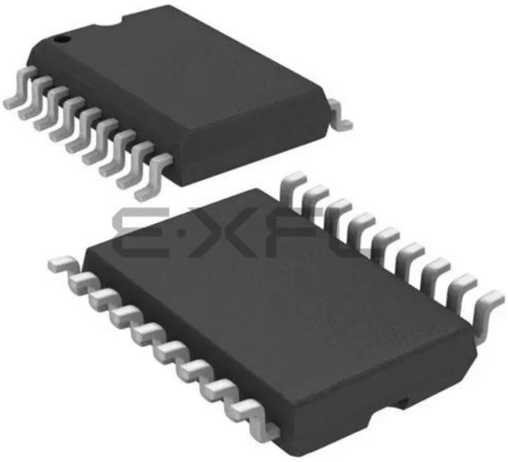
Microchip Technology - DSPIC30F2011-30I/SO Datasheet



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	30 MIPs
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	12
Program Memory Size	12KB (4K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-SOIC (0.295", 7.50mm Width)
Supplier Device Package	18-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f2011-30i-so

NOTES:

The core does not support a multi-stage instruction pipeline. However, a single-stage instruction prefetch mechanism is used, which accesses and partially decodes instructions a cycle ahead of execution, in order to maximize available execution time. Most instructions execute in a single cycle with certain exceptions.

The core features a vectored exception processing structure for traps and interrupts, with 62 independent vectors. The exceptions consist of up to 8 traps (of which 4 are reserved) and 54 interrupts. Each interrupt is prioritized based on a user-assigned priority between 1 and 7 (1 being the lowest priority and 7 being the highest), in conjunction with a predetermined 'natural order'. Traps have fixed priorities ranging from 8 to 15.

2.2 Programmer's Model

The programmer's model is shown in Figure 2-1 and consists of 16 x 16-bit working registers (W0 through W15), 2 x 40-bit accumulators (ACCA and ACCB), STATUS register (SR), Data Table Page register (TBLPAG), Program Space Visibility Page register (PSVPAG), DO and REPEAT registers (DOSTART, DOEND, DCOUNT and RCOUNT) and Program Counter (PC). The working registers can act as data, address or offset registers. All registers are memory mapped. W0 acts as the W register for file register addressing.

Some of these registers have a shadow register associated with each of them, as shown in Figure 2-1. The shadow register is used as a temporary holding register and can transfer its contents to or from its host register upon the occurrence of an event. None of the shadow registers are accessible directly. The following rules apply for transfer of registers into and out of shadows.

- `PUSH.S` and `POP.S`
  W0, W1, W2, W3, SR (DC, N, OV, Z and C bits only) are transferred.
- `DO` instruction
  DOSTART, DOEND, DCOUNT shadows are pushed on loop start and popped on loop end.

When a byte operation is performed on a working register, only the Least Significant Byte (LSB) of the target register is affected. However, a benefit of memory mapped working registers is that both the Least and Most Significant Bytes (MSB) can be manipulated through byte-wide data memory space accesses.

2.2.1 SOFTWARE STACK POINTER/ FRAME POINTER

The dsPIC® DSC devices contain a software stack. W15 is the dedicated Software Stack Pointer (SP), which is automatically modified by exception processing and subroutine calls and returns. However, W15 can be referenced by any instruction in the same manner as all other W registers. This simplifies the reading, writing and manipulation of the Stack Pointer (e.g., creating stack frames).

Note: In order to protect against misaligned stack accesses, W15<0> is always clear.

W15 is initialized to 0x0800 during a Reset. The user may reprogram the SP during initialization to any location within data space.

W14 has been dedicated as a Stack Frame Pointer, as defined by the `LNK` and `ULNK` instructions. However, W14 can be referenced by any instruction in the same manner as all other W registers.

2.2.2 STATUS REGISTER

The dsPIC DSC core has a 16-bit STATUS register (SR), the LSB of which is referred to as the SR Low byte (SRL) and the MSB as the SR High byte (SRH). See Figure 2-1 for SR layout.

SRL contains all the MCU ALU operation Status flags (including the Z bit), as well as the CPU Interrupt Priority Level Status bits, IPL<2:0>, and the Repeat Active Status bit, RA. During exception processing, SRL is concatenated with the MSB of the PC to form a complete word value which is then stacked.

The upper byte of the STATUS register contains the DSP Adder/Subtracter Status bits, the DO Loop Active bit (DA) and the Digit Carry (DC) Status bit.

2.2.3 PROGRAM COUNTER

The program counter is 23 bits wide; bit 0 is always clear. Therefore, the PC can address up to 4M instruction words.

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).

