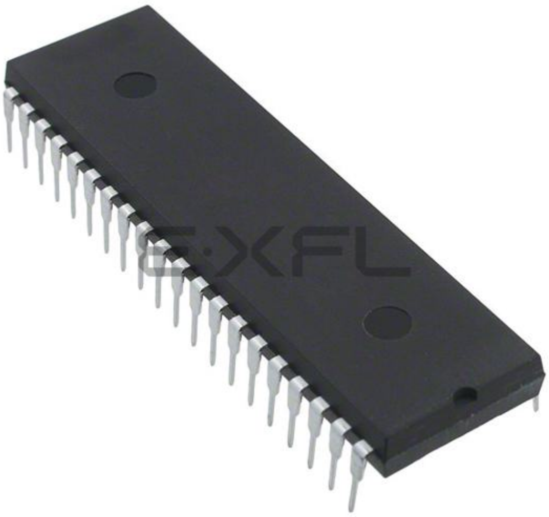
Microchip Technology - PIC18LF47K42-E/P Datasheet



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	64MHz
Connectivity	I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT
Number of I/O	36
Program Memory Size	128KB (64K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 35x12b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°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/pic18lf47k42-e-p

PIC18(L)F2X/4X/5XK42 FAMILY TYPES

Device	Data Sheet Index	Program Flash Memory (KB)	Data EEPROM (B)	Data SRAM (bytes)	I/O Pins	12-bit ADC^2 (ch)	5-bit DAC	Comparator	8-bit/ (with HLT) /16-bit Timer	Window Watchdog Timer (WWDT)	Signal Measurement Timer (SMT)	CCP/10-bit PWM	CWG	NCO	CLC	Zero-Cross Detect	Direct Memory Access (DMA) (ch)	Memory Access Partition	Vectored Interrupts	UART	I^{2}C/SPI	Peripheral Pin Select	Peripheral Module Disable	Debug[1]
PIC18(L)F24K42	A	16	256	1024	25	24	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F25K42	A	32	256	2048	25	24	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F26K42	B	64	1024	4096	25	24	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F27K42	B	128	1024	8192	25	24	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F45K42	B	32	256	2048	36	35	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F46K42	B	64	1024	4096	36	35	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F47K42	B	128	1024	8192	36	35	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F55K42	B	32	256	2048	44	43	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F56K42	B	64	1024	4096	44	43	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I
PIC18(L)F57K42	B	128	1024	8192	44	43	1	2	3/4	Y	Y	4/4	3	1	4	Y	2	Y	Y	2	2/1	Y	Y	I

Note 1: I – Debugging integrated on chip.

Data Sheet Index:

Unshaded devices are not described in this document.

A:	DS40001869	PIC18(L)F24/25K42 Data Sheet, 28-Pin
B:	DS40001919	PIC18(L)F26/27/45/46/47/55/56/57K42 Data Sheet, 28/40/44/48-Pin

> **Note:** For other small form-factor package availability and marking information, visit **http://www.microchip.com/packaging** or contact your local sales office.

EXAMPLE 9-3: SETTING UP VECTORED INTERRUPTS USING MPASM

```
ISR_TMR0:  CODE       0x8C0                ; ISR code at 0x08C0 in PFM
        BANKSEL        PIR0                 ; Select bank for PIR0
        BCF            PIR3, TMR0IF         ; Clear TMR0IF
        BTG            LATC, 0, ACCESS      ; Code to execute in ISR
        RETFIE         1                    ; Return from ISR

InterruptInit:
        BANKSEL        INTCON0              ; Select bank for INTCON0
        BSF            INTCON0, GIEH        ; Enable high priority interrupts
        BSF            INTCON0, GIEL        ; Enable low priority interrupts
        BSF            INTCON0, IPEN_INTCON0 ; Enable interrupt priority

        BANKSEL        PIE0                 ; Select bank for PIE0
        BSF            PIE3, TMR0IE         ; Enable TMR0 interrupt
        BSF            PIE4, TMR1IE         ; Enable TMR1 interrupt

        BCF            IPR3, TMR0IP         ; Make TMR0 interrupt low priority
        RETURN         1

VectorTableInit:
        ; Set IVTBASE (optional - default is 0x000008)
        MOVLW          0x00                 ; This is optional
        MOVWF          IVTBASEU, ACCESS     ; If not included, then the
        MOVLW          0x40                 ; hardware default value of
        MOVWF          IVTBASEH, ACCESS     ; 0x0008 will be taken.
        MOVLW          0x08
        MOVWF          IVTBASEL, ACCESS

        ; TMR0 vector at IVTBASE + 2*(TMR0 vector number i.e. 31) = 0x4046
        MOVLW          0x00                 ; Load TBLPTR with the
        MOVWF          TBLPTRU, ACCESS      ; PFM memory location to be
        MOVLW          0x40                 ; written to.
        MOVWF          TBLPTRH, ACCESS
        MOVLW          0x46
        MOVWF          TBLPTRL, ACCESS

        ; Write the contents of TMR0 vector location
        ; ISR_TMR0_ADDRESS >> 2 = 0x08C0 >> 2 = 0x0230
        MOVLW          0x30                 ; Low byte first
        MOVWF          TABLAT, ACCESS
        TBLWT*+                             ; Write to temp table latch

        MOVLW          0x02                 ; High byte next
        MOVWF          TABLAT, ACCESS
        TBLWT*+                             ; Write to temp table latch

        ; Write to PFM now using NVMCON
        BANKSEL        NVMCON1              ; Select bank for NVMCON1
        MOVLW          0x84                 ; Setting to write to PFM
        MOVWF          NVMCON1

        MOVLW          0x55                 ; Required unlock sequence
        MOVWF          NVMCON2
        MOVLW          0xAA
        MOVWF          NVMCON2
        BSF            NVMCON1, WR          ; Start writing to PFM

        BTFSC          NVMCON1, WR          ; Wait for write to complete
        GOTO           $-2

        RETURN         1
```

