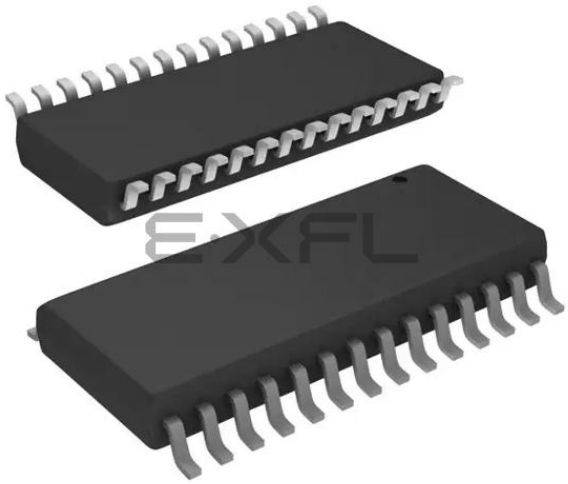


Welcome to E-XFL.COM

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"

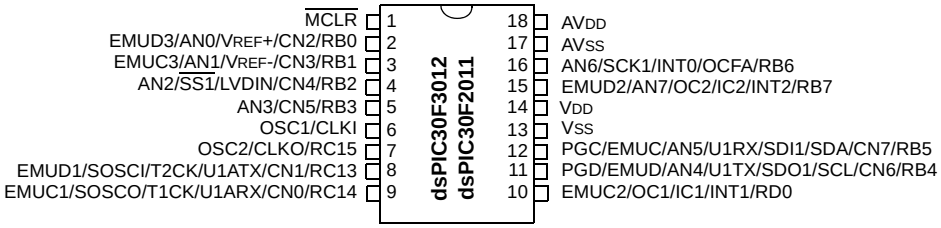
Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	20
Program Memory Size	24KB (8K x 24)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 10x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	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/dspic30f3013-20i-so

dsPIC30F2011/2012/3012/3013 Sensor Family

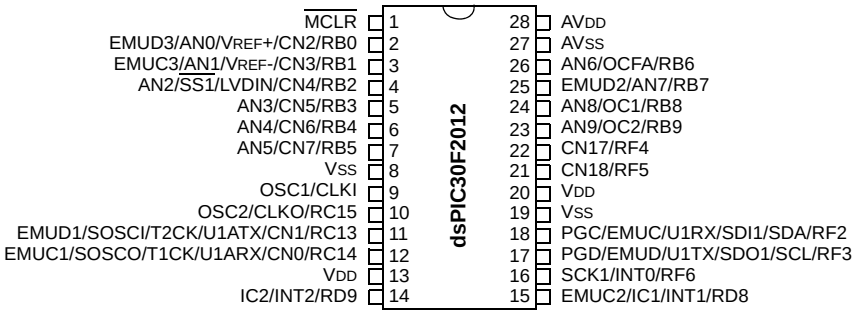
Device	Pins	Program Memory		SRAM Bytes	EEPROM Bytes	Timer 16-bit	Input Cap	Output Comp/Std PWM	A/D 12-bit 200 Ksps	UART	SPI	I^2C^{TM}
		Bytes	Instructions									
dsPIC30F2011	18	12K	4K	1024	–	3	2	2	8 ch	1	1	1
dsPIC30F3012	18	24K	8K	2048	1024	3	2	2	8 ch	1	1	1
dsPIC30F2012	28	12K	4K	1024	–	3	2	2	10 ch	1	1	1
dsPIC30F3013	28	24K	8K	2048	1024	3	2	2	10 ch	2	1	1

Pin Diagrams

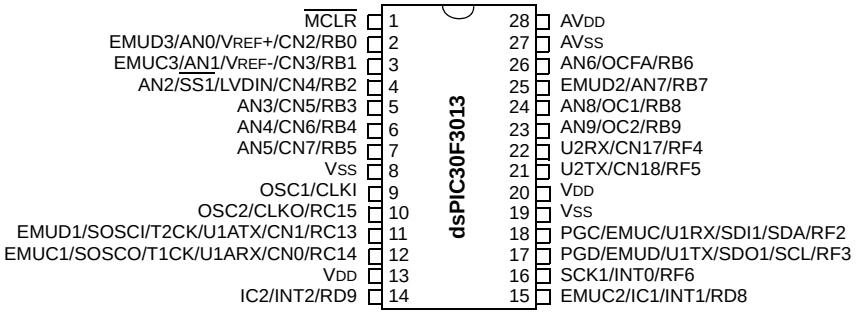
18-Pin PDIP and SOIC



28-Pin PDIP and SOIC



28-Pin SPDIP and SOIC



4.1.3 MOVE AND ACCUMULATOR INSTRUCTIONS

Move instructions and the DSP accumulator class of instructions provide a greater degree of addressing flexibility than other instructions. In addition to the addressing modes supported by most MCU instructions, move and accumulator instructions also support Register Indirect with Register Offset Addressing mode, also referred to as Register Indexed mode.

Note:	For the MOV instructions, the addressing mode specified in the instruction can differ for the source and destination EA. However, the 4-bit Wb (register offset) field is shared between both source and destination (but typically only used by one).

In summary, the following addressing modes are supported by move and accumulator instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- Register Indirect with Register Offset (Indexed)
- Register Indirect with Literal Offset
- 8-bit Literal
- 16-bit Literal

Note:	Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

4.1.4 MAC INSTRUCTIONS

The dual source operand DSP instructions (CLR, ED, EDAC, MAC, MPY, MPY.N, MOVSAC and MSC), also referred to as MAC instructions, utilize a simplified set of addressing modes to allow the user to effectively manipulate the data pointers through register indirect tables.

The two source operand prefetch registers must belong to the set {W8, W9, W10, W11}. For data reads, W8 and W9 are always directed to the X RAGU. W10 and W11 are always directed to the Y AGU. The effective addresses generated (before and after modification) must, therefore, be valid addresses within X data space for W8 and W9 and Y data space for W10 and W11.

Note:	Register Indirect with Register Offset addressing is only available for W9 (in X space) and W11 (in Y space).

In summary, the following addressing modes are supported by the MAC class of instructions:

- Register Indirect
- Register Indirect Post-modified by 2
- Register Indirect Post-modified by 4
- Register Indirect Post-modified by 6
- Register Indirect with Register Offset (Indexed)

4.1.5 OTHER INSTRUCTIONS

Besides the various addressing modes outlined above, some instructions use literal constants of various sizes. For example, BRA (branch) instructions use 16-bit signed literals to specify the branch destination directly, whereas the DISI instruction uses a 14-bit unsigned literal field. In some instructions, such as ADD Acc, the source of an operand or result is implied by the opcode itself. Certain operations, such as NOP, do not have any operands.

