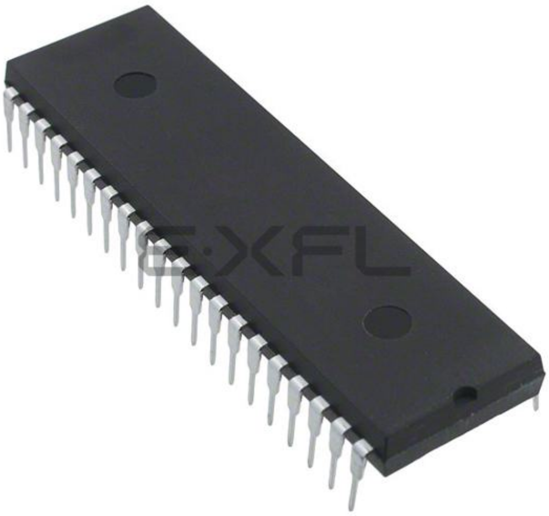
Microchip Technology - PIC18F4431-I/P Datasheet



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	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 ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	40-DIP (0.600", 15.24mm)
Supplier Device Package	40-PDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f4431-i-p

4.3 Sleep Mode

The power-managed Sleep mode in the PIC18F2331/2431/4331/4431 devices is identical to the legacy Sleep mode offered in all other PIC devices. It is entered by clearing the IDLEN bit (the default state on device Reset) and executing the `SLEEP` instruction. This shuts down the selected oscillator (Figure 4-5). All clock source status bits are cleared.

Entering the Sleep mode from any other mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source, selected by the SCS<1:0> bits, becomes ready (see Figure 4-6), or it will be clocked from the internal oscillator block if either the Two-Speed Start-up or the Fail-Safe Clock Monitor is enabled (see **Section 23.0 "Special Features of the CPU"**). In either case, the OSTS bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

4.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

If the IDLEN bit is set to a '`1`' when a `SLEEP` instruction is executed, the peripherals will be clocked from the clock source selected using the SCS<1:0> bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a `SLEEP` instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of TCSD (Parameter 38, Table 26-8) while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC_IDLE mode, the internal oscillator block will clock the CPU and peripherals (in other words, RC_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS<1:0> bits.

FIGURE 4-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE

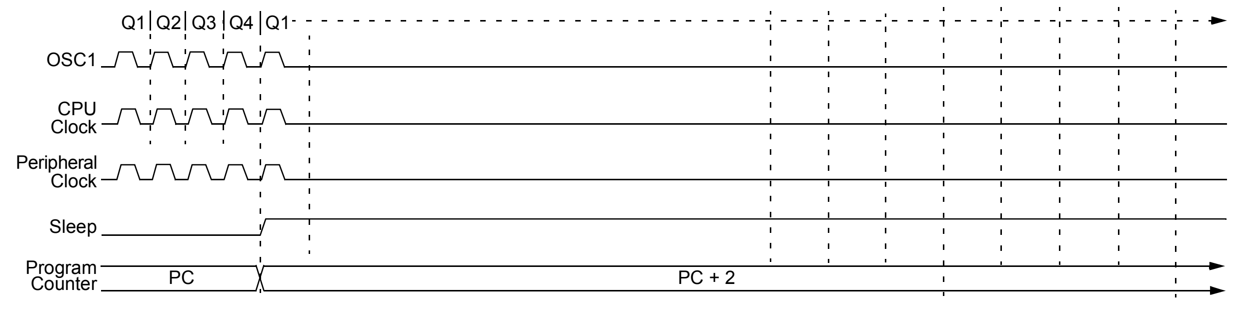
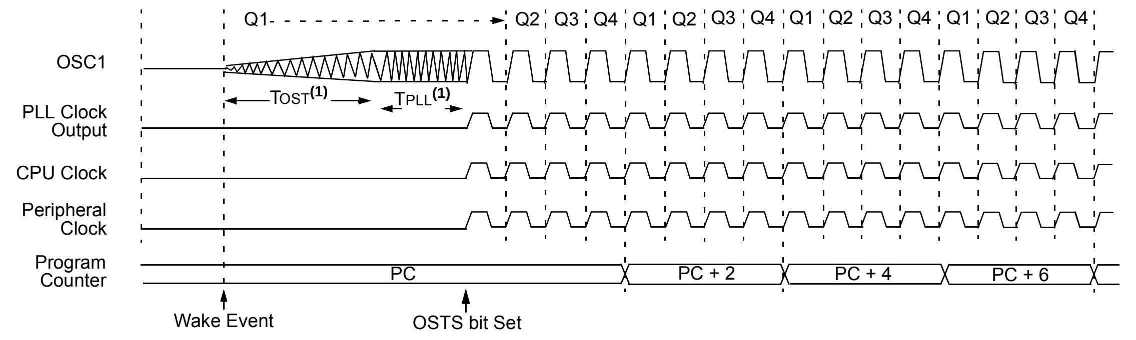


FIGURE 4-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)



Note 1: TOST = 1024 TOSC; TPLL = 2 ms (approx). These intervals are not shown to scale.