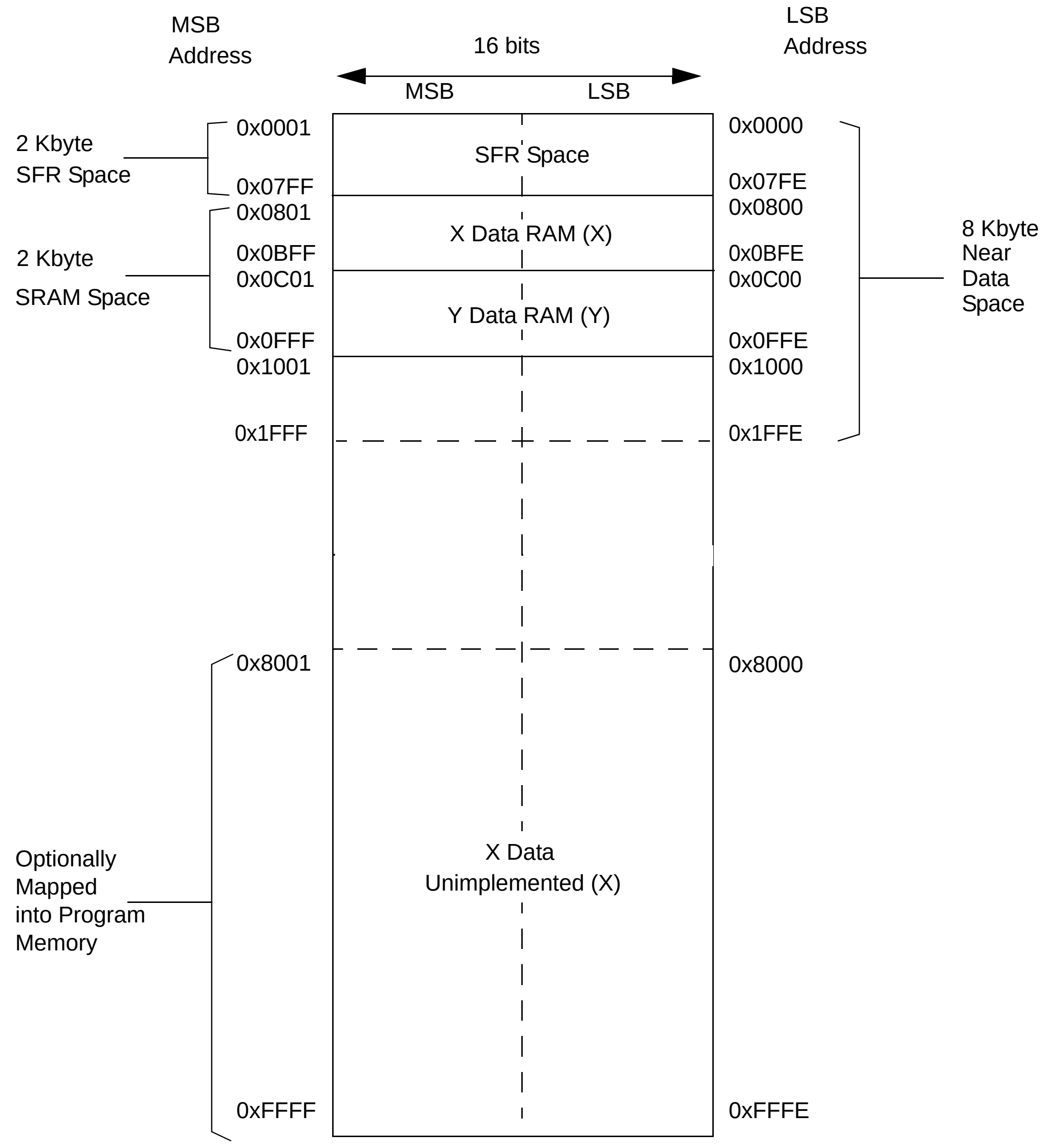
3.1 Program Address Space

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.

FIGURE 3-7: dsPIC30F3012/3013 DATA SPACE MEMORY MAP



5.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. A programming operation is nominally 2 msec in duration and the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation and the WR bit is automatically cleared when the operation is finished.

5.6.1 PROGRAMMING ALGORITHM FOR PROGRAM FLASH

The user can erase or program one row of program Flash memory at a time. The general process is:

1. Read one row of program Flash (32 instruction words) and store into data RAM as a data “image”.
2. Update the data image with the desired new data.
3. Erase program Flash row.
    a) Set up NVMCON register for multi-word, program Flash, erase, and set WREN bit.
    b) Write address of row to be erased into NVMADRU/NVMDR.
    c) Write 0x55 to NVMKEY.
    d) Write 0xAA to NVMKEY.
    e) Set the WR bit. This begins erase cycle.
    f) CPU stalls for the duration of the erase cycle.
    g) The WR bit is cleared when erase cycle ends.
4. Write 32 instruction words of data from data RAM “image” into the program Flash write latches.
5. Program 32 instruction words into program Flash.
    a) Set up NVMCON register for multi-word, program Flash, program, and set WREN bit.
    b) Write 0x55 to NVMKEY.
    c) Write 0xAA to NVMKEY.
    d) Set the WR bit. This begins program cycle.
    e) CPU stalls for duration of the program cycle.
    f) The WR bit is cleared by the hardware when program cycle ends.
6. Repeat steps 1 through 5 as needed to program desired amount of program Flash memory.

5.6.2 ERASING A ROW OF PROGRAM MEMORY

Example 5-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

EXAMPLE 5-1: ERASING A ROW OF PROGRAM MEMORY

```
; Setup NVMCON for erase operation, multi word write
; program memory selected, and writes enabled
       MOV     #0x4041,W0                    ;
       MOV     W0,NVMCON                     ; Init NVMCON SFR
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR),W0        ;
       MOV     W0,NVMADRU                    ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR),W0      ; Intialize in-page EA[15:0] pointer
       MOV     W0, NVMADR                    ; Initialize NVMADR SFR
       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 the erase sequence
       NOP                                   ; Insert two NOPs after the erase
       NOP                                   ; command is asserted
```

7.3 Input Change Notification Module

The input change notification module provides the dsPIC30F devices the ability to generate interrupt requests to the processor, in response to a change of state on selected input pins. This module is capable of detecting input change of states even in Sleep mode, when the clocks are disabled. There are up to 10 external signals (CN0 through CN7, CN17 and CN18) that may be selected (enabled) for generating an interrupt request on a change of state.

TABLE 7-7: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F2011/3012 (BITS 7-0)

SFR Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	`0000 0000 0000 0000`
CNEN2	00C2	—	—	—	—	—	—	—	—	`0000 0000 0000 0000`
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	`0000 0000 0000 0000`
CNPU2	00C6	—	—	—	—	—	—	—	—	`0000 0000 0000 0000`

Legend: — = unimplemented bit, read as ‘`0`’

TABLE 7-8: INPUT CHANGE NOTIFICATION REGISTER MAP FOR dsPIC30F2012/3013 (BITS 7-0)

SFR Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
CNEN1	00C0	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	`0000 0000 0000 0000`
CNEN2	00C2	—	—	—	—	—	CN18IE	CN17IE	—	`0000 0000 0000 0000`
CNPU1	00C4	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	`0000 0000 0000 0000`
CNPU2	00C6	—	—	—	—	—	CN18PUE	CN17PUE	—	`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.

Address Error Trap:

This trap is initiated when any of the following circumstances occurs:

1. A misaligned data word access is attempted.
2. A data fetch from our unimplemented data memory location is attempted.
3. A data access of an unimplemented program memory location is attempted.
4. An instruction fetch from vector space is attempted.

