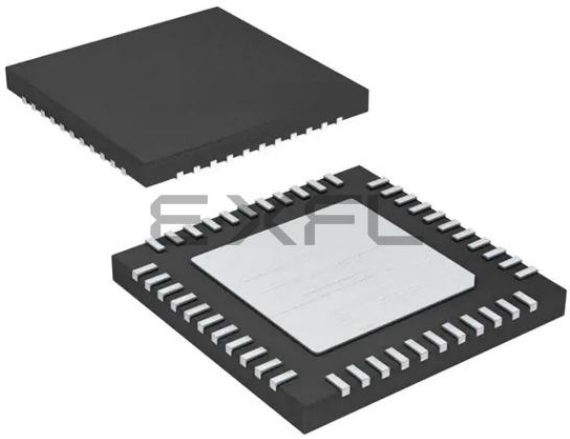


Welcome to E-XFL.COM

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	64MHz
Connectivity	ECANbus, I²C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	35
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.6K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 11x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°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/pic18f45k80-i-ml

TABLE 1-5: PIC18F4XK80 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP	QFN/ TQFP			
$\overline{MCLR}$/RE3	1	18			
$\overline{MCLR}$			I	ST	Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.
RE3			I	ST	General purpose, input only pin.
OSC1/CLKIN/RA7	13	30			
OSC1			I	ST	Oscillator crystal input.
CLKIN			I	CMOS	External clock source input. Always associated with pin function, OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
RA7			I/O	ST/ CMOS	General purpose I/O pin.
OSC2/CLKOUT/RA6	14	31			
OSC2			O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKOUT			O	—	In certain oscillator modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA6			I/O	ST/ CMOS	General purpose I/O pin.

Legend: I^2C™ = I^2C/SMBus input buffer
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output

TABLE 1-5: PIC18F4XK80 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PDIP	QFN/ TQFP			
RC7/CANRX/RX1/DT1/ CCP4	26	1			
RC7			I/O	ST/ CMOS	Digital I/O.
CANRX			I	ST	CAN bus RX.
RX1			I	ST	EUSART asynchronous receive.
DT1			I/O	ST	EUSART synchronous data. (See related TX2/CK2.)
CCP4			I/O	ST	Capture 4 input/Compare 4 output/PWM4 output.

Legend: I^2C™ = I^2C/SMBus input buffer
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output

TABLE 1-6: PIC18F6XK80 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Num	Pin Type	Buffer Type	Description
				PORTA is a bidirectional I/O port.
RA0/CVREF/AN0/ ULPWU	29			
RA0		I/O	ST/ CMOS	General purpose I/O pin.
CVREF		O	Analog	Comparator reference voltage output.
AN0		I	Analog	Analog Input 0.
ULPWU		I	Analog	Ultra Low-Power Wake-up input.
RA1/AN1/C1INC	30			
RA1		I/O	ST/ CMOS	Digital I/O.
AN1		I	Analog	Analog Input 1.
C1INC		I	Analog	Comparator 1 Input C.
RA2/VREF-/AN2/C2INC	31			
RA2		I/O	ST/ CMOS	Digital I/O.
VREF-		I	Analog	A/D reference voltage (low) input.
AN2		I	Analog	Analog Input 2.
C2INC		I	Analog	Comparator 2 Input C.
RA3/VREF+/AN3	32			
RA3		I/O	ST/ CMOS	Digital I/O.
VREF+		I	Analog	A/D reference voltage (high) input.
AN3		I	Analog	Analog Input 3.
RA5/AN4/HLVDIN/ T1CKI/$\overline{SS}$	34			
RA5		I/O	ST/ CMOS	Digital I/O.
AN4		I	Analog	Analog Input 4.
HLVDIN		I	Analog	High/Low-Voltage Detect input.
T1CKI		I	ST	Timer1 clock input.
$\overline{SS}$		I	ST	SPI slave select input.

Legend: I^2C™ = I^2C/SMBus input buffer
ST = Schmitt Trigger input with CMOS levels
I = Input
P = Power
CMOS = CMOS compatible input or output
Analog = Analog input
O = Output

3.2 Control Registers

The OSCCON register (Register 3-1) controls the main aspects of the device clock's operation. It selects the oscillator type to be used, which of the power-managed modes to invoke and the output frequency of the INTOSC source. It also provides status on the oscillators.

The OSCTUNE register (Register 3-3) controls the tuning and operation of the internal oscillator block. It also implements the PLLEN bit which controls the operation of the Phase Locked Loop (PLL) (see **Section 3.5.3 "PLL Frequency Multiplier"**).

REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER

R/W-0	R/W-1	R/W-1	R/W-0	R(1)	R-0	R/W-0	R/W-0
IDLEN	IRCF2(2)	IRCF1(2)	IRCF0(2)	OSTS	HFIOFS	SCS1(4)	SCS0(4)
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 **IDLEN:** Idle Enable bit
- 1 = Device enters an Idle mode when a SLEEP instruction is executed
- 0 = Device enters Sleep mode when a SLEEP instruction is executed

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits(2)
- 111 = HF-INTOSC output frequency is used (16 MHz)
- 110 = HF-INTOSC/2 output frequency is used (8 MHz, default)
- 101 = HF-INTOSC/4 output frequency is used (4 MHz)
- 100 = HF-INTOSC/8 output frequency is used (2 MHz)
- 011 = HF-INTOSC/16 output frequency is used (1 MHz)

If INTSRC = 0 and MFIOSEL = 0:(3,5)
- 010 = HF-INTOSC/32 output frequency is used (500 kHz)
- 001 = HF-INTOSC/64 output frequency is used (250 kHz)
- 000 = LF-INTOSC output frequency is used (31.25 kHz)(6)

If INTSRC = 1 and MFIOSEL = 0:(3,5)
- 010 = HF-INTOSC/32 output frequency is used (500 kHz)
- 001 = HF-INTOSC/64 output frequency is used (250 kHz)
- 000 = HF-INTOSC/512 output frequency is used (31.25 kHz)

If INTSRC = 0 and MFIOSEL = 1:(3,5)
- 010 = MF-INTOSC output frequency is used (500 kHz)
- 001 = MF-INTOSC/2 output frequency is used (250 kHz)
- 000 = LF-INTOSC output frequency is used (31.25 kHz)(6)

If INTSRC = 1 and MFIOSEL = 1:(3,5)
- 010 = MF-INTOSC output frequency is used (500 kHz)
- 001 = MF-INTOSC/2 output frequency is used (250 kHz)
- 000 = MF-INTOSC/16 output frequency is used (31.25 kHz)

bit 3 **OSTS:** Oscillator Start-up Timer Time-out Status bit(1)
- 1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running, as defined by FOSC<3:0>
- 0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready – device is running from internal oscillator (HF-INTOSC, MF-INTOSC or LF-INTOSC)

Note 1: The Reset state depends on the state of the IESO Configuration bit (CONFIG1H<7>).
2: Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.
3: The source is selected by the INTSRC bit (OSCTUNE<7>).
4: Modifying these bits will cause an immediate clock source switch.
5: INTSRC = OSCTUNE<7> and MFIOSEL = OSCCON2<0>.
6: This is the lowest power option for an internal source.

8.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADRH:EEADR register pair, clear the EEPGD control bit (EECON1<7>) and then set control bit, RD (EECON1<0>). The data is available after one instruction cycle, in the EEDATA register. It can be read after one NOP instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

The basic process is shown in Example 8-1.

8.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADRH:EEADR register pair and the data written to the EEDATA register. The sequence in Example 8-2 must be followed to initiate the write cycle.

The write will not begin if this sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment.

Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, EECON1, EEADRH:EEADR and EEDATA cannot be modified. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt or poll this bit; EEIF must be cleared by software.

8.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

Note:	Self-write execution to Flash and EEPROM memory cannot be done while running in LP Oscillator (low-power) mode. Executing a self-write will put the device into High-Power mode.

14.2 Timer1 Operation

The Timer1 module is an 8 or 16-bit incrementing counter that is accessed through the TMR1H:TMR1L register pair.

When used with an internal clock source, the module is a timer and increments on every instruction cycle. When used with an external clock source, the module can be used as either a timer or counter. It increments on every selected edge of the external source.

Timer1 is enabled by configuring the TMR1ON and TMR1GE bits in the T1CON and T1GCON registers, respectively.

When SOSC is selected as Crystal mode (by the SOSCSELx bits), the RC1/SOSCI and RC0/SOSCO/SCLKI pins become inputs. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

14.3 Clock Source Selection

The TMR1CS<1:0> and SOSCEN bits of the T1CON register are used to select the clock source for Timer1. Table 14-1 displays the clock source selections.

14.3.1 INTERNAL CLOCK SOURCE

When the internal clock source is selected, the TMR1H:TMR1L register pair will increment on multiples of FOSC as determined by the Timer1 prescaler.

14.3.2 EXTERNAL CLOCK SOURCE

When the external clock source is selected, the Timer1 module may work as a timer or a counter.

When enabled to count, Timer1 is incremented on the rising edge of the external clock input, T1CKI. Either of these external clock sources can be synchronized to the microcontroller system clock or they can run asynchronously.

When used as a timer with a clock oscillator, an external, 32.768 kHz crystal can be used in conjunction with the dedicated internal oscillator circuit.

Note: In Counter mode, a falling edge must be registered by the counter prior to the first incrementing rising edge after any one or more of the following conditions:

- Timer1 is enabled after POR Reset
- Write to TMR1H or TMR1L
- Timer1 is disabled
- Timer1 is disabled (TMR1ON = 0)

When T1CKI is high, Timer1 is enabled (TMR1ON = 1) when T1CKI is low.

