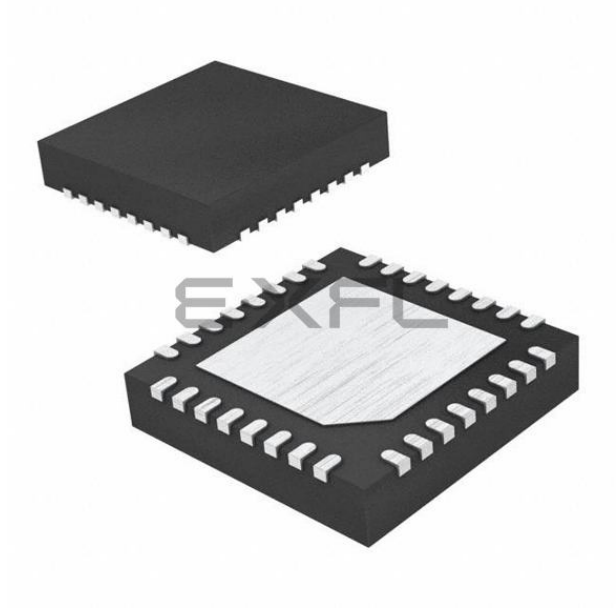
Microchip Technology - DSPIC30F1010-20E/MM Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

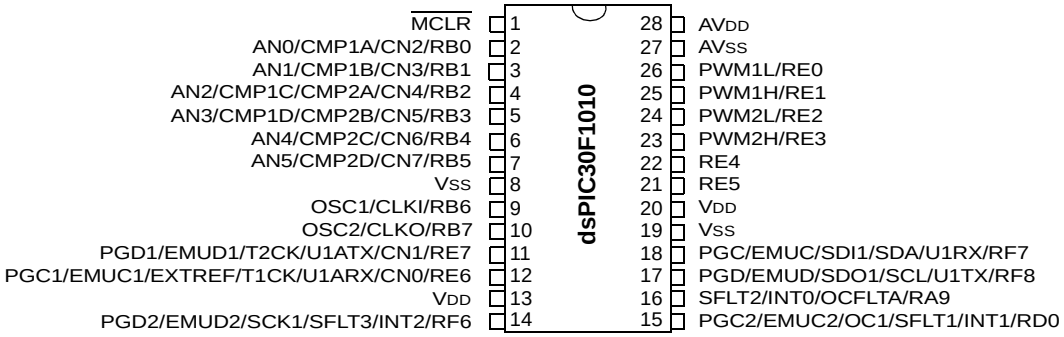
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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"

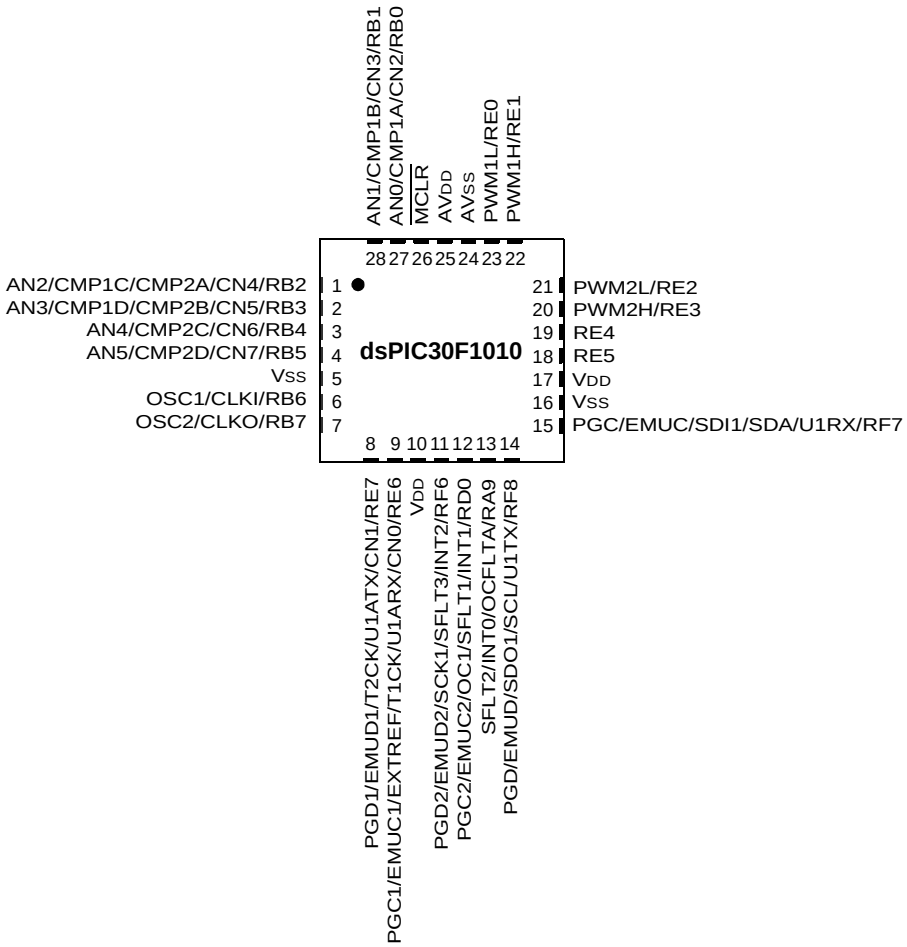
Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	20 MIPS
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	21
Program Memory Size	6KB (2K x 24)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 5.5V
Data Converters	A/D 6x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VQFN Exposed Pad
Supplier Device Package	28-QFN-S (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f1010-20e-mm

Pin Diagrams

28-Pin SDIP and SOIC



28-Pin QFN-S



2.2 Programmer's Model

The programmer's model is shown in Figure 2-1 and consists of 16x16-bit working registers (W0 through W15), 2x40-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 (MSBs) 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), and will be 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.