Note:	In the `MAC` class of instructions, wherein the data space is split into X and Y data space, unimplemented X space includes all of Y space, and unimplemented Y space includes all of X space.

5. Execution of a “`BRA #literal`” instruction or a “`GOTO #literal`” instruction, where `literal` is an unimplemented program memory address.
6. Executing instructions after modifying the PC to point the unimplemented program memory addresses. The PC may be modified by loading a value into the stack and executing a `RETURN` instruction.

Stack Error Trap:

This trap is initiated under the following conditions:

- The Stack Pointer is loaded with a value which is greater than the (user programmable) limit value written into the SPLIM register (stack overflow).
- The Stack Pointer is loaded with a value which is less than 0x0800 (simple stack underflow).

Oscillator Fail Trap:

This trap is initiated if the external oscillator fails and operation becomes reliant on an internal RC backup.

8.3.2 HARD AND SOFT TRAPS

It is possible that multiple traps can become active within the same cycle (e.g., a misaligned word stack write to an overflowed address). In such a case, the fixed priority shown in Figure 8-2 is implemented, which may require the user to check if other traps are pending, in order to completely correct the Fault.

Soft traps include exceptions of priority level 8 through level 11, inclusive. The arithmetic error trap (level 11) falls into this category of traps.

Hard traps include exceptions of priority level 12 through level 15, inclusive. The address error (level 12), stack error (level 13) and oscillator error (level 14) traps fall into this category.

Each hard trap that occurs must be acknowledged before code execution of any type can continue. If a lower priority hard trap occurs while a higher priority trap is pending, acknowledged, or is being processed, a hard trap conflict occurs.

The device is automatically Reset in a hard trap conflict condition. The TRAPR Status bit (RCON<15>) is set when the Reset occurs, so that the condition may be detected in software.

9.0 TIMER1 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 16-bit general purpose Timer1 module and associated operational modes. Figure 9-1 depicts the simplified block diagram of the 16-bit Timer1 module. The following sections provide detailed descriptions including setup and Control registers, along with associated block diagrams for the operational modes of the timers.

The Timer1 module is a 16-bit timer that serves as the time counter for the real-time clock or operates as a free-running interval timer/counter. The 16-bit timer has the following modes:

- 16-bit Timer
- 16-bit Synchronous Counter
- 16-bit Asynchronous Counter

These operational characteristics are supported:

- Timer gate operation
- Selectable prescaler settings
- Timer operation during CPU Idle and Sleep modes
- Interrupt on 16-bit Period register match or falling edge of external gate signal

These operating modes are determined by setting the appropriate bit(s) in the 16-bit SFR, T1CON. Figure 9-1 presents a block diagram of the 16-bit timer module.

16-bit Timer Mode: In the 16-bit Timer mode, the timer increments on every instruction cycle up to a match value preloaded into the Period register PR1, then resets to '`0`' and continues to count.

When the CPU goes into the Idle mode, the timer stops incrementing unless the TSIDL (T1CON<13>) bit = `0`. If TSIDL = `1`, the timer module logic resumes the incrementing sequence on termination of CPU Idle mode.

16-bit Synchronous Counter Mode: In the 16-bit Synchronous Counter mode, the timer increments on the rising edge of the applied external clock signal which is synchronized with the internal phase clocks. The timer counts up to a match value preloaded in PR1, then resets to '`0`' and continues.

When the CPU goes into the Idle mode, the timer stops incrementing unless the respective TSIDL bit = `0`. If TSIDL = `1`, the timer module logic resumes the incrementing sequence upon termination of the CPU Idle mode.

16-bit Asynchronous Counter Mode: In the 16-bit Asynchronous Counter mode, the timer increments on every rising edge of the applied external clock signal. The timer counts up to a match value preloaded in PR1, then resets to '`0`' and continues.

When the timer is configured for the Asynchronous mode of operation and the CPU goes into the Idle mode, the timer stops incrementing if TSIDL = `1`.