TABLE 14-1: TIMER1 CLOCK SOURCE SELECTION

TMR1CS1	TMR1CS0	SOSCEN	Clock Source
0	1	x	Clock Source (FOSC)
0	0	x	Instruction Clock (FOSC/4)
1	0	0	External Clock on T1CKI Pin
1	0	1	Oscillator Circuit on SOSCI/SOSCO Pins

14.8.3 TIMER1 GATE TOGGLE MODE

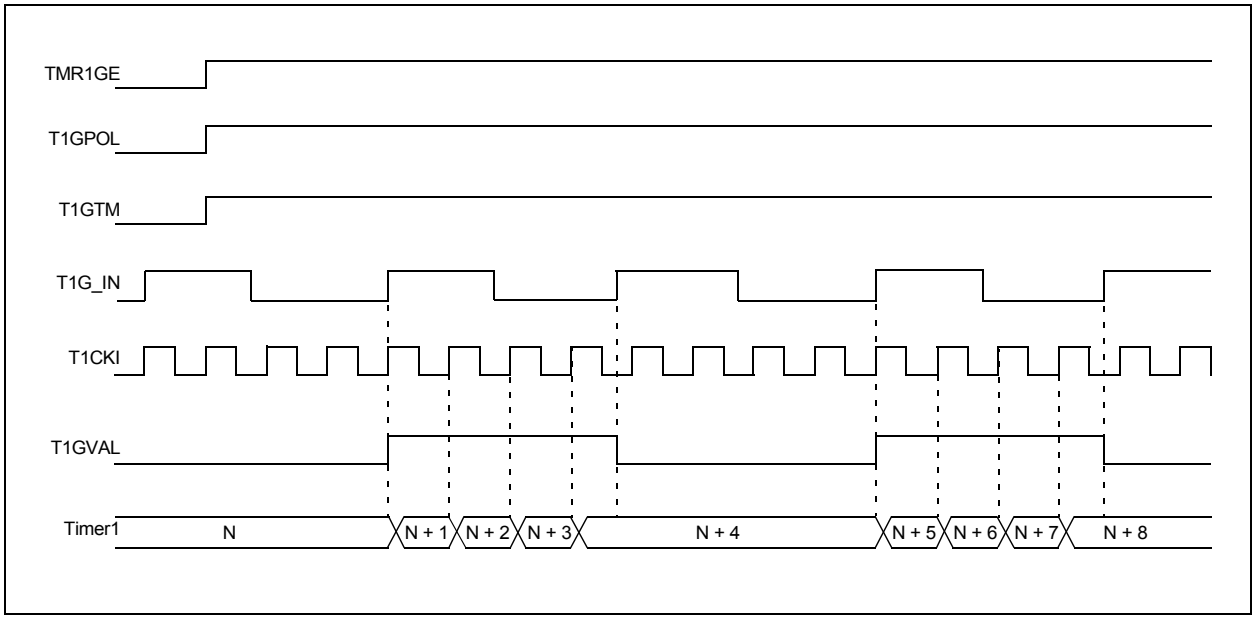
When Timer1 Gate Toggle mode is enabled, it is possible to measure the full cycle length of a Timer1 gate signal, as opposed to the duration of a single level pulse.

The Timer1 gate source is routed through a flip-flop that changes state on every incrementing edge of the signal. (For timing details, see Figure 14-5.)

The T1GVAL bit (T1GCON<2>) indicates when the Toggled mode is active and the timer is counting.

The Timer1 Gate Toggle mode is enabled by setting the T1GTM bit (T1GCON<5>). When T1GTM is cleared, the flip-flop is cleared and held clear. This is necessary in order to control which edge is measured.

FIGURE 14-5: TIMER1 GATE TOGGLE MODE



REGISTER 16-3: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-0	U-0	RW-1	R/W-0	U-0	R-x	R/W-0
—	SOSCRUN	—	SOSCDRV(1)	SOSCGO	—	MFIOFS	MFIOSEL
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 **Unimplemented:** Read as '0'

bit 6 **SOSCRUN:** SOSC Run Status bit
- 1 = System clock comes from a secondary SOSC
- 0 = System clock comes from an oscillator other than SOSC

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

bit 4 **SOSCDRV:** Secondary Oscillator Drive Control bit(1)
- 1 = High-power SOSC circuit selected
- 0 = Low/high-power select is done via the SOSCSEL<1:0> Configuration bits

bit 3 **SOSCGO:** Oscillator Start Control bit
- 1 = Oscillator is running even if no other sources are requesting it
- 0 = Oscillator is shut off if no other sources are requesting it (When the SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.)

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

bit 1 **MFIOFS:** MF-INTOSC Frequency Stable bit
- 1 = MF-INTOSC is stable
- 0 = MF-INTOSC is not stable

bit 0 **MFIOSEL:** MF-INTOSC Select bit
- 1 = MF-INTOSC is used in place of HF-INTOSC frequencies of 500 kHz, 250 kHz and 31.25 kHz
- 0 = MF-INTOSC is not used

Note 1: When SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.