6.0 MEMORY ORGANIZATION

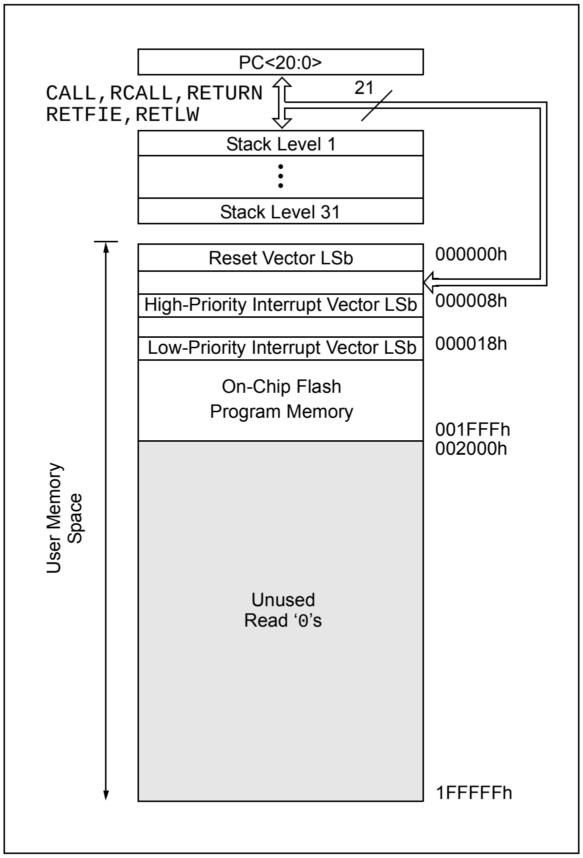
There are three memory types in enhanced MCU devices. These memory types are:

- Program Memory
- Data RAM
- Data EEPROM

As Harvard architecture devices, the data and program memories use separate buses, enabling concurrent access of the two memory spaces. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

Additional detailed information on the operation of the Flash program memory is provided in **Section 8.0 "Flash Program Memory"**. Data EEPROM is discussed separately in **Section 7.0 "Data EEPROM Memory"**.

FIGURE 6-1: PROGRAM MEMORY MAP AND STACK FOR PIC18F2331/4331



6.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter that can address a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all '`0`'s (a `NOP` instruction).

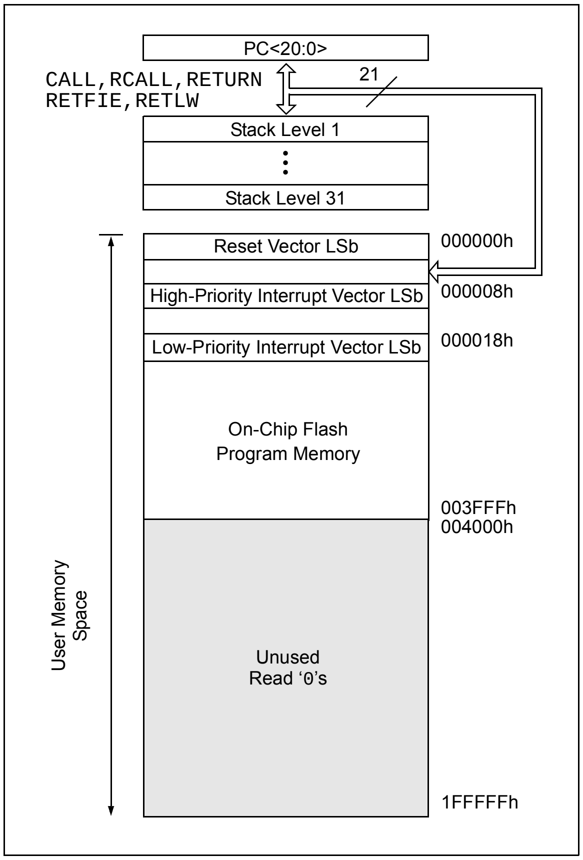
The PIC18F2331/4331 devices each have 8 Kbytes of Flash memory and can store up to 4,096 single-word instructions.

The PIC18F2431/4431 devices each have 16 Kbytes of Flash memory and can store up to 8,192 single-word instructions.

PIC18 devices have two interrupt vectors. The Reset vector address is at 000000h and the interrupt vector addresses are at 000008h and 000018h.

The program memory maps for PIC18F2331/4331 and PIC18F2431/4431 devices are shown in Figure 6-1 and Figure 6-2, respectively.

FIGURE 6-2: PROGRAM MEMORY MAP AND STACK FOR PIC18F2431/4431



EXAMPLE 8-3: WRITING TO FLASH PROGRAM MEMORY

```
           MOVLW  D'64'                      ; number of bytes in erase block
           MOVWF  COUNTER
           MOVLW  BUFFER_ADDR_HIGH           ; point to buffer
           MOVWF  FSR0H
           MOVLW  BUFFER_ADDR_LOW
           MOVWF  FSR0L
           MOVLW  CODE_ADDR_UPPER            ; Load TBLPTR with the base
           MOVWF  TBLPTRU                    ; address of the memory block
           MOVLW  CODE_ADDR_HIGH
           MOVWF  TBLPTRH
           MOVLW  CODE_ADDR_LOW              ; 6 LSB = 0
           MOVWF  TBLPTRL
READ_BLOCK
           TBLRD*+                           ; read into TABLAT, and inc
           MOVF   TABLAT,W                   ; get data
           MOVWF  POSTINC0                   ; store data and increment FSR0
           DECFSZ COUNTER                    ; done?
           BRA    READ_BLOCK                 ; repeat
MODIFY_WORD
           MOVLW  DATA_ADDR_HIGH             ; point to buffer
           MOVWF  FSR0H
           MOVLW  DATA_ADDR_LOW
           MOVWF  FSR0L
           MOVLW  NEW_DATA_LOW               ; update buffer word and increment FSR0
           MOVWF  POSTINC0
           MOVLW  NEW_DATA_HIGH              ; update buffer word
           MOVWF  INDF0
ERASE_BLOCK
           MOVLW  CODE_ADDR_UPPER            ; load TBLPTR with the base
           MOVWF  TBLPTRU                    ; address of the memory block
           MOVLW  CODE_ADDR_HIGH
           MOVWF  TBLPTRH
           MOVLW  CODE_ADDR_LOW              ; 6 LSB = 0
           MOVWF  TBLPTRL
           BCF    EECON1, CFGS               ; point to PROG/EEPROM memory
           BSF    EECON1, EEPGD              ; point to Flash program memory
           BSF    EECON1, WREN               ; enable write to memory
           BSF    EECON1, FREE               ; enable Row Erase operation
           BCF    INTCON, GIE                ; disable interrupts
           MOVLW  55h                        ; Required sequence
           MOVWF  EECON2                     ; write 55h
           MOVLW  0AAh
           MOVWF  EECON2                     ; write 0AAh
           BSF    EECON1, WR                 ; start erase (CPU stall)
           NOP
           BSF    INTCON, GIE                ; re-enable interrupts
WRITE_BUFFER_BACK
           MOVLW  8                          ; number of write buffer groups of 8 bytes
           MOVWF  COUNTER_HI
           MOVLW  BUFFER_ADDR_HIGH           ; point to buffer
           MOVWF  FSR0H
           MOVLW  BUFFER_ADDR_LOW
           MOVWF  FSR0L
PROGRAM_LOOP
           MOVLW  8                          ; number of bytes in holding register
           MOVWF  COUNTER
WRITE_WORD_TO_HREGS
           MOVF   POSTINC0,F                 ; get low byte of buffer data and increment FSR0
           MOVWF  TABLAT                     ; present data to table latch
           TBLWT+*                           ; short write
                                             ; to internal TBLWT holding register, increment
                                             ; TBLPTR
           DECFSZ COUNTER                    ; loop until buffers are full
           GOTO   WRITE_WORD_TO_HREGS
```