4.2 Modulo Addressing

Modulo Addressing is a method of providing an automated means to support circular data buffers using hardware. The objective is to remove the need for software to perform data address boundary checks when executing tightly looped code, as is typical in many DSP algorithms.

Modulo Addressing can operate in either data or program space (since the data pointer mechanism is essentially the same for both). One circular buffer can be supported in each of the X (which also provides the pointers into program space) and Y data spaces. Modulo Addressing can operate on any W register pointer. However, it is not advisable to use W14 or W15 for Modulo Addressing since these two registers are used as the Stack Frame Pointer and Stack Pointer, respectively.

In general, any particular circular buffer can only be configured to operate in one direction, as there are certain restrictions on the buffer Start address (for incrementing buffers), or end address (for decrementing buffers) based upon the direction of the buffer.

The only exception to the usage restrictions is for buffers that have a power-of-2 length. As these buffers satisfy the Start and the end address criteria, they can operate in a Bidirectional mode (i.e., address boundary checks are performed on both the lower and upper address boundaries).

5.0 FLASH PROGRAM MEMORY

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).

The dsPIC30F family of devices contains internal program Flash memory for executing user code. There are two methods by which the user can program this memory:

1. Run-Time Self-Programming (RTSP)
2. In-Circuit Serial Programming™ (ICSP™)

5.1 In-Circuit Serial Programming (ICSP)

dsPIC30F devices can be serially programmed while in the end application circuit. This is simply done with two lines for Programming Clock and Programming Data (which are named PGC and PGD respectively), and three other lines for Power (V_{DD}), Ground (V_{SS}) and Master Clear ($\overline{MCLR}$). This allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

5.2 Run-Time Self-Programming (RTSP)

RTSP is accomplished using TBLRD (table read) and TBLWT (table write) instructions.

With RTSP, the user may erase program memory, 32 instructions (96 bytes) at a time and can write program memory data, 32 instructions (96 bytes) at a time.

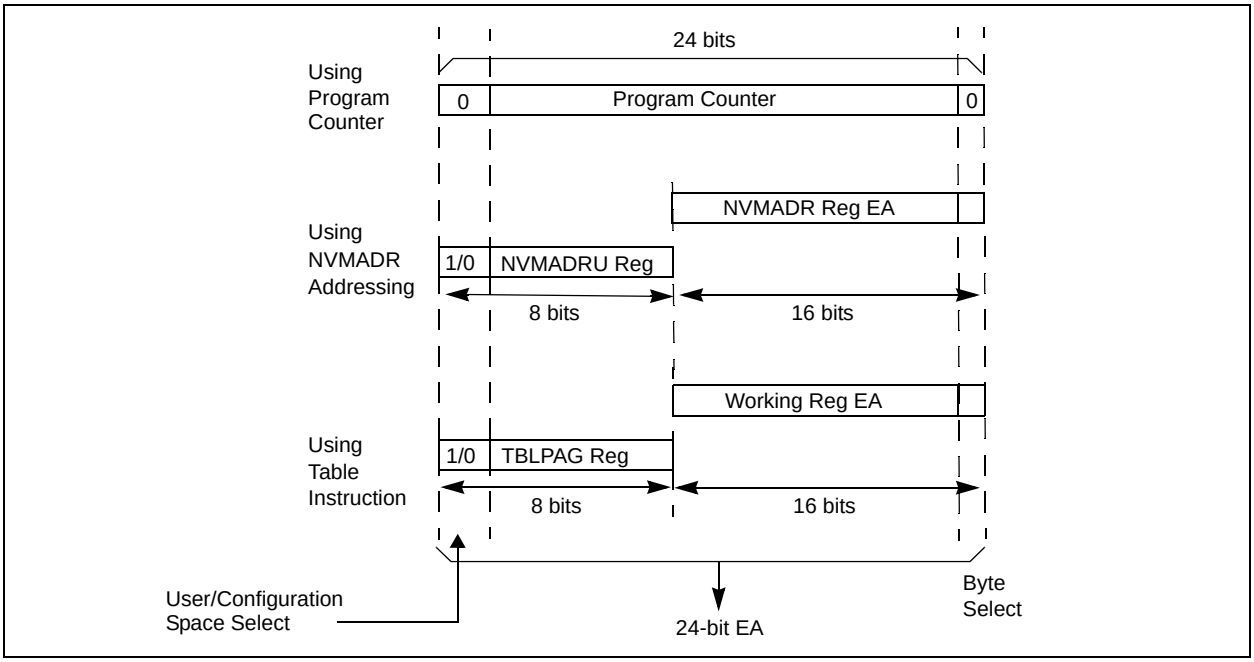
5.3 Table Instruction Operation Summary

The TBLRDL and the TBLWTL instructions are used to read or write to bits<15:0> of program memory. TBLRDL and TBLWTL can access program memory in Word or Byte mode.

The TBLRDH and TBLWTH instructions are used to read or write to bits<23:16> of program memory. TBLRDH and TBLWTH can access program memory in Word or Byte mode.

A 24-bit program memory address is formed using bits<7:0> of the TBLPAG register and the Effective Address (EA) from a W register specified in the table instruction, as shown in Figure 5-1.

FIGURE 5-1: ADDRESSING FOR TABLE AND NVM REGISTERS



6.0 DATA EEPROM MEMORY

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).

The data EEPROM memory is readable and writable during normal operation over the entire VDD range. The data EEPROM memory is directly mapped in the program memory address space.

The four SFRs used to read and write the program Flash memory are used to access data EEPROM memory, as well. As described in **Section 5.5 "Control Registers"**, these registers are:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

The EEPROM data memory allows read and write of single words and 16-word blocks. When interfacing to data memory, NVMADR, in conjunction with the NVMADRU register, are used to address the EEPROM location being accessed. TBLRDL and TBLWTL instructions are used to read and write data EEPROM. The dsPIC30F devices have up to 8 Kbytes (4K words) of data EEPROM with an address range from 0x7FF000 to 0x7FFFFE.

A word write operation should be preceded by an erase of the corresponding memory location(s). The write typically requires 2 ms to complete, but the write time varies with voltage and temperature.

A program or erase operation on the data EEPROM does not stop the instruction flow. The user is responsible for waiting for the appropriate duration of time before initiating another data EEPROM write/ erase operation. Attempting to read the data EEPROM while a programming or erase operation is in progress results in unspecified data.

Control bit WR initiates write operations similar to program Flash writes. This bit cannot be cleared, only set, in software. They are cleared in hardware at the completion of the write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation.