A shutdown condition is indicated by the ECCP1ASE (Auto-Shutdown Event Status) bit (ECCP1AS<7>). If the bit is a '0', the PWM pins are operating normally. If the bit is a '1', the PWM outputs are in the shutdown state.

When a shutdown event occurs, two things happen:

- The ECCP1ASE bit is set to '1'. The ECCP1ASE will remain set until cleared in firmware or an auto-restart occurs. (See **Section 20.4.5 "Auto-Restart Mode"**.)
- The enabled PWM pins are asynchronously placed in their shutdown states. The PWM output pins are grouped into pairs (P1A/P1C) and (P1B/P1D). The state of each pin pair is determined by the PSS1ACx and PSS1BDx bits (ECCP1AS<3:2> and <1:0>, respectively).

Each pin pair may be placed into one of three states:

- Drive logic '1'
- Drive logic '0'
- Tri-state (high-impedance)

REGISTER 20-3: ECCP1AS: ECCP1 AUTO-SHUTDOWN 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
ECCP1ASE	ECCP1AS2	ECCP1AS1	ECCP1AS0	PSS1AC1	PSS1AC0	PSS1BD1	PSS1BD0
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 **ECCP1ASE:** ECCP Auto-Shutdown Event Status bit
1 = A shutdown event has occurred; ECCP outputs are in a shutdown state
0 = ECCP outputs are operating

bit 6-4 **ECCP1AS<2:0>:** ECCP Auto-Shutdown Source Select bits
000 = Auto-shutdown is disabled
001 = Comparator C1OUT output is high
010 = Comparator C2OUT output is high
011 = Either Comparator C1OUT or C2OUT is high
100 = VIL on FLT0 pin
101 = VIL on FLT0 pin or Comparator C1OUT output is high
110 = VIL on FLT0 pin or Comparator C2OUT output is high
111 = VIL on FLT0 pin or Comparator C1OUT or Comparator C2OUT is high

bit 3-2 **PSS1AC<1:0>:** P1A and P1C Pins Shutdown State Control bits
00 = Drive pins, P1A and P1C, to '0'
01 = Drive pins, P1A and P1C, to '1'
1x = Pins, P1A and P1C, tri-state

bit 1-0 **PSS1BD<1:0>:** P1B and P1D Pins Shutdown State Control bits
00 = Drive pins, P1B and P1D, to '0'
01 = Drive pins, P1B and P1D, to '1'
1x = Pins, P1B and P1D, tri-state

Note 1: The auto-shutdown condition is a level-based signal, not an edge-based signal. As long as the level is present, the auto-shutdown will persist.
2: Writing to the ECCP1ASE bit is disabled while an auto-shutdown condition persists.
3: Once the auto-shutdown condition has been removed and the PWM restarted (either through firmware or auto-restart), the PWM signal will always restart at the beginning of the next PWM period.

After the A/D module has been configured as desired, the selected channel must be acquired before the conversion can start. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 23.3 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

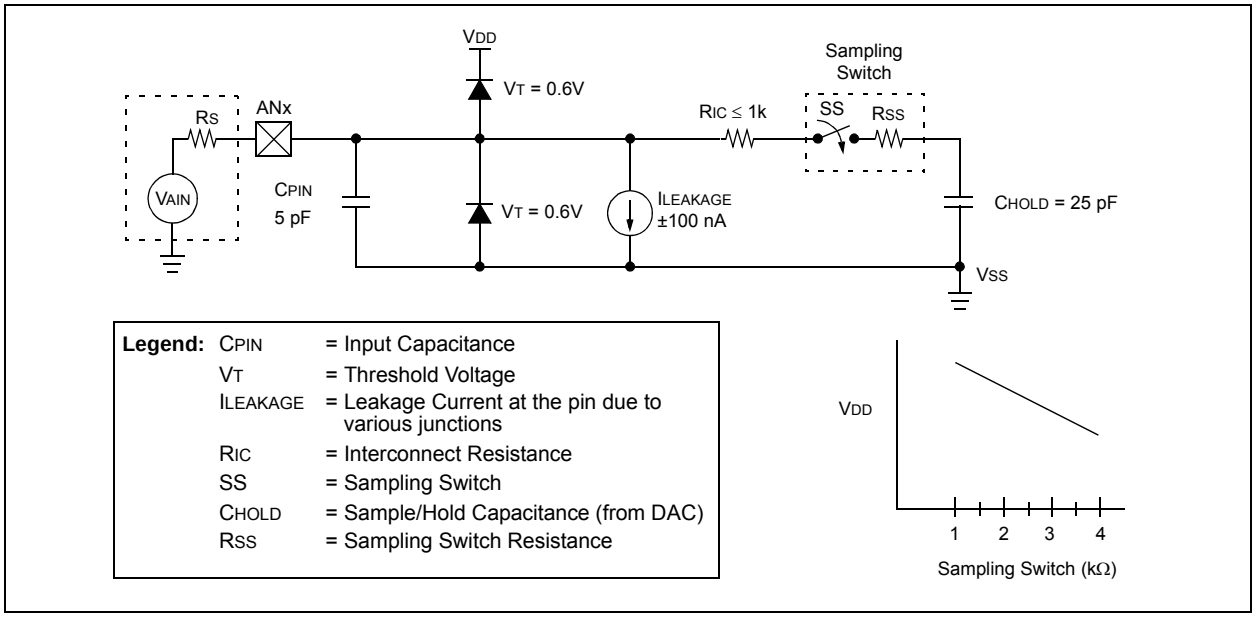
To do an A/D conversion, follow these steps:

1. Configure the A/D module:
   - Configure the required A/D pins as analog pins (ANCON0 and ANCON1)
   - Set the voltage reference (ADCON1)
   - Select the A/D positive and negative input channels (ADCON0 and ADCON1)
   - Select the A/D acquisition time (ADCON2)
   - Select the A/D conversion clock (ADCON2)
   - Turn on the A/D module (ADCON0)
2. Configure the A/D interrupt (if desired):
   - Clear the ADIF bit (PIR1<6>)
   - Set the ADIE bit (PIE1<6>)
   - Set the GIE bit (INTCON<7>)
3. Wait the required acquisition time (if required).
4. Start the conversion:
   - Set the GO/$\overline{\text{DONE}}$ bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
   - Polling for the GO/$\overline{\text{DONE}}$ bit to be cleared

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL) and, if required, clear bit, ADIF.
7. For the next conversion, begin with Step 1 or 2, as required.

   The A/D conversion time per bit is defined as T_{AD}. Before the next acquisition starts, a minimum wait of 2 T_{AD} is required.

FIGURE 23-5: ANALOG INPUT MODEL



REGISTER 27-24: BnSIDH: TX/RX BUFFER 'n' STANDARD IDENTIFIER REGISTERS, HIGH BYTE IN RECEIVE MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 0](1)

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
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 7-0 **SID<10:3>:** Standard Identifier bits (if EXIDE (BnSIDL<3>) = 0)
Extended Identifier bits, EID<28:21> (if EXIDE = 1).

Note 1: These registers are available in Mode 1 and 2 only.

REGISTER 27-25: BnSIDH: TX/RX BUFFER 'n' STANDARD IDENTIFIER REGISTERS, HIGH BYTE IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1](1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
SID10	SID9	SID8	SID7	SID6	SID5	SID4	SID3
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 7-0 **SID<10:3>:** Standard Identifier bits (if EXIDE (BnSIDL<3>) = 0)
Extended Identifier bits, EID<28:21> (if EXIDE = 1).

Note 1: These registers are available in Mode 1 and 2 only.

27.5.4 PROGRAMMABLE AUTO-RTR BUFFERS

In Mode 1 and 2, any of six programmable transmit/receive buffers may be programmed to automatically respond to predefined RTR messages without user firmware intervention. Automatic RTR handling is enabled by setting the TX2EN bit in the BSEL0 register and the RTREN bit in the BnCON register. After this setup, when an RTR request is received, the TXREQ bit is automatically set and the current buffer content is automatically queued for transmission as a RTR response. As with all transmit buffers, once the TXREQ bit is set, buffer registers become read-only and any writes to them will be ignored.

The following outlines the steps required to automatically handle RTR messages:

1. Set buffer to Transmit mode by setting the TXnEN bit to '1' in the BSEL0 register.
2. At least one acceptance filter must be associated with this buffer and preloaded with the expected RTR identifier.
3. Bit, RTREN in the BnCON register, must be set to '1'.
4. Buffer must be preloaded with the data to be sent as a RTR response.

Normally, user firmware will keep buffer data registers up to date. If firmware attempts to update the buffer while an automatic RTR response is in the process of transmission, all writes to buffers are ignored.

27.6 CAN Message Transmission

27.6.1 INITIATING TRANSMISSION

For the MCU to have write access to the message buffer, the TXREQ bit must be clear, indicating that the message buffer is clear of any pending message to be transmitted. At a minimum, the SIDH, SIDL and DLC registers must be loaded. If data bytes are present in the message, the Data registers must also be loaded. If the message is to use extended identifiers, the EIDH:EIDL registers must also be loaded and the EXIDE bit set.

To initiate message transmission, the TXREQ bit must be set for each buffer to be transmitted. When TXREQ is set, the TXABT, TXLARB and TXERR bits will be cleared. To successfully complete the transmission, there must be at least one node with matching baud rate on the network.

Setting the TXREQ bit does not initiate a message transmission; it merely flags a message buffer as ready for transmission. Transmission will start when the device detects that the bus is available. The device will then begin transmission of the highest priority message that is ready.

When the transmission has completed successfully, the TXREQ bit will be cleared, the TXBnIF bit will be set and an interrupt will be generated if the TXBnIE bit is set.