14.0 CYCLIC REDUNDANCY CHECK (CRC) MODULE WITH MEMORY SCANNER

The Cyclic Redundancy Check (CRC) module provides a software-configurable hardware-implemented CRC checksum generator. This module includes the following features:

- Any standard CRC up to 16 bits can be used
- Configurable Polynomial
- Any seed value up to 16 bits can be used
- Standard and reversed bit order available
- Augmented zeros can be added automatically or by the user
- Memory scanner for fast CRC calculations on program/Data EEPROM memory user data
- Software loadable data registers for communication CRC's

14.1 CRC Module Overview

The CRC module provides a means for calculating a check value of program/Data EEPROM memory. The CRC module is coupled with a memory scanner for faster CRC calculations. The memory scanner can automatically provide data to the CRC module. The CRC module can also be operated by directly writing data to SFRs, without using a scanner.

14.5 CRC Check Value

The CRC check value will be located in the CRCACC registers after the CRC calculation has finished. The check value will depend on two mode settings of the CRCCON0 register: ACCM and SHIFTM. When the ACCM bit is set, the CRC module augments the data with a number of zeros equal to the length of the polynomial to align the final check value. When the ACCM bit is not set, the CRC will stop at the end of the data. A number of zeros equal to the length of the polynomial can then be entered into CRCDAT to find the same check value as augmented mode. Alternatively, the expected check value can be entered at this point to make the final result equal '0'.

When the CRC check value is computed with the SHIFTM bit set, selecting LSb first, and the ACCM bit is also set, then the final value in the CRCACC registers will be reversed such that the LSb will be in the MSb position and vice versa. This is the expected check value in bit reversed form. If you are creating a check value to be appended to a data stream, then a bit reversal must be performed on the final value to achieve the correct checksum. You can use the CRC to do this reversal by the following method:

- Save the CRCACC value in user RAM space
- Clear the CRCACC registers
- Clear the CRCXOR registers
- Write the saved CRCACC value to the CRCDAT input.

The properly oriented check value will be in the CRCACC registers as the result.

14.6 CRC Interrupt

The CRC will generate an interrupt when the BUSY bit transitions from 1 to 0. The CRCIF Interrupt Flag is set every time the BUSY bit transitions, regardless of whether or not the CRC interrupt is enabled. The CRCIF bit can only be cleared in software.

14.7 Configuring the CRC

The following steps illustrate how to properly configure the CRC.

1. Determine if the automatic program memory scan will be used with the scanner or manual calculation through the SFR interface and perform the actions specified in **Section 14.4 "CRC Data Sources"**, depending on which decision was made.
2. If desired, seed a starting CRC value into the CRCACCH/L registers.
3. Program the CRCXORH/L registers with the desired generator polynomial.
4. Program the DLEN<3:0> bits of the CRCCON1 register with the length of the data word - 1 (refer to Example 14-1). This determines how many times the shifter will shift into the accumulator for each data word.
5. Program the PLEN<3:0> bits of the CRCCON1 register with the length of the polynomial -2 (refer to Example 14-1).
6. Determine whether shifting in trailing zeros is desired and set the ACCM bit of the CRCCON0 register appropriately.
7. Likewise, determine whether the MSb or LSb should be shifted first and write the SHIFTM bit of the CRCCON0 register appropriately.
8. Write the GO bit of the CRCCON0 register to begin the shifting process.

9a. If manual SFR entry is used, monitor the FULL bit of the CRCCON0 register. When FULL = 0, another word of data can be written to the CRCDATH/L registers, keeping in mind that CRCDATH should be written first if the data has more than eight bits, as the shifter will begin upon the CRCDATL register being written.

9b. If the scanner is used, the scanner will automatically load words into the CRCDATH/L registers as needed, as long as the GO bit is set.

10a. If manual entry is used, monitor the CRCIF (and BUSY bit to determine when the completed CRC calculation can be read from CRCACCH/L registers.

10b. If using the memory scanner, monitor the SCANIF (or the GO bit) for the scanner to finish pushing information into the CRCDAT registers. After the scanner is completed, monitor the BUSY bit to determine that the CRC has been completed and the check value can be read from the CRCACC registers. If both the interrupt flags are set and the BUSY and GO bits are cleared, the completed CRC calculation can be read from the CRCACCH/L registers.

TABLE 17-2: PPS OUTPUT REGISTER DETAILS

RxyPPS<5:0>	Pin Rxy Output Source	Device Configuration													
		PIC18(L)F26/27K42			PIC18(L)F45/46/47K42					PIC18(L)F55/56/57K42					
0b01 0010 - 0b01 0001	Reserved														
0b01 0000	PWM8	A	—	C	A	—	—	D	—	A	—	—	D	—	—
0b00 1111	PWM7	A	—	C	A	—	C	—	—	—	—	C	—	—	F
0b00 1110	PWM6	A	—	C	A	—	—	D	—	A	—	—	D	—	—
0b00 1101	PWM5	A	—	C	A	—	C	—	—	A	—	—	—	—	F
0b00 1100	CCP4	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 1011	CCP3	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 1010	CCP2	—	B	C	—	B	C	—	—	—	—	C	—	—	F
0b00 1001	CCP1	—	B	C	—	B	C	—	—	—	—	C	—	—	F
0b00 1000	CWG1D	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 0111	CWG1C	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 0110	CWG1B	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 0101	CWG1A	—	B	C	—	B	C	—	—	—	B	C	—	—	—
0b00 0100	CLC4OUT	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 0011	CLC3OUT	—	B	C	—	B	—	D	—	—	B	—	D	—	—
0b00 0010	CLC2OUT	A	—	C	A	—	C	—	—	A	—	—	—	—	F
0b00 0001	CLC1OUT	A	—	C	A	—	C	—	—	A	—	—	—	—	F
0b00 0000	LATxy	A	B	C	A	B	C	D	E	A	B	C	D	E	F

18.0 INTERRUPT-ON-CHANGE

PORTA, PORTB, PORTC and pin RE3 of PORTE can be configured to operate as Interrupt-on-Change (IOC) pins on PIC18(L)F26/27/45/46/47/55/56/57K42 family devices. An interrupt can be generated by detecting a signal that has either a rising edge or a falling edge. Any individual port pin, or combination of port pins, can be configured to generate an interrupt. The interrupt-on-change module has the following features:

- Interrupt-on-Change enable (Master Switch)
- Individual pin configuration
- Rising and falling edge detection
- Individual pin interrupt flags

Figure 18-1 is a block diagram of the IOC module.