REGISTER 5-4: IFS1: INTERRUPT FLAG STATUS REGISTER 1

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0
AC3IF	AC2IF	AC1IF	—	CNIF	—	—	—
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	PSEMIF	INT2IF	INT1IF
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **AC3IF:** Analog Comparator #3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 14 **AC2IF:** Analog Comparator #2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 13 **AC1IF:** Analog Comparator #1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 12 **Unimplemented:** Read as '0'

bit 11 **CNIF:** Input Change Notification Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 10-7 **Unimplemented:** Read as '0'

bit 6 **PWM4IF:** Pulse Width Modulation Generator #4 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 5 **PWM3IF:** Pulse Width Modulation Generator #3 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 4 **PWM2IF:** Pulse Width Modulation Generator #2 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 3 **PWM1IF:** Pulse Width Modulation Generator #1 Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 2 **PSEMIF:** PWM Special Event Match Interrupt Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 1 **INT2IF:** External Interrupt 2 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

bit 0 **INT1IF:** External Interrupt 1 Flag Status bit
1 = Interrupt request has occurred
0 = Interrupt request has not occurred

REGISTER 5-17: IPC8: INTERRUPT PRIORITY CONTROL REGISTER 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	AC4IP<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-3 **Unimplemented:** Read as ‘0’

bit 2-0 **AC4IP<2:0>:** Analog Comparator 4 Interrupt Priority bits

111 = Interrupt is priority 7 (highest priority interrupt)

•

•

•

001 = Interrupt is priority 1

000 = Interrupt source is disabled

TABLE 6-2: dsPIC30F2023 PORT 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
TRISA	02C0	—	—	—	—	TRISA11	TRISA10	TRIS9	TRISA8	—	—	—	—	—	—	—	—	0000 1111 0000 0000
PORTA	02C2	—	—	—	—	RA11	RA10	RA9	RA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATA	02C4	—	—	—	—	LATA11	LATA10	LATA9	LATA8	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISB	02C6	—	—	—	—	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRIS6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 1111 1111 1111
PORTB	02C8	—	—	—	—	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CA	—	—	—	—	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD1	TRISD0	0000 0000 0000 0011
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD1	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	—	TRSE7	TRSE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0000 1111 1111
PORTE	02DA	—	—	—	—	—	—	—	—	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	—	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02DE	TRISF15	TRISF14	—	—	—	—	—	TRISF8	TRISF7	TRISF6	—	—	TRISF3	TRISF2	—	—	1100 0001 1100 1100
PORTF	02E0	RF15	RF14	—	—	—	—	—	RF8	RF7	RF6	—	—	RF3	RF2	—	—	0000 0000 0000 0000
LATF	02E2	LATF15	LATF14	—	—	—	—	—	LATF8	LATF7	LATF6	—	—	LATF3	LATF2	—	—	0000 0000 0000 0000
TRISG	02E4	—	—	—	—	—	—	—	—	—	—	—	—	TRISG3	TRISG2	—	—	0000 0000 0000 1100
PORTG	02E6	—	—	—	—	—	—	—	—	—	—	—	—	RG3	RG2	—	—	0000 0000 0000 0000
LATG	02E8	—	—	—	—	—	—	—	—	—	—	—	—	LATG3	LATG2	—	—	0000 0000 0000 0000

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

TABLE 6-3: dsPIC30F1010/202X INPUT CHANGE NOTIFICATION 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
CNEN1	0060	—	—	—	—	—	—	—	—	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNPU1	0064	—	—	—	—	—	—	—	—	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000

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

7.6.3 LOADING WRITE LATCHES

Example 7-2 shows a sequence of instructions that can be used to load the 96 bytes of write latches. 32 TBLWTL and 32 TBLWTH instructions are needed to load the write latches selected by the table pointer.

EXAMPLE 7-2: LOADING WRITE LATCHES

```
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
       MOV     #0x0000,W0                      ;
       MOV     W0,TBLPAG                       ; Initialize PM Page Boundary SFR
       MOV     #0x6000,W0                      ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
       MOV     #LOW_WORD_0,W2                  ;
       MOV     #HIGH_BYTE_0,W3                 ;
       TBLWTL  W2,[W0]                         ; Write PM low word into program latch
       TBLWTH  W3,[W0++]                       ; Write PM high byte into program latch
; 1st_program_word
       MOV     #LOW_WORD_1,W2                  ;
       MOV     #HIGH_BYTE_1,W3                 ;
       TBLWTL  W2,[W0]                         ; Write PM low word into program latch
       TBLWTH  W3,[W0++]                       ; Write PM high byte into program latch
;  2nd_program_word
       MOV     #LOW_WORD_2,W2                  ;
       MOV     #HIGH_BYTE_2,W3                 ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                      ; Write PM high byte into program latch
       •
       •
       •
; 31st_program_word
       MOV     #LOW_WORD_31,W2                 ;
       MOV     #HIGH_BYTE_31,W3                ;
       TBLWTL  W2, [W0]                        ; Write PM low word into program latch
       TBLWTH  W3, [W0++]                      ; Write PM high byte into program latch
```

Note: In Example 7-2, the contents of the upper byte of W3 have no effect.

7.6.4 INITIATING THE PROGRAMMING SEQUENCE

For protection, the write initiate sequence for NVMKEY must be used to allow any erase or program operation to proceed. After the programming command has been executed, the user must wait for the programming time until programming is complete. The two instructions following the start of the programming sequence should be NOPs.

EXAMPLE 7-3: INITIATING A PROGRAMMING SEQUENCE

```
       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
```

REGISTER 12-4: MDC: PWM MASTER DUTY CYCLE REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC<7:0>							
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-0 Master PWM Duty Cycle Value bits**(1)**

Note 1: The minimum value for this register is 0x0008 and the maximum value is 0xFFEF.

REGISTER 12-5: PWMCONx: PWM CONTROL REGISTER

HS/HC-0	HS/HC-0	HS/HC-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS
bit 15							bit 8

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
DTC<1:0>		—	—	—	—	XPRES	IUE
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15 **FLTSTAT:** Fault Interrupt Status
1 = Fault Interrupt is pending
0 = No Fault Interrupt is pending
This bit is cleared by setting FLTIEN = 0.