The WREN bit, when set, allows a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation. In these situations, following Reset, the user can check the WRERR bit and rewrite the location. The address register NVMADR remains unchanged.

Note: Interrupt flag bit NVMIF in the IFS0 register is set when write is complete. It must be cleared in software.

6.1 Reading the Data EEPROM

A TBLRD instruction reads a word at the current program word address. This example uses W0 as a pointer to data EEPROM. The result is placed in register W4 as shown in Example 6-1.

EXAMPLE 6-1: DATA EEPROM READ

```
MOV     #LOW_ADDR_WORD,W0  ; Init Pointer
MOV     #HIGH_ADDR_WORD,W1
MOV     W1,TBLPAG
TBLRDL  [ W0 ], W4         ; read data EEPROM
```

EXAMPLE 6-5: DATA EEPROM BLOCK WRITE

```
    MOV        #LOW_ADDR_WORD,W0  ; Init pointer
    MOV        #HIGH_ADDR_WORD,W1
    MOV        W1,TBLPAG
    MOV        #data1,W2          ; Get 1st data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data2,W2          ; Get 2nd data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data3,W2          ; Get 3rd data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data4,W2          ; Get 4th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data5,W2          ; Get 5th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data6,W2          ; Get 6th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data7,W2          ; Get 7th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data8,W2          ; Get 8th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data9,W2          ; Get 9th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data10,W2         ; Get 10th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data11,W2         ; Get 11th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data12,W2         ; Get 12th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data13,W2         ; Get 13th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data14,W2         ; Get 14th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data15,W2         ; Get 15th data
    TBLWTL     W2,[ W0]++         ; write data
    MOV        #data16,W2         ; Get 16th data
    TBLWTL     W2,[ W0]++         ; write data. The NVMADR captures last table access address.
    MOV        #0x400A,W0         ; Select data EEPROM for multi word op
    MOV        W0,NVMCON          ; Operate Key to allow program operation
    DISI       #5                 ; Block all interrupts with priority <7 for
                                  ; next 5 instructions
    MOV        #0x55,W0
    MOV        W0,NVMKEY          ; Write the 0x55 key
    MOV        #0xAA,W1
    MOV        W1,NVMKEY          ; Write the 0xAA key
    BSET       NVMCON,#WR         ; Start write cycle
    NOP
    NOP
```

6.4 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

6.5 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared; also, the Power-up Timer prevents EEPROM write.

The write initiate sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch, or software malfunction.

8.0 INTERRUPTS

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).

The dsPIC30F sensor family has up to 21 interrupt sources and 4 processor exceptions (traps) which must be arbitrated based on a priority scheme.

The CPU is responsible for reading the Interrupt Vector Table (IVT) and transferring the address contained in the interrupt vector to the program counter. The interrupt vector is transferred from the program data bus into the program counter via a 24-bit wide multiplexer on the input of the program counter.

The Interrupt Vector Table (IVT) and Alternate Interrupt Vector Table (AIVT) are placed near the beginning of program memory (0x000004). The IVT and AIVT are shown in Figure 8-1.

The interrupt controller is responsible for pre-processing the interrupts and processor exceptions before they are presented to the processor core. The peripheral interrupts and traps are enabled, prioritized and controlled using centralized Special Function Registers (SFRs):

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0>
  All interrupt request flags are maintained in these three registers. The flags are set by their respective peripherals or external signals and they are cleared via software.
- IEC0<15:0>, IEC1<15:0>, IEC2<15:0>
  All interrupt enable control bits are maintained in these three registers. These control bits are used to individually enable interrupts from the peripherals or external signals.
- IPC0<15:0> through IPC10<7:0>
  The user assignable priority level associated with each of these 41 interrupts is held centrally in these eleven registers.
- IPL<3:0>
  The current CPU priority level is explicitly stored in the IPL bits. IPL<3> is present in the CORCON register, whereas IPL<2:0> are present in the STATUS register (SR) in the processor core.
- INTCON1<15:0>, INTCON2<15:0>
  Global interrupt control functions are derived from these two registers. INTCON1 contains the control and status flags for the processor exceptions. The INTCON2 register controls the external interrupt request signal behavior and the use of the alternate vector table.

Note:	Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

All interrupt sources can be user assigned to one of 7 priority levels, 1 through 7, through the IPCx registers. Each interrupt source is associated with an interrupt vector, as shown in Table 8-1. Levels 7 and 1 represent the highest and lowest maskable priorities, respectively.

Note:	Assigning a priority level of '0' to an interrupt source is equivalent to disabling that interrupt.

If the NSTDIS bit (INTCON1<15>) is set, nesting of interrupts is prevented. Thus, if an interrupt is currently being serviced, processing of a new interrupt is prevented even if the new interrupt is of higher priority than the one currently being serviced.

Note:	The IPL bits become read-only whenever the NSTDIS bit has been set to '1'.

Certain interrupts have specialized control bits for features like edge or level triggered interrupts, interrupt-on-change, etc. Control of these features remains within the peripheral module which generates the interrupt.

The `DISI` instruction can be used to disable the processing of interrupts of priorities 6 and lower for a certain number of instructions, during which the DISI bit (INTCON2<14>) remains set.

When an interrupt is serviced, the PC is loaded with the address stored in the vector location in program memory that corresponds to the interrupt. There are 63 different vectors within the IVT (refer to Table 8-1). These vectors are contained in locations 0x000004 through 0x0000FE of program memory (refer to Table 8-1). These locations contain 24-bit addresses, and in order to preserve robustness, an address error trap takes place if the PC attempts to fetch any of these words during normal execution. This prevents execution of random data as a result of accidentally decrementing a PC into vector space, accidentally mapping a data space address into vector space, or the PC rolling over to 0x000000 after reaching the end of implemented program memory space. Execution of a `GOTO` instruction to this vector space also generates an address error trap.

14.12.2 I²C MASTER RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (I2CCON<3>). The I²C module must be Idle before the RCEN bit is set, otherwise the RCEN bit will be disregarded. The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin ACK and data are shifted into the I2CRSR on the rising edge of each clock.

14.12.3 BAUD RATE GENERATOR

In I²C Master mode, the reload value for the BRG is located in the I2CBRG register. When the BRG is loaded with this value, the BRG counts down to '0' and stops until another reload has taken place. If clock arbitration is taking place, for instance, the BRG is reloaded when the SCL pin is sampled high.

As per the I²C standard, FSCK may be 100 kHz or 400 kHz. However, the user can specify any baud rate up to 1 MHz. I2CBRG values of '0' or '1' are illegal.