If the message transmission fails, the TXREQ will remain set, indicating that the message is still pending for transmission and one of the following condition flags will be set. If the message started to transmit but encountered an error condition, the TXERR and the IRXIF bits will be set and an interrupt will be generated. If the message lost arbitration, the TXLARB bit will be set.

27.6.2 ABORTING TRANSMISSION

The MCU can request to abort a message by clearing the TXREQ bit associated with the corresponding message buffer (TXBnCON<3> or BnCON<3>). Setting the ABAT bit (CANCON<4>) will request an abort of all pending messages. If the message has not yet started transmission, or if the message started but is interrupted by loss of arbitration or an error, the abort will be processed. The abort is indicated when the module sets the TXABT bit for the corresponding buffer (TXBnCON<6> or BnCON<6>). If the message has started to transmit, it will attempt to transmit the current message fully. If the current message is transmitted fully and is not lost to arbitration or an error, the TXABT bit will not be set because the message was transmitted successfully. Likewise, if a message is being transmitted during an abort request and the message is lost to arbitration or an error, the message will not be retransmitted and the TXABT bit will be set, indicating that the message was successfully aborted.

Once an abort is requested by setting the ABAT or TXABT bits, it cannot be cleared to cancel the abort request. Only CAN module hardware or a POR condition can clear it.

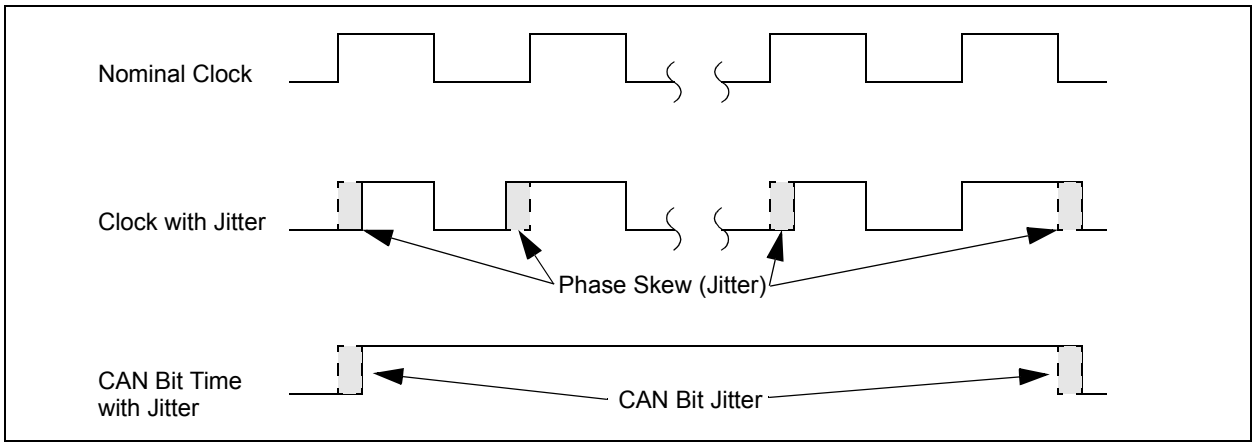
27.9.1 EXTERNAL CLOCK, INTERNAL CLOCK AND MEASURABLE JITTER IN HS-PLL BASED OSCILLATORS

The microcontroller clock frequency generated from a PLL circuit is subject to a jitter, also defined as Phase Jitter or Phase Skew. For its PIC18 Enhanced microcontrollers, Microchip specifies phase jitter (P_{jitter}) as being 2% (Gaussian distribution, within 3 standard deviations, see Parameter F13 in Table 31-7) and Total Jitter (T_{jitter}) as being 2 * P_{jitter}.

The CAN protocol uses a bit-stuffing technique that inserts a bit of a given polarity following five bits with the opposite polarity. This gives a total of 10 bits transmitted without resynchronization (compensation for jitter or phase error).

Given the random nature of the added jitter error, it can be shown that the total error caused by the jitter tends to cancel itself over time. For a period of 10 bits, it is necessary to add only two jitter intervals to correct for jitter induced error: one interval in the beginning of the 10-bit period and another at the end. The overall effect is shown in Figure 27-5.

FIGURE 27-5: EFFECTS OF PHASE JITTER ON THE MICROCONTROLLER CLOCK AND CAN BIT TIME



Once these considerations are taken into account, it is possible to show that the relation between the jitter and the total frequency error can be defined as:

EQUATION 27-4: JITTER AND TOTAL FREQUENCY ERROR

$$\Delta f = \frac{T_{jitter}}{10 \times NBT} = \frac{2 \times P_{jitter}}{10 \times NBT}$$

where jitter is expressed in terms of time and NBT is the Nominal Bit Time.

