 | 0 | 0 | 0 | u | u |
| $\overline{\text{MCLR}}$ Reset during Sleep | 0x000000 | u | u | 1 | u | 0 | 0 | 1 | u | u |
| $\overline{\text{MCLR}}$ Reset during Idle | 0x000000 | u | u | 1 | u | 0 | 1 | 0 | u | u |
| WDT Time-out Reset | 0x000000 | u | u | 0 | 0 | 1 | 0 | 0 | u | u |
| WDT Wake-up | PC + 2 | u | u | u | u | 1 | u | 1 | u | u |
| Interrupt Wake-up from Sleep | PC + 2(1) | u | u | u | u | u | u | 1 | u | u |
| Clock Failure Trap | 0x000004 | u | u | u | u | u | u | u | u | u |
| Trap Reset | 0x000000 | 1 | u | u | u | u | u | u | u | u |
| Illegal Operation Reset | 0x000000 | u | 1 | u | u | u | u | u | u | u |

**Legend:** u = unchanged

**Note 1:** When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

**TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Base Instr # | Assembly Mnemonic | Assembly Syntax | | Description | # of words | # of cycles | Status Flags Affected |
|---|---|---|---|---|---|---|---|
| 11 | BTSS | BTSS | f,#bit4 | Bit Test f, Skip if Set | 1 | 1 (2 or 3) | None |
| | | BTSS | Ws,#bit4 | Bit Test Ws, Skip if Set | 1 | 1 (2 or 3) | None |
| 12 | BTST | BTST | f,#bit4 | Bit Test f | 1 | 1 | Z |
| | | BTST.C | Ws,#bit4 | Bit Test Ws to C | 1 | 1 | C |
| | | BTST.Z | Ws,#bit4 | Bit Test Ws to Z | 1 | 1 | Z |
| | | BTST.C | Ws,Wb | Bit Test Ws<Wb> to C | 1 | 1 | C |
| | | BTST.Z | Ws,Wb | Bit Test Ws<Wb> to Z | 1 | 1 | Z |
| 13 | BTSTS | BTSTS | f,#bit4 | Bit Test then Set f | 1 | 1 | Z |
| | | BTSTS.C | Ws,#bit4 | Bit Test Ws to C, then Set | 1 | 1 | C |
| | | BTSTS.Z | Ws,#bit4 | Bit Test Ws to Z, then Set | 1 | 1 | Z |
| 14 | CALL | CALL | lit23 | Call subroutine | 2 | 2 | None |
| | | CALL | Wn | Call indirect subroutine | 1 | 2 | None |
| 15 | CLR | CLR | f | f = 0x0000 | 1 | 1 | None |
| | | CLR | WREG | WREG = 0x0000 | 1 | 1 | None |
| | | CLR | Ws | Ws = 0x0000 | 1 | 1 | None |
| | | CLR | Acc,Wx,Wxd,Wy,Wyd,AWB | Clear Accumulator | 1 | 1 | OA,OB,SA,SB |
| 16 | CLRWDT | CLRWDT | | Clear Watchdog Timer | 1 | 1 | WDTO,Sleep |
| 17 | COM | COM | f | f = $\overline{f}$ | 1 | 1 | N,Z |
| | | COM | f,WREG | WREG = $\overline{f}$ | 1 | 1 | N,Z |
| | | COM | Ws,Wd | Wd = $\overline{Ws}$ | 1 | 1 | N,Z |