FIGURE 9-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM

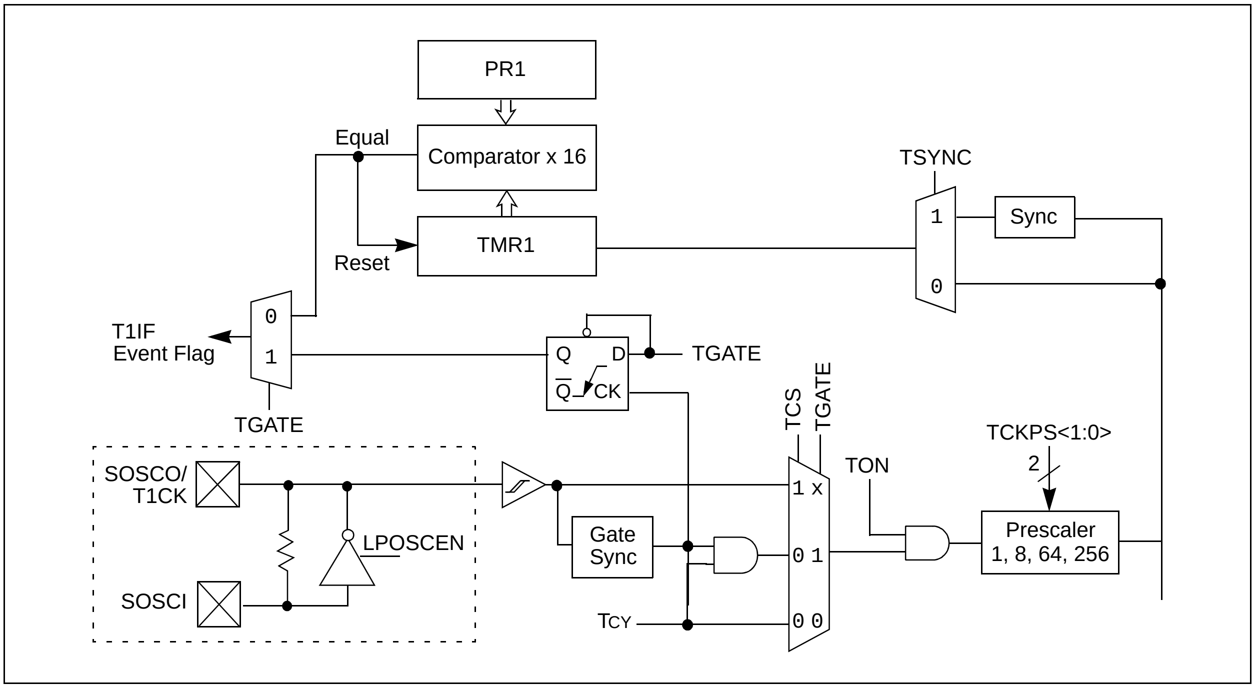


TABLE 10-1: TIMER2/3 REGISTER MAP

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
TMR2	0106	Timer2 Register																`uuuu uuuu uuuu uuuu`
TMR3HLD	0108	Timer3 Holding Register (for 32-bit timer operations only)																`uuuu uuuu uuuu uuuu`
TMR3	010A	Timer3 Register																`uuuu uuuu uuuu uuuu`
PR2	010C	Period Register 2																`1111 1111 1111 1111`
PR3	010E	Period Register 3																`1111 1111 1111 1111`
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	`0000 0000 0000 0000`
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	`0000 0000 0000 0000`

Legend: u = uninitialized bit; — = unimplemented bit, read as ‘`0`’

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

12.0 OUTPUT COMPARE 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 output compare module and associated operational modes. The features provided by this module are useful in applications requiring operational modes, such as:

- Generation of Variable Width Output Pulses
- Power Factor Correction

Figure 12-1 depicts a block diagram of the output compare module.

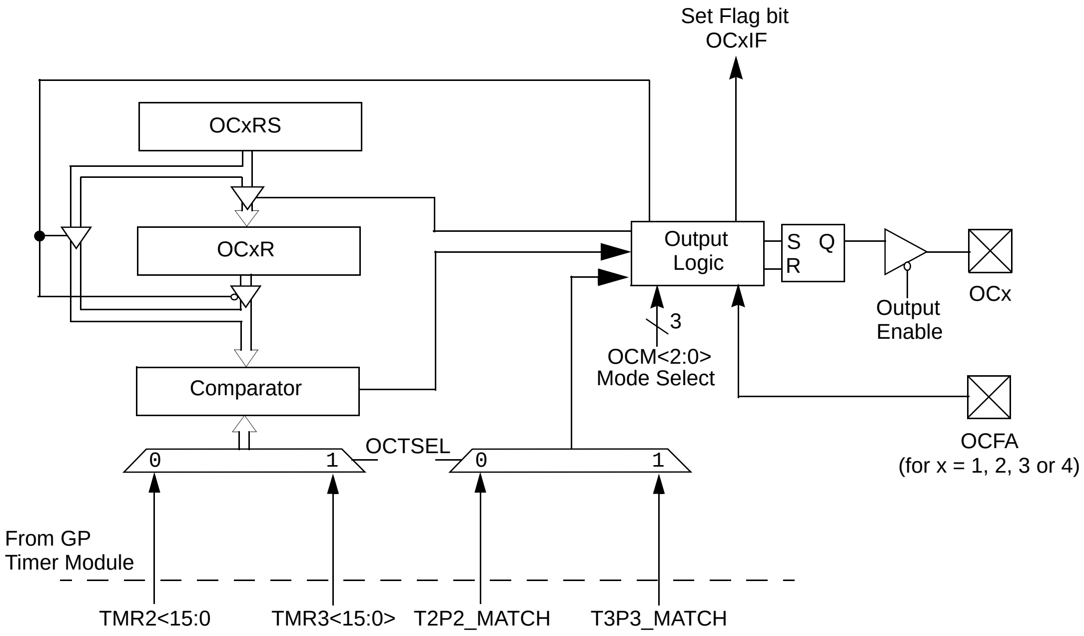
The key operational features of the output compare module include:

- Timer2 and Timer3 Selection mode
- Simple Output Compare Match mode
- Dual Output Compare Match mode
- Simple PWM mode
- Output Compare During Sleep and Idle modes
- Interrupt on Output Compare/PWM Event

These operating modes are determined by setting the appropriate bits in the 16-bit OC1CON and OC2CON registers. The dsPIC30F2011/2012/3012/3013 devices have 2 compare channels.

OCxRS and OCxR in Figure 12-1 represent the Dual Compare registers. In the Dual Compare mode, the OCxR register is used for the first compare and OCxRS is used for the second compare.

FIGURE 12-1: OUTPUT COMPARE MODE BLOCK DIAGRAM[1]



Note 1: Where 'x' is shown, reference is made to the registers associated with the respective output compare channel (1 or 2).

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.

17.2.7 FAIL-SAFE CLOCK MONITOR

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate even in the event of an oscillator failure. The FSCM function is enabled by appropriately programming the FCKSM Configuration bits (clock switch and monitor selection bits) in the FOSC Device Configuration register. If the FSCM function is enabled, the LPRC internal oscillator will run at all times (except during Sleep mode) and will not be subject to control by the SWDTEN bit.

In the event of an oscillator failure, the FSCM will generate a clock failure trap event and will switch the system clock over to the FRC oscillator. The user will then have the option to either attempt to restart the oscillator or execute a controlled shutdown. The user may decide to treat the trap as a warm Reset by simply loading the Reset address into the oscillator fail trap vector. In this event, the CF (Clock Fail) bit (OSCCON<3>) is also set whenever a clock failure is recognized.

In the event of a clock failure, the WDT is unaffected and continues to run on the LPRC clock.

If the oscillator has a very slow start-up time coming out of POR, BOR or Sleep, it is possible that the PWRT timer will expire before the oscillator has started. In such cases, the FSCM will be activated and the FSCM will initiate a clock failure trap, and the COSC<2:0> bits are loaded with FRC oscillator selection. This will effectively shut-off the original oscillator that was trying to start.

The user may detect this situation and restart the oscillator in the clock fail trap ISR.