Note: Software must clear the interrupt status here, and the corresponding IFS bit in Interrupt Controller.

bit 14 **CLSTAT:** Current-Limit Interrupt Status bit
1 = Current-limit interrupt is pending
0 = No current-limit interrupt is pending
This bit is cleared by setting CLIEN = 0.

Note: Software must clear the interrupt status here, and the corresponding IFS bit in Interrupt Controller.

bit 13 **TRGSTAT:** Trigger Interrupt Status bit
1 = Trigger interrupt is pending
0 = No trigger interrupt is pending
This bit is cleared by setting TRGIEN = 0.

bit 12 **FLTIEN:** Fault Interrupt Enable bit
1 = Fault interrupt enabled
0 = Fault interrupt disabled and FLTSTAT bit is cleared

REGISTER 12-9: ALTDTRx: PWM ALTERNATE DEAD-TIME REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	ALTDTRx<13:8>					
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
ALTDTR <7:2>						—	—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-14 **Unimplemented:** Read as ‘0’

bit 13-2 **ALTDTRx<13:2>:** Unsigned 12-bit Dead-Time Value bits for PWMx Dead-Time Unit bits

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

REGISTER 12-10: TRGCONx: PWM TRIGGER CONTROL REGISTER

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TRGDIV<2:0>			—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TRGSTRT<5:0>					
bit 7							bit 0

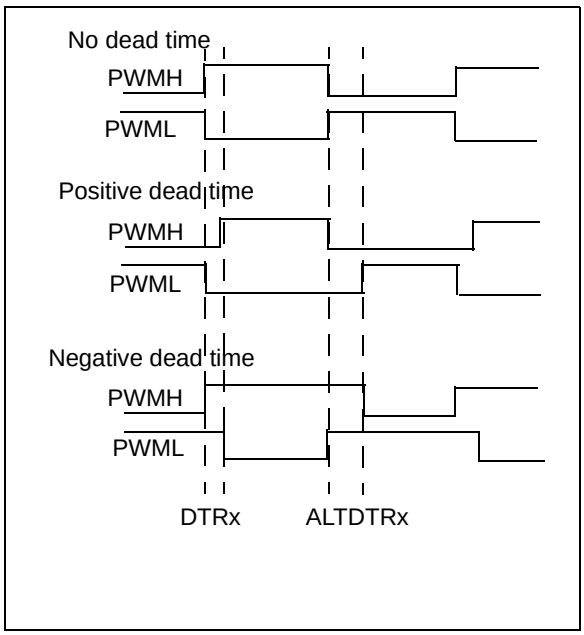
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’	
-n = Value at POR	‘1’ = Bit is set	‘0’ = Bit is cleared	x = Bit is unknown

bit 15-13 **TRGDIV<2:0>:** Trigger Output Divider bits
000 = Trigger output for every trigger event
001 = Trigger output for every 2nd trigger event
010 = Trigger output for every 3rd trigger event
011 = Trigger output for every 4th trigger event
100 = Trigger output for every 5th trigger event
101 = Trigger output for every 6th trigger event
110 = Trigger output for every 7th trigger event
111 = Trigger output for every 8th trigger event

bit 12-6 **Unimplemented:** Read as ‘0’

bit 5-0 **TRGSTRT<5:0>:** Trigger Postscaler Start Enable Select bits
This value specifies the ROLL counter value needed for a match that will then enable the trigger postscaler logic to begin counting trigger events.

FIGURE 12-17: DUAL DEAD-TIME WAVEFORMS



12.14.3 DEAD-TIME RANGES

The amount of dead time provided by each dead-time unit is selected by specifying a 12-bit unsigned value in the DTRx registers. The 12-bit dead-time counters clock at four times the instruction execution rate. The Least Significant one bit of the dead-time value are processed by the Fine Adjust PWM module.

Table 12-3 shows example dead-time ranges as a function of the device operating frequency.

TABLE 12-3: EXAMPLE DEAD-TIME RANGES

MIPS	Resolution	Dead-Time Range
30	4.16 ns	0-17.03 µsec
20	6.25 ns	0-25.59 µsec

12.14.4 DEAD-TIME INSERTION TIMING

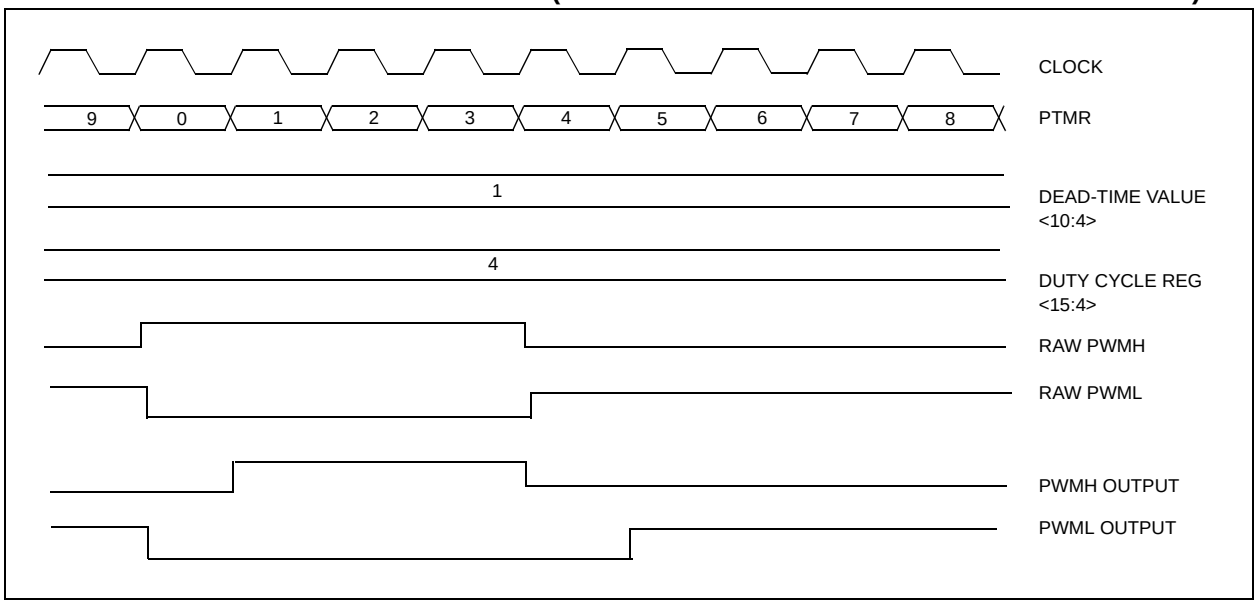
Figure 12-18 shows how the dead-time insertion for complementary signals is accomplished.