18.1 Enabling the Module

To allow individual port pins to generate an interrupt, the IOCIE bit of the PIEx register must be set. If the IOCIE bit is disabled, the edge detection on the pin will still occur, but an interrupt will not be generated.

18.2 Individual Pin Configuration

For each port pin, a rising edge detector and a falling edge detector are present. To enable a pin to detect a rising edge, the associated bit of the IOCxP register is set. To enable a pin to detect a falling edge, the associated bit of the IOCxN register is set.

A pin can be configured to detect rising and falling edges simultaneously by setting both associated bits of the IOCxP and IOCxN registers, respectively.

18.3 Interrupt Flags

The IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits located in the IOCAF, IOCBF, IOCCF and IOCEF registers respectively, are status flags that correspond to the interrupt-on-change pins of the associated port. If an expected edge is detected on an appropriately enabled pin, then the status flag for that pin will be set, and an interrupt will be generated if the IOCIE bit is set. The IOCIF bit of the PIR0 register reflects the status of all IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits.

18.4 Clearing Interrupt Flags

The individual status flags, (IOCAFx, IOCBFx, IOCCFx and IOCEF3 bits), can be cleared by resetting them to zero. If another edge is detected during this clearing operation, the associated status flag will be set at the end of the sequence, regardless of the value actually being written.

In order to ensure that no detected edge is lost while clearing flags, only AND operations masking out known changed bits should be performed. The following sequence is an example of what should be performed.

EXAMPLE 18-1: CLEARING INTERRUPT FLAGS (PORTA EXAMPLE)

```
MOVLW   0xff
XORWF   IOCAF, W
ANDWF   IOCAF, F
```

18.5 Operation in Sleep

The interrupt-on-change interrupt sequence will wake the device from Sleep mode, if the IOCIE bit is set.

If an edge is detected while in Sleep mode, the IOCxF register will be updated prior to the first instruction executed out of Sleep.

REGISTER 21-2: TxGCON: TIMERx GATE CONTROL REGISTER

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R-x	U-0	U-0
GE	GPOL	GTM	GSPM	GGO/$\overline{\text{DONE}}$	GVAL	—	—
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 **GE:** Timerx Gate Enable bit

If TMRxON = 1:
1 = Timerx counting is controlled by the Timerx gate function
0 = Timerx is always counting
If TMRxON = 0:
This bit is ignored

bit 6 **GPOL:** Timerx Gate Polarity bit

1 = Timerx gate is active-high (Timerx counts when gate is high)
0 = Timerx gate is active-low (Timerx counts when gate is low)

bit 5 **GTM:** Timerx Gate Toggle Mode bit

1 = Timerx Gate Toggle mode is enabled
0 = Timerx Gate Toggle mode is disabled and Toggle flip-flop is cleared
Timerx Gate Flip Flop Toggles on every rising edge

bit 4 **GSPM:** Timerx Gate Single Pulse Mode bit

1 = Timerx Gate Single Pulse mode is enabled and is controlling Timerx gate)
0 = Timerx Gate Single Pulse mode is disabled

bit 3 **GGO/$\overline{\text{DONE}}$:** Timerx Gate Single Pulse Acquisition Status bit

1 = Timerx Gate Single Pulse Acquisition is ready, waiting for an edge
0 = Timerx Gate Single Pulse Acquisition has completed or has not been started.
This bit is automatically cleared when TxGSPM is cleared.

bit 2 **GVAL:** Timerx Gate Current State bit

Indicates the current state of the Timerx gate that could be provided to TMRxH:TMRxL
Unaffected by Timerx Gate Enable (TMRxGE)