| 18 | CP | CP | f | Compare f with WREG | 1 | 1 | C,DC,N,OV,Z |
| | | CP | Wb,#lit5 | Compare Wb with lit5 | 1 | 1 | C,DC,N,OV,Z |
| | | CP | Wb,Ws | Compare Wb with Ws (Wb - Ws) | 1 | 1 | C,DC,N,OV,Z |
| 19 | CP0 | CP0 | f | Compare f with 0x0000 | 1 | 1 | C,DC,N,OV,Z |
| | | CP0 | Ws | Compare Ws with 0x0000 | 1 | 1 | C,DC,N,OV,Z |
| 20 | CPB | CPB | f | Compare f with WREG, with Borrow | 1 | 1 | C,DC,N,OV,Z |
| | | CPB | Wb,#lit5 | Compare Wb with lit5, with Borrow | 1 | 1 | C,DC,N,OV,Z |
| | | CPB | Wb,Ws | Compare Wb with Ws, with Borrow (Wb - Ws - $\overline{C}$) | 1 | 1 | C,DC,N,OV,Z |
| 21 | CPSEQ | CPSEQ | Wb, Wn | Compare Wb with Wn, skip if = | 1 | 1 (2 or 3) | None |
| 22 | CPSGT | CPSGT | Wb, Wn | Compare Wb with Wn, skip if > | 1 | 1 (2 or 3) | None |
| 23 | CPSLT | CPSLT | Wb, Wn | Compare Wb with Wn, skip if < | 1 | 1 (2 or 3) | None |
| 24 | CPSNE | CPSNE | Wb, Wn | Compare Wb with Wn, skip if ≠ | 1 | 1 (2 or 3) | None |
| 25 | DAW | DAW | Wn | Wn = decimal adjust Wn | 1 | 1 | C |
| 26 | DEC | DEC | f | f = f - 1 | 1 | 1 | C,DC,N,OV,Z |
| | | DEC | f,WREG | WREG = f -1 | 1 | 1 | C,DC,N,OV,Z |
| | | DEC | Ws,Wd | Wd = Ws - 1 | 1 | 1 | C,DC,N,OV,Z |
| 27 | DEC2 | DEC2 | f | f = f - 2 | 1 | 1 | C,DC,N,OV,Z |
| | | DEC2 | f,WREG | WREG = f -2 | 1 | 1 | C,DC,N,OV,Z |
| | | DEC2 | Ws,Wd | Wd = Ws - 2 | 1 | 1 | C,DC,N,OV,Z |
| 28 | DISI | DISI | #lit14 | Disable Interrupts for k instruction cycles | 1 | 1 | None |
| 29 | DIV | DIV.S | Wm,Wn | Signed 16/16-bit Integer Divide | 1 | 18 | N,Z,C, OV |
| | | DIV.SD | Wm,Wn | Signed 32/16-bit Integer Divide | 1 | 18 | N,Z,C, OV |
| | | DIV.U | Wm,Wn | Unsigned 16/16-bit Integer Divide | 1 | 18 | N,Z,C, OV |
| | | DIV.UD | Wm,Wn | Unsigned 32/16-bit Integer Divide | 1 | 18 | N,Z,C, OV |
| 30 | DIVF | DIVF | Wm,Wn | Signed 16/16-bit Fractional Divide | 1 | 18 | N,Z,C, OV |
| 31 | DO | DO | #lit14,Expr | Do code to PC+Expr, lit14 + 1 times | 2 | 2 | None |
| | | DO | Wn,Expr | Do code to PC+Expr, (Wn) + 1 times | 2 | 2 | None |
| 32 | ED | ED | Wm*Wm,Acc,Wx,Wy,Wxd | Euclidean Distance (no accumulate) | 1 | 1 | OA,OB,OAB, SA,SB,SAB |
| 33 | EDAC | EDAC | Wm*Wm,Acc,Wx,Wy,Wxd | Euclidean Distance | 1 | 1 | OA,OB,OAB, SA,SB,SAB |

**TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)**

| Base Instr # | Assembly Mnemonic | Assembly Syntax | | Description | # of words | # of cycles | Status Flags Affected |
|---|---|---|---|---|---|---|---|
| 52 | NEG | NEG | Acc | Negate Accumulator | 1 | 1 | OA,OB,OAB, SA,SB,SAB |
| | | NEG | f | f = $\overline{f}$ + 1 | 1 | 1 | C,DC,N,OV,Z |
| | | NEG | f,WREG | WREG = $\overline{f}$ + 1 | 1 | 1 | C,DC,N,OV,Z |
| | | NEG | Ws,Wd | Wd = $\overline{Ws}$ + 1 | 1 | 1 | C,DC,N,OV,Z |
| 53 | NOP | NOP | | No Operation | 1 | 1 | None |
| | | NOPR | | No Operation | 1 | 1 | None |
| 54 | POP | POP | f | Pop f from Top-of-Stack (TOS) | 1 | 1 | None |
| | | POP | Wdo | Pop from Top-of-Stack (TOS) to Wdo | 1 | 1 | None |
| | | POP.D | Wnd | Pop from Top-of-Stack (TOS) to W(nd):W(nd + 1) | 1 | 2 | None |
| | | POP.S | | Pop Shadow Registers | 1 | 1 | All |
| 55 | PUSH | PUSH | f | Push f to Top-of-Stack (TOS) | 1 | 1 | None |
| | | PUSH | Wso | Push Wso to Top-of-Stack (TOS) | 1 | 1 | None |
| | | PUSH.D | Wns | Push W(ns):W(ns + 1) to Top-of-Stack (TOS) | 1 | 2 | None |
| | | PUSH.S | | Push Shadow Registers | 1 | 1 | None |
| 56 | PWRSAV | PWRSAV | #lit1 | Go into Sleep or Idle mode | 1 | 1 | WDTO,Sleep |
| 57 | RCALL | RCALL | Expr | Relative Call | 1 | 2 | None |
| | | RCALL | Wn | Computed Call | 1 | 2 | None |
| 58 | REPEAT | REPEAT | #lit14 | Repeat Next Instruction lit14 + 1 times | 1 | 1 | None |
| | | REPEAT | Wn | Repeat Next Instruction (Wn) + 1 times | 1 | 1 | None |
| 59 | RESET | RESET | | Software device Reset | 1 | 1 | None |
| 60 | RETFIE | RETFIE | | Return from interrupt | 1 | 3 (2) | None |
| 61 | RETLW | RETLW | #lit10,Wn | Return with literal in Wn | 1 | 3 (2) | None |
| 62 | RETURN | RETURN | | Return from Subroutine | 1 | 3 (2) | None |
| 63 | RLC | RLC | f | f = Rotate Left through Carry f | 1 | 1 | C,N,Z |
| | | RLC | f,WREG | WREG = Rotate Left through Carry f | 1 | 1 | C,N,Z |
| | | RLC | Ws,Wd | Wd = Rotate Left through Carry Ws | 1 | 1 | C,N,Z |
| 64 | RLNC | RLNC | f | f = Rotate Left (No Carry) f | 1 | 1 | N,Z |
| | | RLNC | f,WREG | WREG = Rotate Left (No Carry) f | 1 | 1 | N,Z |
| | | RLNC | Ws,Wd | Wd = Rotate Left (No Carry) Ws | 1 | 1 | N,Z |
| 65 | RRC | RRC | f | f = Rotate Right through Carry f | 1 | 1 | C,N,Z |
| | | RRC | f,WREG | WREG = Rotate Right through Carry f | 1 | 1 | C,N,Z |
| | | RRC | Ws,Wd | Wd = Rotate Right through Carry Ws | 1 | 1 | C,N,Z |
| 66 | RRNC | RRNC | f | f = Rotate Right (No Carry) f | 1 | 1 | N,Z |
| | | RRNC | f,WREG | WREG = Rotate Right (No Carry) f | 1 | 1 | N,Z |
| | | RRNC | Ws,Wd | Wd = Rotate Right (No Carry) Ws | 1 | 1 | N,Z |
| 67 | SAC | SAC | Acc,#Slit4,Wdo | Store Accumulator | 1 | 1 | None |
| | | SAC.R | Acc,#Slit4,Wdo | Store Rounded Accumulator | 1 | 1 | None |
| 68 | SE | SE | Ws,Wnd | Wnd = sign extended Ws | 1 | 1 | C,N,Z |
| 69 | SETM | SETM | f | f = 0xFFFF | 1 | 1 | None |
| | | SETM | WREG | WREG = 0xFFFF | 1 | 1 | None |
| | | SETM | Ws | Ws = 0xFFFF | 1 | 1 | None |
| 70 | SFTAC | SFTAC | Acc,Wn | Arithmetic Shift Accumulator by (Wn) | 1 | 1 | OA,OB,OAB, SA,SB,SAB |
| | | SFTAC | Acc,#Slit6 | Arithmetic Shift Accumulator by Slit6 | 1 | 1 | OA,OB,OAB, SA,SB,SAB |
| 71 | SL | SL | f | f = Left Shift f | 1 | 1 | C,N,OV,Z |
| | | SL | f,WREG | WREG = Left Shift f | 1 | 1 | C,N,OV,Z |
| | | SL | Ws,Wd | Wd = Left Shift Ws | 1 | 1 | C,N,OV,Z |
| | | SL | Wb,Wns,Wnd | Wnd = Left Shift Wb by Wns | 1 | 1 | N,Z |
| | | SL | Wb,#lit5,Wnd | Wnd = Left Shift Wb by lit5 | 1 | 1 | N,Z |