Upon a clock failure detection, the FSCM module will initiate a clock switch to the FRC oscillator as follows:

1. The COSC bits (OSCCON<14:12>) are loaded with the FRC oscillator selection value.
2. CF bit is set (OSCCON<3>).
3. OSWEN control bit (OSCCON<0>) is cleared.

For the purpose of clock switching, the clock sources are sectioned into four groups:

- Primary (with or without PLL)
- Secondary
- Internal FRC
- Internal LPRC

The user can switch between these functional groups but cannot switch between options within a group. If the primary group is selected, then the choice within the group is always determined by the FPR<4:0> Configuration bits.

The OSCCON register holds the Control and Status bits related to clock switching.

- COSC<2:0>: Read-only bits always reflect the current oscillator group in effect.
- NOSC<2:0>: Control bits which are written to indicate the new oscillator group of choice.
  - On POR and BOR, COSC<2:0> and NOSC<2:0> are both loaded with the Configuration bit values FOS<2:0>.
- LOCK: The LOCK bit indicates a PLL lock.
- CF: Read-only bit indicating if a clock fail detect has occurred.
- OSWEN: Control bit changes from a ‘`0`’ to a ‘`1`’ when a clock transition sequence is initiated. Clearing the OSWEN control bit will abort a clock transition in progress (used for hang-up situations).

If Configuration bits FCKSM<1:0> = `1x`, then the clock switching and Fail-Safe Clock monitoring functions are disabled. This is the default Configuration bit setting.

If clock switching is disabled, then the FOS<2:0> and FPR<4:0> bits directly control the oscillator selection and the COSC<2:0> bits do not control the clock selection. However, these bits will reflect the clock source selection.

Note: The application should not attempt to switch to a clock of frequency lower than 100 kHz when the Fail-Safe Clock Monitor is enabled. If such clock switching is performed, the device may generate an oscillator fail trap and switch to the Fast RC oscillator.

17.2.8 PROTECTION AGAINST ACCIDENTAL WRITES TO OSCCON

A write to the OSCCON register is intentionally made difficult because it controls clock switching and clock scaling.

To write to the OSCCON low byte, the following code sequence must be executed without any other instructions in between:

```
Byte Write 0x46 to OSCCON low
Byte Write 0x57 to OSCCON low
```

Byte write is allowed for one instruction cycle. Write the desired value or use bit manipulation instruction.

To write to the OSCCON high byte, the following instructions must be executed without any other instructions in between:

```
Byte Write 0x78 to OSCCON high
Byte Write 0x9A to OSCCON high
```

Byte write is allowed for one instruction cycle. Write the desired value or use bit manipulation instruction.