REGISTER 10-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

R/W-0	U-0	U-0	R/W-0	U-0	R/W-0	U-0	R/W-0
OSCFIF	—	—	EEIF	—	LVDIF	—	CCP2IF
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 **OSCFIF:** Oscillator Fail Interrupt Flag bit

`1` = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)
`0` = Device clock operating

bit 6-5 **Unimplemented:** Read as ‘`0`’

bit 4 **EEIF:** EEPROM or Flash Write Operation Interrupt Flag bit

`1` = The write operation is complete (must be cleared in software)
`0` = The write operation is not complete or has not been started

bit 3 **Unimplemented:** Read as ‘`0`’

bit 2 **LVDIF:** Low-Voltage Detect Interrupt Flag bit

`1` = The supply voltage has fallen below the specified LVD voltage (must be cleared in software)
`0` = The supply voltage is greater than the specified LVD voltage

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

bit 0 **CCP2IF:** CCP2 Interrupt Flag bit

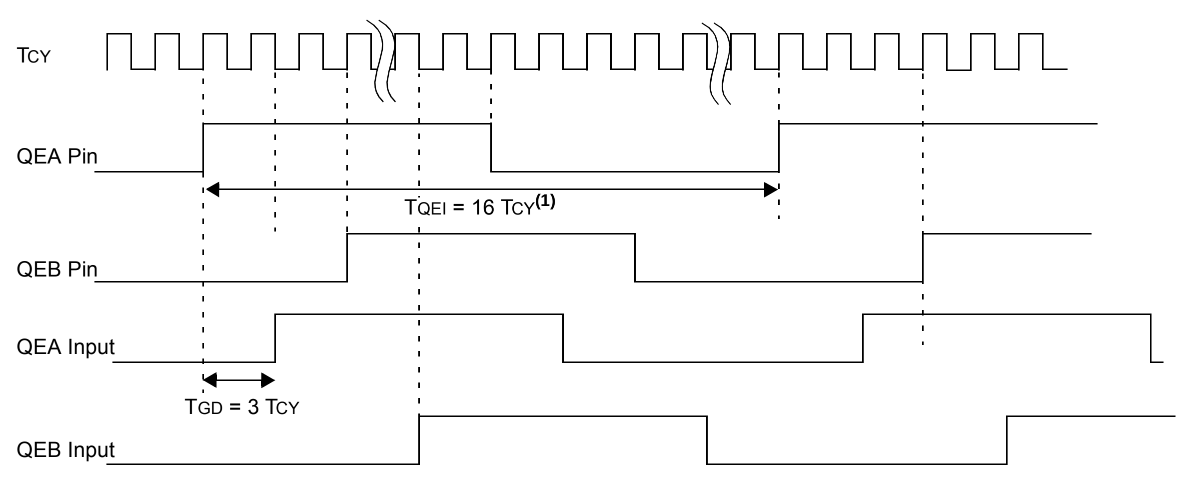
Capture mode:
`1` = A TMR1 register capture occurred (must be cleared in software)
`0` = No TMR1 register capture occurred

Compare mode:
`1` = A TMR1 register compare match occurred (must be cleared in software)
`0` = No TMR1 register compare match occurred

PWM mode:
Not used in this mode.

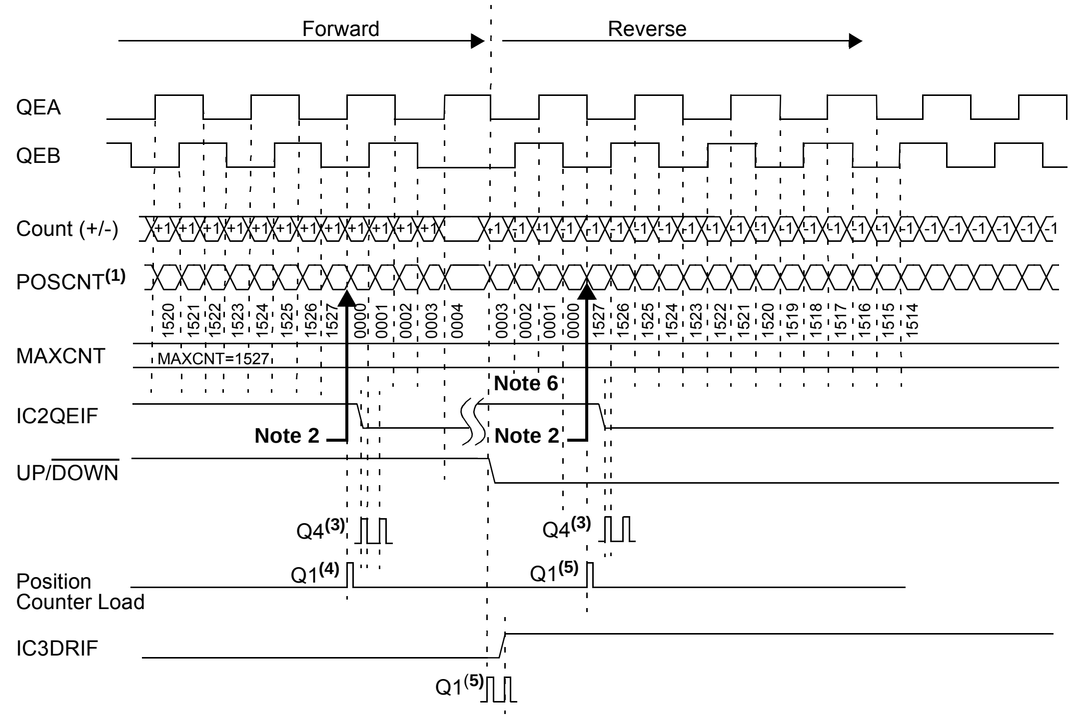
NOTES:

FIGURE 17-9: QEI INPUTS WHEN SAMPLED BY THE FILTER (DIVIDE RATIO = 1:1)



Note 1: The module design allows a quadrature frequency of up to FQEI = FCY/16.

FIGURE 17-10: QEI MODULE RESET TIMING ON PERIOD MATCH



Note 1: The POSCNT register is shown in QEI x4 Update mode (POSCNT increments on every rising and every falling edge of QEA and QEB input signals). Asynchronous external QEA and QEB inputs are synchronized to the TCY clock by the input sampling FF in the noise filter (see Figure 17-14).

2: When POSCNT = MAXCNT, POSCNT is reset to ‘0’ on the next QEA rising edge. POSCNT is set to MAXCNT when POSCNT = `0` (when decrementing), which occurs on the next QEA falling edge.

3: IC2QEIF is generated on the Q4 rising edge.

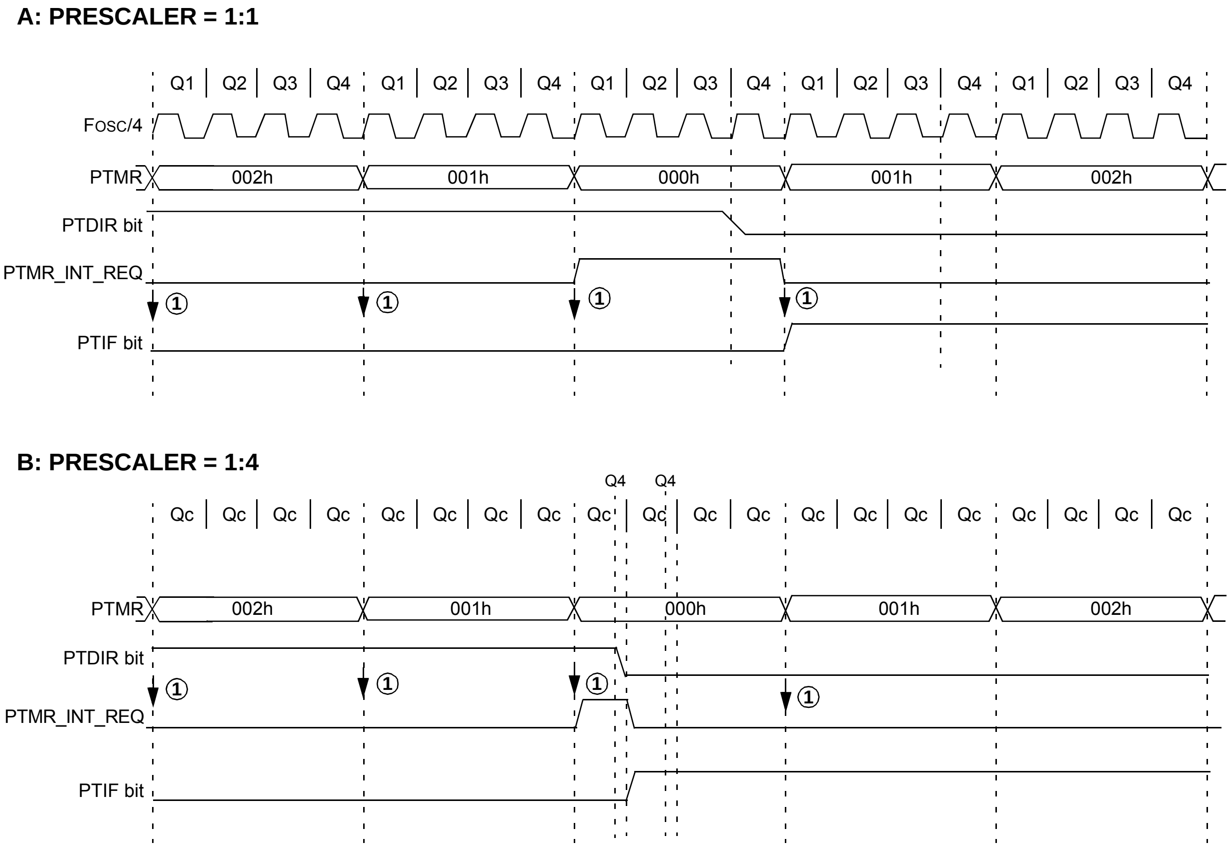
4: Position counter is loaded with ‘`0`’ (which is a rollover event in this case) on POSCNT = MAXCNT.

5: Position counter is loaded with MAXCNT value (1527h) on underflow.

6: IC2QEIF must be cleared in software.

NOTES:

FIGURE 18-7: PWM TIME BASE INTERRUPT, CONTINUOUS UP/DOWN COUNT MODE



Note 1: Interrupt flag bit, PTIF, is sampled here (every Q1).