EQUATION 14-1: SERIAL CLOCK RATE

$$I2CBRG = \left(\frac{FCY}{FSCL} - \frac{FCY}{1{,}111{,}111} \right) - 1$$

14.12.4 CLOCK ARBITRATION

Clock arbitration occurs when the master de-asserts the SCL pin (SCL allowed to float high) during any receive, transmit, or Restart/Stop condition. When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of I2CBRG and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device.

14.12.5 MULTI-MASTER COMMUNICATION, BUS COLLISION, AND BUS ARBITRATION

Multi-master operation support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high while another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the MI2CIF pulse and reset the master portion of the I²C port to its Idle state.

If a transmit was in progress when the bus collision occurred, the transmission is halted, the TBF flag is cleared, the SDA and SCL lines are de-asserted and a value can now be written to I2CTRN. When the user services the I²C master event Interrupt Service Routine, if the I²C bus is free (i.e., the P bit is set), the user can resume communication by asserting a Start condition.

If a Start, Restart, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are de-asserted, and the respective control bits in the I2CCON register are cleared to '0'. When the user services the bus collision Interrupt Service Routine, and if the I²C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins, and if a Stop condition occurs, the MI2CIF bit will be set.

A write to the I2CTRN will start the transmission of data at the first data bit regardless of where the transmitter left off when bus collision occurred.

In a multi-master environment, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I²C bus can be taken when the P bit is set in the I2CSTAT register, or the bus is Idle and the S and P bits are cleared.

14.13 I²C Module Operation During CPU Sleep and Idle Modes

14.13.1 I²C OPERATION DURING CPU SLEEP MODE

When the device enters Sleep mode, all clock sources to the module are shut down and stay at logic '0'. If Sleep occurs in the middle of a transmission and the state machine is partially into a transmission as the clocks stop, then the transmission is aborted. Similarly, if Sleep occurs in the middle of a reception, then the reception is aborted.

14.13.2 I²C OPERATION DURING CPU IDLE MODE

For the I²C, the I2CSIDL bit selects if the module will stop on Idle or continue on Idle. If I2CSIDL = 0, the module will continue operation on assertion of the Idle mode. If I2CSIDL = 1, the module will stop on Idle.

17.2.3 LP OSCILLATOR CONTROL

Enabling the LP oscillator is controlled with two elements:

- The current oscillator group bits COSC<2:0>.
- The LPOSCEN bit (OSCCON register).

The LP oscillator is on (even during Sleep mode) if LPOSCEN = 1. The LP oscillator is the device clock if:

- COSC<2:0> = 000 (LP selected as main osc.) and
- LPOSCEN = 1

Keeping the LP oscillator on at all times allows for a fast switch to the 32 kHz system clock for lower power operation. Returning to the faster main oscillator will still require a start-up time

17.2.4 PHASE LOCKED LOOP (PLL)

The PLL multiplies the clock which is generated by the primary oscillator or Fast RC oscillator. The PLL is selectable to have either gains of x4, x8, and x16. Input and output frequency ranges are summarized in Table 17-3.

TABLE 17-3: PLL FREQUENCY RANGE

FIN	PLL Multiplier	FOUT
4 MHz-10 MHz	x4	16 MHz-40 MHz
4 MHz-10 MHz	x8	32 MHz-80 MHz
4 MHz-7.5 MHz	x16	64 MHz-120 MHz

The PLL features a lock output which is asserted when the PLL enters a phase locked state. Should the loop fall out of lock (e.g., due to noise), the lock signal will be rescinded. The state of this signal is reflected in the read-only LOCK bit in the OSCCON register.

17.2.5 FAST RC OSCILLATOR (FRC)

The FRC oscillator is a fast (7.37 MHz ±2% nominal) internal RC oscillator. This oscillator is intended to provide reasonable device operating speeds without the use of an external crystal, ceramic resonator, or RC network. The FRC oscillator can be used with the PLL to obtain higher clock frequencies.

The dsPIC30F operates from the FRC oscillator whenever the current oscillator selection control bits in the OSCCON register (OSCCON<14:12>) are set to '001'.

The four bit field specified by TUN<3:0> (OSCTUN <3:0>) allows the user to tune the internal fast RC oscillator (nominal 7.37 MHz). The user can tune the FRC oscillator within a range of +10.5% (840 kHz) and -12% (960 kHz) in steps of 1.50% around the factory calibrated setting, as shown in Table 17-4.

Note: OSCTUN functionality has been provided to help customers compensate for temperature effects on the FRC frequency over a wide range of temperatures. The tuning step size is an approximation and is neither characterized nor tested.

If OSCCON<14:12> are set to '111' and FPR<4:0> are set to '00001', '01010' or '00011', a PLL multiplier of 4, 8 or 16 (respectively) is applied.

Note: When a 16x PLL is used, the FRC frequency must not be tuned to a frequency greater than 7.5 MHz.

TABLE 17-4: FRC TUNING

TUN<3:0> Bits	FRC Frequency
0111	+ 10.5%
0110	+ 9.0%
0101	+ 7.5%
0100	+ 6.0%
0011	+ 4.5%
0010	+ 3.0%
0001	+ 1.5%
0000	Center Frequency (oscillator is running at calibrated frequency)
1111	- 1.5%
1110	- 3.0%
1101	- 4.5%
1100	- 6.0%
1011	- 7.5%
1010	- 9.0%
1001	- 10.5%
1000	- 12.0%

17.2.6 LOW-POWER RC OSCILLATOR (LPRC)

The LPRC oscillator is a component of the Watchdog Timer (WDT) and oscillates at a nominal frequency of 512 kHz. The LPRC oscillator is the clock source for the Power-up Timer (PWRT) circuit, WDT and clock monitor circuits. It may also be used to provide a low-frequency clock source option for applications where power consumption is critical and timing accuracy is not required.

The LPRC oscillator is always enabled at a Power-on Reset because it is the clock source for the PWRT. After the PWRT expires, the LPRC oscillator will remain on if one of the following is true:

- The Fail-Safe Clock Monitor is enabled
- The WDT is enabled
- The LPRC oscillator is selected as the system clock via the COSC<2:0> control bits in the OSCCON register

If one of the above conditions is not true, the LPRC will shut-off after the PWRT expires.

Note 1: OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<4:0>).

2: OSC1 pin cannot be used as an I/O pin even if the secondary oscillator or an internal clock source is selected at all times.