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

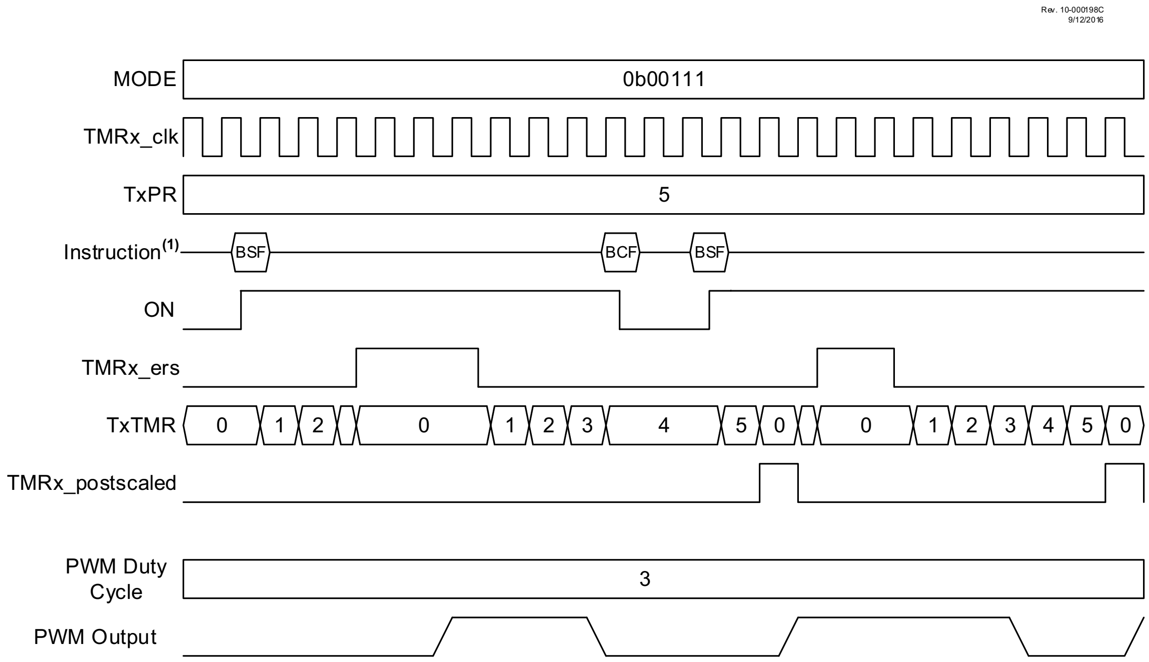
22.5.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the level triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMR2_ers, as shown in Figure 22-7. Selecting MODE<4:0> = `00110` will cause the timer to reset on a low level external signal. Selecting MODE<4:0> = `00111` will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMR2_ers = `1`. ON is controlled by BSF and BCF instructions. When ON=`0` the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the T2PR value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the T2PR match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.

FIGURE 22-7: LEVEL TRIGGERED HARDWARE LIMIT MODE TIMING DIAGRAM (MODE = `00111`)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

22.5.8 LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODES

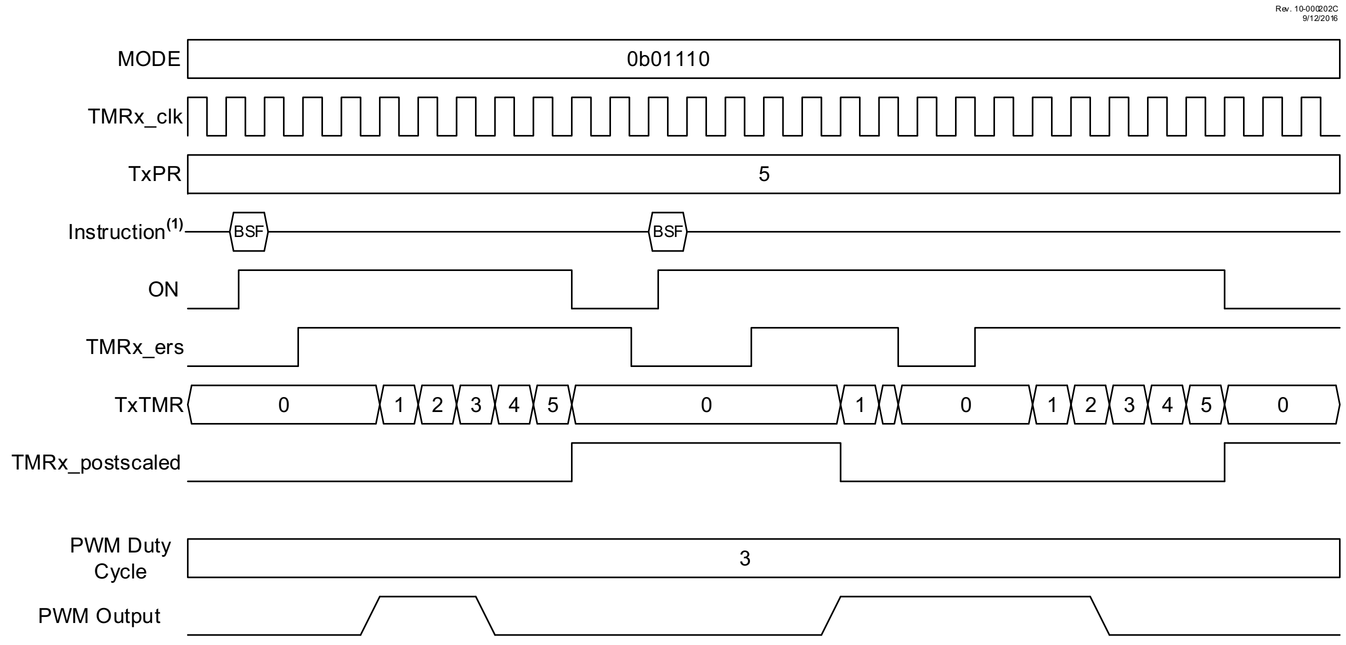
In Level Triggered One-Shot mode, the timer count is reset on the external signal level and starts counting on the rising/falling edge of the transition from reset level to the active level while the ON bit is set. Reset levels are selected as follows:

- Low reset level (MODE<4:0> = `01110`)
- High reset level (MODE<4:0> = `01111`)

When the timer count matches the T2PR period count, the timer is reset and the ON bit is cleared. When the ON bit is cleared by either a T2PR match or by software control a new external signal edge is required after the ON bit is set to start the counter.

When Level Triggered Reset One-Shot mode is used in conjunction with the CCP PWM operation, the PWM drive goes active with the external signal edge that starts the timer. The PWM drive goes inactive when the timer count equals the CCPRx pulse-width count. The PWM drive does not go active when the timer count clears at the T2PR period count match.

FIGURE 22-11: LOW LEVEL RESET, EDGE-TRIGGERED HARDWARE LIMIT ONE-SHOT MODE TIMING DIAGRAM (MODE = `01110`)



Note 1: BSF and BCF represent Bit-Set File and Bit-Clear File instructions executed by the CPU to set or clear the ON bit of TxCON. CPU execution is asynchronous to the timer clock input.