REGISTER 19-2: SSPCON: SYNCHRONOUS SERIAL PORT CONTROL 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
WCOL	SSPOV(1)	SSPEN(2)	CKP	SSPM3(3)	SSPM2(3)	SSPM1(3)	SSPM0(3)
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 **WCOL:** Write Collision Detect bit

`1` = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)
`0` = No collision

bit 6 **SSPOV**: Receive Overflow Indicator bit(1)

In SPI mode:
`1` = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow. In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.
`0` = No overflow

In I^{2}C™ mode:
`1` = A byte is received while the SSPBUF register is still holding the previous byte. SSPOV is a “don’t care” in Transmit mode. SSPOV must be cleared in software in either mode.
`0` = No overflow

bit 5 **SSPEN**: Synchronous Serial Port Enable bit(2)

In SPI mode:
`1` = Enables serial port and configures SCK, SDO and SDI as serial port pins
`0` = Disables serial port and configures these pins as I/O port pins

In I^{2}C mode:
`1` = Enables the serial port and configures the SDA and SCL pins as serial port pins
`0` = Disables serial port and configures these pins as I/O port pins
In both modes, when enabled, these pins must be properly configured as input or output.

bit 4 **CKP**: Clock Polarity Select bit

In SPI mode:
`1` = Idle state for clock is a high level
`0` = Idle state for clock is a low level

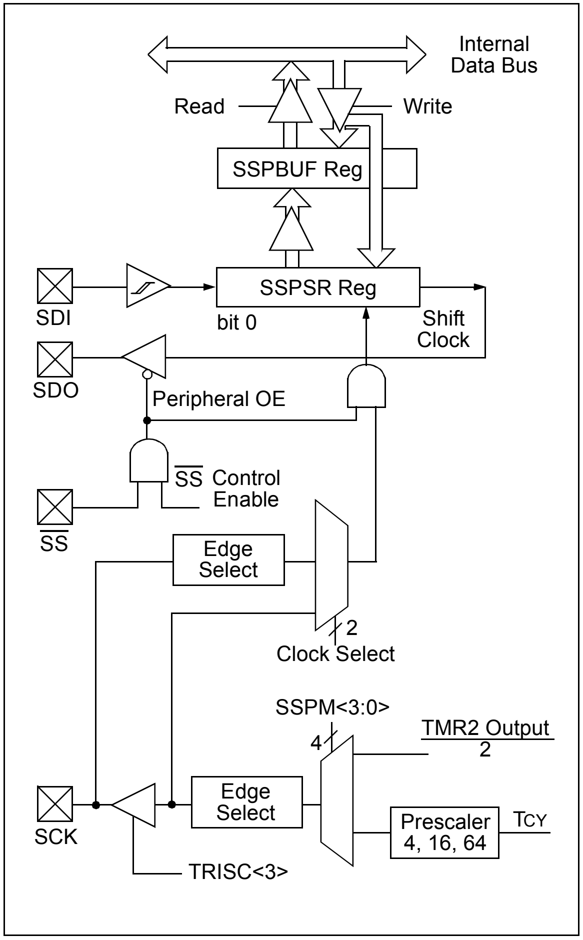
In I^{2}C mode:
SCK release control.
`1` = Enables clock
`0` = Holds clock low (clock stretch). (Used to ensure data setup time.)

Note 1: In Master mode, the overflow bit is not set since each new reception (and transmission) is initiated by writing to the SSPBUF register.

2: When enabled, these pins must be properly configured as inputs or outputs.

3: Bit combinations not specifically listed here are either reserved or implemented in I^{2}C™ mode only.

FIGURE 19-1: SSP BLOCK DIAGRAM (SPI MODE)



To enable the serial port, SSP Enable bit, SSPEN (SSPCON<5>), must be set. To reset or reconfigure SPI mode, clear bit SSPEN, reinitialize the SSPCON register and then set bit SSPEN. This configures the SDI, SDO, SCK and $\overline{\text{SS}}$ pins as serial port pins. For the pins to behave as the serial port function, they must have their data direction bits (in the TRISC register) appropriately programmed. That is:

- Serial Data Out (SDO) – RC7/RX/DT/SDO or RD1/SDO
- SDI must have TRISC<4> or TRISD<2> set
- SDO must have TRISC<7> or TRISD<1> cleared
- SCK (Master mode) must have TRISC<5> or TRISD<3> cleared
- SCK (Slave mode) must have TRISC<5> or TRISD<3> set
- $\overline{\text{SS}}$ must have TRISA<6> set

Note 1: When the SPI is in Slave mode, with the $\overline{\text{SS}}$ pin control enabled, (SSPCON<3:0> = `0100`), the SPI module will reset if the $\overline{\text{SS}}$ pin is set to V_{DD}.

2: If the SPI is used in Slave mode with CKE = `1`, then the $\overline{\text{SS}}$ pin control must be enabled.

3: When the SPI is in Slave mode with $\overline{\text{SS}}$ pin control enabled (SSPCON<3:0> = `0100`), the state of the $\overline{\text{SS}}$ pin can affect the state read back from the TRISC<6> bit. The peripheral OE signal from the SSP module into PORTC controls the state that is read back from the TRISC<6> bit (see **Section 11.3 "PORTC, TRISC and LATC Registers"** for information on PORTC). If Read-Modify-Write instructions, such as BSF, are performed on the TRISC register while the $\overline{\text{SS}}$ pin is high, this will cause the TRISC<6> bit to be set, thus disabling the SDO output.

20.2.3 AUTO-BAUD RATE DETECT

The Enhanced USART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 20-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Rate Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX signal, the RX signal is timing the BRG. In ABD mode, the internal Baud Rate Generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Detect must receive a byte with the value of 55h (ASCII “U”, which is also the LIN/J2602 bus Sync character) in order to calculate the proper bit rate. The measurement takes over both a low and a high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRG begins counting up, using the preselected clock source on the first rising edge of RX. After eight bits on the RX pin, or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH:SPBRG registers. Once the 5th edge is seen (should correspond to the Stop bit), the ABDEN bit is automatically cleared.

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. The BRG clock can be configured by the BRG16 and BRGH bits. The BRG16 bit must be set to use both SPBRG and SPBRGH as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes by checking for 00h in the SPBRGH register. Refer to Table 20-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in Idle. The RCIF interrupt is set once the fifth rising edge on RX is detected. The value in the RCREG needs to be read to clear the RCIF interrupt. RCREG content should be discarded.

Note 1: If the WUE bit is set with the ABDEN bit, Auto-Baud Rate Detection will occur on the byte *following* the Break character (see **Section 20.3.4 “Auto-Wake-up on Sync Break Character”**).

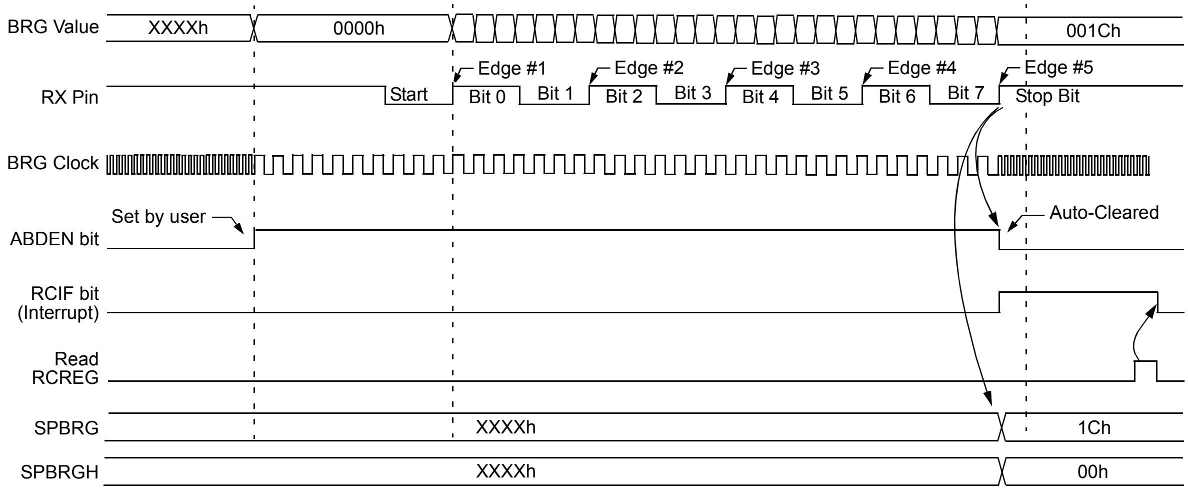
2: It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Rate Detection feature.

3: To maximize baud rate range, setting the BRG16 bit is recommended if the auto-baud feature is used.

TABLE 20-4: BRG COUNTER CLOCK RATES

BRG16	BRGH	BRG Counter Clock
0	0	FOSC/512
0	1	FOSC/256
1	0	FOSC/128
1	1	FOSC/32

FIGURE 20-1: AUTOMATIC BAUD RATE CALCULATION(1)



Note 1: The ABD sequence requires the EUSART module to be configured in Asynchronous mode and WUE = `0`.