FIGURE 17-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

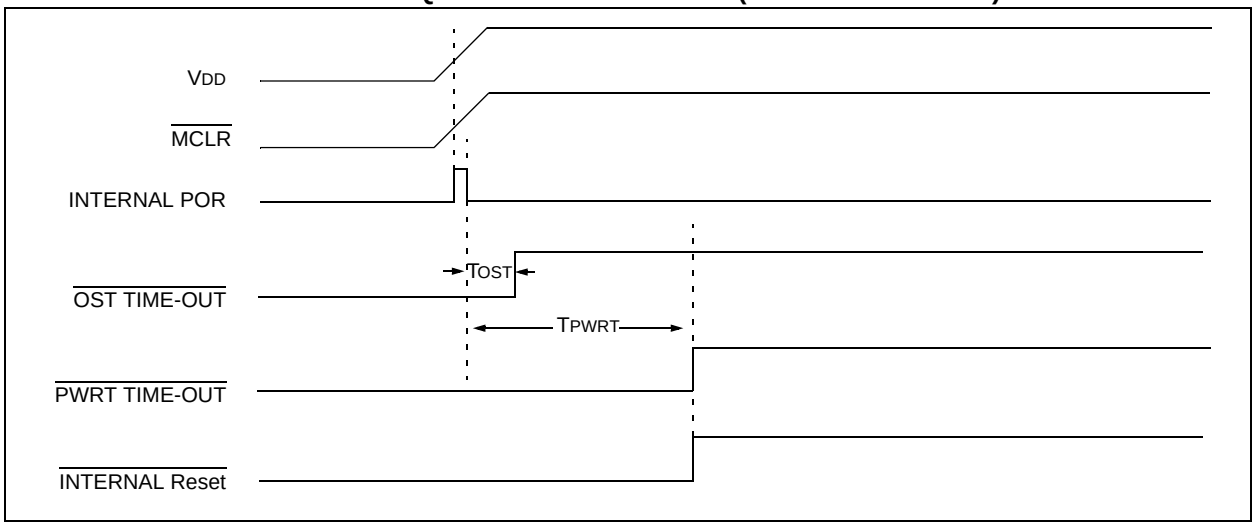


FIGURE 17-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

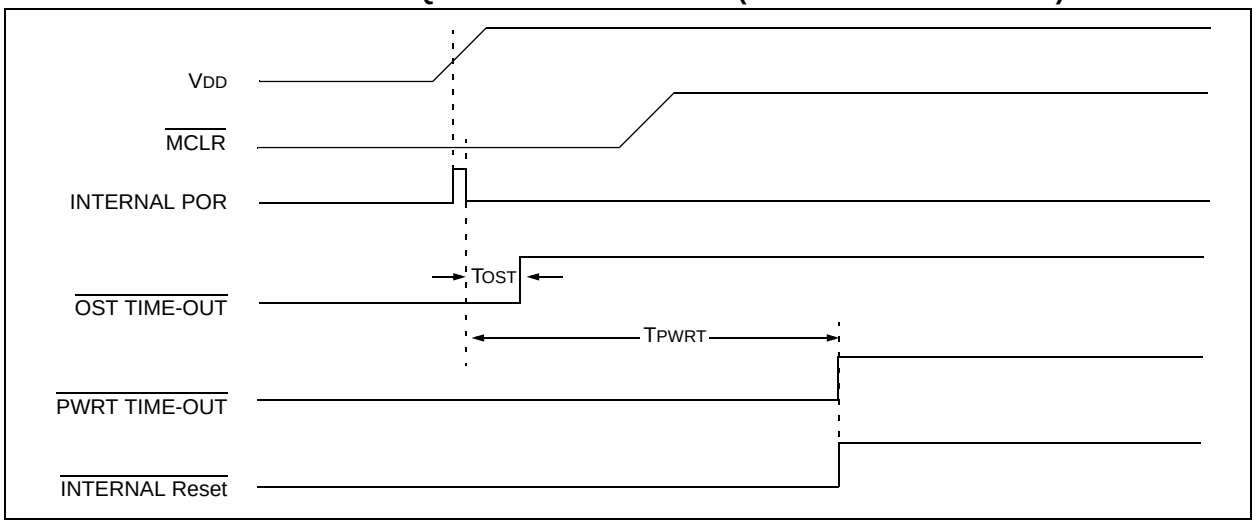


FIGURE 17-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

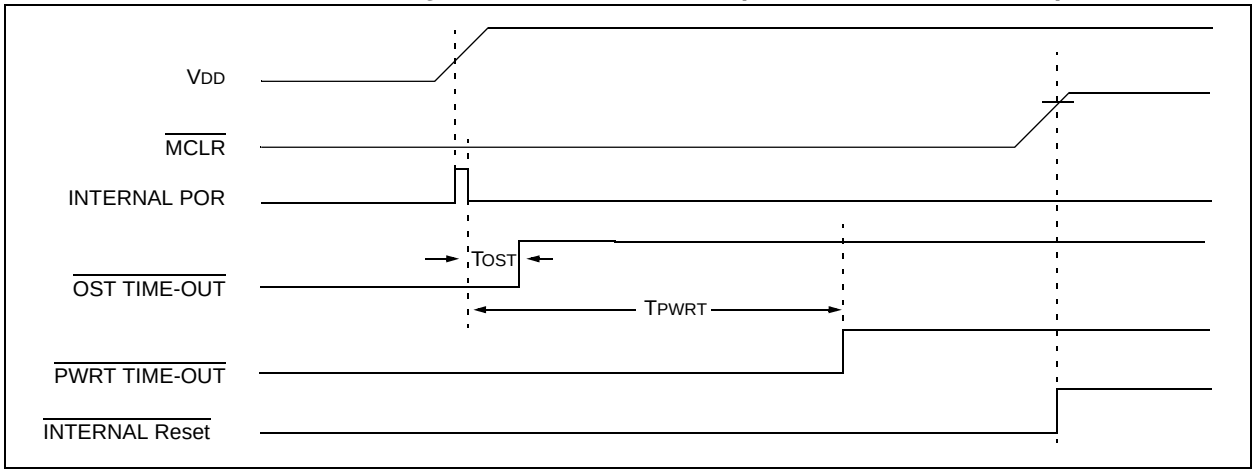


TABLE 17-7: SYSTEM INTEGRATION REGISTER MAP

SFR Name	Address	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
RCON	0740	TRAPR	IOPUWR	BGST	LVDEN	LVDL<3:0>				EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	(Note 1)
OSCCON	0742	—	COSC<2:0>			—	NOSC<2:0>			POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN	(Note 2)
OSCTUN	0744	—	—	—	—	—	—	—	—	—	—	—	—	TUN3	TUN2	TUN1	TUN0	(Note 2)
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	—	—	I2CMD	U2MD(3)	U1MD	—	SPI1MD	—	—	ADCMD	0000 0000 0000 0000
PMD2	0772	—	—	—	—	—	—	IC2MD	IC1MD	—	—	—	—	—	—	OC2MD	OC1MD	0000 0000 0000 0000

Legend: — = unimplemented bit, read as '0'

Note 1: Reset state depends on type of reset.