12.14.5 DEAD-TIME DISTORTION

For small PWM duty cycles, the ratio of dead time to the active PWM time may become large. In this case, the inserted dead time introduces distortion into waveforms produced by the PWM module. The user can ensure that dead-time distortion is minimized by keeping the PWM duty cycle at least three times larger than the dead time.

A similar effect occurs for duty cycles at or near 100%. The maximum duty cycle used in the application should be chosen such that the minimum inactive time of the signal is at least three times larger than the dead time.

FIGURE 12-18: DEAD-TIME INSERTION (PWM OUTPUT SIGNAL TIMING MAY BE DELAYED)



EXAMPLE 12-1: CODE EXAMPLE FOR CONFIGURING PWM CHANNEL 1

.

Note: This code example does not illustrate configuration of various fault modes for the PWM module. It is intended as a quick start guide for setting up the PWM Module.

```
mov #0x0400, w0              ; PWM Module is disabled, continue operation in
mov w0, PTCON                ; idle mode, special event interrupt disabled,
                             ; immediate period updates enabled, no external
                             ; synchronization

; Set the PWM Period
mov #0x094D, w0              ; Select period to be approximately 2.5usec
mov w0, PTPER                ; PLL Frequency is ~480MHz. This equates to a
                             ; clocke period of 2.1nsec. The PWM period and
                             ; duty cycle registers are triggered on both +ve
                             ; and -ve edges of the PLL clock. Therefore,
                             ; one count of the PTPER and PDCx registers
                             ; equals 1.05nsec.
                             ; So, to achieve a PWM period of 2.5usec, we
                             ; choose PTPER = 0x094D

mov #0x0000, w0              ; no phase shift for this PWM Channel
mov w0, PHASE1               ; This register is used for generating variable
                             ; phase PWM



; Select individual Duty Cycle Control
mov #0x0001, w0              ; Fault interrupt disabled, Current Limit
mov w0, PWMCON1              ; interrupt disabled, trigger interrupt,
                             ; disabled, Primary time base provides timing,
                             ; DC1 provides duty cycle information, positive
                             ; dead time applied, no external PWM reset,
                             ; Enable immediate duty cycle updates

; Code for PWM Current Limit and Fault Inputs
mov #0x0003, w0
mov w0, FCLCON1              ; Disable current limit and fault inputs


; Code for PWM Output Control
mov #0xC000, w0              ; PWM1H and PWM1L is controlled by PWM module
mov w0, IOCON1               ; Output polarities are active high, override
                             ; disabled

; Duty Cycle Setting
mov #0x04A6, w0              ; To achieve a duty cycle of 50%, we choose
mov w0, PDC1                 ; the PDC1 value = 0.5*(PWM Period)
                             ; The ON time for the PWM = 1.25usec
                             ; The Duty Cycle Register will provide
                             ; positive duty cycle to the PWMxH outputs
                             ; when output polarities are active high
                             ; (see IOCON1 register)

; Dead Time Setting
mov #0x0040, w0              ; Dead time ~ 67nsec
mov w0, DTR1                 ; Hex(40) = decimal(64)
                             ; So, Dead time = 64*1.05nsec = 67.2nsec
                             ; Note that the last 2 bits are unimplemented,
                             ; therefore the dead time register can achieve a
                             ; a resolution of about 4nsec.
mov w0, ALTDTR1              ; Load the same value in ALTDTR1 register

bset PTCON, #15              ; turn ON PWM module
```

12.23.1 FAULT INTERRUPTS

The FLTIENx bits in the PWMCONx registers determine if an interrupt will be generated when the FLTx input is asserted high. The FLTMOD bits in the FCLCONx register determines how the PWM generator and its outputs respond to the selected Fault input pin. The FLTDAT<1:0> bits in the IOCONx registers supply the data values to be assigned to the PWMxH,L pins in the advent of a Fault.

The Fault pin logic can operate separately from the PWM logic as an external interrupt pin. If the faults are disabled from affecting the PWM generators in the FCLCONx register, then the Fault pin can be used as a general purpose interrupt pin.

12.23.2 FAULT STATES

The IOCONx register has two bits that determine the state of each PWMx I/O pin when they are overridden by a Fault input. When these bits are cleared, the PWM I/O pin is driven to the inactive state. If the bit is set, the PWM I/O pin is driven to the active state. The active and inactive states are referenced to the polarity defined for each PWM I/O pin (HPOL and LPOL polarity control bits).

12.23.3 FAULT INPUT MODES

The Fault input pin has two modes of operation:

- **Latched Mode:** When the Fault pin is asserted, the PWM outputs go to the states defined in the FLTDAT bits in the IOCONx registers. The PWM outputs remain in this state until the Fault pin is deasserted AND the corresponding interrupt flag has been cleared in software. When both of these actions have occurred, the PWM outputs return to normal operation at the beginning of the next PWM cycle boundary. If the FLTSTAT bit is cleared before the Fault condition ends, the PWM module waits until the Fault pin is no longer asserted to restore the outputs. Software can clear the FLTSTAT bit by writing a zero to the FLTIEN bit.
- **Cycle-by-Cycle Mode:** When the Fault input pin is asserted, the PWM outputs remain in the deasserted PWM state for as long as the Fault pin is asserted. For Complementary Output modes, PWMH is low (deasserted) and PWML is high (asserted). After the Fault pin is driven high, the PWM outputs return to normal operation at the beginning of the following PWM cycle.