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

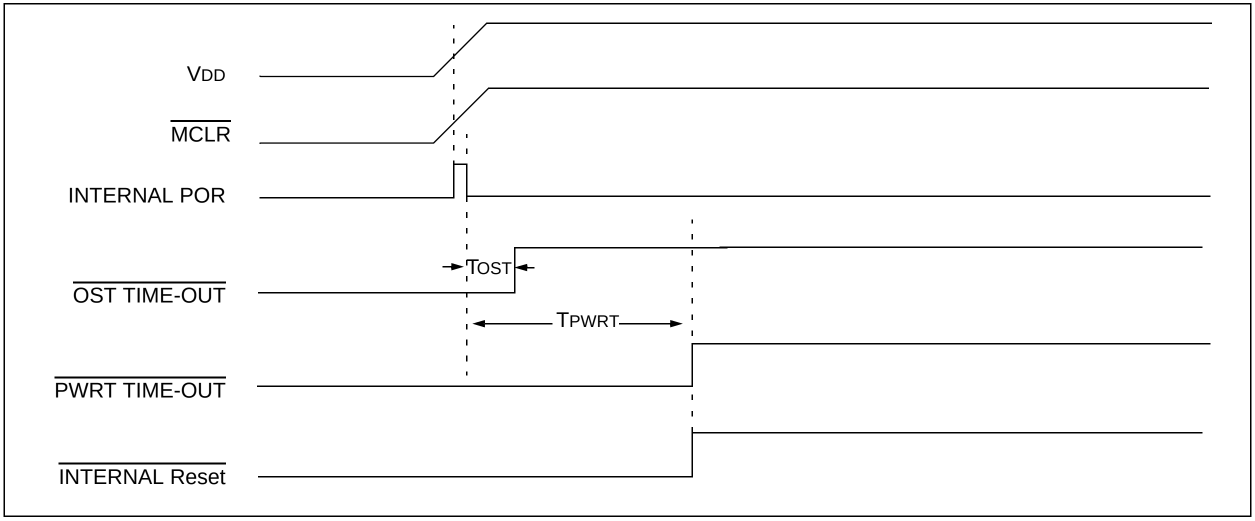


FIGURE 17-4: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 1

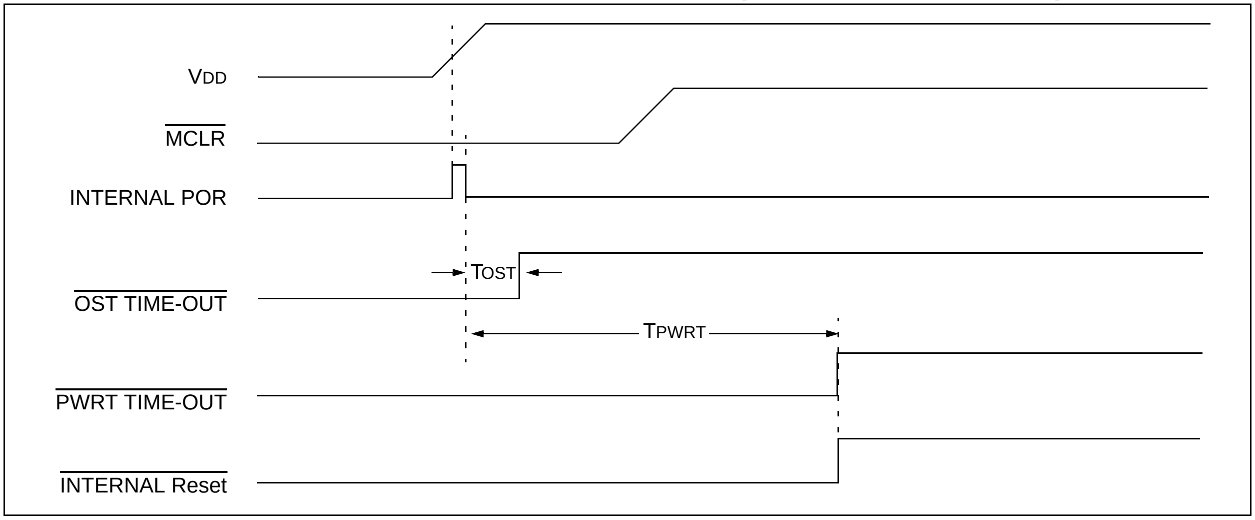


FIGURE 17-5: TIME-OUT SEQUENCE ON POWER-UP ($\overline{\text{MCLR}}$ NOT TIED TO V_{DD}): CASE 2

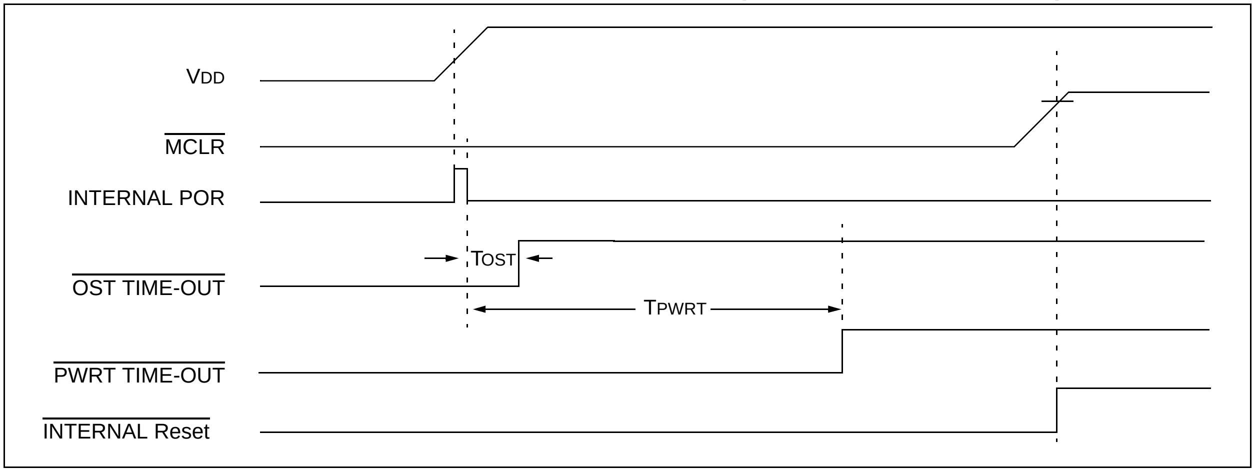


TABLE 18-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
66	RRNC	`RRNC f`	f = Rotate Right (No Carry) f	1	1	N,Z
		`RRNC f,WREG`	WREG = Rotate Right (No Carry) f	1	1	N,Z
		`RRNC Ws,Wd`	Wd = Rotate Right (No Carry) Ws	1	1	N,Z
67	SAC	`SAC Acc,#Slit4,Wdo`	Store Accumulator	1	1	None
		`SAC.R Acc,#Slit4,Wdo`	Store Rounded Accumulator	1	1	None
68	SE	`SE Ws,Wnd`	Wnd = sign-extended Ws	1	1	C,N,Z
69	SETM	`SETM f`	f = 0xFFFF	1	1	None
		`SETM WREG`	WREG = 0xFFFF	1	1	None
		`SETM Ws`	Ws = 0xFFFF	1	1	None
70	SFTAC	`SFTAC Acc,Wn`	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		`SFTAC Acc,#Slit6`	Arithmetic Shift Accumulator by Slit6	1	1	OA,OB,OAB, SA,SB,SAB
71	SL	`SL f`	f = Left Shift f	1	1	C,N,OV,Z
		`SL f,WREG`	WREG = Left Shift f	1	1	C,N,OV,Z
		`SL Ws,Wd`	Wd = Left Shift Ws	1	1	C,N,OV,Z
		`SL Wb,Wns,Wnd`	Wnd = Left Shift Wb by Wns	1	1	N,Z
		`SL Wb,#lit5,Wnd`	Wnd = Left Shift Wb by lit5	1	1	N,Z
72	SUB	`SUB Acc`	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		`SUB f`	f = f - WREG	1	1	C,DC,N,OV,Z
		`SUB f,WREG`	WREG = f - WREG	1	1	C,DC,N,OV,Z
		`SUB #lit10,Wn`	Wn = Wn - lit10	1	1	C,DC,N,OV,Z
		`SUB Wb,Ws,Wd`	Wd = Wb - Ws	1	1	C,DC,N,OV,Z
		`SUB Wb,#lit5,Wd`	Wd = Wb - lit5	1	1	C,DC,N,OV,Z
73	SUBB	`SUBB f`	f = f - WREG - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBB f,WREG`	WREG = f - WREG - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBB #lit10,Wn`	Wn = Wn - lit10 - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBB Wb,Ws,Wd`	Wd = Wb - Ws - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBB Wb,#lit5,Wd`	Wd = Wb - lit5 - ($\overline{C}$)	1	1	C,DC,N,OV,Z
74	SUBR	`SUBR f`	f = WREG - f	1	1	C,DC,N,OV,Z
		`SUBR f,WREG`	WREG = WREG - f	1	1	C,DC,N,OV,Z
		`SUBR Wb,Ws,Wd`	Wd = Ws - Wb	1	1	C,DC,N,OV,Z
		`SUBR Wb,#lit5,Wd`	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
75	SUBBR	`SUBBR f`	f = WREG - f - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBBR f,WREG`	WREG = WREG -f - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBBR Wb,Ws,Wd`	Wd = Ws - Wb - ($\overline{C}$)	1	1	C,DC,N,OV,Z
		`SUBBR Wb,#lit5,Wd`	Wd = lit5 - Wb - ($\overline{C}$)	1	1	C,DC,N,OV,Z
76	SWAP	`SWAP.b Wn`	Wn = nibble swap Wn	1	1	None
		`SWAP Wn`	Wn = byte swap Wn	1	1	None
77	TBLRDH	`TBLRDH Ws,Wd`	Read Prog<23:16> to Wd<7:0>	1	2	None
78	TBLRDL	`TBLRDL Ws,Wd`	Read Prog<15:0> to Wd	1	2	None
79	TBLWTH	`TBLWTH Ws,Wd`	Write Ws<7:0> to Prog<23:16>	1	2	None
80	TBLWTL	`TBLWTL Ws,Wd`	Write Ws to Prog<15:0>	1	2	None
81	ULNK	`ULNK`	Unlink frame pointer	1	1	None
82	XOR	`XOR f`	f = f .XOR. WREG	1	1	N,Z
		`XOR f,WREG`	WREG = f .XOR. WREG	1	1	N,Z
		`XOR #lit10,Wn`	Wd = lit10 .XOR. Wd	1	1	N,Z
		`XOR Wb,Ws,Wd`	Wd = Wb .XOR. Ws	1	1	N,Z
		`XOR Wb,#lit5,Wd`	Wd = Wb .XOR. lit5	1	1	N,Z
83	ZE	`ZE Ws,Wnd`	Wnd = Zero-extend Ws	1	1	C,Z,N