2: Reset state depends on Configuration bits.

3: Only available on dsPIC30F3013 devices.

TABLE 17-8: DEVICE CONFIGURATION REGISTER MAP

Name	Address	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
FOSC	F80000	FCKSM<1:0>		—	—	—	FOS<2:0>			—	—	—	FPR<4:0>				
FWDT	F80002	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
FBORPOR	F80004	MCLREN	—	—	—	—	PWMPIN(1)	HPOL(1)	LPOL(1)	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
FBS	F80006	—	—	Reserved(2)		—	—	—	Reserved(2)	—	—	—	—	Reserved(2)			
FSS	F80008	—	—	Reserved(2)		—	—	Reserved(2)		—	—	—	—	Reserved(2)			
FGS	F8000A	—	—	—	—	—	—	—	—	—	—	—	—	—	Reserved(3)	GCP	GWRP
FICD	F8000C	BKBUG	COE	—	—	—	—	—	—	—	—	—	—	—	—	ICS<1:0>	

Legend: — = unimplemented bit, read as '0'

Note 1: These bits are reserved (read as '1' and must be programmed as '1').

2: Reserved bits read as '1' and must be programmed as '1'.

3: The FGS<2> bit is a read-only copy of the GCP bit (FGS<1>).

19.0 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

19.1 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.

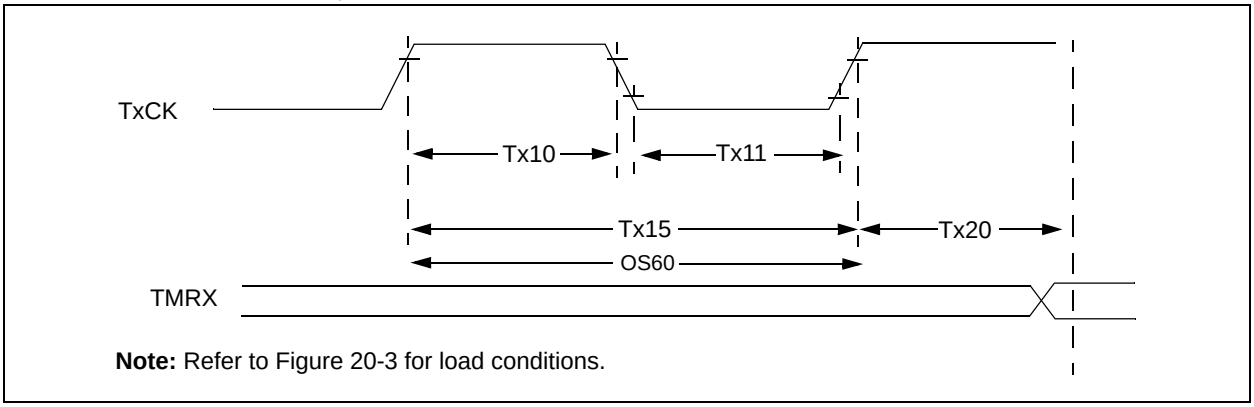
TABLE 20-12: DC CHARACTERISTICS: PROGRAM AND EEPROM

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(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.

FIGURE 20-8: TYPE A, B AND C TIMER EXTERNAL CLOCK TIMING CHARACTERISTICS



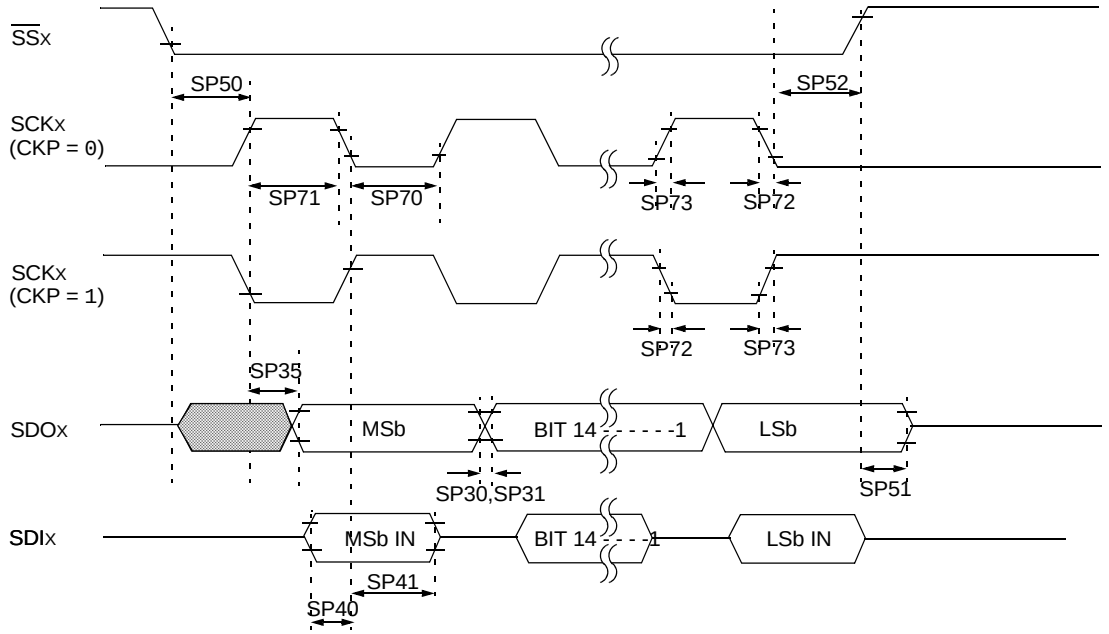
Note: Refer to Figure 20-3 for load conditions.

TABLE 20-23: TYPE A TIMER (TIMER1) EXTERNAL CLOCK TIMING REQUIREMENTS

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**	**Typ**	**Max**	**Units**	**Conditions**
TA10	TTXH	TxCK High Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA11	TTXL	TxCK Low Time	Synchronous, no prescaler	0.5 TCY + 20	—	—	ns	Must also meet parameter TA15
			Synchronous, with prescaler	10	—	—	ns	
			Asynchronous	10	—	—	ns	
TA15	TTXP	TxCK Input Period	Synchronous, no prescaler	TCY + 10	—	—	ns	
			Synchronous, with prescaler	Greater of: 20 ns or (TCY + 40)/N	—	—	—	N = prescale value (1, 8, 64, 256)
			Asynchronous	20	—	—	ns	
OS60	Ft1	SOSC1/T1CK oscillator input frequency range (oscillator enabled by setting bit TCS (T1CON, bit 1))		DC	—	50	kHz	
TA20	TCKEXTMRL	Delay from External TxCK Clock Edge to Timer Increment		0.5 TCY	—	1.5 TCY	—	

Note: Timer1 is a Type A.