FIGURE 20-2: EUSART TRANSMIT BLOCK DIAGRAM

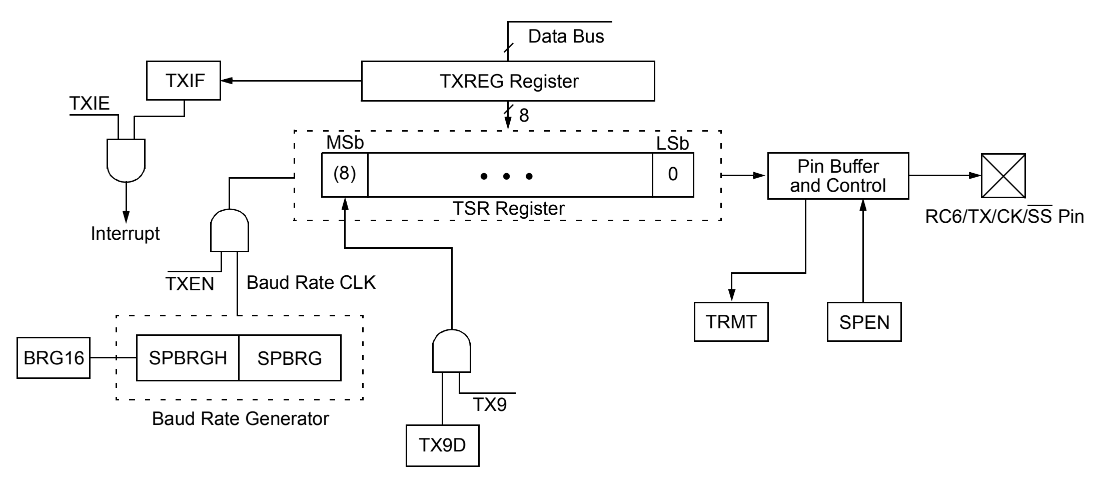


FIGURE 20-3: ASYNCHRONOUS TRANSMISSION

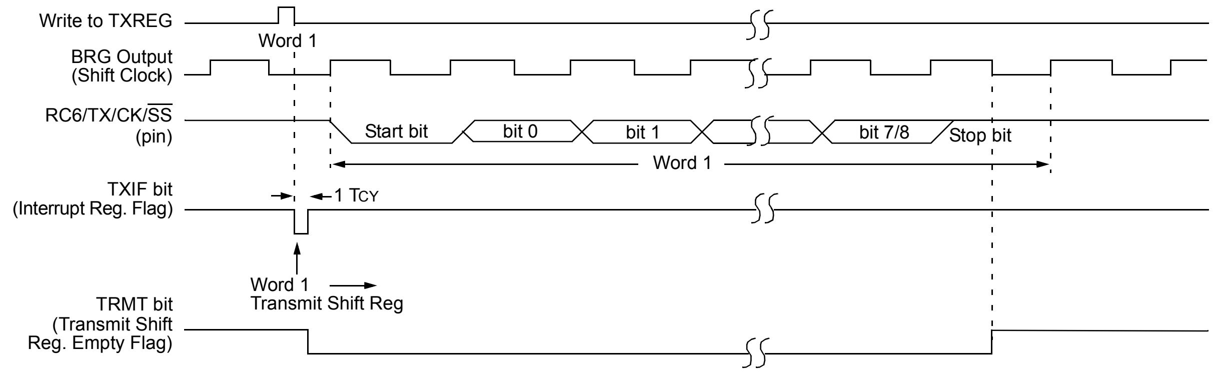
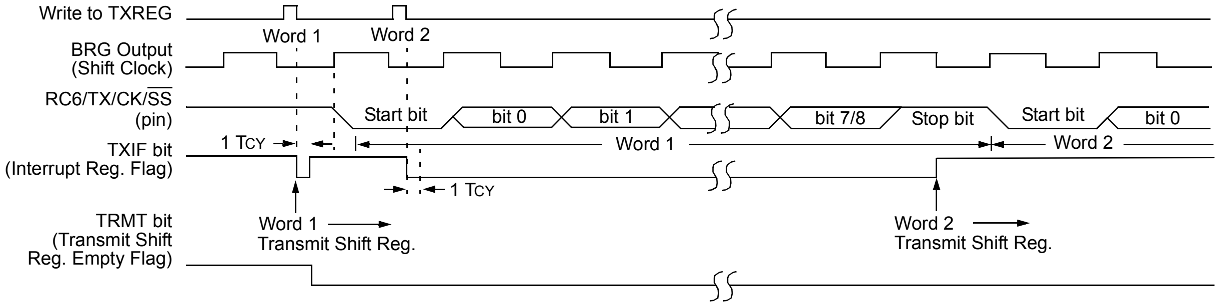


FIGURE 20-4: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

22.0 LOW-VOLTAGE DETECT (LVD)

PIC18F2331/2431/4331/4431 devices have a Low-Voltage Detect module (LVD), a programmable circuit that enables the user to specify a device voltage trip point. If the device experiences an excursion below the trip point, an interrupt flag is set. If the interrupt is enabled, the program execution will branch to the interrupt vector address and the software can then respond to the interrupt.

The Low-Voltage Detect Control register (Register 22-1) completely controls the operation of the LVD module. This allows the circuitry to be "turned off" by the user under software control, which minimizes the current consumption for the device.

The block diagram for the LVD module is shown in Figure 22-1.

The module is enabled by setting the LVDEN bit, but the circuitry requires some time to stabilize each time that it is enabled. The IRVST bit is a read-only bit used to indicate when the circuit is stable. The module can only generate an interrupt after the circuit is stable and the IRVST bit is set. The module monitors for drops in VDD below a predetermined set point.

REGISTER 22-1: LVDCON: LOW-VOLTAGE DETECT CONTROL REGISTER

U-0	U-0	R-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1
—	—	IRVST	LVDEN	LVDL3[1]	LVDL2[1]	LVDL1[1]	LVDL0[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 7-6 **Unimplemented:** Read as '0'

bit 5 **IRVST:** Internal Reference Voltage Stable Flag bit

1 = Indicates that the Low-Voltage Detect logic will generate the interrupt flag at the specified voltage range
0 = Indicates that the Low-Voltage Detect logic will not generate the interrupt flag at the specified voltage range and the LVD interrupt should not be enabled

bit 4 **LVDEN:** Low-Voltage Detect Power Enable bit

1 = Enables LVD, powers up LVD circuit
0 = Disables LVD, powers down LVD circuit

bit 3-0 **LVDL<3:0>:** Low-Voltage Detection Limit bits[1]

1111 = External analog input is used (input comes from the LVDIN pin)
1110 = Maximum setting
.
.
.
0010 = Minimum setting
0001 = Reserved
0000 = Reserved

Note 1: LVDL<3:0> bit modes, which result in a trip point below the valid operating voltage of the device, are not tested.

NOTES:

POP — Pop Top of Return Stack

Syntax:	[*label*] POP
Operands:	None
Operation:	(TOS) → bit bucket
Status Affected:	None

Encoding:

`0000`	`0000`	`0000`	`0110`

Description:	The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack. This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	POP TOS value	No operation

Example:

```
POP
GOTO    NEW
```

Before Instruction

TOS	=	0x0031A2
Stack (1 level down)	=	0x014332

After Instruction

TOS	=	0x014332
PC	=	NEW

PUSH — Push Top of Return Stack

Syntax:	[*label*] PUSH
Operands:	None
Operation:	(PC + 2) → TOS
Status Affected:	None

Encoding:

`0000`	`0000`	`0000`	`0101`

Description:	The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows to implement a software stack by modifying TOS, and then push it onto the return stack.
Words:	1
Cycles:	1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	PUSH PC + 2 onto return stack	No operation	No operation

Example:

```
PUSH
```

Before Instruction

TOS	=	0x00345A
PC	=	0x000124

After Instruction

PC	=	0x000126
TOS	=	0x000126
Stack (1 level down)	=	0x00345A

RRNCF Rotate Right f (No Carry)

Syntax:	[*label*] RRNCF f [,d [,a]]
Operands:	$0 \le f \le 255$ $d \in [0,1]$ $a \in [0,1]$
Operation:	(f<n>) → dest<n – 1>, (f<0>) → dest<7>
Status Affected:	N, Z
Encoding:	`0100` `00da` `ffff` `ffff`
Description:	The contents of register, 'f', are rotated one bit to the right. If 'd' is '`0`', the result is placed in W. If 'd' is '`1`', the result is placed back in register, 'f'. If 'a' is '`0`', the Access Bank will be selected, overriding the BSR value. If 'a' is '`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 to destination