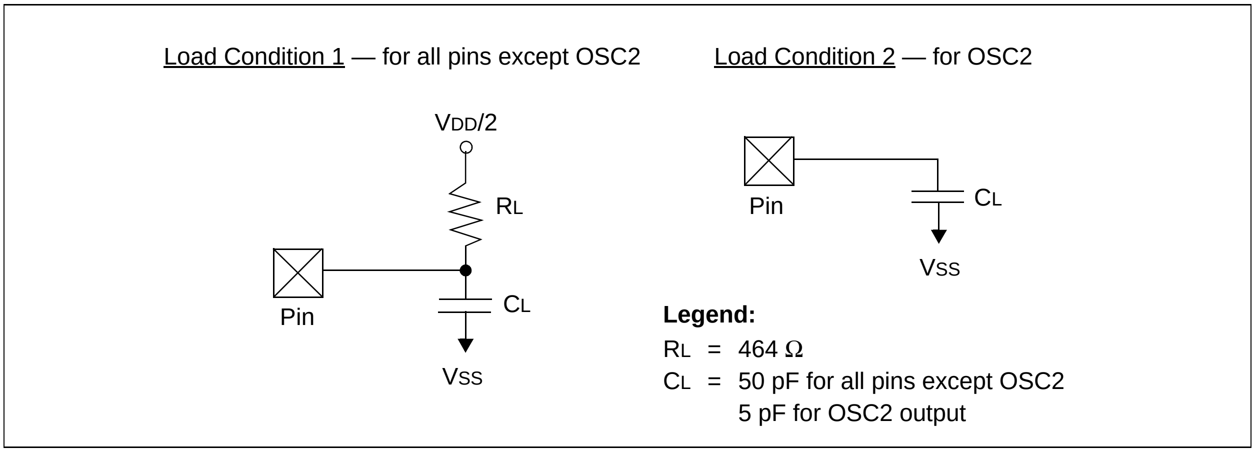
20.2 AC Characteristics and Timing Parameters

The information contained in this section defines dsPIC30F AC characteristics and timing parameters.

TABLE 20-13: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 2.5V to 5.5V** **(unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended Operating voltage V_{DD} range as described in **Section 20.1 “DC Characteristics”**.

FIGURE 20-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



Legend:

R_L = 464 Ω

C_L = 50 pF for all pins except OSC2

5 pF for OSC2 output

FIGURE 20-4: EXTERNAL CLOCK TIMING

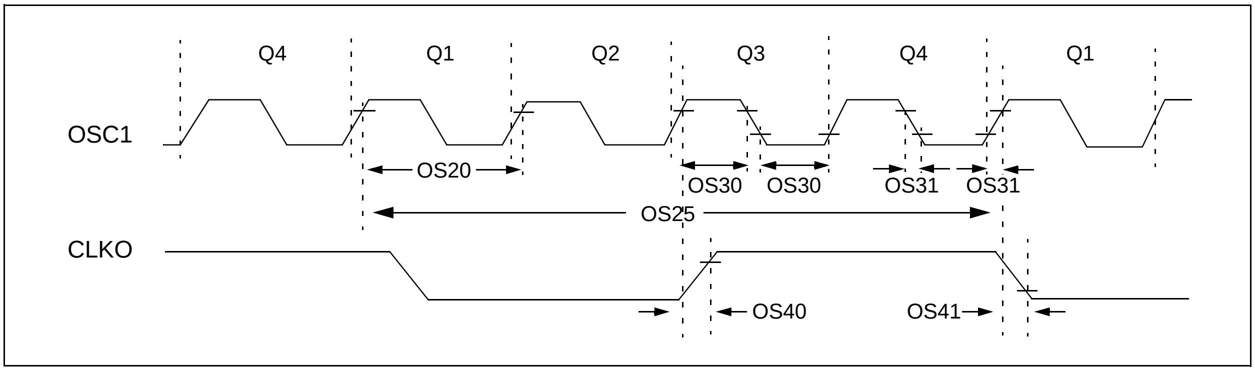
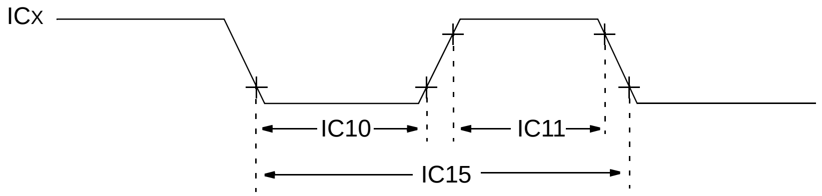