FIGURE 20-14: SPI MODULE SLAVE MODE (CKE = 0) TIMING CHARACTERISTICS



Note: Refer to Figure 20-3 for load conditions.

TABLE 20-31: SPI MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS

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(1)**	**Min**	**Typ(2)**	**Max**	**Units**	**Conditions**
SP70	TscL	SCKx Input Low Time	30	—	—	ns	—
SP71	TscH	SCKx Input High Time	30	—	—	ns	—
SP72	TscF	SCKx Input Fall Time(3)	—	10	25	ns	—
SP73	TscR	SCKx Input Rise Time(3)	—	10	25	ns	—
SP30	TdoF	SDOx Data Output Fall Time(3)	—	—	—	ns	See DO32
SP31	TdoR	SDOx Data Output Rise Time(3)	—	—	—	ns	See DO31
SP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	—	30	ns	—
SP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	20	—	—	ns	—
SP50	TssL2scH, TssL2scL	$\overline{SS}$x↓to SCKx↑ or SCKx↓Input	120	—	—	ns	—
SP51	TssH2doZ	$\overline{SS}$x↑ to SDOx Output high impedance(3)	10	—	50	ns	—
SP52	TscH2ssH TscL2ssH	$\overline{SS}$x after SCK Edge	1.5 TCY +40	—	—	ns	—

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

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

3: Assumes 50 pF load on all SPI pins.

FIGURE 20-18: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS (SLAVE MODE)

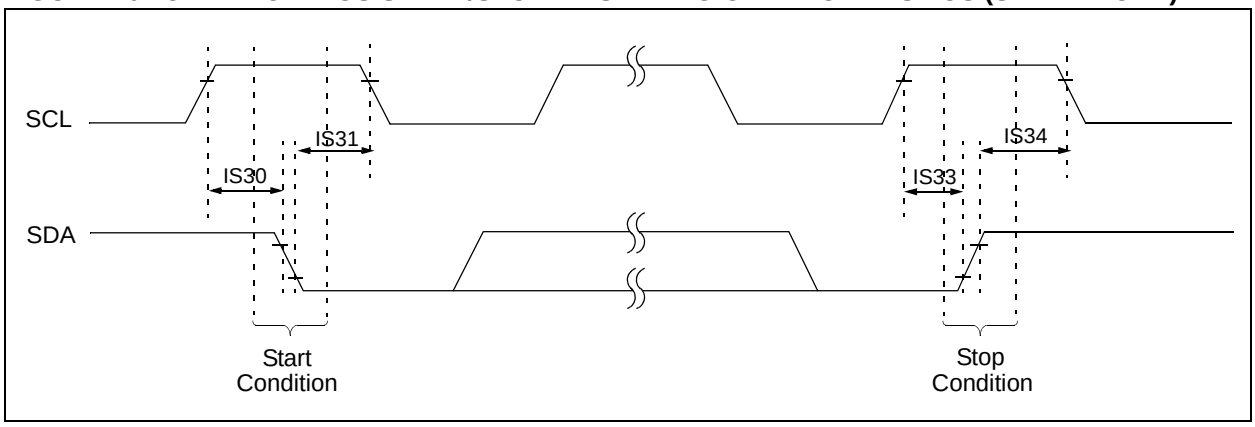


FIGURE 20-19: I²C™ BUS DATA TIMING CHARACTERISTICS (SLAVE MODE)

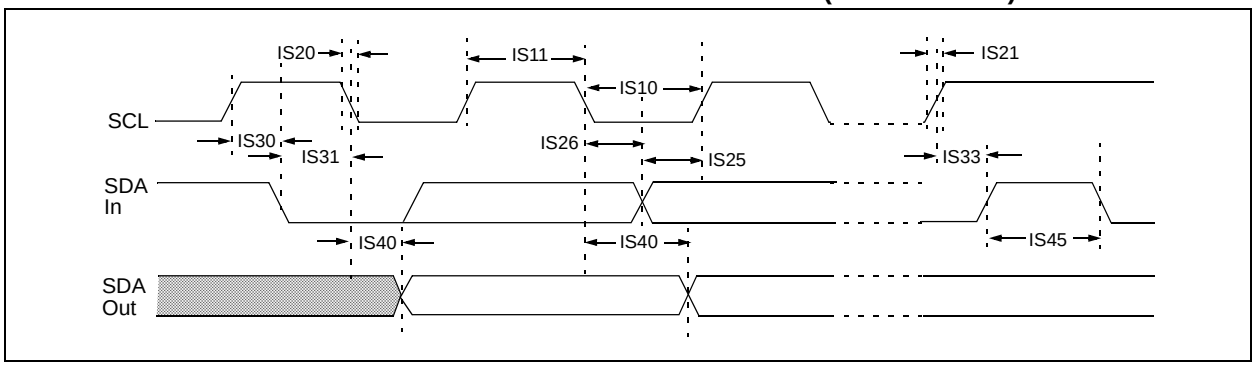


TABLE 20-34: I²C™ BUS DATA TIMING REQUIREMENTS (SLAVE MODE)

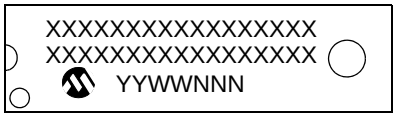
AC CHARACTERISTICS			**Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)** Operating temperature -40°C ≤TA ≤+85°C for Industrial -40°C ≤TA ≤+125°C for Extended				
Param No.	**Symbol**	**Characteristic**		**Min**	**Max**	**Units**	**Conditions**
IS10	TLO:SCL	Clock Low Time	100 kHz mode	4.7	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	1.3	—	µs	Device must operate at a minimum of 10 MHz.
			1 MHz mode(1)	0.5	—	µs	
IS11	THI:SCL	Clock High Time	100 kHz mode	4.0	—	µs	Device must operate at a minimum of 1.5 MHz
			400 kHz mode	0.6	—	µs	Device must operate at a minimum of 10 MHz
			1 MHz mode(1)	0.5	—	µs	
IS20	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(1)	—	100	ns	
IS21	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(1)	—	300	ns	

Note 1: Maximum pin capacitance = 10 pF for all I²C™ pins (for 1 MHz mode only).