REGISTER 28-8: NCO1INCU: NCO1 INCREMENT REGISTER – UPPER BYTE[(1)]

U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	INC<19:16>			
bit 7							bit 0

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-4 **Unimplemented:** Read as ‘0’

bit 3-0 **INC<19:16>:** NCO1 Increment, Upper Byte

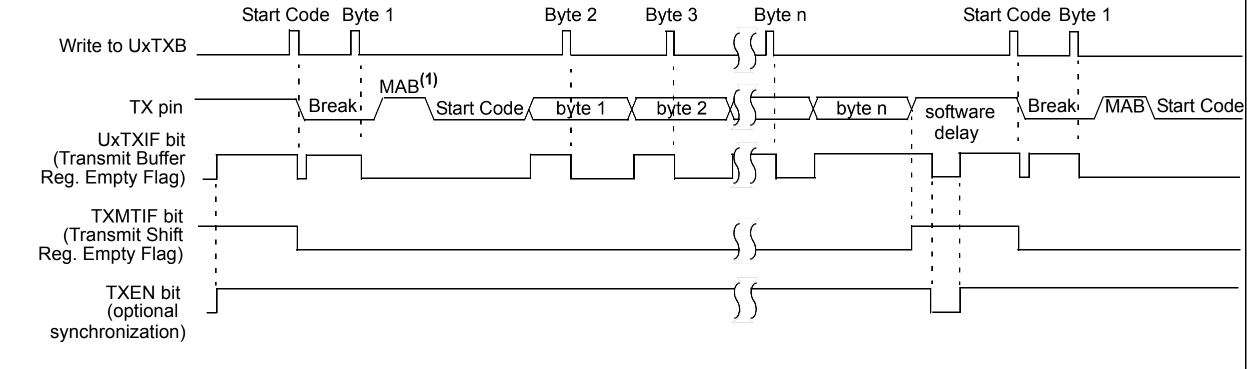
Note 1: The logical increment spans NCO1INCU:NCO1INCH:NCO1INCL.

TABLE 28-1: SUMMARY OF REGISTERS ASSOCIATED WITH NCO

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
NCO1CON	N1EN	—	N1OUT	N1POL	—	—	—	N1PFM	453
NCO1CLK	N1PWS<2:0>			—	—	N1CKS<2:0>			454
NCO1ACCL	NCO1ACC<7:0>								455
NCO1ACCH	NCO1ACC<15:8>								455
NCO1ACCU	—	—	—	—	NCO1ACC<19:16>				456
NCO1INCL	NCO1INC<7:0>								456
NCO1INCH	NCO1INC<15:8>								456
NCO1INCU	—	—	—	—	NCO1INC<19:16>				457

Legend: — = unimplemented read as ‘0’. Shaded cells are not used for NCO module.

FIGURE 31-6: DMX TRANSMIT SEQUENCE



Note 1: The MAB period is fixed at 3-bits period.

31.5 LIN Modes (UART1 only)

LIN is a protocol used primarily in automotive applications. The LIN network consists of two kinds of software processes: a Master process and a Slave process. Each network has only one Master process and one or more Slave processes.

From a physical layer point of view, the UART on one processor may be driven by both a Master and a Slave process, as long as only one Master process exists on the network.

A LIN transaction consists of a Master process followed by a Slave process. The Slave process may involve more than one Slave where one is transmitting and the other(s) are receiving. The transaction begins by the following Master process transmission sequence:

1. Break
2. Delimiter bit
3. Sync Field
4. PID byte

The PID determines which Slave processes are expected to respond to the Master. When the PID byte is complete, the TX output remains in the Idle state. One or more of the Slave processes may respond to the Master process. If no one responds within the inter-byte period, the Master is free to start another transmission. The inter-byte period is timed by software using a means other than the UART.

The Slave process follows the Master process. When the Slave software recognizes the PID then that Slave process responds by either transmitting the required response or by receiving the transmitted data. Only Slave processes send data. Therefore, Slave processes receiving data are receiving that of another Slave process.

When a Slave sends data, the Slave UART automatically calculates the checksum for the transmitted bytes as they are sent and appends the inverted checksum byte to the slave response.

When a Slave receives data, the checksum is accumulated on each byte as it is received using the same algorithm as the sending process. The last byte, which is the inverted checksum value calculated by the sending process, is added to the locally calculated checksum by the UART. The check passes when the result is all ‘1’s, otherwise the check fails and the CERIF bit is set.

Two methods for computing the checksum are available: legacy and enhanced. The legacy checksum includes only the data bytes. The enhanced checksum includes the PID and the data. The C0EN control bit in the UxCON2 register determines the checksum method. Setting C0EN to ‘1’ selects the enhanced method. Software must select the appropriate method before the Start bit of the checksum byte is received.

31.5.1 LIN MASTER/SLAVE MODE

The LIN Master mode includes capabilities to generate Slave processes. The Master process stops at the PID transmission. Any data that is transmitted in Master/Slave mode is done as a Slave process. LIN Master/Slave mode is configured by the following settings:

- MODE<3:0> = 1100
- TXEN = 1
- RXEN = 1
- UxBRGH:L = Value to achieve desired baud rate
- TXPOL = 0 (for high Idle state)
- STP = desired Stop bits selection
- C0EN = desired checksum mode
- RxyPPS = TX pin selection code
- TX pin TRIS control = 0
- ON = 1

Note: The TXEN bit must be set before the Master process is received and remain set while in LIN mode whether or not the slave process is a transmitter.

31.9 Stop Bits

The number of Stop bits is user selectable with the STP bits in the UxCON2 register.The STP bits affect all modes of operation.

Stop bits selections include:

- 1 transmit with receive verify on first
- 1.5 transmit with receive verify on first
- 2 transmit with receive verify on both
- 2 transmit with receive verify on first only

In all modes, except DALI, the transmitter is idle for the number of Stop bit periods between each consecutively transmitted word. In DALI, the Stop bits are generated after the last bit in the transmitted data stream.