## 21.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers and dsPIC® digital signal controllers are supported with a full range of software and hardware development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Compilers/Assemblers/Linkers
  - MPLAB C Compiler for Various Device Families
  - HI-TECH C for Various Device Families
  - MPASM™ Assembler
  - MPLINK™ Object Linker/ MPLIB™ Object Librarian
  - MPLAB Assembler/Linker/Librarian for Various Device Families
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debuggers
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- Device Programmers
  - PICkit™ 2 Programmer
  - MPLAB PM3 Device Programmer
- Low-Cost Demonstration/Development Boards, Evaluation Kits, and Starter Kits

### 21.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16/32-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - In-Circuit Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either C or assembly)
- One-touch compile or assemble, and download to emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (C or assembly)
  - Mixed C and assembly
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

## 22.1 DC Characteristics

**TABLE 22-1: OPERATING MIPS VS. VOLTAGE**

| VDD Range | Temp Range | Max MIPS | |
|---|---|---|---|
| | | dsPIC30F2010-30I | dsPIC30F2010-20E |
| 4.5-5.5V | -40°C to 85°C | 30 | — |
| 4.5-5.5V | -40°C to 125°C | — | 20 |
| 3.0-3.6V | -40°C to 85°C | 20 | — |
| 3.0-3.6V | -40°C to 125°C | — | 15 |
| 2.5-3.0V | -40°C to 85°C | 10 | — |

**TABLE 22-2: THERMAL OPERATING CONDITIONS**

| Rating | Symbol | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| dsPIC30F2010-30I | | | | | |
| Operating Junction Temperature Range | TJ | -40 | — | +125 | °C |
| Operating Ambient Temperature Range | TA | -40 | — | +85 | °C |
| dsPIC30F2010-20E | | | | | |
| Operating Junction Temperature Range | TJ | -40 | — | +150 | °C |
| Operating Ambient Temperature Range | TA | -40 | — | +125 | °C |
| Power Dissipation:<br>Internal chip power dissipation:<br>$P_{INT} = V_{DD} \times \left( I_{DD} - \sum I_{OH} \right)$<br>I/O Pin power dissipation:<br>$P_{I/O} = \sum(\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum(V_{OL} \times I_{OL})$ | PD | PINT + PI/O | | | W |
| Maximum Allowed Power Dissipation | PDMAX | (TJ - TA)/θJA | | | W |

**TABLE 22-3: THERMAL PACKAGING CHARACTERISTICS**

| Characteristic | Symbol | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|
| Package Thermal Resistance, 28-pin SOIC (SO) | θJA | 48.3 | — | °C/W | 1 |
| Package Thermal Resistance, 28-pin QFN | θJA | 33.7 | — | °C/W | 1 |
| Package Thermal Resistance, 28-pin SPDIP (SP) | θJA | 42 | — | °C/W | 1 |