For example, assume a CAN bit rate of 125 Kb/s, which gives an NBT of 8 µs. For a 16 MHz clock generated from a 4x PLL, the jitter at this clock frequency is:

EQUATION 27-5: 16 MHz CLOCK FROM 4x PLL JITTER:

$$2\% \times \frac{1}{16\ MHz} = \frac{0.02}{16 \times 10^6} = 1.25 ns$$

and resultant frequency error is:

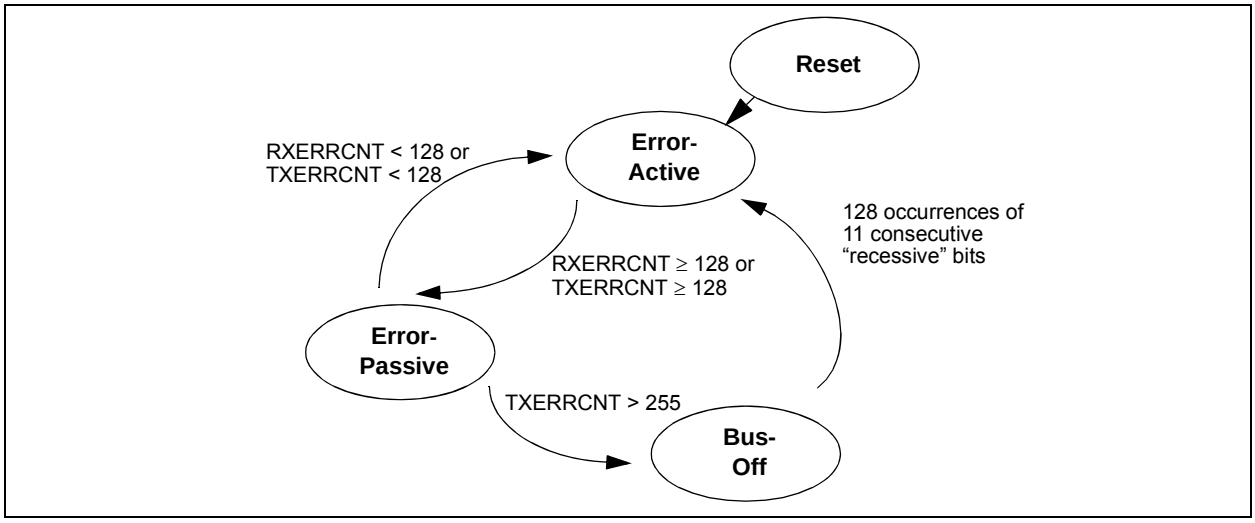
EQUATION 27-6: RESULTANT FREQUENCY ERROR:

$$\frac{2 \times (1.25 \times 10^{-9})}{10 \times (8 \times 10^{-6})} = 3.125 \times 10^{-5} = 0.0031\%$$

The PIC18F66K80 family devices are error-active if both error counters are below the error-passive limit of 128. They are error-passive if at least one of the error counters equals or exceeds 128. They go to bus-off if the transmit error counter equals or exceeds the bus-off limit of 256. The devices remain in this state until the bus-off recovery sequence is finished. The bus-off recovery sequence consists of 128 occurrences of 11 consecutive recessive bits (see Figure 27-8). Note that the CAN module, after going bus-off, will recover back to error-active without any intervention by the MCU if the bus remains Idle for 128 x 11 bit times. If this is not desired, the error Interrupt Service Routine should address this. The current Error mode of the CAN module can be read by the MCU via the COMSTAT register.

Additionally, there is an Error State Warning flag bit, EWARN, which is set if at least one of the error counters equals or exceeds the error warning limit of 96. EWARN is reset if both error counters are less than the error warning limit.

FIGURE 27-8: ERROR MODES STATE DIAGRAM



27.15 CAN Interrupts

The module has several sources of interrupts. Each of these interrupts can be individually enabled or disabled. The PIR5 register contains interrupt flags. The PIE5 register contains the enables for the 8 main interrupts. A special set of read-only bits in the CANSTAT register, the ICODE bits, can be used in combination with a jump table for efficient handling of interrupts.

All interrupts have one source, with the exception of the error interrupt and buffer interrupts in Mode 1 and 2. Any of the error interrupt sources can set the error interrupt flag. The source of the error interrupt can be determined by reading the Communication Status register, COMSTAT. In Mode 1 and 2, there are two interrupt enable/disable and flag bits – one for all transmit buffers and the other for all receive buffers.

The interrupts can be broken up into two categories: receive and transmit interrupts.