Example 1: `RRNCF REG, 1, 0`

Before Instruction
REG = `1101 0111`

After Instruction
REG = `1110 1011`

Example 2: `RRNCF REG, W`

Before Instruction
W = ?
REG = `1101 0111`

After Instruction
W = `1110 1011`
REG = `1101 0111`

SETF Set f

Syntax:	[*label*] SETF f [,a]
Operands:	$0 \le f \le 255$ $a \in [0,1]$
Operation:	FFh → f
Status Affected:	None
Encoding:	`0110` `100a` `ffff` `ffff`
Description:	The contents of the specified register are set to FFh. If 'a' is '`0`', the Access Bank will be selected, overriding the BSR value. If 'a' is '`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: `SETF REG`

Before Instruction
REG = 0x5A

After Instruction
REG = 0xFF

26.2 DC Characteristics: Power-Down and Supply Current PIC18F2331/2431/4331/4431 (Industrial, Extended) PIC18LF2331/2431/4331/4431 (Industrial) (Continued)

PIC18LF2331/2431/4331/4431 (Industrial)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial
PIC18F2331/2431/4331/4431 (Industrial, Extended)	**Standard Operating Conditions (unless otherwise stated)** Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended

Param No.	Device	Typ	Max	Units	Conditions		
	Supply Current (IDD)$^{(2,3)}$						
	PIC18LF2X31/4X31	4.7	8	μA	-40°C	VDD = 2.0V	FOSC = 31 kHz (**RC_IDLE** mode, Internal oscillator source)
		5.0	8	μA	+25°C		
		5.8	11	μA	+85°C		
	PIC18LF2X31/4X31	7.0	11	μA	-40°C	VDD = 3.0V	
		7.8	11	μA	+25°C		
		8.7	15	μA	+85°C		
	All devices	12	16	μA	-40°C	VDD = 5.0V	
		14	16	μA	+25°C		
		14	22	μA	+85°C		
		200	850	μA	+125°C		
	PIC18LF2X31/4X31	75	150	μA	-40°C	VDD = 2.0V	FOSC = 1 MHz (**RC_IDLE** mode, Internal oscillator source)
		85	150	μA	+25°C		
		95	150	μA	+85°C		
	PIC18LF2X31/4X31	110	180	μA	-40°C	VDD = 3.0V	
		125	180	μA	+25°C		
		135	180	μA	+85°C		
	All devices	180	300	μA	-40°C	VDD = 5.0V	
		195	300	μA	+25°C		
		200	300	μA	+85°C		
		300	750	μA	+125°C		
	PIC18LF2X31/4X31	175	275	μA	-40°C	VDD = 2.0V	FOSC = 4 MHz (**RC_IDLE** mode, Internal oscillator source)
		185	275	μA	+25°C		
		195	275	μA	+85°C		
	PIC18LF2X31/4X31	265	375	μA	-40°C	VDD = 3.0V	
		280	375	μA	+25°C		
		300	375	μA	+85°C		
	All devices	475	800	μA	-40°C	VDD = 5.0V	
		500	800	μA	+25°C		
		505	800	μA	+85°C		
		0.7	1.6	mA	+125°C		

Legend: Shading of rows is to assist in readability of the table.

Note 1: The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode, with all I/O pins in high-impedance state and tied to VDD or VSS, and all features that add delta current disabled (such as WDT, Timer1 Oscillator, BOR, etc.).

2: The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption.
The test conditions for all IDD measurements in active operation mode are:
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to VDD;
$\overline{\text{MCLR}}$ = VDD; WDT enabled/disabled as specified.

3: For RC oscillator configurations, current through REXT is not included. The current through the resistor can be estimated by the formula: Ir = VDD/2REXT (mA) with REXT in kΩ.

4: Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.

26.3 DC Characteristics: PIC18F2331/2431/4331/4431 (Industrial, Extended) PIC18LF2331/2431/4331/4431 (Industrial) (Continued)

DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended			
Param No.	**Symbol**	**Characteristic**	**Min**	**Max**	**Units**	**Conditions**
	V_{OL}	**Output Low Voltage**				
D080		I/O Ports	—	0.6	V	I_{OL} = 8.5 mA, V_{DD} = 4.5V, -40°C to +85°C
D083		OSC2/CLKO (RC, RCIO, EC, ECIO modes)	—	0.6	V	I_{OL} = 1.6 mA, V_{DD} = 4.5V, -40°C to +85°C
	V_{OH}	**Output High Voltage**(3)				
D090		I/O Ports	V_{DD} – 0.7	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V, -40°C to +85°C
D092		OSC2/CLKO (RC, RCIO, EC, ECIO modes)	V_{DD} – 0.7	—	V	I_{OH} = -1.3 mA, V_{DD} = 4.5V, -40°C to +85°C
		Capacitive Loading Specs on Output Pins				
D100	C_{OSC2}	OSC2 Pin	—	15	pF	In XT, HS and LP modes when external clock is used to drive OSC1
D101	C_{IO}	All I/O Pins and OSC2 (in RC mode)	—	50	pF	To meet the AC Timing Specifications
D102	C_B	SCL, SDA	—	400	pF	I^{2}C™ Specification

Note 1: In RC oscillator configuration, the OSC1/CLKI pin is a Schmitt Trigger input. It is not recommended that the PIC® device be driven with an external clock while in RC mode.

2: The leakage current on the $\overline{\text{MCLR}}$ pin is strongly dependent on the applied voltage level. The specified levels represent normal operating conditions. Higher leakage current may be measured at different input voltages.

3: Negative current is defined as current sourced by the pin.

INDEX