**Note 1:** Junction to ambient thermal resistance, Theta-ja (θJA) numbers are achieved by package simulations.

**TABLE 22-10: ELECTRICAL CHARACTERISTICS: BOR**

| DC CHARACTERISTICS | | | Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)<br>Operating temperature -40°C ≤TA ≤+85°C for Industrial<br>-40°C ≤TA ≤+125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | **Min** | **Typ(1)** | **Max** | **Units** | **Conditions** |
| BO10 | VBOR | BOR Voltage(2) on VDD transition high to low | BORV = 11(3) | — | — | — | V | Not in operating range |
| | | | BORV = 10 | 2.6 | — | 2.71 | V | — |
| | | | BORV = 01 | 4.1 | — | 4.4 | V | — |
| | | | BORV = 00 | 4.58 | — | 4.73 | V | — |
| BO15 | VBHYS | | | — | 5 | — | mV | — |

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

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

**3:** 11 values not in usable operating range.

**TABLE 22-11: DC CHARACTERISTICS: PROGRAM AND EEPROM**

| DC CHARACTERISTICS | | | Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)<br>Operating temperature -40°C ≤TA ≤+85°C for Industrial<br>-40°C ≤TA ≤+125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | **Min** | **Typ(1)** | **Max** | **Units** | **Conditions** |
| | | **Data EEPROM Memory(2)** | | | | | |
| D120 | ED | Byte Endurance | 100K | 1M | — | E/W | -40° C ≤TA ≤+85°C |
| D121 | VDRW | VDD for Read/Write | VMIN | — | 5.5 | V | Using EECON to read/write VMIN = Minimum operating voltage |
| D122 | TDEW | Erase/Write Cycle Time | 0.8 | 2 | 2.6 | ms | RTSP |
| D123 | TRETD | Characteristic Retention | 40 | 100 | — | Year | Provided no other specifications are violated |
| D124 | IDEW | IDD During Programming | — | 10 | 30 | mA | Row Erase |
| | | **Program Flash Memory(2)** | | | | | |
| D130 | EP | Cell Endurance | 10K | 100K | — | E/W | -40° C ≤TA ≤+85°C |
| D131 | VPR | VDD for Read | VMIN | — | 5.5 | V | VMIN = Minimum operating voltage |
| D132 | VEB | VDD for Bulk Erase | 4.5 | — | 5.5 | V | |
| D133 | VPEW | VDD for Erase/Write | 3.0 | — | 5.5 | V | |
| D134 | TPEW | Erase/Write Cycle Time | 0.8 | 2 | 2.6 | ms | RTSP |
| D135 | TRETD | Characteristic Retention | 40 | 100 | — | Year | Provided no other specifications are violated |
| D137 | IPEW | IDD During Programming | — | 10 | 30 | mA | Row Erase |
| D138 | IEB | IDD During Programming | — | 10 | 30 | mA | Bulk Erase |

**Note 1:** Data in "Typ" column is at 5V, 25°C unless otherwise stated.

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

**TABLE 22-38: 10-BIT HIGH-SPEED A/D MODULE SPECIFICATIONS (CONTINUED)**

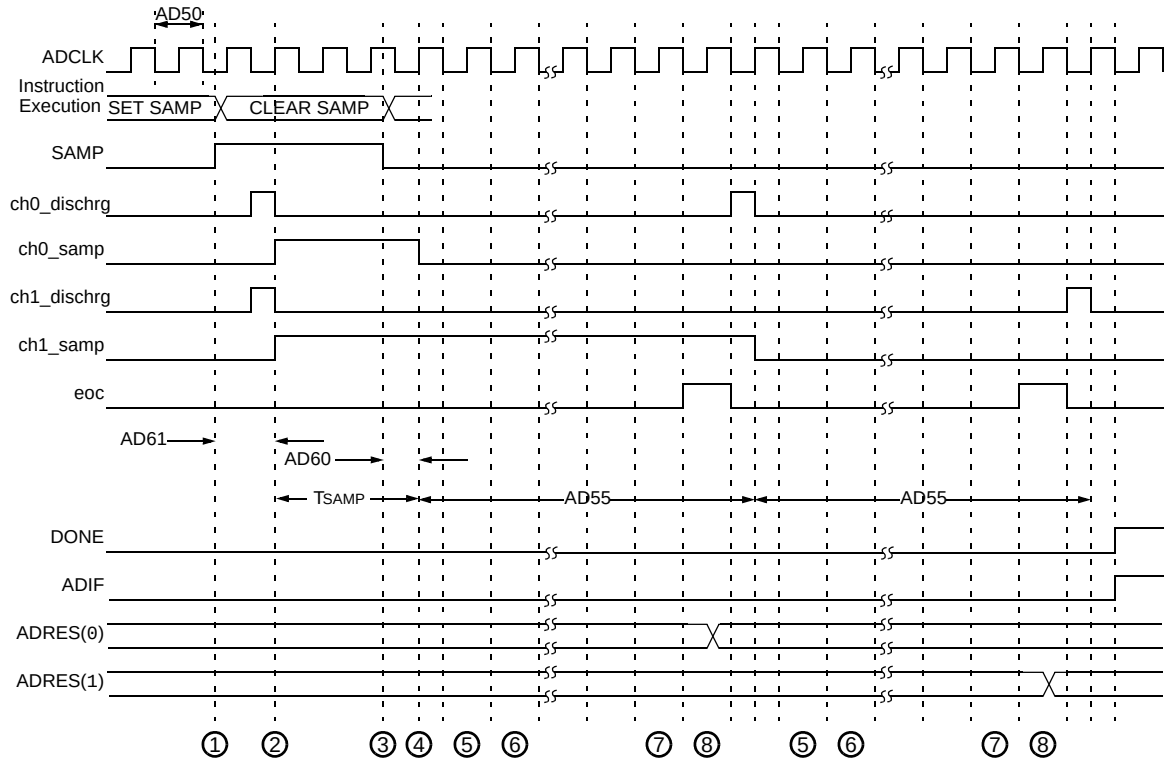
| AC CHARACTERISTICS | | | Standard Operating Conditions: 2.7V to 5.5V (unless otherwise stated)<br>Operating temperature -40°C ≤TA ≤+85°C for Industrial<br>-40°C ≤TA ≤+125°C for Extended | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Symbol** | **Characteristic** | **Min.** | **Typ** | **Max.** | **Units** | **Conditions** |
| AD24 | EOFF | Offset Error | ±1 | ±2 | ±3 | LSb | VINL = AVSS = VREFL = 0V, AVDD = VREFH = 5V |
| AD24A | EOFF | Offset Error | ±1 | ±2 | ±3 | LSb | VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V |
| AD25 | — | Monotonicity(2) | — | — | — | — | Guaranteed |
| **Dynamic Performance** | | | | | | | |
| AD30 | THD | Total Harmonic Distortion | — | -64 | -67 | dB | — |
| AD31 | SINAD | Signal to Noise and Distortion | — | 57 | 58 | dB | — |
| AD32 | SFDR | Spurious Free Dynamic Range | — | 67 | 71 | dB | — |
| AD33 | FNYQ | Input Signal Bandwidth | — | — | 500 | kHz | — |
| AD34 | ENOB | Effective Number of Bits | 9.29 | 9.41 | — | bits | — |

**Note 1:** Because the sample caps will eventually lose charge, clock rates below 10 kHz can affect linearity performance, especially at elevated temperatures.

**2:** The A/D conversion result never decreases with an increase in the input voltage, and has no missing codes.

**3:** Measurements were taken with external VREF+ and VREF- used as the ADC voltage references.

**FIGURE 22-24: 10-BIT HIGH-SPEED A/D CONVERSION TIMING CHARACTERISTICS (CHPS = 01, SIMSAM = 0, ASAM = 0, SSRC = 000)**




① – Software sets ADCON. SAMP to start sampling.

② – Sampling starts after discharge period.
TSAMP is described in **Section 17. "10-bit A/D Converter"** (DS70064) of the *"dsPIC30F Family Reference Manual"* (DS70

③ – Software clears ADCON. SAMP to start conversion.

④ – Sampling ends, conversion sequence starts.

⑤ – Convert bit 9.

⑥ – Convert bit 8.

⑦ – Convert bit 0.

⑧ – One TAD for end of conversion.

Enable Bits ..... 88
Fault States ..... 88
Modes ..... 88
Cycle-by-Cycle ..... 88
Latched ..... 88
PWM Operation During CPU Idle Mode ..... 89
PWM Operation During CPU Sleep Mode ..... 89
PWM Output and Polarity Control ..... 88
Output Pin Control ..... 88
PWM Output Override ..... 87
Complementary Output Mode ..... 87
Synchronization ..... 87
PWM Period ..... 84
PWM Special Event Trigger ..... 89
Postscaler ..... 89
PWM Time Base ..... 83
Continuous Up/Down Counting Modes ..... 83
Double Update Mode ..... 84
Free Running Mode ..... 83
Postscaler ..... 84
Prescaler ..... 84
Single-Shot Mode ..... 83
PWM Update Lockout ..... 88

**Q**

QEA/QEB Input Characteristics ..... 169
QEI Module
External Clock Timing Requirements ..... 165
Index Pulse Timing Characteristics ..... 170
Index Pulse Timing Requirements ..... 170
Operation During CPU Idle Mode ..... 78
Operation During CPU Sleep Mode ..... 77
Register Map ..... 79
Timer Operation During CPU Idle Mode ..... 78
Timer Operation During CPU Sleep Mode ..... 77
Quadrature Decoder Timing Requirements ..... 169
Quadrature Encoder Interface (QEI) Module ..... 75
Quadrature Encoder Interface Interrupts ..... 78
Quadrature Encoder Interface Logic ..... 76

**R**

Reader Response ..... 200
Reset ..... 121, 127
Reset Sequence ..... 39
Reset Sources ..... 39
Reset Timing Characteristics ..... 161
Reset Timing Requirements ..... 161
Resets
BOR, Programmable ..... 129
POR ..... 127
Operating without FSCM and PWRT ..... 129
POR with Long Crystal Start-up Time ..... 129
RTSP Operation ..... 44

