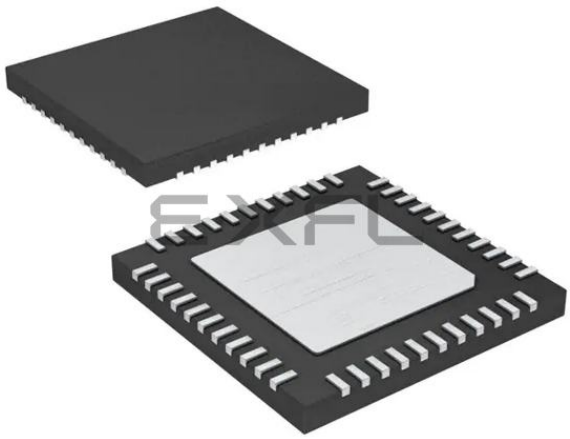


Welcome to E-XFL.COM

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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, Power Control PWM, QEI, POR, PWM, WDT
Number of I/O	36
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	768 x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 9x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4431-e-ml

TABLE 1-3: PIC18F4331/4431 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/PWM0	33	8	9			
RB0				I/O	TTL	Digital I/O.
PWM0				O	TTL	PWM Output 0.
RB1/PWM1	34	9	10			
RB1				I/O	TTL	Digital I/O.
PWM1				O	TTL	PWM Output 1.
RB2/PWM2	35	10	11			
RB2				I/O	TTL	Digital I/O.
PWM2				O	TTL	PWM Output 2.
RB3/PWM3	36	11	12			
RB3				I/O	TTL	Digital I/O.
PWM3				O	TTL	PWM Output 3.
RB4/KBI0/PWM5	37	14	14			
RB4				I/O	TTL	Digital I/O.
KBI0				I	TTL	Interrupt-on-change pin.
PWM5				O	TTL	PWM Output 5.
RB5/KBI1/PWM4/ PGM	38	15	15			
RB5				I/O	TTL	Digital I/O.
KBI1				I	TTL	Interrupt-on-change pin.
PWM4				O	TTL	PWM Output 4.
PGM				I/O	ST	Single-Supply ICSP™ Programming entry pin.
RB6/KBI2/PGC	39	16	16			
RB6				I/O	TTL	Digital I/O.
KBI2				I	TTL	Interrupt-on-change pin.
PGC				I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	40	17	17			
RB7				I/O	TTL	Digital I/O.
KBI3				I	TTL	Interrupt-on-change pin.
PGD				I/O	ST	In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL; RC7 is the alternate pin for SDO.

2: RD4 is the alternate pin for $\overline{\text{FLTA}}$.

3: RD5 is the alternate pin for PWM4.

TABLE 1-3: PIC18F4331/4431 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	TQFP	QFN			
						PORTE is a bidirectional I/O port.
RE0/AN6	8	25	25			
RE0				I/O	ST	Digital I/O.
AN6				I	Analog	Analog Input 6.
RE1/AN7	9	26	26			
RE1				I/O	ST	Digital I/O.
AN7				I	Analog	Analog Input 7.
RE2/AN8	10	27	27			
RE2				I/O	ST	Digital I/O.
AN8				I	Analog	Analog Input 8.
VSS	12, 31	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	7, 28	7, 8, 28, 29	P	—	Positive supply for logic and I/O pins.
NC	—	12, 13, 33, 34	13	NC	NC	No connect.

Legend: TTL = TTL compatible input; CMOS = CMOS compatible input or output; ST = Schmitt Trigger input with CMOS levels; I = Input; O = Output; P = Power

Note 1: RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL; RC7 is the alternate pin for SDO.

2: RD4 is the alternate pin for $\overline{\text{FLTA}}$.

3: RD5 is the alternate pin for PWM4.

2.5 External Oscillator Pins

Many microcontrollers have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator (refer to **Section 3.0 "Oscillator Configurations"** for details).

The oscillator circuit should be placed on the same side of the board as the device. Place the oscillator circuit close to the respective oscillator pins with no more than 0.5 inch (12 mm) between the circuit components and the pins. The load capacitors should be placed next to the oscillator itself, on the same side of the board.

Use a grounded copper pour around the oscillator circuit to isolate it from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed.

Layout suggestions are shown in Figure 2-4. In-line packages may be handled with a single-sided layout that completely encompasses the oscillator pins. With fine-pitch packages, it is not always possible to completely surround the pins and components. A suitable solution is to tie the broken guard sections to a mirrored ground layer. In all cases, the guard trace(s) must be returned to ground.

In planning the application's routing and I/O assignments, ensure that adjacent port pins and other signals in close proximity to the oscillator are benign (i.e., free of high frequencies, short rise and fall times, and other similar noise).

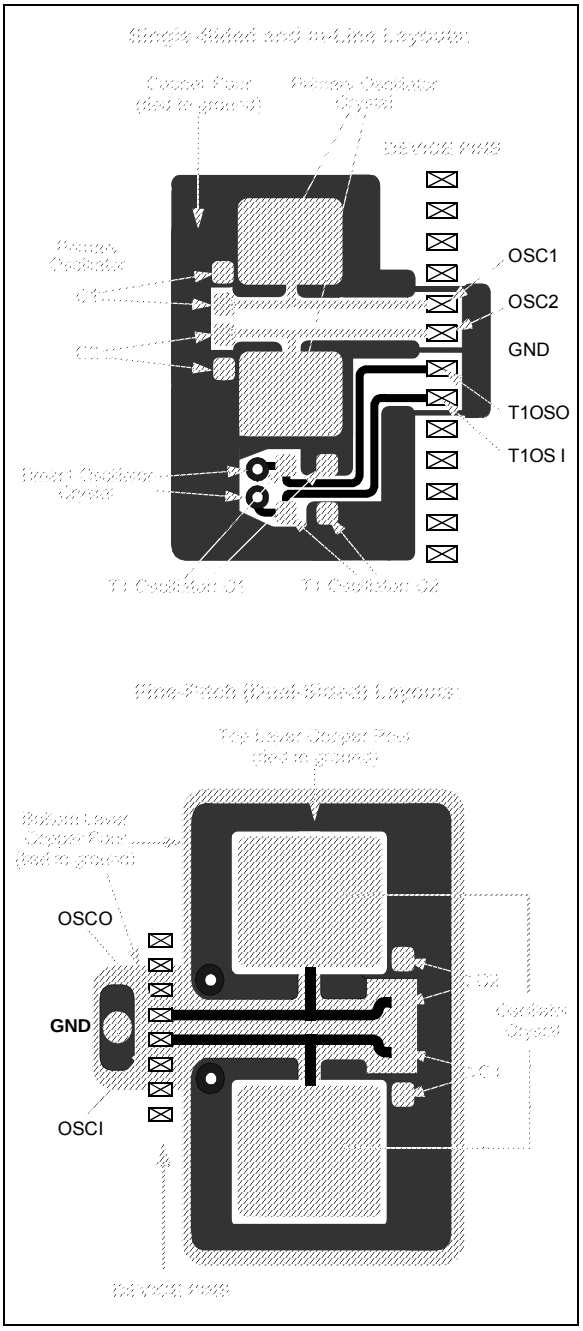
For additional information and design guidance on oscillator circuits, please refer to these Microchip Application Notes, available at the corporate web site (www.microchip.com):

- *AN826, "Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices"*
- *AN849, "Basic PICmicro® Oscillator Design"*
- *AN943, "Practical PICmicro® Oscillator Analysis and Design"*
- *AN949, "Making Your Oscillator Work"*

2.6 Unused I/Os

Unused I/O pins should be configured as outputs and driven to a logic low state. Alternatively, connect a 1 kΩ to 10 kΩ resistor to VSS on unused pins and drive the output to logic low.

FIGURE 2-3: SUGGESTED PLACEMENT OF THE OSCILLATOR CIRCUIT



5.5 Device Reset Timers

PIC18F2331/2431/4331/4431 devices incorporate three separate on-chip timers that help regulate the Power-on Reset process. Their main function is to ensure that the device clock is stable before code is executed. These timers are:

- Power-up Timer (PWRT)
- Oscillator Start-up Timer (OST)
- PLL Lock Time-out

5.5.1 POWER-UP TIMER (PWRT)

The Power-up Timer (PWRT) of PIC18F2331/2431/4331/4431 devices is an 11-bit counter that uses the INTRC source as the clock input. This yields an approximate time interval of 2,048 x 32 μs = 65.6 ms.

While the PWRT is counting, the device is held in Reset. The power-up time delay depends on the INTRC clock and will vary from chip to chip due to temperature and process variation. See DC Parameter 33 for details.

The PWRT is enabled by clearing the PWRTEN Configuration bit.

5.5.2 OSCILLATOR START-UP TIMER (OST)

The Oscillator Start-up Timer (OST) provides a 1,024 oscillator cycle (from OSC1 input) delay after the PWRT delay is over (Parameter 33). This ensures that the crystal oscillator or resonator has started and stabilized.

The OST time-out is invoked only for XT, LP, HS and HSPLL modes, and on Power-on Reset or on exit from most power-managed modes.

5.5.3 PLL LOCK TIME-OUT

With the PLL enabled in its PLL mode, the time-out sequence following a Power-on Reset is slightly different from other oscillator modes. A separate timer is used to provide a fixed time-out that is sufficient for the PLL to lock to the main oscillator frequency. This PLL Lock Time-out (T_{PLL}) is typically 2 ms and follows the oscillator start-up time-out.

5.5.4 TIME-OUT SEQUENCE

On power-up, the time-out sequence is as follows:

1. After the POR pulse has cleared, the PWRT time-out is invoked (if enabled).
2. Then, the OST is activated.

The total time-out will vary based on oscillator configuration and the status of the PWRT. Figure 5-3 through Figure 5-7 depict time-out sequences on power-up, with the Power-up Timer enabled and the device operating in HS Oscillator mode. Figure 5-3 through Figure 5-6 also apply to devices operating in XT or LP modes.

For devices in RC mode, and with the PWRT disabled, there will be no time-out at all. Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, all time-outs will expire. Bringing $\overline{MCLR}$ high will begin execution immediately (Figure 5-5). This is useful for testing purposes or synchronization of more than one PIC18FXXXX device operating in parallel.

TABLE 5-1: TIME-OUT IN VARIOUS SITUATIONS

Oscillator Configuration	Power-up(2) and Brown-out		Exit From Power-Managed Mode
	$\overline{PWRTEN}$ = 0	$\overline{PWRTEN}$ = 1	
HSPLL	66 ms(1) + 1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)	1024 TOSC + 2 ms(2)
HS, XT, LP	66 ms(1) + 1024 TOSC	1024 TOSC	1024 TOSC
EC, ECIO	66 ms(1)	—	—
RC, RCIO	66 ms(1)	—	—
INTIO1, INTIO2	66 ms(1)	—	—

Note 1: 66 ms (65.5 ms) is the nominal Power-up Timer (PWRT) delay.
2: 2 ms is the nominal time required for the 4x PLL to lock.

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
ADRESH	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADRESL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
ADCON1	2331	2431	4331	4431	00-0 0000	00-0 0000	uu-u uuuu
ADCON2	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
ADCON3	2331	2431	4331	4431	00-0 0000	00-0 0000	uu-u uuuu
ADCHS	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
CCPR1H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
CCPR2H	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR2L	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP2CON	2331	2431	4331	4431	--00 0000	--00 0000	--uu uuuu
ANSEL1	2331	2431	4331	4431	---- ---1	---- ---1	---- ---u
ANSEL0	2331	2431	4331	4431	1111 1111	1111 1111	uuuu uuuu
T5CON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
QEICON	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SPBRGH	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
SPBRG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
RCREG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TXREG	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
TXSTA	2331	2431	4331	4431	0000 -010	0000 -010	uuuu -uuu
RCSTA	2331	2431	4331	4431	0000 000x	0000 000x	uuuu uuuu
BAUDCON	2331	2431	4331	4431	-1-1 0-00	-1-1 0-00	-u-u u-uu
EEADR	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
EEDATA	2331	2431	4331	4431	0000 0000	0000 0000	uuuu uuuu
EECON2	2331	2431	4331	4431	0000 0000	0000 0000	0000 0000
EECON1	2331	2431	4331	4431	xx-0 x000	uu-0 u000	uu-0 u000
IPR3	2331	2431	4331	4431	---1 1111	---1 1111	---u uuuu
PIE3	2331	2431	4331	4431	---0 0000	---0 0000	---u uuuu
PIR3	2331	2431	4331	4431	---0 0000	---0 0000	---u uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 3 of PORTE and LATE are enabled if $\overline{\text{MCLR}}$ functionality is disabled. When not enabled as the PORTE pin, they are disabled and read as '0'. The 28-pin devices do not have only RE3 implemented.

TABLE 5-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices				Power-on Reset, Brown-out Reset	$\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
CAP3BUFL/ MAXCNTL	2331	2431	4331	4431	xxxx xxxx	uuuu uuuu	uuuu uuuu
CAP1CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
CAP2CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
CAP3CON	2331	2431	4331	4431	-0-- 0000	-0-- 0000	-u-- uuuu
DFLTCON	2331	2431	4331	4431	-000 0000	-000 0000	-uuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 5-2 for Reset value for specific condition.

5: Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read '0'.

6: Bit 3 of PORTE and LATE are enabled if $\overline{\text{MCLR}}$ functionality is disabled. When not enabled as the PORTE pin, they are disabled and read as '0'. The 28-pin devices do not have only RE3 implemented.

6.1.1 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and contained in three 8-bit registers. The low byte, known as the PCL register, is both readable and writable. The high byte (PCH register) contains the PC<15:8> bits and is not directly readable or writable.

Updates to the PCH register are performed through the PCLATH register. The upper byte is the PCU register and contains the bits, PC<20:16>. This register is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the program counter by any operation that writes to the PCL. Similarly, the upper two bytes of the program counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 6.1.4.1 "Computed GOTO"**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of the PCL is fixed to a value of '0'. The PC increments by two to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

6.1.2 RETURN ADDRESS STACK

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, with the Stack Pointer initialized to 00000b after all Resets. There is no RAM associated with Stack Pointer, 00000b. This is only a Reset value. During a CALL type instruction, causing a push onto the stack, the Stack Pointer is first incremented and the RAM location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). During a RETURN type instruction, causing a pop from the stack, the contents of the RAM location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

The stack space is not part of either program or data space. The Stack Pointer is readable and writable, and the address on the top of the stack is readable and writable through the Top-of-Stack (TOS) Special Function Registers. Data can also be pushed to, or popped from, the stack using the Top-of-Stack SFRs. Status bits indicate if the stack is full, has overflowed or underflowed.

6.1.2.1 Top-of-Stack Access

The top of the stack is readable and writable. Three register locations, TOSU, TOSH and TOSL, hold the contents of the stack location pointed to by the STKPTR register (Figure 6-3). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU, TOSH and TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can replace the TOSU, TOSH and TOSL and do a return.

The user must disable the global interrupt enable bits while accessing the stack to prevent inadvertent stack corruption.

6.1.2.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 6-1) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. At Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 23.1 "Configuration Bits"** for a description of the device Configuration bits.) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and STKPTR will remain at 31.

11.4 PORTD, TRISD and LATD Registers

Note: PORTD is only available on PIC18F4331/4431 devices.

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction register is TRISD. Setting a TRISD bit (= 1) will make the corresponding PORTD pin an input (i.e., put the corresponding output driver in a high-impedance mode). Clearing a TRISD bit (= 0) will make the corresponding PORTD pin an output (i.e., put the contents of the output latch on the selected pin).

The Data Latch register (LATD) is also memory mapped. Read-modify-write operations on the LATD register read and write the latched output value for PORTD.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

Note: On a Power-on Reset, these pins are configured as digital inputs.

PORTD includes PWM<7:6> complementary fourth channel PWM outputs. PWM4 is the complementary output of PWM5 (the third channel), which is multiplexed with the RB5 pin. This output can be used as the alternate output using the PWM4MX Configuration bit in CONFIG3H when the Single-Supply Programming pin (PGM) is used on RB5.

RD1, RD2 and RD3 can be used as the alternate output for SDO, SDI/SDA and SCK/SCL using the SSPMX Configuration bit in CONFIG3H.

RD4 an be used as the alternate output for $\overline{\text{FLTA}}$ using the FLTAMX Configuration bit in CONFIG3H.

EXAMPLE 11-4: INITIALIZING PORTD

```
CLRF    PORTD   ; Initialize PORTD by
                ; clearing output
                ; data latches
CLRF    LATD    ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0xCF    ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISD   ; Set RD<3:0> as inputs
                ; RD<5:4> as outputs
                ; RD<7:6> as inputs
```

17.1.2 PERIOD MEASUREMENT MODE

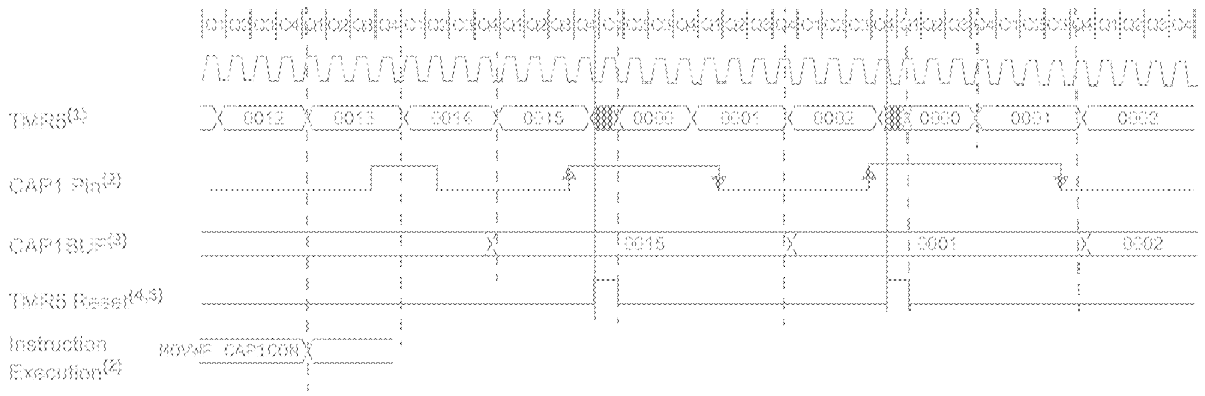
The Period Measurement mode is selected by setting CAPxM<3:0> = 0101. In this mode, the value of Timer5 is latched into the CAPxBUF register on the rising edge of the input capture trigger and Timer5 is subsequently reset to 0000h (optional by setting CAPxREN = 1) on the next TCY (see capture and Reset relationship in Figure 17-4).

17.1.3 PULSE-WIDTH MEASUREMENT MODE

The Pulse-Width Measurement mode can be configured for two different edge sequences, such that the pulse width is based on either the falling to rising edge of the CAPx input pin (CAPxM<3:0> = 0110), or on the rising to falling edge (CAPxM<3:0> = 0111).

Timer5 is always reset on the edge when the measurement is first initiated. For example, when the measurement is based on the falling to rising edge, Timer5 is first reset on the falling edge, and thereafter, the timer value is captured on the rising edge. Upon entry into the Pulse-Width Measurement mode, the very first edge detected on the CAPx pin is always captured. The TMR5 value is reset on the first active edge (see Figure 17-5).

FIGURE 17-5: PULSE-WIDTH MEASUREMENT MODE TIMING



Note 1: TMR5 is a synchronous time base input to the input capture; prescaler = 1:1. It increments on every Q1 rising edge.

2: IC1 is configured in Pulse-Width Measurement mode (CAP1M<3:0> = 0111, rising to falling pulse-width measurement). No noise filter on CAP1 input is used. The MOVWF instruction loads CAP1CON when W = 0111.

3: TMR5 value is latched by CAP1BUF on TCY rising edge. In the event that a write to TMR5 coincides with an input capture event, the write will always take precedence. All Input Capture Buffers, CAP1BUF, CAP2BUF and CAP3BUF, are updated with the incremented value of the time base on the next TCY clock edge when the capture event takes place (see Note 4 when Reset occurs).

4: TMR5 Reset is normally an asynchronous Reset signal to TMR5. When used in Pulse-Width Measurement mode, it is always present on the edge that first initiates the pulse-width measurement (i.e., when configured in the rising to falling Pulse-Width Measurement mode), it is active on each rising edge detected. In the falling to rising Pulse-Width Measurement mode, it is active on each falling edge detected.

5: TMR5 Reset pulse is activated on the capture edge. The CAP1REN bit has no bearing in this mode.

REGISTER 18-4: PWMCON1: PWM CONTROL REGISTER 1

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
SEVOPS3	SEVOPS2	SEVOPS1	SEVOPS0	SEVTDIR	—	UDIS	OSYNC
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 7-4 **SEVOPS<3:0>:** PWM Special Event Trigger Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
.
.
.
1111 = 1:16 Postscale

bit 3 **SEVTDIR:** Special Event Trigger Time Base Direction bit
1 = A Special Event Trigger will occur when the PWM time base is counting downwards
0 = A Special Event Trigger will occur when the PWM time base is counting upwards

bit 2 **Unimplemented:** Read as '0'

bit 1 **UDIS:** PWM Update Disable bit
1 = Updates from Duty Cycle and Period Buffer registers are disabled
0 = Updates from Duty Cycle and Period Buffer registers are enabled

bit 0 **OSYNC:** PWM Output Override Synchronization bit
1 = Output overrides via the OVDCON register are synchronized to the PWM time base
0 = Output overrides via the OVDCON register are asynchronous

18.3.1 FREE-RUNNING MODE

In the Free-Running mode, the PWM Time Base registers (PTMRL and PTMRH) will begin counting upwards until the value in the PWM Time Base Period register, PTPER (PTPERL and PTPERH), is matched. The PTMR registers will be reset on the following input clock edge and the time base will continue counting upwards as long as the PTEN bit remains set.

18.3.2 SINGLE-SHOT MODE

In the Single-Shot mode, the PWM time base will begin counting upwards when the PTEN bit is set. When the value in the PTMR register matches the PTPER register, the PTMR register will be reset on the following input clock edge and the PTEN bit will be cleared by the hardware to halt the time base.

18.3.3 CONTINUOUS UP/DOWN COUNT MODES

In Continuous Up/Down Count modes, the PWM time base counts upwards until the value in the PTPER register matches with the PTMR register. On the following input clock edge, the timer counts downwards. The PTDIR bit in the PTCON1 register is read-only and indicates the counting direction. The PTDIR bit is set when the timer counts downwards.

Note: Since the PWM compare outputs are driven to the active state when the PWM time base is counting downwards and matches the duty cycle value, the PWM outputs are held inactive during the first half of the first period of the Continuous Up/Down Count mode until PTMR begins to count down from the PTPER value.

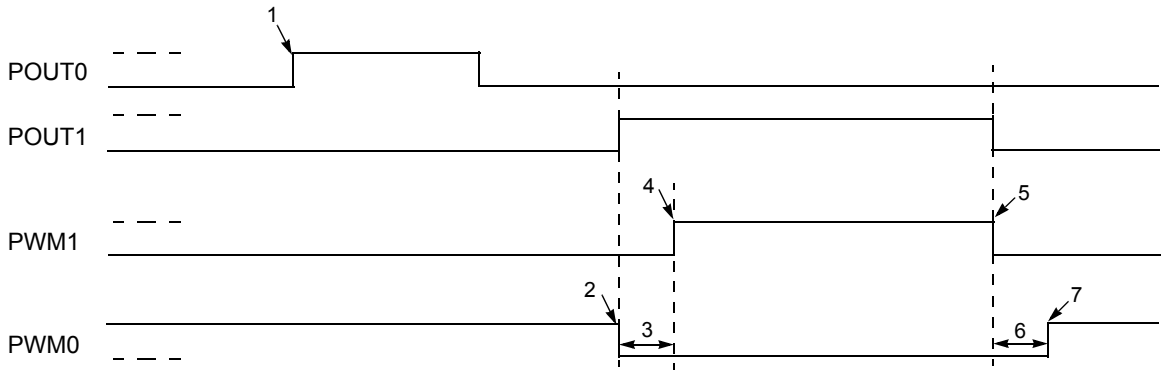
18.3.4 PWM TIME BASE PRESCALER

The input clock to PTMR (FOSC/4) has prescaler options of 1:1, 1:4, 1:16 or 1:64. These are selected by control bits, PTCKPS<1:0>, in the PTCON0 register. The prescaler counter is cleared when any of the following occurs:

- Write to the PTMR register
- Write to the PTCON (PTCON0 or PTCON1) register
- Any device Reset

Note: The PTMR register is not cleared when PTCONx is written.

FIGURE 18-20: PWM OVERRIDE BITS IN COMPLEMENTARY MODE



Assume: POVD0 = 0; POVD1 = 0; PMOD0 = 0

1. Even override bits have no effect in Complementary mode.
2. Odd override bit is activated, which causes the even PWM to deactivate.
3. Dead-time insertion.
4. Odd PWM activated after the dead time.
5. Odd override bit is deactivated, which causes the odd PWM to deactivate.
6. Dead-time insertion.
7. Even PWM is activated after the dead time.

19.0 SYNCHRONOUS SERIAL PORT (SSP) MODULE

19.1 SSP Module Overview

The Synchronous Serial Port (SSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The SSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C™)

An overview of I^2C operations and additional information on the SSP module can be found in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

Refer to application note *AN578, "Use of the SSP Module in the I^2C™ Multi-Master Environment"* (DS00578).

19.2 SPI Mode

This section contains register definitions and operational characteristics of the SPI module. Additional information on the SPI module can be found in the *"PIC® Mid-Range MCU Family Reference Manual"* (DS33023).

SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO)
- Serial Data In (SDI)
- Serial Clock (SCK)

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ($\overline{SS}$)

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits in the SSPCON (SSPCON<5:0>) and SSPSTAT<7:6> registers. These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock polarity (Idle state of SCK)
- Clock edge (output data on rising/falling edge of SCK)
- Clock rate (Master mode only)
- Slave Select mode (Slave mode only)

FIGURE 19-2: SPI MODE TIMING, MASTER MODE

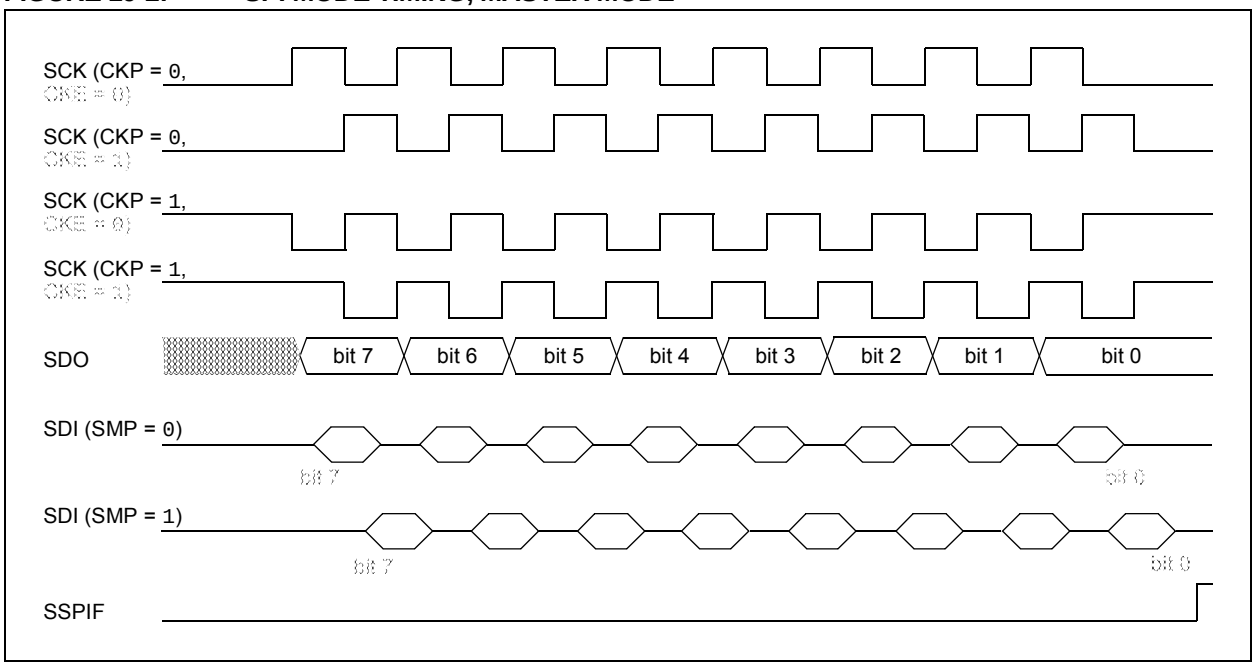
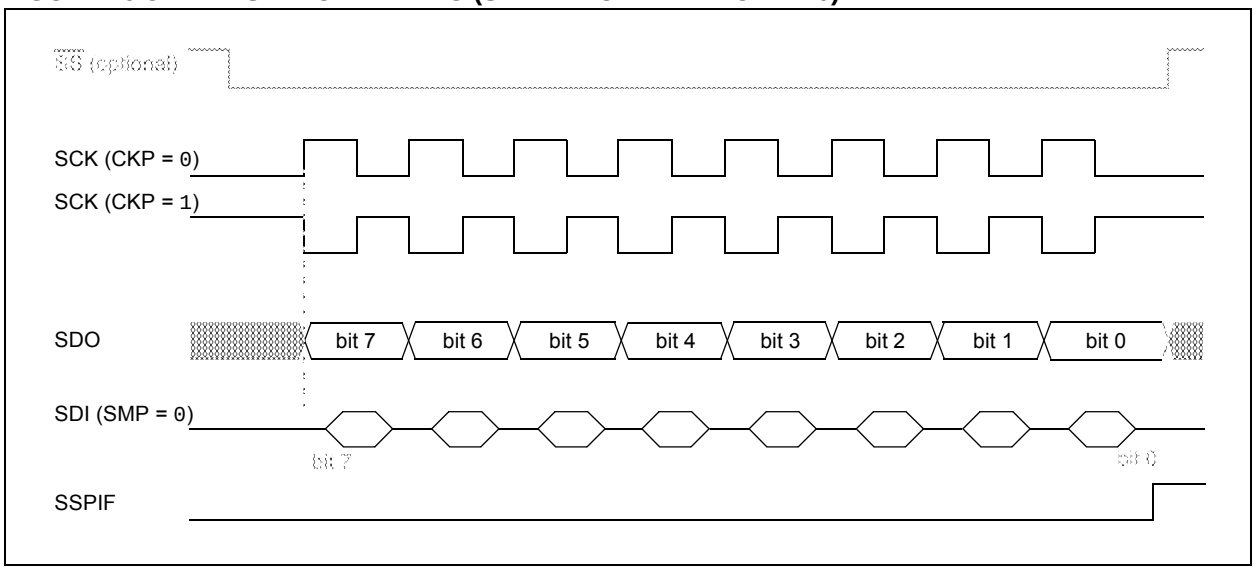


FIGURE 19-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)



REGISTER 23-2: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

U-0	U-0	U-0	U-0	R/P-1	R/P-1	R/P-1	R/P-1
—	—	—	—	BORV1	BORV0	BOREN(1)	$\overline{\text{PWRTEN}}$(1)
bit 7							bit 0

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0'

bit 3-2 **BORV<1:0>:** Brown-out Reset Voltage bits
- 11 = Reserved
- 10 = V_{BOR} set to 2.7V
- 01 = V_{BOR} set to 4.2V
- 00 = V_{BOR} set to 4.5V

bit 1 **BOREN:** Brown-out Reset Enable bit(1)
- 1 = Brown-out Reset is enabled
- 0 = Brown-out Reset is disabled

bit 0 **$\overline{\text{PWRTEN}}$:** Power-up Timer Enable bit(1)
- 1 = PWRT is disabled
- 0 = PWRT is enabled

Note 1: Having BOREN = 1 does not automatically override the $\overline{\text{PWRTEN}}$ to '0', nor automatically enables the Power-up Timer.

REGISTER 23-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3(1,2)	CP2(1,2)	CP1(2)	CP0(2)
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit(1,2)
- 1 = Block 3 is not code-protected
- 0 = Block 3 is code-protected

bit 2 **CP2:** Code Protection bit(1,2)
- 1 = Block 2 is not code-protected
- 0 = Block 2 is code-protected

bit 1 **CP1:** Code Protection bit(2)
- 1 = Block 1 is not code-protected
- 0 = Block 1 is code-protected

bit 0 **CP0:** Code Protection bit(2)
- 1 = Block 0 is not code-protected
- 0 = Block 0 is code-protected

Note 1: Unimplemented in PIC18F2331/4331 devices; maintain this bit set.
2: Refer to Figure 23-5 for block boundary addresses.

REGISTER 23-8: CONFIG5H: CONFIGURATION REGISTER 5 HIGH (BYTE ADDRESS 300009h)

R/C-1	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
CPD(1)	CPB(1)	—	—	—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		U = Unchanged from programmed state

bit 7 **CPD:** Data EEPROM Code Protection bit(1)
- 1 = Data EEPROM is not code-protected
- 0 = Data EEPROM is code-protected

bit 6 **CPB:** Boot Block Code Protection bit(1)
- 1 = Boot Block is not code-protected
- 0 = Boot Block is code-protected

bit 5-0 **Unimplemented:** Read as '0'

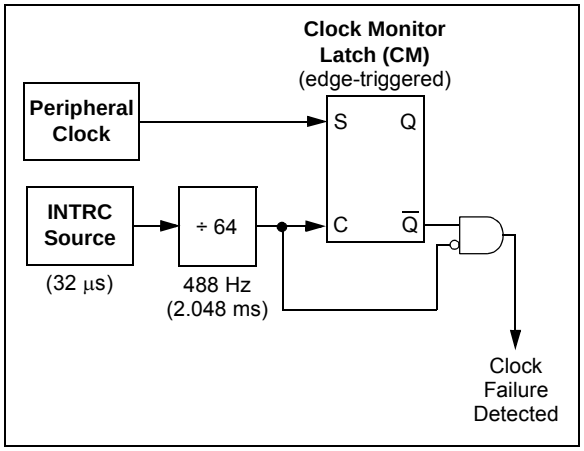
Note 1: Refer to Figure 23-5 for block boundary addresses.

23.4 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation, in the event of an external oscillator failure, by automatically switching the system clock to the internal oscillator block. The FSCM function is enabled by setting the Fail-Safe Clock Monitor Enable bit, FCMEN (CONFIG1H<6>).

When FSCM is enabled, the INTRC oscillator runs at all times to monitor clocks to peripherals and provide an instant backup clock in the event of a clock failure. Clock monitoring (shown in Figure 23-3) is accomplished by creating a sample clock signal, which is the INTRC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral system clock and the sample clock are presented as inputs to the Clock Monitor latch (CM). The CM is set on the falling edge of the system clock source, but cleared on the rising edge of the sample clock.

FIGURE 23-3: FSCM BLOCK DIAGRAM



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while the CM is still set, a clock failure has been detected (Figure 23-4). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- the system clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- the WDT is reset.

Since the postscaler frequency from the internal oscillator block may not be sufficiently stable, it may be desirable to select another clock configuration and enter an alternate power-managed mode (see **Section 23.3.1 "Special Considerations for Using Two-Speed Start-up"** and **Section 4.1.4 "Multiple Sleep Commands"** for more details). This can be done to attempt a partial recovery or execute a controlled shutdown.

To use a higher clock speed on wake-up, the INTOSC or postscaler clock sources can be selected to provide a higher clock speed by setting bits, IRCF<2:0>, immediately after Reset. For wake-ups from Sleep, the INTOSC or postscaler clock sources can be selected by setting the IRCF<2:0> bits prior to entering Sleep mode.

Adjustments to the internal oscillator block using the OSCTUNE register also affect the period of the FSCM by the same factor. This can usually be neglected, as the clock frequency being monitored is generally much higher than the sample clock frequency.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

23.4.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTOSC clock when a clock failure is detected. Depending on the frequency selected by the IRCF<2:0> bits, this may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock Monitor events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

23.4.2 EXITING FAIL-SAFE OPERATION

The fail-safe condition is terminated by either a device Reset, or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 1H (with any required start-up delays that are required for the oscillator mode, such as the OST or PLL timer). The INTOSC multiplexer provides the system clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock system source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexer. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

Entering a power-managed mode by loading the OSCCON register and executing a SLEEP instruction will clear the fail-safe condition. When the fail-safe condition is cleared, the clock monitor will resume monitoring the peripheral clock.

MOVLW **Move Literal to W**

Syntax: [*label*] MOVLW k

Operands: $0 \le k \le 255$

Operation: $k \rightarrow W$

Status Affected: None

Encoding:

0000	1110	kkkk	kkkk

Description: The 8-bit literal, ‘k’, is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read literal ‘k’	Process Data	Write to W

Example: MOVLW 0x5A

After Instruction
W = 0x5A

MOVWF **Move W to f**

Syntax: [*label*] MOVWF f [,a]

Operands: $0 \le f \le 255$
$a \in [0,1]$

Operation: $(W) \rightarrow f$

Status Affected: None

Encoding:

0110	111a	ffff	ffff

Description: Move data from W to register, ‘f’. Location, ‘f’, can be anywhere in the 256-byte bank. If ‘a’ is ‘0’, the Access Bank will be selected, overriding the BSR value. If ‘a’ = 1, then the bank will be selected as per the BSR value.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register ‘f’	Process Data	Write register ‘f’

Example: MOVWF REG

Before Instruction
W = 0x4F
REG = 0xFF
After Instruction
W = 0x4F
REG = 0x4F

FIGURE 26-14: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)

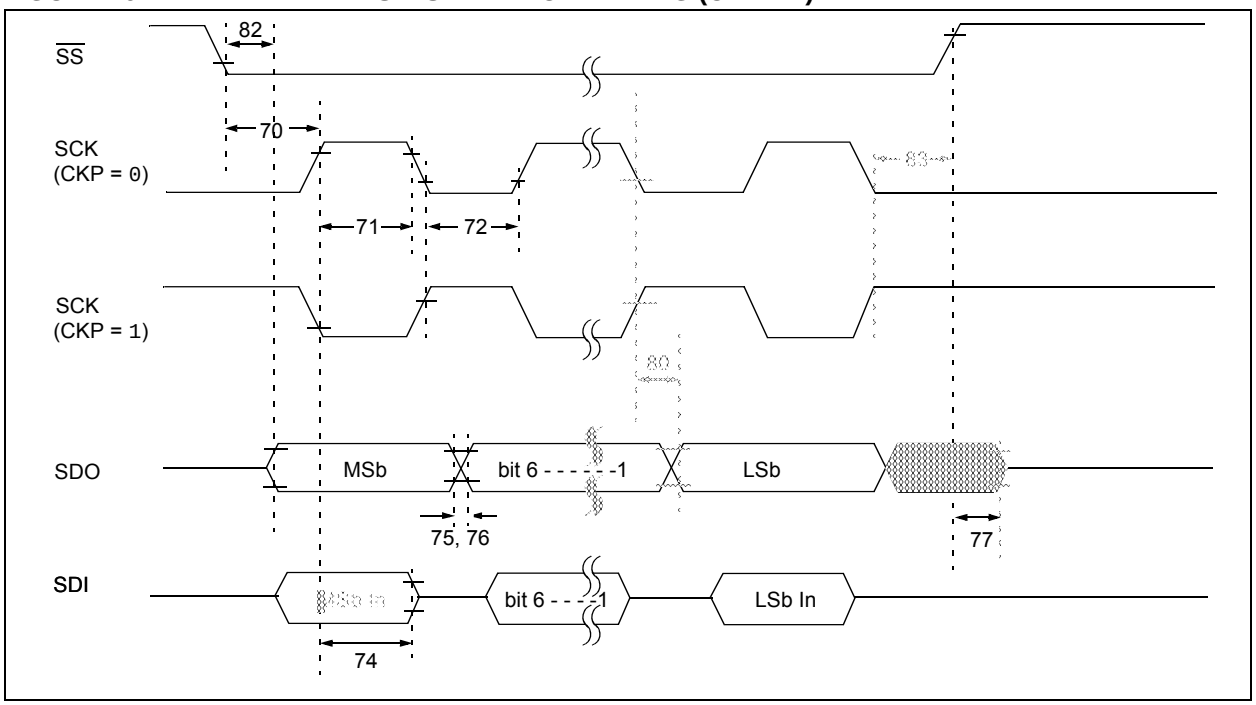


TABLE 26-14: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)

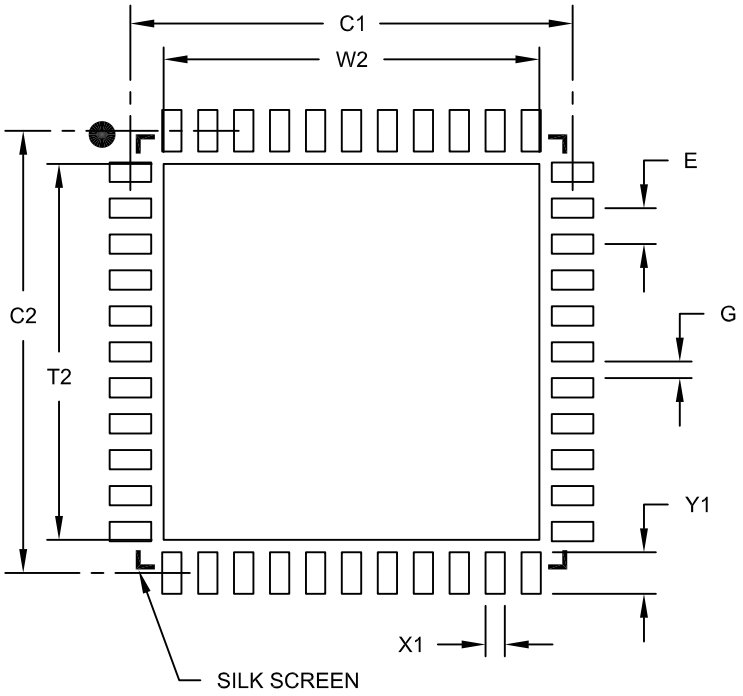
Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input		TCY	—	ns	
71	TscH	SCK Input High Time	Continuous	1.25 TCY + 30	—	ns	
71A			Single byte	40	—	ns	**(Note 1)**
72	TscL	SCK Input Low Time	Continuous	1.25 TCY + 30	—	ns	
72A			Single byte	40	—	ns	**(Note 1)**
73A	TB2B	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 TCY + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		40	—	ns	
75	TdoR	SDO Data Output Rise Time	PIC18FXX31	—	25	ns	
			PIC18LFXX31	—	45	ns	
76	TdoF	SDO Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SS}$ ↑ to SDO Output High-Impedance		10	50	ns	
80	TscH2doV, TscL2doV	SDO Data Output Valid after SCK Edge	PIC18FXX31	—	50	ns	
			PIC18LFXX31	—	100	ns	
82	TssL2doV	SDO Data Output Valid after $\overline{SS}$ ↓ Edge	PIC18FXX31	—	50	ns	
			PIC18LFXX31	—	100	ns	
83	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK Edge		1.5 TCY + 40	—	ns	

Note 1: Requires the use of Parameter 73A.

2: Only if Parameter 71A and 72A are used.

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