FIGURE 20-9: INPUT CAPTURE (CAPx) TIMING CHARACTERISTICS



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

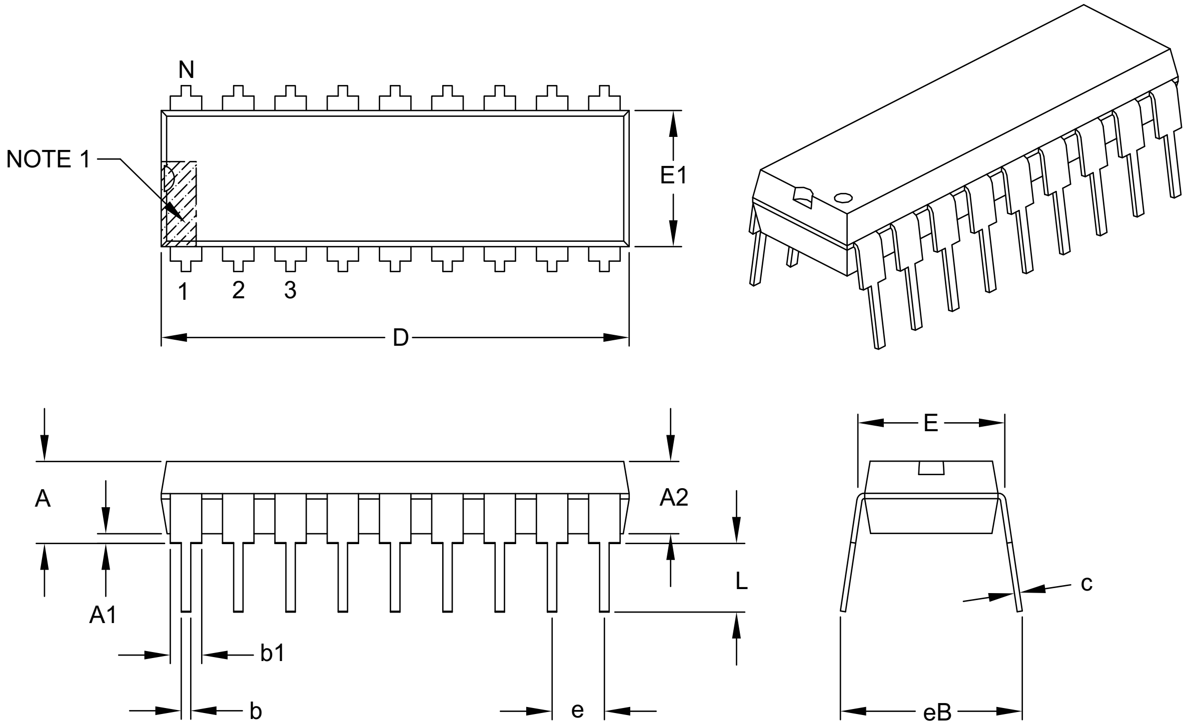
TABLE 20-26: INPUT CAPTURE TIMING REQUIREMENTS

AC CHARACTERISTICS				**Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated)** Operating temperature -40°C ≤ T_A ≤ +85°C for Industrial -40°C ≤ T_A ≤ +125°C for Extended			
Param No.	**Symbol**	**Characteristic[1]**		**Min**	**Max**	**Units**	**Conditions**
IC10	TccL	ICx Input Low Time	No Prescaler	0.5 T_{CY} + 20	—	ns	
			With Prescaler	10	—	ns	
IC11	TccH	ICx Input High Time	No Prescaler	0.5 T_{CY} + 20	—	ns	
			With Prescaler	10	—	ns	
IC15	TccP	ICx Input Period		(2 T_{CY} + 40)/N	—	ns	N = prescale value (1, 4, 16)

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

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

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



	Units	INCHES		
	Dimension Limits	MIN	NOM	MAX
Number of Pins	N	18		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.300	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.880	.900	.920
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.014
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

Notes:

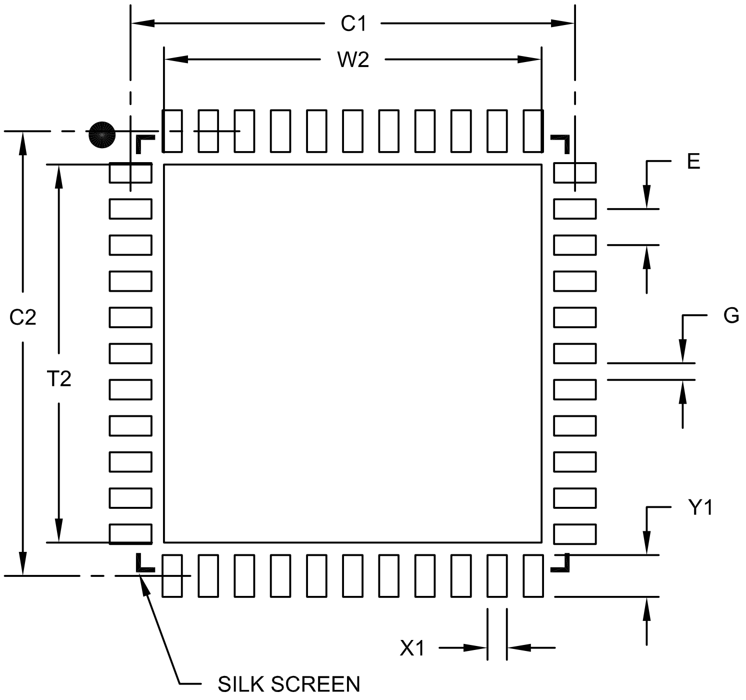
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

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

Microchip Technology Drawing C04-007B

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

READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

TO: Technical Publications Manager
Total Pages Sent ________
RE: Reader Response
From: Name ____
Company ____
Address ____
City / State / ZIP / Country ________________________________
Telephone: (_______) _________ - _________
FAX: (______) _________ - _________
Application (optional):
Would you like a reply? ___Y ___N
Device: dsPIC30F2011/2012/3012/3013
Literature Number: DS70139G

Questions:

1. What are the best features of this document?

____

____

2. How does this document meet your hardware and software development needs?

____

____

3. Do you find the organization of this document easy to follow? If not, why?

____

____

4. What additions to the document do you think would enhance the structure and subject?

____

____

5. What deletions from the document could be made without affecting the overall usefulness?

____

____

6. Is there any incorrect or misleading information (what and where)?

____

____

7. How would you improve this document?

____

____