**S**

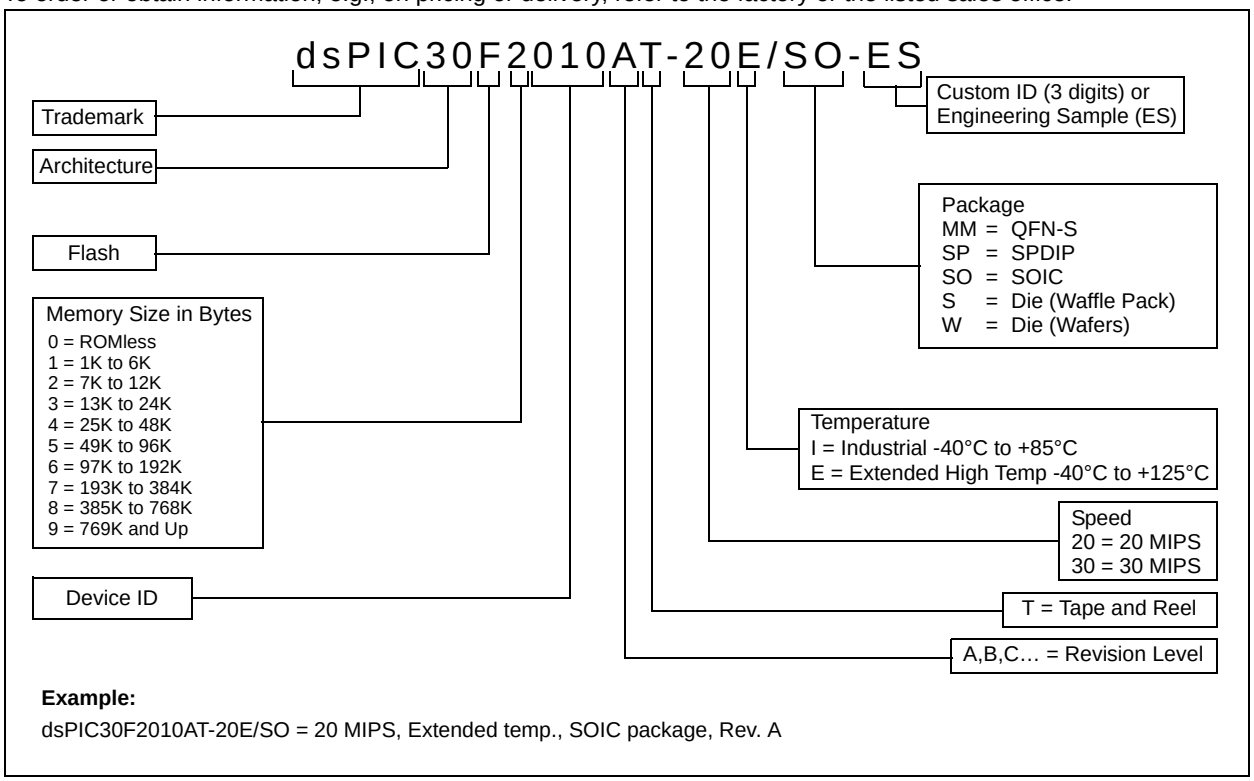
Serial Peripheral Interface. *See* SPI
Simple Capture Event Mode
Capture Buffer Operation ..... 68
Capture Prescaler ..... 68
Hall Sensor Mode ..... 68
Input Capture in CPU Idle Mode ..... 69
Timer2 and Timer3 Selection Mode ..... 68
Simple OC/PWM Mode Timing Requirements ..... 167
Simple Output Compare Match Mode ..... 72
Simple PWM Mode ..... 72
Input Pin Fault Protection ..... 72
Period ..... 73
Single Pulse PWM Operation ..... 87
Software Simulator (MPLAB SIM) ..... 145
Software Stack Pointer, Frame Pointer ..... 12
CALL Stack Frame ..... 27
SPI ..... 91
SPI Mode
Slave Select Synchronization ..... 93
SPI1 Register Map ..... 94
SPI Module ..... 91
Framed SPI Support ..... 92
Operating Function Description ..... 91
SDOx Disable ..... 91
Timing Characteristics
Master Mode (CKE = 0) ..... 171
Master Mode (CKE = 1) ..... 172
Slave Mode (CKE = 1) ..... 173, 174
Timing Requirements
Master Mode (CKE = 0) ..... 171
Master Mode (CKE = 1) ..... 172
Slave Mode (CKE = 0) ..... 173
Slave Mode (CKE = 1) ..... 175
Word and Byte Communication ..... 91
SPI Operation During CPU Idle Mode ..... 93
SPI Operation During CPU Sleep Mode ..... 93
STATUS Register ..... 12
Subtracter ..... 16
Data Space Write Saturation ..... 18
Overflow and Saturation ..... 16
Round Logic ..... 17
Write Back ..... 17
Symbols used in Opcode Descriptions ..... 136
System Integration ..... 121
Overview ..... 121
Register Map ..... 133

**T**

Temperature and Voltage Specifications
AC ..... 156
DC ..... 148
Timer1 Module ..... 57
16-bit Asynchronous Counter Mode ..... 57
16-bit Synchronous Counter Mode ..... 57
16-bit Timer Mode ..... 57
Gate Operation ..... 58
Interrupt ..... 59
Operation During Sleep Mode ..... 58
Prescaler ..... 58
Real-Time Clock ..... 59
RTC Interrupts ..... 59
RTC Oscillator Operation ..... 59
Register Map ..... 60
Timer2 and Timer3 Selection Mode ..... 72
Timer2/3 Module ..... 61
32-bit Synchronous Counter Mode ..... 61
32-bit Timer Mode ..... 61
ADC Event Trigger ..... 64
Gate Operation ..... 64
Interrupt ..... 64
Operation During Sleep Mode ..... 64
Register Map ..... 65
Timer Prescaler ..... 64

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.




**Example:**

dsPIC30F2010AT-20E/SO = 20 MIPS, Extended temp., SOIC package, Rev. A