The input is checked for the idle level in the middle of the first Stop bit, when receive verify on first is selected, as well as in the middle of the second Stop bit, when verify on both is selected. If any Stop bit verification indicates a non-idle level, the framing error FERIF bit is set for the received word.

31.9.1 DELAYED UXRXIF

When operating in Half-Duplex mode, where the microcontroller needs to reverse the transceiver direction after a reception, it may be more convenient to hold off the UxRXIF interrupt until the end of the Stop bits to avoid line contention. The user selects when the UxRXIF interrupt occurs with the STPMD bit in the UxFIFO register. When STPMD is '1', the UxRXIF occurs at the end of the last Stop bit. When STPMD is '0', UxRXIF occurs when the received byte is stored in the receive FIFO. When STP<1:0> = 10, the store operation is performed in the middle of the second Stop bit, otherwise, it is performed in the middle of the first Stop bit. The FERIF and PERIF interrupts are not delayed with STPMD. Only UxRXIF is delayed when STPMD is set and should be the only indicator for reversing transceiver direction.

31.10 Operation after FIFO overflow

The Receive Shift Register (RSR) can be configured to stop or continue running during a receive FIFO overflow condition. Stopped operation is the Legacy mode.

When the RSR continues to run during an overflow condition, the first word received after clearing the overflow will always be valid.

When the RSR is stopped during an overflow condition, synchronization with the Start bits is lost. Therefore, the first word received after the overflow is cleared may start in the middle of a word.

Operation during overflow is selected with the RUNOVF bit in the UxCON2 register. Setting the RUNOVF bit selects the run during overflow method.

31.11 Receive and Transmit Buffers

The UART uses small buffer areas to transmit and receive data. These are sometimes referred to as FIFOs.

The receiver has a Receive Shift Register (RSR) and two buffer registers. The buffer at the top of the FIFO (earliest byte to enter the FIFO) is by retrieved by reading the UxRXB register.

The transmitter has one Transmit Shift Register (TSR) and one buffer register. Writes to UxTXB go to the transmit buffer then immediately to the TSR, if it is empty. When the TSR is not empty, writes to UxTXB are held then transferred to the TSR when it becomes available.

31.11.1 FIFO STATUS

The UxFIFO register contains several status bits for determining the state of the receive and transmit buffers.

The RXBE bit indicates that the receive FIFO is empty. This bit is essentially the inverse of UxRXIF. The RXBF bit indicates that the receive FIFO is full.

The transmitter has only one buffer register so the status bits are essentially a copy and inverse of the UxTXIF bit. The TXBE bit indicates that the buffer is empty (same as UxTXIF) and the TXBF bit indicates that the buffer is full (UxTXIF inverse). A third transmitter status bit, TXWRE (transmit write error), is set whenever a UxTXB write is performed when the TXBF bit is set. This indicates that the write was unsuccessful.

31.11.2 FIFO RESET

All modes support resetting the receive and transmit buffers.

The receive buffer is flushed and all unread data discarded when the RXBE bit in the UxFIFO register is written to '1'. The MOVWF instruction with the TXBE bit cleared should be used to avoid inadvertently clearing a byte pending in the TSR when UxTXB is empty.

Data written to UxTXB when TXEN is low will be held in the Transmit Shift Register (TSR) then sent when TXEN is set. The transmit buffer and inactive TSR are flushed by setting the TXBE bit in the UxFIFO register. Setting TXBE while a character is actively transmitting from the TSR will complete the transmission without being flushed.

Clearing the ON bit will discard all received data and transmit data pending in the TSR and UxTXB.

REGISTER 31-16: UxP3H: UART PARAMETER 3 HIGH REGISTER

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0
—	—	—	—	—	—	—	P3<8>
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-6 **Unimplemented:** Read as ‘0’

bit 0 **P3<8>:** Most Significant Bit of Parameter 3

DMX mode:
Most Significant bit of last address of receive block

Other modes:
Not used

REGISTER 31-17: UxP3L: UART PARAMETER 3 LOW REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
P3<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **P3<7:0>:** Least Significant Bits of Parameter 3

DMX mode:
Least Significant Byte of last address of receive block

LIN Slave mode:
Number of data bytes to receive

Asynchronous Address mode:
Receiver address mask. Received address is XOR’d with UxP2L then AND’d with UxP3L
Match occurs when result is zero

Other modes:
Not used

REGISTER 32-2: SPIxINTE: SPI INTERRUPT ENABLE REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0
SRMTIE	TCZIE	SOSIE	EOSIE	—	RXOIE	TXUIE	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’

bit 7 **SRMTIE**: Shift Register Empty Interrupt Enable bit
1 = Enables the Shift Register Empty Interrupt
0 = Disables the Shift Register Empty Interrupt

bit 6 **TCZIE**: Transfer Counter is Zero Interrupt Enable bit
1 = Enables the Transfer Counter is Zero Interrupt
0 = Disables the Transfer Counter is Zero Interrupt

bit 5 **SOSIE**: Start of Slave Select Interrupt Enable bit
1 = Enables the Start of Slave Select Interrupt
0 = Disables the Start of Slave Select Interrupt

bit 4 **EOSIE**: End of Slave Select Interrupt Enable bit
1 = Enables the End of Slave Select Interrupt
0 = Disables the End of Slave Select Interrupt

bit 3 **Unimplemented**: Read as ‘0’

bit 2 **RXOIE**: Receiver Overflow Interrupt Enable bit
1 = Enables the Receiver Overflow Interrupt
0 = Disables the Receiver Overflow Interrupt

bit 1 **TXUIE**: Transmitter Underflow Interrupt Enable bit
1 = Enables the Transmitter Underflow Interrupt
0 = Disables the Transmitter Underflow Interrupt

bit 0 **Unimplemented**: Read as ‘0’

REGISTER 32-3: SPIxTCNTL – SPI TRANSFER COUNTER LSB REGISTER

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’

bit 7-0 TCNT<7:0>

BMODE = 0

Bits 10-3 of the Transfer Counter, counting the total number of bits to transfer

BMODE = 1

Bits 7-0 of the Transfer Counter, counting the total number of bytes to transfer

Note: This register should not be written to while a transfer is in progress (BUSY bit of SPIxCON2 is set).