The operating mode for each Fault input pin is selected using the FLTMOD<1:0> control bits in the FCLCONx register.

12.23.4 FAULT ENTRY

The response of the PWM pins to the Fault input pins is always asynchronous with respect to the device clock signals. That is, the PWM outputs should immediately go to the states defined in the FLTDAT register bits without any interaction from the dsPIC DSC device or software.

Refer to **Section 12.28 “Fault and Current-Limit Override Issues with Dead-Time Logic”** for information regarding data sensitivity and behavior in response to current-limit or Fault events.

12.23.5 FAULT EXIT

The restoration of the PWM signals after a Fault condition has ended must occur at a PWM cycle boundary to ensure proper synchronization of PWM signal edges and manual signal overrides. The next PWM cycle begins when the PTMRx value is zero.

12.23.6 FAULT EXIT WITH PTMR DISABLED

There is a special case for exiting a Fault condition when the PWM time base is disabled (PTEN = 0). When a Fault input is programmed for Cycle-by-Cycle mode, the PWM outputs are immediately restored to normal operation when the Fault input pin is deasserted. The PWM outputs should return to their default programmed values. (The time base is disabled, so there is no reason to wait for the beginning of the next PWM cycle.)

When a Fault input is programmed for Latched mode, the PWM outputs are restored immediately when the Fault input pin is deasserted AND the FSTAT bit has been cleared in software.

12.23.7 FAULT PIN SOFTWARE CONTROL

The Fault pin can be controlled manually in software. Since the Fault input is shared with a PORT I/O pin, the PORT pin can be configured as an output by clearing the corresponding TRIS bit. When the PORT bit for the pin is cleared, the Fault input will be activated.

Note:	The user should use caution when controlling the Fault inputs in software. If the TRIS bit for the Fault pin is cleared and the PORT bit is set high, then the Fault input cannot be driven externally.

12.36 EXTERNAL SYNCHRONIZATION FEATURES

In large power conversion systems, it is often desirable to be able to synchronize multiple power controllers to ensure that “beat frequencies” are not generated within the system, or as a means to ensure “quiet” periods during which current and voltage measurements can be made.

dsPIC30F202X devices (excluding 28-pin packages) have input and/or output pins that provide the capability to either synchronize the SMPS dsPIC DSC device with an external device or have external devices synchronized to the SMPS dsPIC DSC. These synchronizing features are enabled via the SYNCIEN and SYNCOEN bits in the PTCON control register in the PWM module.

The SYNCPOL bit in the PTCON register selects whether the rising edge or the falling edge of the SYNCI signal is the active edge. The SYNCPOL bit in the PTCON register also selects whether the SYNCO output pulse is low active or high active.

The SYNCSRC<2:0> bits in the PTCON register specify the source for the SYNCI signal.

If the SYNCI feature is enabled, the primary time base counter is reset when an active SYNCI edge is detected. If the SYNCO feature is enabled, an output pulse is generated when the primary time base counter rolls over at the end of a PWM cycle.

The recommended SYNCI pulse width should be more than 100 nsec. The expected SYNCO output pulse width will be approximately 100 nsec.

When using the SYNCI feature, it is recommended that the user program the period register with a period value that is slightly longer than the expected period of the external synchronization input signal. This provides protection in case the SYNCI signal is not received due to noise or external component failure. With a reasonable period value programmed into the PTPER register, the local power conversion process should remain operational even if the global synchronization signal is not received.

12.37 CPU LOAD STAGGERING

The SMPS dsPIC DSC has the ability to stagger the individual trigger comparison operations. This feature helps to level the processor’s workload to minimize situations where the processor is overloaded.

Assume a situation where there are four PWM channels controlling four independent voltage outputs. Assume further that each PWM generator is operating at 1000 kHz (1 µsec period) and each control loop is operating at 125 kHz (8 µsec).

The TRGDIV<2:0> bits in each TRGCONx register will be set to ‘111’, which selects that every 8th trigger comparison match will generate a trigger signal to the ADC to capture data and begin a conversion process.

If the stagger-in-time feature did not exist, all of the requests from all of the PWM trigger registers might occur at the same time. If this “pile-up” were to happen, some data sample might become stale (outdated) by the time the data for all four channels can be processed.

With the stagger-in-time feature, the trigger signals are spaced out over time (during succeeding PWM periods) so that all of the data is processed in an orderly manner.

The ROLL counter is a counter connected to the primary time base counter. The ROLL counter is incremented each time the primary time base counter reaches terminal count (period rollover).

The stagger-in-time feature is controlled by the TRGSTRT<5:0> bits in the TRGCONx registers. The TRGSTRT<5:0> bits specify the count value of the ROLL counter that must be matched before an individual trigger comparison module in each of the PWM generators can begin to count the trigger comparison events as specified by the TRGDIV<2:0> bits in the PWMCONx registers.

So, in our example with the four PWM generators, the first PWM’s TRGSTRT<5:0> bits would be ‘000’, the second PWM’s TRGSTRT bits would be set to ‘010’, the third PWM’s TRGSTRT bits would be set to ‘100’ and the fourth PWM’s TRGSTRT bits would be set to ‘110’. Therefore, over a total of eight PWM cycles, the four separate control loops could be run each with their own 2-µsec time period.

12.38 EXTERNAL TRIGGER BLANKING

Using the LEB<9:3> bits in the LEBCONx registers, the PWM module has the capability to blank (ignore) the external current and Fault inputs for a period of 0 to 1024 nsec. This feature is useful if power transistor turn-on induced transients make current sensing difficult at the start of a PWM cycle.

TABLE 12-4: POWER SUPPLY PWM REGISTER MAP

File Name	ADR	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	All Resets
PTCON	0400	PTEN	—	PTSIDL	SESTAT	SEIEN	EIPU	SYNCPOL	SYNCOEN	SYNCEN	SYNCSRC<2:0>			SEVTPS<3:0>				0000
PTPER	0402	PTPER<15:3>													—	—	—	FFF0
MDC	0404	MDC<15:0>																0000
SEVTCMP	0406	SEVTCMP<15:3>													—	—	—	0000
PWMCON1	0408	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON1	040A	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON1	040C	—	—	—	CLSRC<3:0>				CLPOL	CLMOD	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC1	040E	PDC1<15:0>																0000
PHASE1	0410	PHASE1<15:2>														—	—	0000
DTR1	0412	—	—	DTR1<13:2>												—	—	0000
ALTDTR1	0414	—	—	ALTDTR1<13:2>												—	—	0000
TRIG1	0416	TRIG<15:3>													—	—	—	0000
TRGCON1	0418	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON1	041A	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON2	041C	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON2	041E	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON2	0420	—	—	—	CLSRC<3:0>				CLPOL	CLMOD	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC2	0422	PDC2<15:0>																0000
PHASE2	0424	PHASE2<15:2>														—	—	0000
DTR2	0426	—	—	DTR2<13:2>												—	—	0000
ALTDTR2	0428	—	—	ALTDTR2<13:2>												—	—	0000
TRIG2	042A	TRIG<15:3>													—	—	—	0000
TRGCON2	042C	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON2	042E	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON3	0430	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON3	0432	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000
FCLCON3	0434	—	—	—	CLSRC<3:0>				CLPOL	CLMOD	FLTSRC<3:0>				FLTPOL	FLTMOD<1:0>		0000
PDC3	0436	PDC3<15:0>																0000
PHASE3	0438	PHASE3<15:2>														—	—	0000
DTR3	043A	—	—	DTR3<13:2>												—	—	0000
ALTDTR3	043C	—	—	ALTDTR3<13:2>												—	—	0000
TRIG3	043E	TRIG<15:3>													—	—	—	0000
TRGCON3	0440	TRGDIV<2:0>			—-	—-	—-	—-	—-	—-	—-	TRGSTRT<5:0>						0000
LEBCON3	0442	PHR	PHF	PLR	PLF	FLTLEBEN	CLLEBEN	LEB<9:3>							—	—	—	0000
PWMCON4	0444	FLTSTAT	CLSTAT	TRGSTAT	FLTIEN	CLIEN	TRGIEN	ITB	MDCS	DTC<1:0>		—	—	—	—	XPRES	IUE	0000
IOCON4	0446	PENH	PENL	POLH	POLL	PMOD<1:0>		OVRENH	OVRENL	OVRDAT<1:0>		FLTDAT<1:0>		CLDAT<1:0>		—	OSYNC	0000

REGISTER 16-3: A/D BASE REGISTER (ADBASE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADBASE<15:8>							
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
ADBASE<7:1>							—
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-1 **ADC Base Register:** This register contains the base address of the user's ADC Interrupt Service Routine jump table. This register, when read, contains the sum of the ADBASE register contents and the encoded value of the PxRDY Status bits.

The encoder logic provides the bit number of the highest priority PxRDY bits where P0RDY is the highest priority, and P5RDY is lowest priority.

Note: The encoding results are shifted left two bits so bits 1-0 of the result are always zero.

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

Note: As an alternative to using the ADBASE Register, the ADCP0-5 ADC Pair Conversion Complete Interrupts (Interrupts 37-42) can be used to invoke A to D conversion completion routines for individual ADC input pairs. Refer to **Section 16.9 "Individual Pair Interrupts"**.

REGISTER 16-4: A/D PORT CONFIGURATION REGISTER (ADPCFG)

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-12 **Unimplemented:** Read as '0'

bit 11-0 **PCFG<11:0>:** A/D Port Configuration Control bits

1 = Port pin in Digital mode, port read input enabled, A/D input multiplexor connected to AVSS

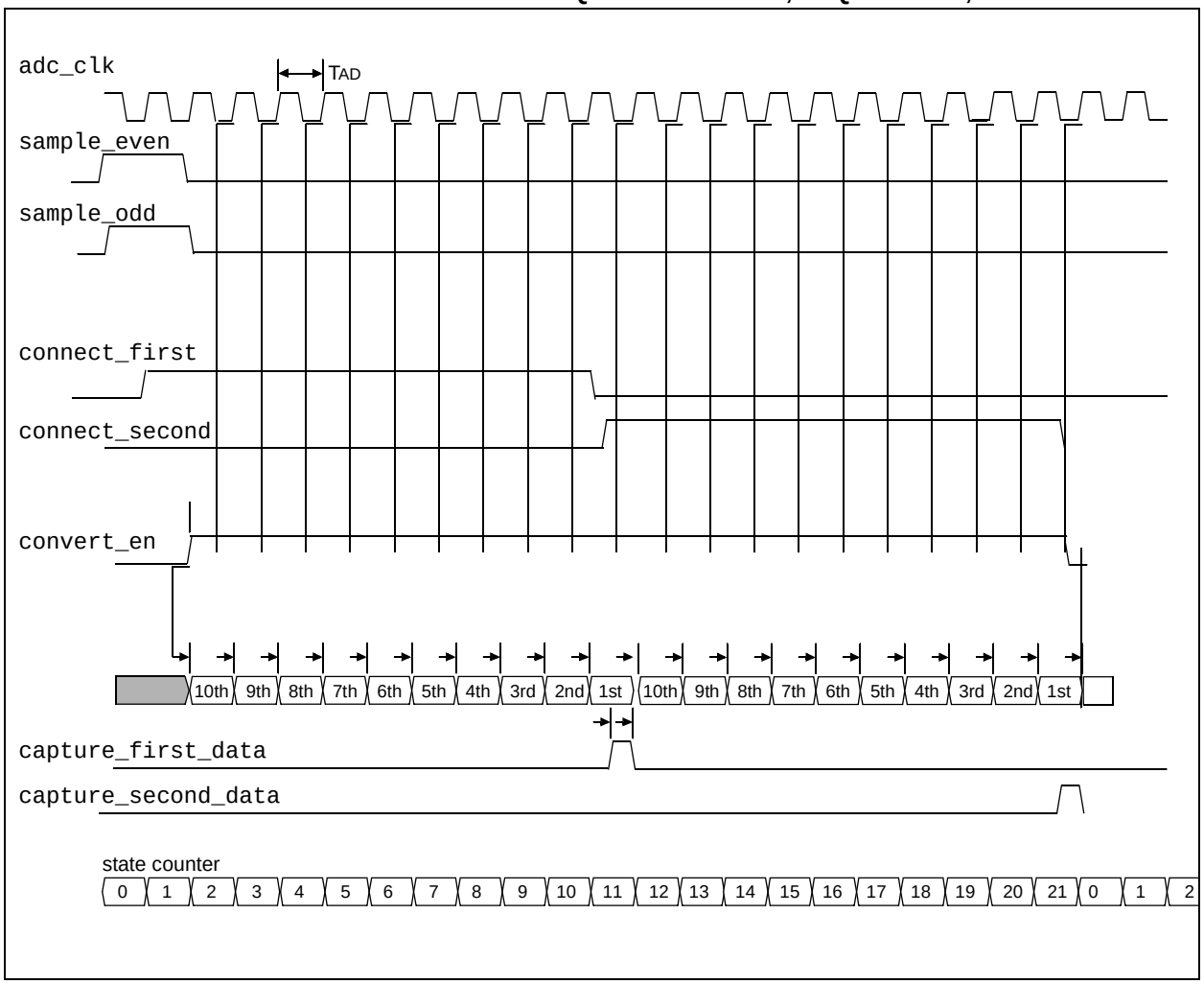
0 = Port pin in Analog mode, port read input disabled, A/D samples pin voltage

16.17 A/D Sample and Convert Timing

The sample and hold circuits assigned to the input pins have their own timing logic that is triggered when an external sample and convert request (from PWM or TMR) is made. The sample and hold circuits have a fixed two clock data sample period. When the sample has been acquired, then the ADC control logic is notified of a pending request, then the conversion is performed as the conversion resources become available.

The ADC module always converts pairs of analog input channels, so a typical conversion process requires 24 clock cycles.

FIGURE 16-3: DETAILED CONVERSION SEQUENCE TIMINGS, SEQSAMP = 0, NOT BUSY



16.18 Module Power-Down Modes

The module has two internal power modes.

When the ADON bit is '1', the module is in Active mode and is fully powered and functional.

When ADON is '0', the module is in Off mode. The state machine for the module is reset, as are all of the pending conversion requests.

To return to the Active mode from Off mode, the user must wait for the bias generators to stabilize. The stabilization time is specified in the electrical specs.

16.19 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the ADC module to be turned off, and any conversion and sampling sequence is aborted. The value that is in the ADCBUFx register is not modified.

The ADCBUFx registers contain unknown data after a Power-on Reset.

16.20 Configuring Analog Port Pins

The use of the ADPCFG and TRIS registers control the operation of the A/D port pins.

The port pins that are desired as analog inputs should have their corresponding TRIS bit set (input). If the TRIS bit is cleared (output), the digital output level (V_{OH} or V_{OL}) will be converted.

Port pins that are desired as analog inputs must have the corresponding ADPCFG bit clear. This will configure the port to disable the digital input buffer. Analog levels on pins where ADPCFG<n> = 1, may cause the digital input buffer to consume excessive current.

If a pin is not configured as an analog input ADPCFG<n> = 1, the analog input is forced to AVss, and conversions of that input do not yield meaningful results.

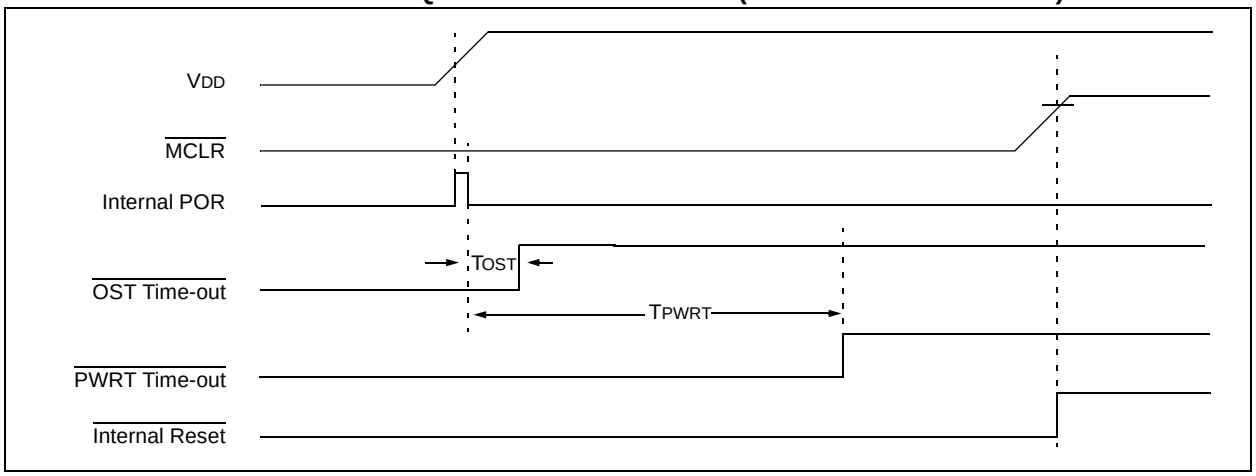
When reading the PORT register, all pins configured as analog input ADPCFG<n> = 0 will read '0'.

The A/D operation is independent of the state of the input selection bits and the TRIS bits.

16.21 Output Formats

The A/D converts 10 bits. The data buffer RAM is 16 bits wide. The ADC data can be read in one of two different formats, as shown in Figure 16-5. The FORM bit selects the format. Each of the output formats translates to a 16-bit result on the data bus.

FIGURE 18-10: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2



18.7.1.1 POR with Long Crystal Start-up Time (with FSCM Enabled)

The oscillator start-up circuitry is not linked to the POR circuitry. Some crystal circuits (especially low frequency crystals) will have a relatively long start-up time. Therefore, one or more of the following conditions is possible after the POR timer and the PWRT have expired:

- The oscillator circuit has not begun to oscillate.
- The Oscillator Start-up Timer has NOT expired (if a crystal oscillator is used).
- The PLL has not achieved a LOCK (if PLL is used).

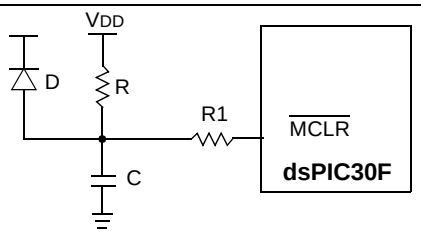
If the FSCM is enabled and one of the above conditions is true, then a clock failure trap will occur. The device will automatically switch to the FRC oscillator and the user can switch to the desired crystal oscillator in the trap, ISR.

18.7.1.2 Operating without FSCM and PWRT

If the FSCM is disabled and the Power-up Timer (PWRT) is also disabled, then the device will exit rapidly from Reset on power-up. If the clock source is FRC or EC, it will be active immediately.

If the FSCM is disabled and the system clock has not started, the device will be in a frozen state at the Reset vector until the system clock starts. From the user's perspective, the device will appear to be in Reset until a system clock is available.

FIGURE 18-11: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)



Note 1: External Power-on Reset circuit is required only if the VDD power-up slope is too slow. The diode D helps discharge the capacitor quickly when VDD powers down.

2: R should be suitably chosen so as to make sure that the voltage drop across R does not violate the device's electrical specification.

3: R1 should be suitably chosen so as to limit any current flowing into MCLR from external capacitor C, in the event of MCLR/VPP pin breakdown due to Electrostatic Discharge (ESD) or Electrical Overstress (EOS).

Note: Dedicated supervisory devices, such as the MCP1XX and MCP8XX, may also be used as an external Power-on Reset circuit.

Table 18-3 shows the Reset conditions for the RCON register. Since the control bits within the RCON register are R/W, the information in the table implies that all the bits are negated prior to the action specified in the condition column.

TABLE 18-3: INITIALIZATION CONDITION FOR RCON REGISTER CASE 1

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

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

Table 18-4 shows a second example of the bit conditions for the RCON register. In this case, it is not assumed the user has set/cleared specific bits prior to action specified in the condition column.

TABLE 18-4: INITIALIZATION CONDITION FOR RCON REGISTER CASE 2

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

Legend: u = unchanged

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

20.6 MPLAB X SIM Software Simulator

The MPLAB X SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB X SIM Software Simulator fully supports symbolic debugging using the MPLAB XC Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

20.7 MPLAB REAL ICE In-Circuit Emulator System

The MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs all 8, 16 and 32-bit MCU, and DSC devices with the easy-to-use, powerful graphical user interface of the MPLAB X IDE.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ-11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB X IDE. MPLAB REAL ICE offers significant advantages over competitive emulators including full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, logic probes, a ruggedized probe interface and long (up to three meters) interconnection cables.

20.8 MPLAB ICD 3 In-Circuit Debugger System

The MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost-effective, high-speed hardware debugger/programmer for Microchip Flash DSC and MCU devices. It debugs and programs PIC Flash microcontrollers and dsPIC DSCs with the powerful, yet easy-to-use graphical user interface of the MPLAB IDE.

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

20.9 PICkit 3 In-Circuit Debugger/ Programmer

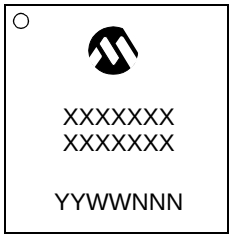
The MPLAB PICkit 3 allows debugging and programming of PIC and dsPIC Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB IDE. The MPLAB PICkit 3 is connected to the design engineer's PC using a full-speed USB interface and can be connected to the target via a Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the Reset line to implement in-circuit debugging and In-Circuit Serial Programming™ (ICSP™).

20.10 MPLAB PM3 Device Programmer

The MPLAB PM3 Device Programmer is a universal, CE compliant device programmer with programmable voltage verification at V_{DDMIN} and V_{DDMAX} for maximum reliability. It features a large LCD display (128 x 64) for menus and error messages, and a modular, detachable socket assembly to support various package types. The ICSP cable assembly is included as a standard item. In Stand-Alone mode, the MPLAB PM3 Device Programmer can read, verify and program PIC devices without a PC connection. It can also set code protection in this mode. The MPLAB PM3 connects to the host PC via an RS-232 or USB cable. The MPLAB PM3 has high-speed communications and optimized algorithms for quick programming of large memory devices, and incorporates an MMC card for file storage and data applications.

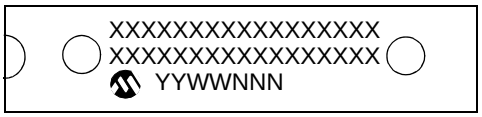
22.0 PACKAGE MARKING INFORMATION

28-Lead QFN-S

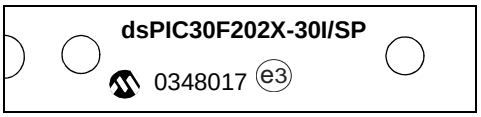


Example

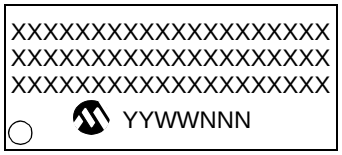
28-Lead PDIP (Skinny DIP)



Example

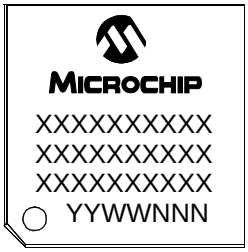


28-Lead SOIC

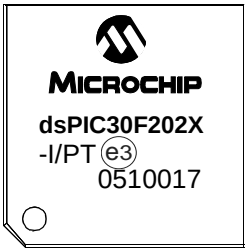


Example

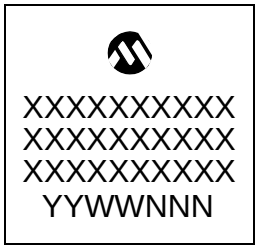
44-Lead TQFP



Example



44-Lead QFN



Example

Legend:	XX...X	Customer-specific information
	e3	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
		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.