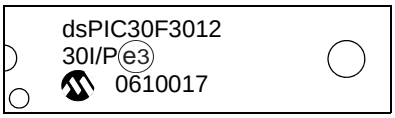
21.0 PACKAGING INFORMATION

21.1 Package Marking Information

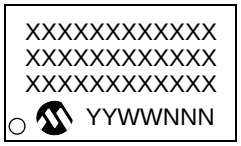
18-Lead PDIP



Example

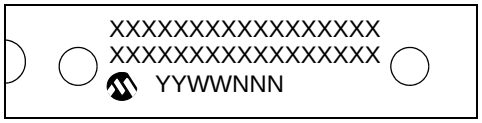


18-Lead SOIC

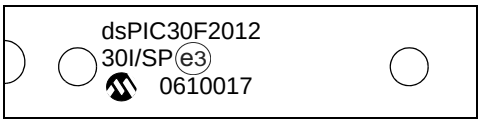


Example

28-Lead SPDIP



Example

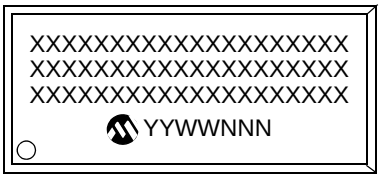


Legend:	XX...X	Customer-specific information
	Y	Year code (last digit of calendar year)
	YY	Year code (last 2 digits of calendar year)
	WW	Week code (week of January 1 is week ‘01’)
	NNN	Alphanumeric traceability code
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)
	*	This package is Pb-free. The Pb-free JEDEC designator ((e3)) can be found on the outer packaging for this package.

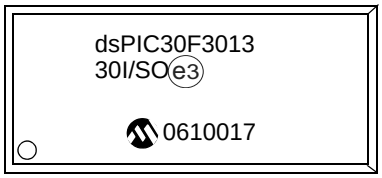
Note: In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

21.2 Package Marking Information (Continued)

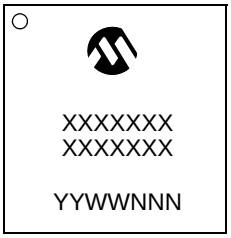
28-Lead SOIC



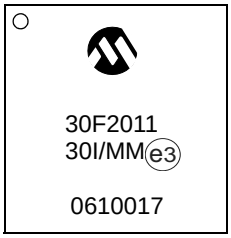
Example



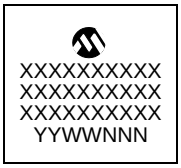
28-Lead QFN-S



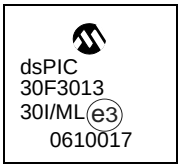
Example



44-Lead QFN

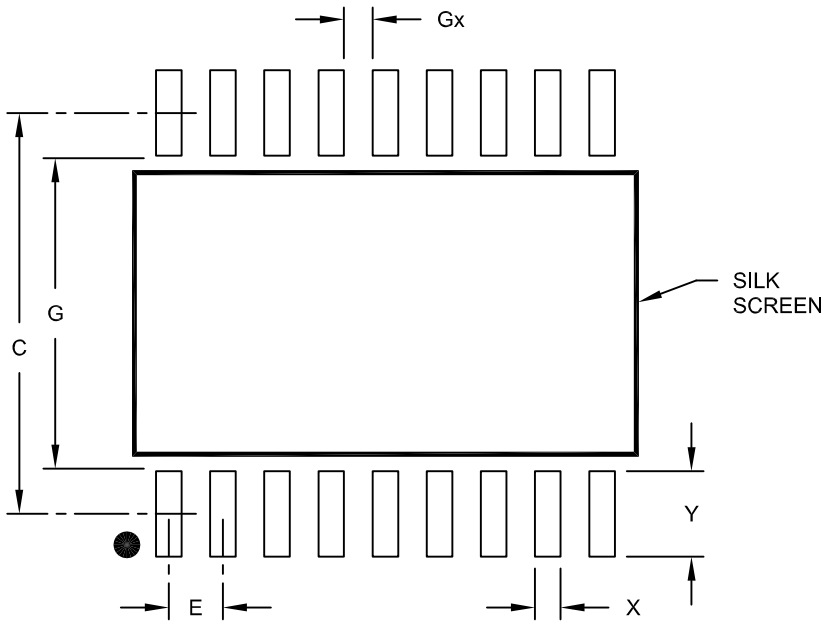


Example



18-Lead Plastic Small Outline (SO) - Wide, 7.50 mm Body [SOIC]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		1.27 BSC	
Contact Pad Spacing	C		9.40	
Contact Pad Width	X			0.60
Contact Pad Length	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

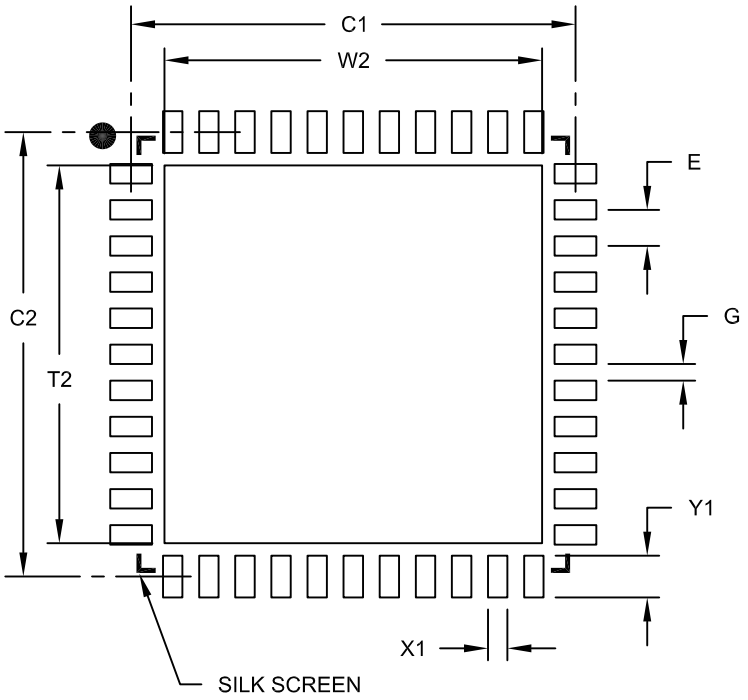
1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2051A

44-Lead Plastic Quad Flat, No Lead Package (ML) – 8x8 mm Body [QFN]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.65 BSC	
Optional Center Pad Width	W2			6.80
Optional Center Pad Length	T2			6.80
Contact Pad Spacing	C1		8.00	
Contact Pad Spacing	C2		8.00	
Contact Pad Width (X44)	X1			0.35
Contact Pad Length (X44)	Y1			0.80
Distance Between Pads	G	0.25		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

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

Microchip Technology Drawing No. C04-2103A

NOTES: