



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	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	20
Program Memory Size	24KB (8K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3013-20i-sp">https://www.e-xfl.com/product-detail/microchip-technology/dspic30f3013-20i-sp</a>

The SA and SB bits are modified each time data passes through the adder/subtractor but can only be cleared by the user. When set, they indicate that the accumulator has overflowed its maximum range (bit 31 for 32-bit saturation or bit 39 for 40-bit saturation) and will be saturated if saturation is enabled. When saturation is not enabled, SA and SB default to bit 39 overflow and thus indicate that a catastrophic overflow has occurred. If the COVTE bit in the INTCON1 register is set, SA and SB bits generate an arithmetic warning trap when saturation is disabled.

The overflow and saturation Status bits can optionally be viewed in the STATUS register (SR) as the logical OR of OA and OB (in bit OAB) and the logical OR of SA and SB (in bit SAB). This allows programmers to check one bit in the STATUS register to determine if either accumulator has overflowed, or one bit to determine if either accumulator has saturated. This would be useful for complex number arithmetic which typically uses both the accumulators.

The device supports three saturation and overflow modes:

1. **Bit 39 Overflow and Saturation:**  
When bit 39 overflow and saturation occurs, the saturation logic loads the maximally positive 9.31 (0x7FFFFFFF) or maximally negative 9.31 value (0x80000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. This is referred to as 'super saturation' and provides protection against erroneous data or unexpected algorithm problems (e.g., gain calculations).
2. **Bit 31 Overflow and Saturation:**  
When bit 31 overflow and saturation occurs, the saturation logic then loads the maximally positive 1.31 value (0x007FFFFFFF) or maximally negative 1.31 value (0x0080000000) into the target accumulator. The SA or SB bit is set and remains set until cleared by the user. When this Saturation mode is in effect, the guard bits are not used, so the OA, OB or OAB bits are never set.
3. **Bit 39 Catastrophic Overflow:**  
The bit 39 overflow Status bit from the adder is used to set the SA or SB bit which remains set until cleared by the user. No saturation operation is performed and the accumulator is allowed to overflow (destroying its sign). If the COVTE bit in the INTCON1 register is set, a catastrophic overflow can initiate a trap exception.

## 2.4.2.2 Accumulator 'Write-Back'

The MAC class of instructions (with the exception of MPY, MPY.N, ED and EDAC) can optionally write a rounded version of the high word (bits 31 through 16) of the accumulator that is not targeted by the instruction into data space memory. The write is performed across the X bus into combined X and Y address space. The following addressing modes are supported:

1. **W13, Register Direct:**  
The rounded contents of the non-target accumulator are written into W13 as a 1.15 fraction.
2. **[W13]+ = 2, Register Indirect with Post-Increment:**  
The rounded contents of the non-target accumulator are written into the address pointed to by W13 as a 1.15 fraction. W13 is then incremented by 2 (for a word write).

## 2.4.2.3 Round Logic

The round logic is a combinational block which performs a conventional (biased) or convergent (unbiased) round function during an accumulator write (store). The Round mode is determined by the state of the RND bit in the CORCON register. It generates a 16-bit, 1.15 data value, which is passed to the data space write saturation logic. If rounding is not indicated by the instruction, a truncated 1.15 data value is stored and the least significant word (lsw) is simply discarded.

Conventional rounding takes bit 15 of the accumulator, zero-extends it and adds it to the ACCxH word (bits 16 through 31 of the accumulator). If the ACCxL word (bits 0 through 15 of the accumulator) is between 0x8000 and 0xFFFF (0x8000 included), ACCxH is incremented. If ACCxL is between 0x0000 and 0x7FFF, ACCxH is left unchanged. A consequence of this algorithm is that over a succession of random rounding operations, the value tends to be biased slightly positive.

Convergent (or unbiased) rounding operates in the same manner as conventional rounding, except when ACCxL equals 0x8000. If this is the case, the LSB (bit 16 of the accumulator) of ACCxH is examined. If it is '1', ACCxH is incremented. If it is '0', ACCxH is not modified. Assuming that bit 16 is effectively random in nature, this scheme will remove any rounding bias that may accumulate.

The SAC and SAC.R instructions store either a truncated (SAC) or rounded (SAC.R) version of the contents of the target accumulator to data memory via the X bus (subject to data saturation, see **Section 2.4.2.4 "Data Space Write Saturation"**). Note that for the MAC class of instructions, the accumulator write-back operation functions in the same manner, addressing combined MCU (X and Y) data space though the X bus. For this class of instructions, the data is always subject to rounding.

## 3.0 MEMORY ORGANIZATION

**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). For more information on the device instruction set and programming, refer to the “*16-bit MCU and DSC Programmer’s Reference Manual*” (DS70157).

### 3.1 Program Address Space

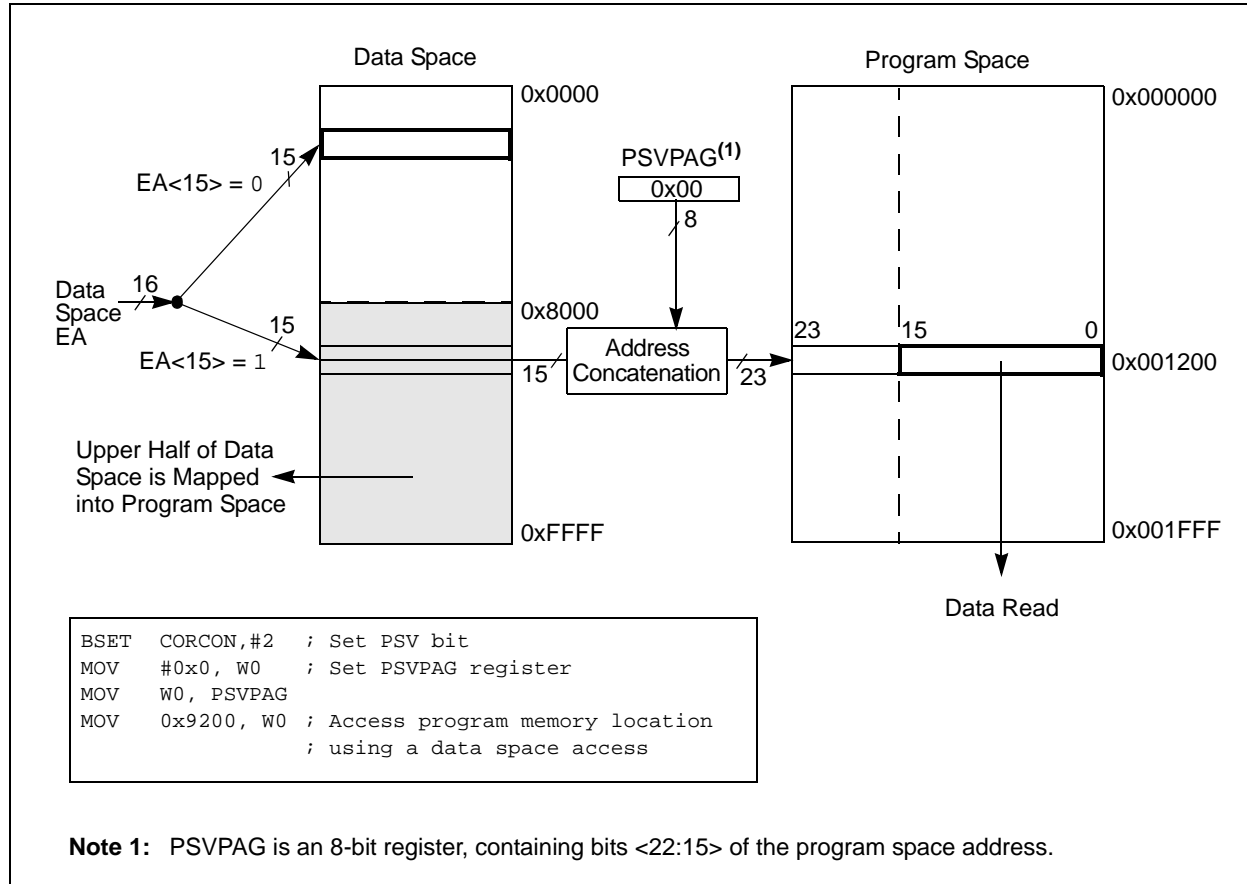
The program address space is 4M instruction words. The program space memory maps for the dsPIC30F2011/2012/3012/3013 devices is shown in Figure 3-1.

Program memory is addressable by a 24-bit value from either the 23-bit PC, table instruction Effective Address (EA), or data space EA, when program space is mapped into data space as defined by Table 3-1. Note that the program space address is incremented by two between successive program words in order to provide compatibility with data space addressing.

User program space access is restricted to the lower 4M instruction word address range (0x000000 to 0x7FFFFE) for all accesses other than TBLRD/TBLWT, which uses TBLPAG<7> to determine user or configuration space access. In Table 3-1, Program Space Address Construction, bit 23 allows access to the Device ID, the User ID and the Configuration bits. Otherwise, bit 23 is always clear.

# dsPIC30F2011/2012/3012/3013

**FIGURE 3-5: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION**



## 8.6 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.

## 8.7 External Interrupt Requests

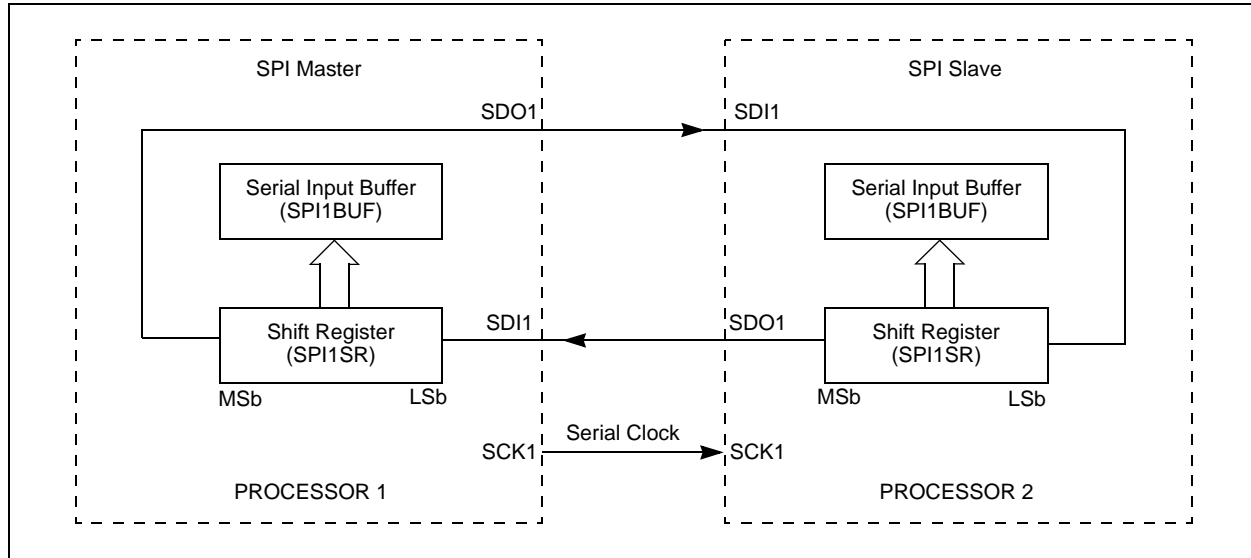
The interrupt controller supports three external interrupt request signals, INT0-INT2. These inputs are edge sensitive; they require a low-to-high or a high-to-low transition to generate an interrupt request. The INTCON2 register has three bits, INT0EP-INT2EP, that select the polarity of the edge detection circuitry.

## 8.8 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 wakes up from Sleep or Idle and begins execution of the ISR needed to process the interrupt request.

**FIGURE 13-2: SPI MASTER/SLAVE CONNECTION**



### 13.3 Slave Select Synchronization

The  $\overline{SS1}$  pin allows a Synchronous Slave mode. The SPI must be configured in SPI Slave mode with  $\overline{SS1}$  pin control enabled ( $SSEN = 1$ ). When the  $\overline{SS1}$  pin is low, transmission and reception are enabled and the SDOx pin is driven. When  $\overline{SS1}$  pin goes high, the SDOx pin is no longer driven. Also, the SPI module is resynchronized, and all counters/control circuitry are reset. Therefore, when the  $\overline{SS1}$  pin is asserted low again, transmission/reception will begin at the MSb even if  $\overline{SS1}$  had been de-asserted in the middle of a transmit/receive.

### 13.4 SPI Operation During CPU Sleep Mode

During Sleep mode, the SPI module is shut down. If the CPU enters Sleep mode while an SPI transaction is in progress, then the transmission and reception is aborted.

The transmitter and receiver will stop in Sleep mode. However, register contents are not affected by entering or exiting Sleep mode.

### 13.5 SPI Operation During CPU Idle Mode

When the device enters Idle mode, all clock sources remain functional. The SPISIDL bit ( $SPI1STAT<13>$ ) selects if the SPI module will stop or continue on idle. If  $SPISIDL = 0$ , the module will continue to operate when the CPU enters Idle mode. If  $SPISIDL = 1$ , the module will stop when the CPU enters Idle mode.

TABLE 13-1: SPI1 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
SPI1STAT	0220	SPIEN	—	SPIIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	Transmit and Receive Buffer																0000 0000 0000 0000

**Legend:** — = unimplemented bit, read as '0'  
**Note:** Refer to the “dsPIC30F Family Reference Manual” (DS70046) for descriptions of register bit fields.

## 14.0 I<sup>2</sup>C™ 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).

The Inter-Integrated Circuit (I<sup>2</sup>C™) module provides complete hardware support for both Slave and Multi-Master modes of the I<sup>2</sup>C serial communication standard, with a 16-bit interface.

This module offers the following key features:

- I<sup>2</sup>C interface supporting both master and slave operation.
- I<sup>2</sup>C Slave mode supports 7-bit and 10-bit addressing.
- I<sup>2</sup>C Master mode supports 7-bit and 10-bit addressing.
- I<sup>2</sup>C port allows bidirectional transfers between master and slaves.
- Serial clock synchronization for I<sup>2</sup>C port can be used as a handshake mechanism to suspend and resume serial transfer (SCLREL control).
- I<sup>2</sup>C supports multi-master operation; detects bus collision and will arbitrate accordingly.

### 14.1 Operating Function Description

The hardware fully implements all the master and slave functions of the I<sup>2</sup>C Standard and Fast mode specifications, as well as 7 and 10-bit addressing.

Thus, the I<sup>2</sup>C module can operate either as a slave or a master on an I<sup>2</sup>C bus.

#### 14.1.1 VARIOUS I<sup>2</sup>C MODES

The following types of I<sup>2</sup>C operation are supported:

- I<sup>2</sup>C slave operation with 7-bit addressing
- I<sup>2</sup>C slave operation with 10-bit addressing
- I<sup>2</sup>C master operation with 7-bit or 10-bit addressing

See the I<sup>2</sup>C programmer's model (Figure 14-1).

#### 14.1.2 PIN CONFIGURATION IN I<sup>2</sup>C MODE

I<sup>2</sup>C has a 2-pin interface; the SCL pin is clock and the SDA pin is data.

#### 14.1.3 I<sup>2</sup>C REGISTERS

I2CCON and I2CSTAT are control and status registers, respectively. The I2CCON register is readable and writable. The lower 6 bits of I2CSTAT are read-only. The remaining bits of the I2CSTAT are read/write.

I2CRSR is the shift register used for shifting data, whereas I2CRCV is the buffer register to which data bytes are written, or from which data bytes are read. I2CRCV is the receive buffer as shown in Figure 14-1. I2CTRN is the transmit register to which bytes are written during a transmit operation, as shown in Figure 14-2.

The I2CADD register holds the slave address. A Status bit, ADD10, indicates 10-bit Address mode. The I2CBRG acts as the Baud Rate Generator reload value.

In receive operations, I2CRSR and I2CRCV together form a double-buffered receiver. When I2CRSR receives a complete byte, it is transferred to I2CRCV and an interrupt pulse is generated. During transmission, the I2CTRN is not double-buffered.

**Note:** Following a Restart condition in 10-bit mode, the user only needs to match the first 7-bit address.

**FIGURE 14-1: PROGRAMMER'S MODEL**

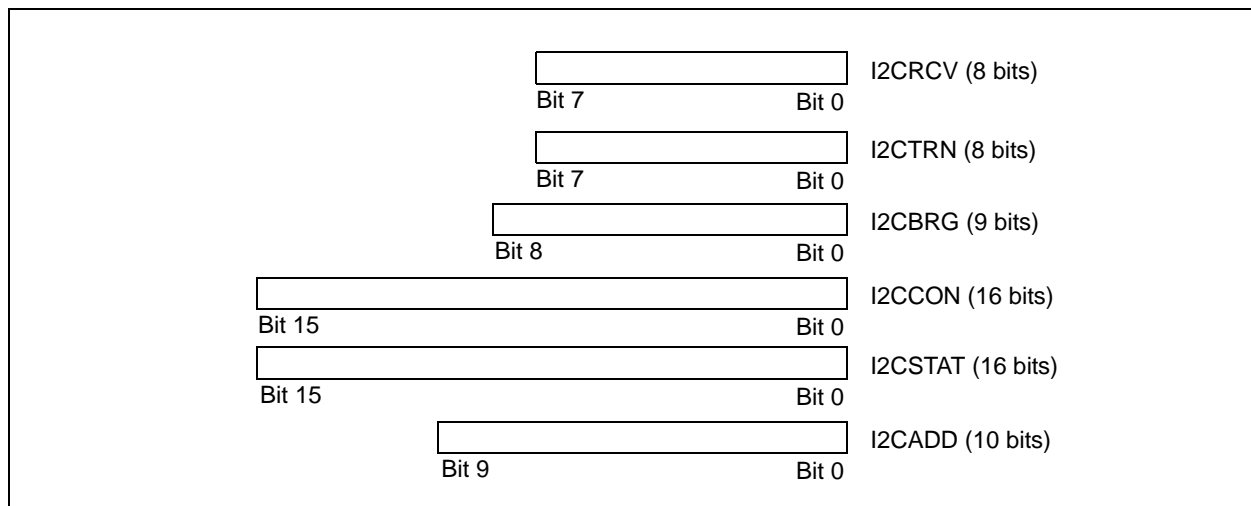




TABLE 14-2: I<sup>2</sup>C REGISTER MAP

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
I2CRCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000 0000 0000 0000
I2CTRN	0202	—	—	—	—	—	—	—	—	Transmit Register								0000 0000 1111 1111
I2CBRG	0204	—	—	—	—	—	—	—	—	Baud Rate Generator								0000 0000 0000 0000
I2CCON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0001 0000 0000 0000
I2CSTAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000 0000 0000 0000
I2CADD	020A	—	—	—	—	—	—	Address Register										0000 0000 0000 0000

**Legend:** — = unimplemented bit, read as '0'  
**Note:** Refer to the "dsPIC30F Family Reference Manual" (DS70046) for descriptions of register bit fields.

## 15.0 UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART) 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).

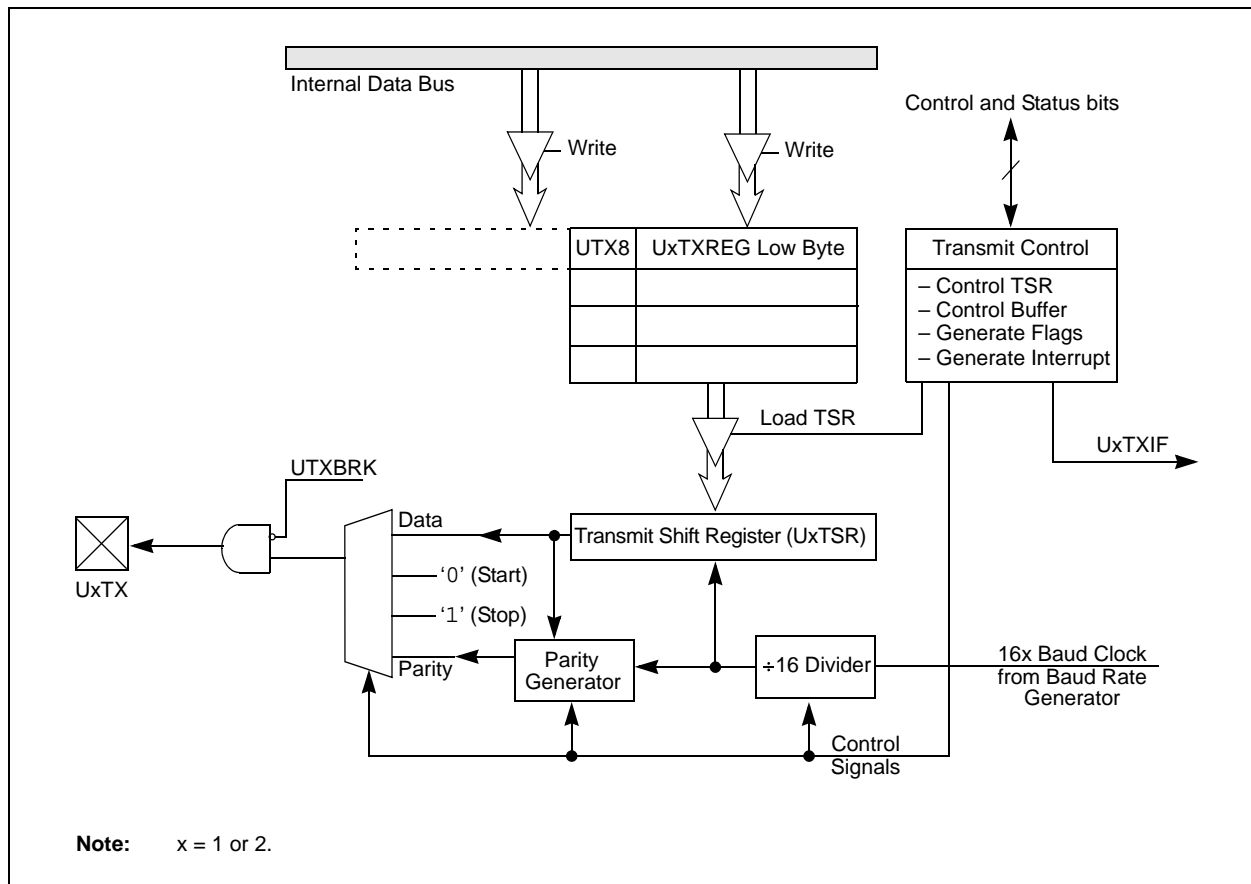
This section describes the Universal Asynchronous Receiver/Transmitter Communications module. The dsPIC30F2011/2012/3012 processors have one UART module (UART1). The dsPIC30F3013 processor has two UART modules (UART1 and UART2).

## 15.1 UART Module Overview

The key features of the UART module are:

- Full-duplex, 8 or 9-bit data communication
- Even, odd or no parity options (for 8-bit data)
- One or two Stop bits
- Fully integrated Baud Rate Generator with 16-bit prescaler
- Baud rates range from 38 bps to 1.875 Mbps at a 30 MHz instruction rate
- 4-word deep transmit data buffer
- 4-word deep receive data buffer
- Parity, framing and buffer overrun error detection
- Support for interrupt only on address detect (9th bit = 1)
- Separate transmit and receive interrupts
- Loopback mode for diagnostic support
- Alternate receive and transmit pins for UART1

**FIGURE 15-1: UART TRANSMITTER BLOCK DIAGRAM**



## 15.2 Enabling and Setting Up UART

### 15.2.1 ENABLING THE UART

The UART module is enabled by setting the UARTEN bit in the UxMODE register (where x = 1 or 2). Once enabled, the UxTX and UxRX pins are configured as an output and an input respectively, overriding the TRIS and LAT register bit settings for the corresponding I/O port pins. The UxTX pin is at logic '1' when no transmission is taking place.

### 15.2.2 DISABLING THE UART

The UART module is disabled by clearing the UARTEN bit in the UxMODE register. This is the default state after any Reset. If the UART is disabled, all I/O pins operate as port pins under the control of the LAT and TRIS bits of the corresponding port pins.

Disabling the UART module resets the buffers to empty states. Any data characters in the buffers are lost and the baud rate counter is reset.

All error and status flags associated with the UART module are reset when the module is disabled. The URXDA, OERR, FERR, PERR, UTXEN, UTXBRK and UTXBF bits are cleared, whereas RIDLE and TRMT are set. Other control bits, including ADDEN, URXISEL<1:0>, UTXISEL, as well as the UxMODE and UxBRG registers, are not affected.

Clearing the UARTEN bit while the UART is active will abort all pending transmissions and receptions and reset the module as defined above. Re-enabling the UART will restart the UART in the same configuration.

### 15.2.3 ALTERNATE I/O

The alternate I/O function is enabled by setting the ALTIO bit (UxMODE<10>). If ALTIO = 1, the UxATX and UxARX pins (alternate transmit and alternate receive pins, respectively) are used by the UART module instead of the UxTX and UxRX pins. If ALTIO = 0, the UxTX and UxRX pins are used by the UART module.

### 15.2.4 SETTING UP DATA, PARITY AND STOP BIT SELECTIONS

Control bits PDSEL<1:0> in the UxMODE register are used to select the data length and parity used in the transmission. The data length may either be 8 bits with even, odd or no parity, or 9 bits with no parity.

The STSEL bit determines whether one or two Stop bits will be used during data transmission.

The default (power-on) setting of the UART is 8 bits, no parity and 1 Stop bit (typically represented as 8, N, 1).

## 15.3 Transmitting Data

### 15.3.1 TRANSMITTING IN 8-BIT DATA MODE

The following steps must be performed to transmit 8-bit data:

1. Set up the UART:  
First, the data length, parity and number of Stop bits must be selected. Then, the transmit and receive interrupt enable and priority bits are setup in the UxMODE and UxSTA registers. Also, the appropriate baud rate value must be written to the UxBRG register.
2. Enable the UART by setting the UARTEN bit (UxMODE<15>).
3. Set the UTXEN bit (UxSTA<10>), thereby enabling a transmission.
4. Write the byte to be transmitted to the lower byte of UxTXREG. The value will be transferred to the Transmit Shift register (UxTSR) immediately and the serial bit stream will start shifting out during the next rising edge of the baud clock. Alternatively, the data byte may be written while UTXEN = 0, following which, the user may set UTXEN. This will cause the serial bit stream to begin immediately because the baud clock will start from a cleared state.
5. A transmit interrupt will be generated, depending on the value of the interrupt control bit UTXISEL (UxSTA<15>).

### 15.3.2 TRANSMITTING IN 9-BIT DATA MODE

The sequence of steps involved in the transmission of 9-bit data is similar to 8-bit transmission, except that a 16-bit data word (of which the upper 7 bits are always clear) must be written to the UxTXREG register.

### 15.3.3 TRANSMIT BUFFER (UxTXB)

The transmit buffer is 9 bits wide and 4 characters deep. Including the Transmit Shift register (UxTSR), the user effectively has a 5-deep FIFO (First-In, First-Out) buffer. The UTXBF bit (UxSTA<9>) indicates whether the transmit buffer is full.

If a user attempts to write to a full buffer, the new data will not be accepted into the FIFO and no data shift will occur within the buffer. This enables recovery from a buffer overrun condition.

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

## 16.1 A/D Result Buffer

The module contains a 16-word dual port read-only buffer, called ADCBUF0...ADCBUFF, to buffer the A/D results. The RAM is 12 bits wide but the data obtained is represented in one of four different 16-bit data formats. The contents of the sixteen A/D Conversion Result Buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

## 16.2 Conversion Operation

After the ADC module has been configured, the sample acquisition is started by setting the SAMP bit. Various sources, such as a programmable bit, timer time-outs and external events, will terminate acquisition and start a conversion. When the A/D conversion is complete, the result is loaded into ADCBUF0...ADCBUFF, and the DONE bit and the A/D interrupt flag, ADIF, are set after the number of samples specified by the SMPI bit. The ADC module can be configured for different interrupt rates as described in **Section 16.3 “Selecting the Conversion Sequence”**.

The following steps should be followed for doing an A/D conversion:

1. Configure the ADC module:
  - Configure analog pins, voltage reference and digital I/O
  - Select A/D input channels
  - Select A/D conversion clock
  - Select A/D conversion trigger
  - Turn on ADC module
2. Configure A/D interrupt (if required):
  - Clear ADIF bit
  - Select A/D interrupt priority
3. Start sampling
4. Wait the required acquisition time
5. Trigger acquisition end, start conversion
6. Wait for A/D conversion to complete, by either:
  - Waiting for the A/D interrupt, or
  - Waiting for the DONE bit to get set
7. Read A/D result buffer; clear ADIF if required

## 16.3 Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the A/D connects inputs to the sample/hold channel, converts a channel, writes the buffer memory and generates interrupts.

The sequence is controlled by the sampling clocks.

The SMPI bits select the number of acquisition/conversion sequences that would be performed before an interrupt occurs. This can vary from 1 sample per interrupt to 16 samples per interrupt.

The BUFM bit will split the 16-word results buffer into two 8-word groups. Writing to the 8-word buffers will be alternated on each interrupt event.

Use of the BUFM bit will depend on how much time is available for the moving of the buffers after the interrupt.

If the processor can quickly unload a full buffer within the time it takes to acquire and convert one channel, the BUFM bit can be '0' and up to 16 conversions (corresponding to the 16 input channels) may be done per interrupt. The processor will have one acquisition and conversion time to move the sixteen conversions.

If the processor cannot unload the buffer within the acquisition and conversion time, the BUFM bit should be '1'. For example, if SMPI<3:0> (ADCON2<5:2>) = 0111, then eight conversions will be loaded into 1/2 of the buffer, following which an interrupt occurs. The next eight conversions will be loaded into the other 1/2 of the buffer. The processor will have the entire time between interrupts to move the eight conversions.

The ALTS bit can be used to alternate the inputs selected during the sampling sequence. The input multiplexer has two sets of sample inputs: MUX A and MUX B. If the ALTS bit is '0', only the MUX A inputs are selected for sampling. If the ALTS bit is '1' and SMPI<3:0> = 0000 on the first sample/convert sequence, the MUX A inputs are selected and on the next acquire/convert sequence, the MUX B inputs are selected.

The CSCNA bit (ADCON2<10>) will allow the multiplexer input to be alternately scanned across a selected number of analog inputs for the MUX A group. The inputs are selected by the ADCSSL register. If a particular bit in the ADCSSL register is '1', the corresponding input is selected. The inputs are always scanned from lower to higher numbered inputs, starting after each interrupt. If the number of inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs are unused.

## 17.3 Reset

The dsPIC30F2011/2012/3012/3013 devices differentiate between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during Sleep
- Watchdog Timer (WDT) Reset (during normal operation)
- Programmable Brown-out Reset (BOR)
- RESET Instruction
- Reset caused by trap lockup (TRAPR)
- Reset caused by illegal opcode or by using an uninitialized W register as an address pointer (IOPUWR)

Different registers are affected in different ways by various Reset conditions. Most registers are not affected by a WDT wake-up since this is viewed as the resumption of normal operation. Status bits from the RCON register are set or cleared differently in different Reset situations, as indicated in Table 17-5. These bits are used in software to determine the nature of the Reset.

A block diagram of the On-Chip Reset Circuit is shown in Figure 17-2.

A  $\overline{\text{MCLR}}$  noise filter is provided in the  $\overline{\text{MCLR}}$  Reset path. The filter detects and ignores small pulses.

Internally generated Resets do not drive  $\overline{\text{MCLR}}$  pin low.

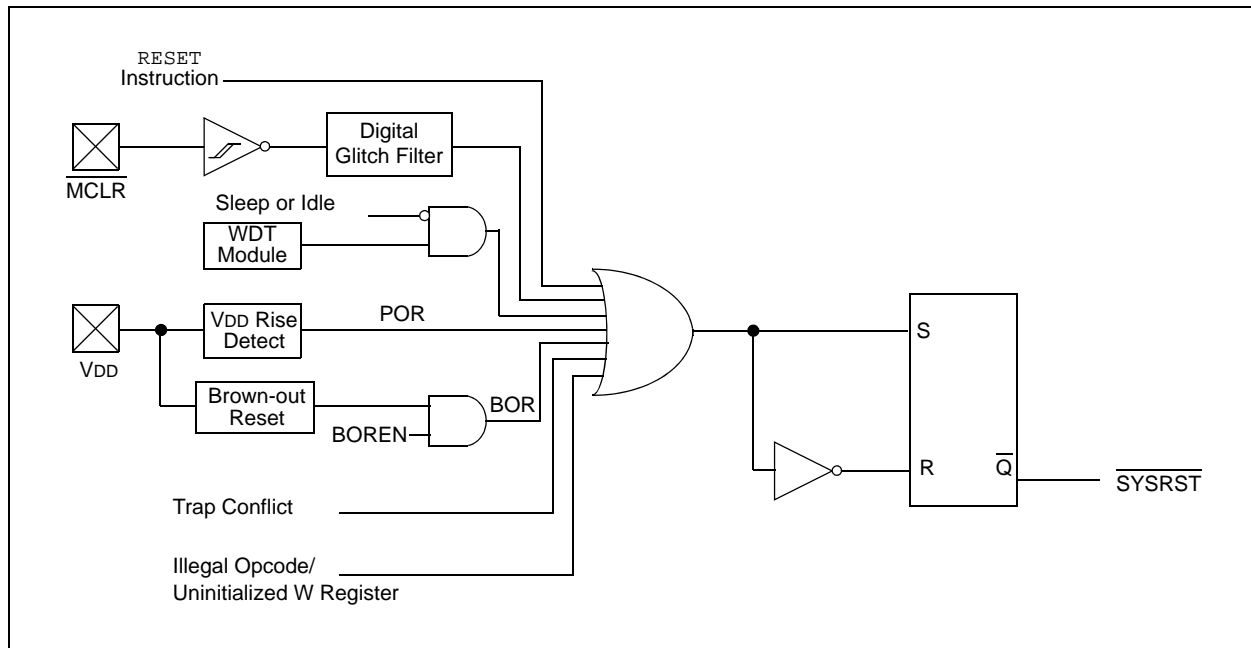
### 17.3.1 POR: POWER-ON RESET

A power-on event will generate an internal POR pulse when a VDD rise is detected. The Reset pulse will occur at the POR circuit threshold voltage (VPOR) which is nominally 1.85V. The device supply voltage characteristics must meet specified starting voltage and rise rate requirements. The POR pulse will reset a POR timer and place the device in the Reset state. The POR also selects the device clock source identified by the oscillator configuration fuses.

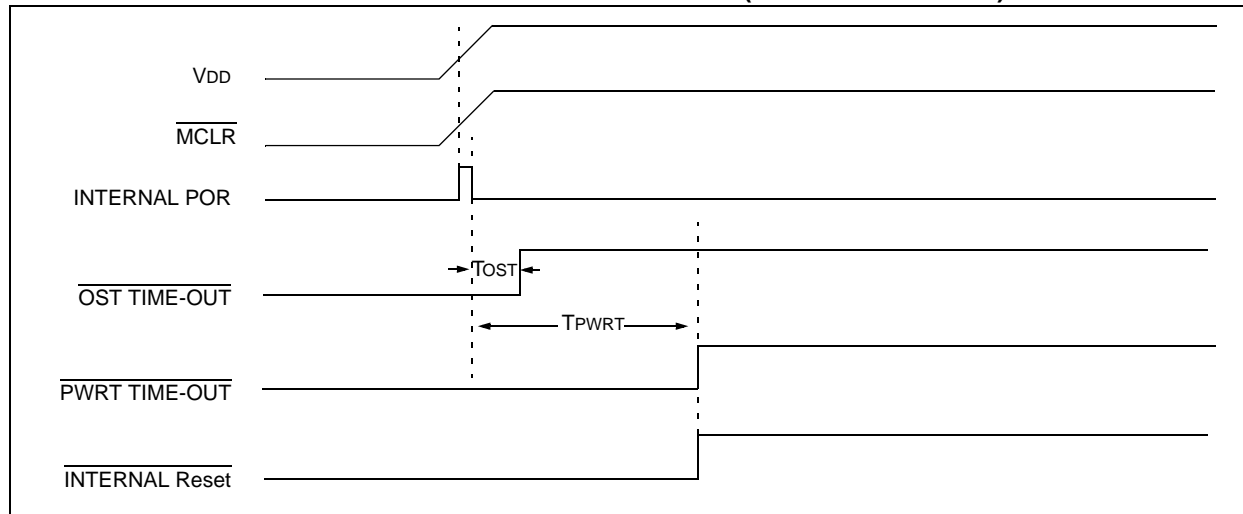
The POR circuit inserts a small delay, TPOR, which is nominally 10  $\mu\text{s}$  and ensures that the device bias circuits are stable. Furthermore, a user selected power-up time-out (TPWRT) is applied. The TPWRT parameter is based on device Configuration bits and can be 0 ms (no delay), 4 ms, 16 ms or 64 ms. The total delay is at device power-up, TPOR + TPWRT. When these delays have expired,  $\overline{\text{SYSRST}}$  will be negated on the next leading edge of the Q1 clock and the PC will jump to the Reset vector.

The timing for the  $\overline{\text{SYSRST}}$  signal is shown in Figure 17-3 through Figure 17-5.

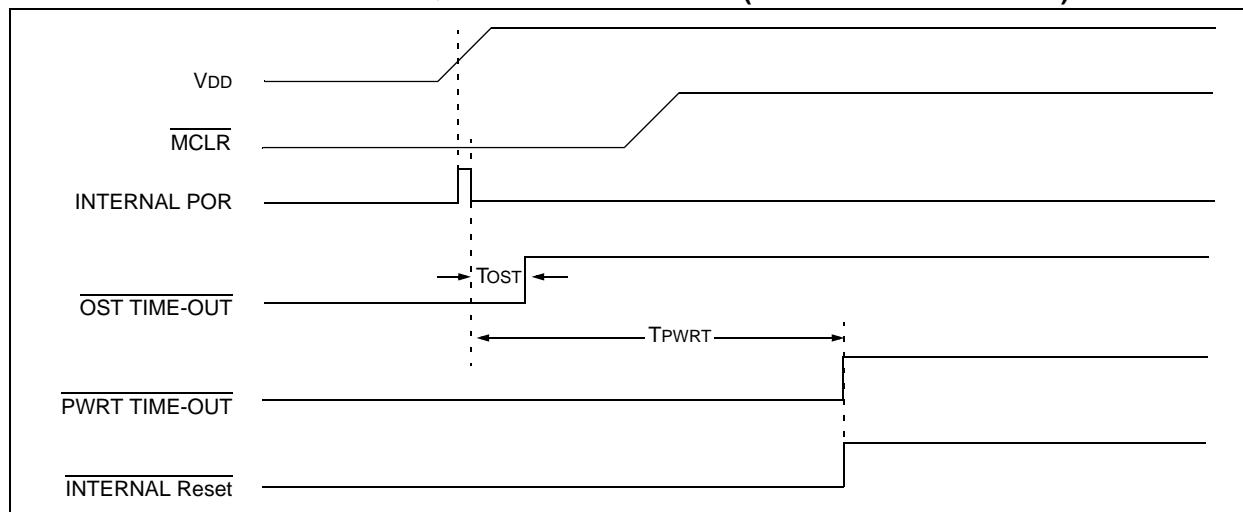
**FIGURE 17-2: RESET SYSTEM BLOCK DIAGRAM**



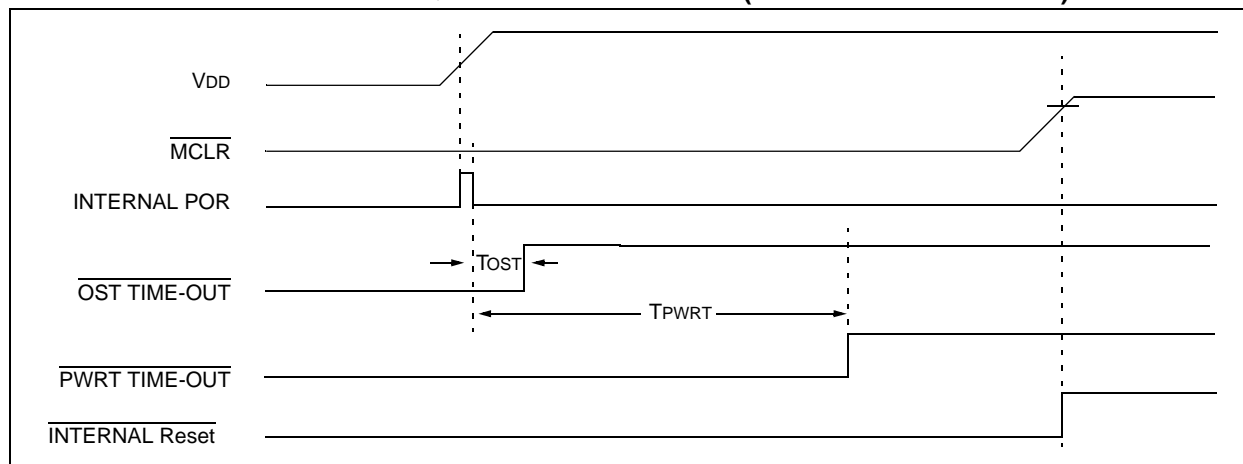
**FIGURE 17-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ )**



**FIGURE 17-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**



**FIGURE 17-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**



Any interrupt that is individually enabled (using the corresponding IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Sleep Status bit in the RCON register is set upon wake-up.

**Note:** In spite of various delays applied (TPOR, TLOCK and TPWRT), the crystal oscillator (and PLL) may not be active at the end of the time-out (e.g., for low-frequency crystals). In such cases, if FSCM is enabled, then the device will detect this as a clock failure and process the clock failure trap, the FRC oscillator will be enabled and the user will have to re-enable the crystal oscillator. If FSCM is not enabled, then the device will simply suspend execution of code until the clock is stable and will remain in Sleep until the oscillator clock has started.

All Resets will wake-up the processor from Sleep mode. Any Reset, other than POR, will set the Sleep Status bit. In a POR, the Sleep bit is cleared.

If the Watchdog Timer is enabled, then the processor will wake-up from Sleep mode upon WDT time-out. The Sleep and WDTO Status bits are both set.

## 17.6.2 IDLE MODE

In Idle mode, the clock to the CPU is shut down while peripherals keep running. Unlike Sleep mode, the clock source remains active.

Several peripherals have a control bit in each module that allows them to operate during Idle.

LPRC Fail-Safe Clock remains active if clock failure detect is enabled.

The processor wakes up from Idle if at least one of the following conditions has occurred:

- any interrupt that is individually enabled (IE bit is '1') and meets the required priority level
- any Reset (POR, BOR, MCLR)
- WDT time-out

Upon wake-up from Idle mode, the clock is re-applied to the CPU and instruction execution begins immediately, starting with the instruction following the PWRSAV instruction.

Any interrupt that is individually enabled (using IE bit) and meets the prevailing priority level will be able to wake-up the processor. The processor will process the interrupt and branch to the ISR. The Idle Status bit in the RCON register is set upon wake-up.

Any Reset other than POR will set the Idle Status bit. On a POR, the Idle bit is cleared.

If Watchdog Timer is enabled, then the processor will wake-up from Idle mode upon WDT time-out. The Idle and WDTO Status bits are both set.

Unlike wake-up from Sleep, there are no time delays involved in wake-up from Idle.

## 17.7 Device Configuration Registers

The Configuration bits in each device Configuration register specify some of the device modes and are programmed by a device programmer, or by using the In-Circuit Serial Programming™ (ICSP™) feature of the device. Each device Configuration register is a 24-bit register, but only the lower 16 bits of each register are used to hold configuration data. There are five device Configuration registers available to the user:

1. FOSC (0xF80000): Oscillator Configuration Register
2. FWDT (0xF80002): Watchdog Timer Configuration Register
3. FBORPOR (0xF80004): BOR and POR Configuration Register
4. FGS (0xF8000A): General Code Segment Configuration Register
5. FICD (0xF8000C): Debug Configuration Register

The placement of the Configuration bits is automatically handled when you select the device in your device programmer. The desired state of the Configuration bits may be specified in the source code (dependent on the language tool used), or through the programming interface. After the device has been programmed, the application software may read the Configuration bit values through the table read instructions. For additional information, please refer to the Programming Specifications of the device.

**Note:** If the code protection Configuration fuse bits (FGS<GCP> and FGS<GWRP>) have been programmed, an erase of the entire code-protected device is only possible at voltages  $V_{DD} \geq 4.5V$ .

## 17.8 Peripheral Module Disable (PMD) Registers

The Peripheral Module Disable (PMD) registers provide a method to disable a peripheral module by stopping all clock sources supplied to that module. When a peripheral is disabled via the appropriate PMD control bit, the peripheral is in a minimum power consumption state. The Control and Status registers associated with the peripheral will also be disabled so writes to those registers will have no effect and read values will be invalid.

A peripheral module will only be enabled if both the associated bit in the PMD register is cleared and the peripheral is supported by the specific dsPIC DSC variant. If the peripheral is present in the device, it is enabled in the PMD register by default.

**Note 1:** If a PMD bit is set, the corresponding module is disabled after a delay of 1 instruction cycle. Similarly, if a PMD bit is cleared, the corresponding module is enabled after a delay of 1 instruction cycle (assuming the module Control registers are already configured to enable module operation).

**2:** In dsPIC30F2011, dsPIC30F3012 and dsPIC30F2012 devices, the U2MD bit is readable and writable and will be read as '1' when set.

## 17.9 In-Circuit Debugger

When MPLAB® ICD 2 is selected as a Debugger, the In-Circuit Debugging functionality is enabled. This function allows simple debugging functions when used with MPLAB IDE. When the device has this feature enabled, some of the resources are not available for general use. These resources include the first 80 bytes of Data RAM and two I/O pins.

One of four pairs of Debug I/O pins may be selected by the user using configuration options in MPLAB IDE. These pin pairs are named EMUD/EMUC, EMUD1/EMUC1, EMUD2/EMUC2 and EMUD3/EMUC3.

In each case, the selected EMUD pin is the Emulation/Debug Data line, and the EMUC pin is the Emulation/Debug Clock line. These pins will interface to the MPLAB ICD 2 module available from Microchip. The selected pair of Debug I/O pins is used by MPLAB ICD 2 to send commands and receive responses, as well as to send and receive data. To use the In-Circuit Debugger function of the device, the design must implement ICSP connections to MCLR, VDD, VSS, PGC, PGD and the selected EMUDx/EMUCx pin pair.

This gives rise to two possibilities:

1. If EMUD/EMUC is selected as the Debug I/O pin pair, then only a 5-pin interface is required, as the EMUD and EMUC pin functions are multiplexed with the PGD and PGC pin functions in all dsPIC30F devices.
2. If EMUD1/EMUC1, EMUD2/EMUC2 or EMUD3/EMUC3 is selected as the Debug I/O pin pair, then a 7-pin interface is required, as the EMUDx/EMUCx pin functions (x = 1, 2 or 3) are not multiplexed with the PGD and PGC pin functions.



## 19.2 MPLAB C Compilers for Various Device Families

The MPLAB C Compiler code development systems are complete ANSI C compilers for Microchip's PIC18, PIC24 and PIC32 families of microcontrollers and the dsPIC30 and dsPIC33 families of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use.

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

## 19.3 HI-TECH C for Various Device Families

The HI-TECH C Compiler code development systems are complete ANSI C compilers for Microchip's PIC family of microcontrollers and the dsPIC family of digital signal controllers. These compilers provide powerful integration capabilities, omniscient code generation and ease of use.

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

The compilers include a macro assembler, linker, pre-processor, and one-step driver, and can run on multiple platforms.

## 19.4 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 IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 19.5 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. 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/librarian 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

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

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC devices. MPLAB C 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 IDE compatibility

# dsPIC30F2011/2012/3012/3013

**TABLE 20-8: DC CHARACTERISTICS: I/O PIN INPUT SPECIFICATIONS**

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended				
Param No.	Symbol	Characteristic	Min	Typ <sup>(1)</sup>	Max	Units	Conditions
DI10	V <sub>IL</sub>	<b>Input Low Voltage<sup>(2)</sup></b> I/O pins: with Schmitt Trigger buffer	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI15		<u>MCLR</u>	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI16		OSC1 (in XT, HS and LP modes)	V <sub>SS</sub>	—	0.2 V <sub>DD</sub>	V	
DI17		OSC1 (in RC mode) <sup>(3)</sup>	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	
DI18		SDA, SCL	V <sub>SS</sub>	—	0.3 V <sub>DD</sub>	V	SM bus disabled
DI19		SDA, SCL	V <sub>SS</sub>	—	0.8	V	SM bus enabled
DI20	V <sub>IH</sub>	<b>Input High Voltage<sup>(2)</sup></b> I/O pins: with Schmitt Trigger buffer	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI25		<u>MCLR</u>	0.8 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI26		OSC1 (in XT, HS and LP modes)	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI27		OSC1 (in RC mode) <sup>(3)</sup>	0.9 V <sub>DD</sub>	—	V <sub>DD</sub>	V	
DI28		SDA, SCL	0.7 V <sub>DD</sub>	—	V <sub>DD</sub>	V	SM bus disabled
DI29		SDA, SCL	2.1	—	V <sub>DD</sub>	V	SM bus enabled
DI30	ICNPU	<b>CNxx Pull-up Current<sup>(2)</sup></b>	50	250	400	μA	V <sub>DD</sub> = 5V, V <sub>PIN</sub> = V <sub>SS</sub>
DI50	I <sub>IL</sub>	<b>Input Leakage Current<sup>(2)(4)(5)</sup></b> I/O ports	—	0.01	±1	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at high impedance
DI51		Analog input pins	—	0.50	—	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , Pin at high impedance
DI55		<u>MCLR</u>	—	0.05	±5	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub>
DI56		OSC1	—	0.05	±5	μA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , XT, HS and LP Osc mode

**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:** In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the dsPIC30F device be driven with an external clock while in RC mode.

**4:** The leakage current on the MCLR pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

**5:** Negative current is defined as current sourced by the pin.

# dsPIC30F2011/2012/3012/3013

**TABLE 20-33: I<sup>2</sup>C™ BUS DATA TIMING REQUIREMENTS (MASTER MODE)**

AC CHARACTERISTICS				Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for Industrial -40°C ≤ TA ≤ +125°C for Extended			
Param No.	Symbol	Characteristic		Min <sup>(1)</sup>	Max	Units	Conditions
IM10	TLO:SCL	Clock Low Time	100 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			400 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM11	THI:SCL	Clock High Time	100 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			400 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM20	TF:SCL	SDA and SCL Fall Time	100 kHz mode	—	300	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 Cb	300	ns	
			1 MHz mode <sup>(2)</sup>	—	100	ns	
IM21	TR:SCL	SDA and SCL Rise Time	100 kHz mode	—	1000	ns	Cb is specified to be from 10 to 400 pF
			400 kHz mode	20 + 0.1 Cb	300	ns	
			1 MHz mode <sup>(2)</sup>	—	300	ns	
IM25	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns	
			400 kHz mode	100	—	ns	
			1 MHz mode <sup>(2)</sup>	—	—	ns	
IM26	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns	
			400 kHz mode	0	0.9	μs	
			1 MHz mode <sup>(2)</sup>	—	—	ns	
IM30	TSU:STA	Start Condition Setup Time	100 kHz mode	Tcy/2 (BRG + 1)	—	μs	Only relevant for Repeated Start condition
			400 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM31	THD:STA	Start Condition Hold Time	100 kHz mode	Tcy/2 (BRG + 1)	—	μs	After this period the first clock pulse is generated
			400 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM33	TSU:STO	Stop Condition Setup Time	100 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			400 kHz mode	Tcy/2 (BRG + 1)	—	μs	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	μs	
IM34	THD:STO	Stop Condition Hold Time	100 kHz mode	Tcy/2 (BRG + 1)	—	ns	
			400 kHz mode	Tcy/2 (BRG + 1)	—	ns	
			1 MHz mode <sup>(2)</sup>	Tcy/2 (BRG + 1)	—	ns	
IM40	TAA:SCL	Output Valid From Clock	100 kHz mode	—	3500	ns	
			400 kHz mode	—	1000	ns	
			1 MHz mode <sup>(2)</sup>	—	—	ns	
IM45	TBF:SDA	Bus Free Time	100 kHz mode	4.7	—	μs	Time the bus must be free before a new transmission can start
			400 kHz mode	1.3	—	μs	
			1 MHz mode <sup>(2)</sup>	—	—	μs	
IM50	CB	Bus Capacitive Loading		—	400	pF	

**Note 1:** BRG is the value of the I<sup>2</sup>C Baud Rate Generator. Refer to **Section 21. “Inter-Integrated Circuit™ (I<sup>2</sup>C)”** (DS70068) in the *dsPIC30F Family Reference Manual* (DS70046).

**2:** Maximum pin capacitance = 10 pF for all I<sup>2</sup>C™ pins (for 1 MHz mode only).

## INDEX

### Numerics

12-bit Analog-to-Digital Converter (A/D) Module ..... 113

### A

A/D .....	113
Aborting a Conversion .....	115
ADCHS Register .....	113
ADCON1 Register .....	113
ADCON2 Register .....	113
ADCON3 Register .....	113
ADCSSL Register .....	113
ADPCFG Register .....	113
Configuring Analog Port Pins .....	60, 119
Connection Considerations .....	119
Conversion Operation .....	114
Effects of a Reset .....	118
Operation During CPU Idle Mode .....	118
Operation During CPU Sleep Mode .....	118
Output Formats .....	118
Power-Down Modes .....	118
Programming the Sample Trigger .....	115
Register Map .....	121
Result Buffer .....	114
Sampling Requirements .....	117
Selecting the Conversion Sequence .....	114
AC Characteristics .....	160
Load Conditions .....	160
AC Temperature and Voltage Specifications .....	160
ADC .....	
Selecting the Conversion Clock .....	115
ADC Conversion Speeds .....	116
Address Generator Units .....	43
Alternate Vector Table .....	69
Analog-to-Digital Converter. <i>See</i> ADC.	
Assembler .....	
MPASM Assembler .....	146
Automatic Clock Stretch .....	100
During 10-bit Addressing (STREN = 1) .....	100
During 7-bit Addressing (STREN = 1) .....	100
Receive Mode .....	100
Transmit Mode .....	100

### B

Bandgap Start-up Time .....	
Requirements .....	166
Timing Characteristics .....	166
Barrel Shifter .....	27
Bit-Reversed Addressing .....	46
Example .....	47
Implementation .....	46
Modifier Values Table .....	47
Sequence Table (16-Entry) .....	47
Block Diagrams .....	
12-bit ADC Functional .....	113
16-bit Timer1 Module .....	73
16-bit Timer2 .....	79
16-bit Timer3 .....	79
32-bit Timer2/3 .....	78
DSP Engine .....	24
dsPIC30F2011 .....	12
dsPIC30F2012 .....	13
dsPIC30F3013 .....	15
External Power-on Reset Circuit .....	131

I <sup>2</sup> C .....	98
Input Capture Mode .....	83
Oscillator System .....	125
Output Compare Mode .....	87
Reset System .....	129
Shared Port Structure .....	59
SPI .....	94
SPI Master/Slave Connection .....	95
UART Receiver .....	106
UART Transmitter .....	105
BOR Characteristics .....	158
BOR. <i>See</i> Brown-out Reset.	
Brown-out Reset .....	
Characteristics .....	158
Timing Requirements .....	165

### C

C Compilers .....	
MPLAB C18 .....	146
CAN Module .....	
I/O Timing Characteristics .....	181
I/O Timing Requirements .....	181
CLKOUT and I/O Timing .....	
Characteristics .....	164
Requirements .....	164
Code Examples .....	
Data EEPROM Block Erase .....	56
Data EEPROM Block Write .....	58
Data EEPROM Read .....	55
Data EEPROM Word Erase .....	56
Data EEPROM Word Write .....	57
Erasing a Row of Program Memory .....	51
Initiating a Programming Sequence .....	52
Loading Write Latches .....	52
Code Protection .....	123
Control Registers .....	50
NVMADR .....	50
NVMADRU .....	50
NVMCON .....	50
NVMKEY .....	50
Core Architecture .....	
Overview .....	19
CPU Architecture Overview .....	19
Customer Change Notification Service .....	205
Customer Notification Service .....	205
Customer Support .....	205

### D

Data Accumulators and Adder/Subtractor .....	25
Data Space Write Saturation .....	27
Overflow and Saturation .....	25
Round Logic .....	26
Write-Back .....	26
Data Address Space .....	35
Alignment .....	38
Alignment (Figure) .....	38
Effect of Invalid Memory Accesses (Table) .....	38
MCU and DSP (MAC Class) Instructions Example ....	37
Memory Map .....	35, 36
Near Data Space .....	39
Software Stack .....	39
Spaces .....	38
Width .....	38
Data EEPROM Memory .....	55
Erasing .....	56
Erasing, Block .....	56

# dsPIC30F2011/2012/3012/3013

## PRODUCT IDENTIFICATION SYSTEM

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