REGISTER 33-6: I2CxSTAT0: I^{2}C STATUS REGISTER 0

R-0	R-0	R-0	R-0	R-0	U-0	U-0	U-0
BFRE[3]	SMA	MMA	R[1, 2]	D	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	HS = Hardware set HC = Hardware clear

bit 7 **BFRE:** Bus Free Status bit[3]

1 = Indicates the I^{2}C bus is idle
Both SCL and SDA have been high for time-out selected by I2CCON2<BFRET<1:0>> bits.
I2CCLK must select a valid clock source for this bit to function.
0 = Bus not idle (When no I2CCLK is selected, this bit remains clear)

bit 6 **SMA:** Slave Module Active Status bit

1 = Set after the 8th falling SCL edge of a received matching 7-bit slave address
Set after the 8th falling SCL edge of a received matching 10-bit slave **low** address
Set after the 8th falling SCL edge of a received matching 10-bit slave **high** w/ read address, only after a previous matching high and low w/ write.
0 = Cleared by any Restart/Stop detected on the bus
Cleared by BTOIF and BCLIF conditions

bit 5 **MMA:** Master Module Active Status bit

1 = Master Mode state machine is active
Set when master state machine asserts a Start on bus
0 = Master state machine is idle
Cleared when BCLIF is set
Cleared when Stop is shifted out by master.
Cleared for BTOIF condition, after the master successfully shifts out a Stop condition.

bit 4 **R:** Read Information bit [1, 2]

1 = Indicates the last matching received (high) address was a Read request
0 = Indicates the last matching received (high) address was a Write

bit 3 **D:** Data bit

1 = Indicates the last byte received or transmitted was data
0 = Indicates the last byte received or transmitted was an address

bit 2-0 **Unimplemented:** Read as 1’b0

Note 1: This bit holds the R bit information following the last received address match. Addresses transmitted by the Master or appearing on the bus without a match do not affect this bit.

2: Clock requests and input from I2CxCLK register are disabled in Slave modes.

3: Software must use the EN bit to force Master or Slave hardware to idle.

REGISTER 36-24: ADACCU: ADC ACCUMULATOR REGISTER UPPER

U-0	U-0	U-0	U-0	U-0	U-0	R/W-x/x	R/W-x/x
—	—	—	—	—	—	ACC<17:16>	
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-2 **Unimplemented**: Read as ‘0’

bit 1-0 **ACC<17:16>**: ADC Accumulator MSB. Upper two bits of accumulator value. See Table 36-2 for more details.

REGISTER 36-25: ADACCH: ADC ACCUMULATOR REGISTER HIGH

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
ACC<15:8>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **ACC<15:8>**: ADC Accumulator middle bits. Middle eight bits of accumulator value. See Table 36-2 for more details.

REGISTER 36-26: ADACCL: ADC ACCUMULATOR REGISTER LOW

R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x	R/W-x/x
ACC<7:0>							
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as ‘0’
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
‘1’ = Bit is set	‘0’ = Bit is cleared	

bit 7-0 **ACC<7:0>**: ADC Accumulator LSB. Lower eight bits of accumulator value. See Table 36-2 for more details.

39.0 HIGH/LOW-VOLTAGE DETECT (HLVD)

The PIC18(L)F26/27/45/46/47/55/56/57K42 family of devices has a High/Low-Voltage Detect module (HLVD). This is a programmable circuit that sets both a device voltage trip point and the direction of change from that point (positive going, negative going or both). If the device experiences an excursion past the trip point in that direction, an interrupt flag is set. If the interrupt is enabled, the program execution branches to the interrupt vector address and the software responds to the interrupt.

Complete control of the HLVD module is provided through the HLVDCON0 and HLVDCON1 register. This allows the circuitry to be “turned off” by the user under software control, which minimizes the current consumption for the device.

The module’s block diagram is shown in Figure 39-1.

Since the HLVD can be software enabled through the EN bit, setting and clearing the enable bit does not produce a false HLVD event glitch. Each time the HLVD module is enabled, the circuitry requires some time to stabilize. The RDY bit (HLVDCON0<4>) is a read-only bit used to indicate when the band gap reference voltages are stable.

The module can only generate an interrupt after the module is turned ON and the band gap reference voltages are ready.

The INTH and INTL bits determine the overall operation of the module. When INTH is set, the module monitors for rises in V_{DD} above the trip point set by the HLVDCON1 register. When INTL is set, the module monitors for drops in V_{DD} below the trip point set by the HLVDCON1 register. When both the INTH and INTL bits are set, any changes above or below the trip point set by the HLVDCON1 register can be monitored.

The OUT bit can be read to determine if the voltage is greater than or less than the voltage level selected by the HLVDCON1 register.

39.6 Operation During Sleep

When enabled, the HLVD circuitry continues to operate during Sleep. If the device voltage crosses the trip point, the HLVDIF bit will be set and the device will wake up from Sleep. Device execution will continue from the interrupt vector address if interrupts have been globally enabled.

39.7 Operation During Idle and Doze Modes

In both Idle and Doze modes, the module is active and events are generated if peripheral is enabled.

39.8 Operation During Freeze

When in Freeze mode, no new event or interrupt can be generated. The state of the LRDY bit is frozen.

Register reads and writes through the CPU interface are allowed.

39.9 Effects of a Reset

A device Reset forces all registers to their Reset state. This forces the HLVD module to be turned off.

TABLE 41-2: INSTRUCTION SET (CONTINUED)

Mnemonic, Operands		Description		Cycles	16-Bit Instruction Word				Status Affected	Notes
					MSb			LSb		
CONTROL INSTRUCTIONS										
BC	n	Branch if Carry		1 (2)	1110	0010	nnnn	nnnn	None	1
BN	n	Branch if Negative		1 (2)	1110	0110	nnnn	nnnn	None	1
BNC	n	Branch if Not Carry		1 (2)	1110	0011	nnnn	nnnn	None	1
BNN	n	Branch if Not Negative		1 (2)	1110	0111	nnnn	nnnn	None	1
BNOV	n	Branch if Not Overflow		1 (2)	1110	0101	nnnn	nnnn	None	1
BNZ	n	Branch if Not Zero		2	1110	0001	nnnn	nnnn	None	1
BOV	n	Branch if Overflow		1 (2)	1110	0100	nnnn	nnnn	None	1
BRA	n	Branch Unconditionally		1 (2)	1101	0nnn	nnnn	nnnn	None	
BZ	n	Branch if Zero		1 (2)	1110	0000	nnnn	nnnn	None	1
CALL	n, s	Call subroutine	1st word	2	1110	110s	nnnn	nnnn	None	2
			2nd word		1111	nnnn	nnnn	nnnn		
GOTO	n	Go to address	1st word	2	1110	1111	nnnn	nnnn	None	2
	—		2nd word		1111	nnnn	nnnn	nnnn		
CALLW	—	W -> PCL and Call subroutine		2	0000	0000	0001	0100	None	1
RCALL	n	Relative Call		2	1101	1nnn	nnnn	nnnn	None	1
RETFIE	s	Return from interrupt enable		2	0000	0000	0001	000s	None	1
RETLW	k	Return with literal in WREG		2	0000	1100	kkkk	kkkk	None	1
RETURN	s	Return from Subroutine		2	0000	0000	0001	001s	None	1
INHERENT INSTRUCTIONS										
CLRWDT	—	Clear Watchdog Timer		1	0000	0000	0000	0100	None	
DAW	—	Decimal Adjust WREG		1	0000	0000	0000	0111	C	
NOP	—	No Operation		1	0000	0000	0000	0000	None	
NOP	—	No Operation		1	1111	xxxx	xxxx	xxxx	None	2
POP	—	Pop top of return stack (TOS)		1	0000	0000	0000	0110	None	
PUSH	—	Push top of return stack (TOS)		1	0000	0000	0000	0101	None	
RESET		Software device Reset		1	0000	0000	1111	1111	All	
SLEEP	—	Go into Standby mode		1	0000	0000	0000	0011	None	

Note 1: If Program Counter (PC) is modified or a conditional test is true, the instruction requires an additional cycle. The extra cycle is executed as a NOP.

2: Some instructions are multi word instructions. The second/third words of these instructions will be decoded as a NOP, unless the first word of the instruction retrieves the information embedded in these 16-bits. This ensures that all program memory locations have a valid instruction.

3: f_s and f_d do not cover the full memory range. 2 MSBs of bank selection are forced to ‘b00 to limit the range of these instructions to lower 4k addressing space.

TABLE 44-4: POWER-DOWN CURRENT (IPD)(1,2)

PIC18LF26/45/46/55/56K42	Standard Operating Conditions (unless otherwise stated)
PIC18F26/45/46/55/56K42	Standard Operating Conditions (unless otherwise stated) VREGPM = 1

Param. No.	Symbol	Device Characteristics	Min.	Typ.†	Max. +85°C	Max. +125°C	Units	Conditions	
								VDD	Note
D200	IPD	IPD Base	—	0.07	2	6	µA	3.0V	
D200 D200A	IPD	IPD Base	—	0.4	4	8	µA	3.0V	
			—	20	38	42	µA	3.0V	VREGPM = 0
D201	IPD_WDT	Low-Frequency Internal Oscillator/ WDT	—	0.9	3.2	7	µA	3.0V	
D201	IPD_WDT	Low-Frequency Internal Oscillator/ WDT	—	1.1	3.2	9	µA	3.0V	
D202	IPD_SOSC	Secondary Oscillator (SOSC)	—	0.75	5	9	µA	3.0V	LP mode
D202	IPD_SOSC	Secondary Oscillator (SOSC)	—	1.0	6.5	10	µA	3.0V	LP mode
D203	IPD_FVR	FVR	—	45	74	75	µA	3.0V	FVRCON = 0x81 or 0x84
D203	IPD_FVR	FVR	—	40	70	76	µA	3.0V	FVRCON = 0x81 or 0x84
D204	IPD_BOR	Brown-out Reset (BOR)	—	9.4	14	18	µA	3.0V	
D204	IPD_BOR	Brown-out Reset (BOR)	—	9.4	15	18	µA	3.0V	
D205	IPD_LPBOR	Low-Power Brown-out Reset (LPBOR)	—	0.2	3	6	µA	3.0V	
D206	IPD_HLVD	High/Low Voltage Detect (HLVD)	—	9.5	14.8	18	µA	3.0V	
D206	IPD_HLVD	High/Low Voltage Detect (HLVD)	—	9.7	14.2	17	µA	3.0V	
D207	IPD_ADCA	ADC - Non-Converting	—	0.1	2	6	µA	3.0V	ADC not converting (4)
D207	IPD_ADCA	ADC - Non-Converting	—	0.1	4	8	µA	3.0V	ADC not converting (4)
D208	IPD_CMP	Comparator	—	33	49	50	µA	3.0V	
D208	IPD_CMP	Comparator	—	30	49	50	µA	3.0V	

† Data in “Typ.” column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: The peripheral current is the sum of the base IDD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max. values should be used when calculating total current consumption.

2: 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 VSS.

3: All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

4: ADC clock source is FRC.