The receive related interrupts are:

- Receive Interrupts
- Wake-up Interrupt
- Receiver Overrun Interrupt
- Receiver Warning Interrupt
- Receiver Error-Passive Interrupt

The transmit related interrupts are:

- Transmit Interrupts
- Transmitter Warning Interrupt
- Transmitter Error-Passive Interrupt
- Bus-Off Interrupt

TABLE 28-1: CONFIGURATION BITS AND DEVICE IDs

File Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default/ Unprogrammed Value
300000h	CONFIG1L	—	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	—	$\overline{\text{RETEN}}$	-1-1 11-1
300001h	CONFIG1H	IESO	FCMEN	—	PLLCFG	FOSC3	FOSC2	FOSC1	FOSC0	00-0 1000
300002h	CONFIG2L	—	BORPWR1	BORWPR0	BORV1	BORV0	BOREN1	BOREN0	$\overline{\text{PWRTEN}}$	-111 1111
300003h	CONFIG2H	—	WDTPS4	WDTPS3	WDTPS2	WDTPS1	WDTPS0	WDTEN1	WDTEN0	-111 1111
300005h	CONFIG3H	MCLRE	—	—	—	MSSPMSK	T3CKMX(1,3)	T0CKMX(1)	CANMX	1--- 1qq1
300006h	CONFIG4L	$\overline{\text{DEBUG}}$	—	—	BBSIZ0	—	—	—	STVREN	1--1 ---1
300008h	CONFIG5L	—	—	—	—	CP3	CP2	CP1	CP0	---- 1111
300009h	CONFIG5H	CPD	CPB	—	—	—	—	—	—	11-- ----
30000Ah	CONFIG6L	—	—	—	—	WRT3	WRT2	WRT1	WRT0	---- 1111
30000Bh	CONFIG6H	WRTD	WRTB	WRTC	—	—	—	—	—	111- ----
30000Ch	CONFIG7L	—	—	—	—	EBTR3	EBTR2	EBTR1	EBTR0	---- 1111
30000Dh	CONFIG7H	—	EBTRB	—	—	—	—	—	—	-1-- ----
3FFFFEh	DEVID1(2)	DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0	xxxx xxxx
3FFFFFh	DEVID2(2)	DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	xxxx xxxx

Legend: x = unknown, u = unchanged, - = unimplemented, q = value depends on condition. Shaded cells are unimplemented, read as '0'.

Note 1: Implemented only on the 64-pin devices (PIC18F6XK80).

2: See Register 28-13 for DEVID1 values. DEVID registers are read-only and cannot be programmed by the user.

3: Maintain as '0' on 28-pin, 40-pin and 44-pin devices.

REGISTER 28-13: DEVID1: DEVICE ID REGISTER 1 FOR THE PIC18F66K80 FAMILY

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
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 7-5 **DEV<2:0>:** Device ID bits

These bits are used with the DEV<10:3> bits in the Device ID Register 2 to identify the part number:
000 = PIC18F46K80, PIC18LF26K80
001 = PIC18F26K80, PIC18LF65K80
010 = PIC18F65K80, PIC18LF45K80
011 = PIC18F45K80, PIC18LF25K80
100 = PIC18F25K80
110 = PIC18LF66K80
111 = PIC18F66K80, PIC18LF46K80

bit 4-0 **REV<4:0>:** Revision ID bits

These bits are used to indicate the device revision.

REGISTER 28-14: DEVID2: DEVICE ID REGISTER 2 FOR THE PIC18F66K80 FAMILY

R	R	R	R	R	R	R	R
DEV10(1)	DEV9(1)	DEV8(1)	DEV7(1)	DEV6(1)	DEV5(1)	DEV4(1)	DEV3(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 7-0 **DEV<10:3>:** Device ID bits(1)

These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

Note 1: These values for DEV<10:3> may be shared with other devices. The specific device is always identified by using the entire DEV<10:0> bit sequence.

BNC **Branch if Not Carry**

Syntax: BNC n

Operands: -128 ≤ n ≤ 127

Operation: if Carry bit is '0',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0011	nnnn	nnnn

Description: If the Carry bit is '0', then the program will branch.

The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BNC Jump

Before Instruction
PC = address (HERE)
After Instruction
If Carry = 0;
PC = address (Jump)
If Carry = 1;
PC = address (HERE + 2)

BNN **Branch if Not Negative**

Syntax: BNN n

Operands: -128 ≤ n ≤ 127

Operation: if Negative bit is '0',
(PC) + 2 + 2n → PC

Status Affected: None

Encoding:

1110	0111	nnnn	nnnn

Description: If the Negative bit is '0', then the program will branch.

The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	Write to PC
No operation	No operation	No operation	No operation

If No Jump:

Q1	Q2	Q3	Q4
Decode	Read literal 'n'	Process Data	No operation

Example: HERE BNN Jump

Before Instruction
PC = address (HERE)
After Instruction
If Negative = 0;
PC = address (Jump)
If Negative = 1;
PC = address (HERE + 2)

MOVSS **Move Indexed to Indexed**

Syntax: MOVSS [z_s], [z_d]

Operands: $0 \le z_s \le 127$
$0 \le z_d \le 127$

Operation: ((FSR2) + z_s) → ((FSR2) + z_d)

Status Affected: None

Encoding:

1st word (source)	1110	1011	1zzz	$zzzz_s$
2nd word (dest.)	1111	xxxx	xzzz	$zzzz_d$

Description: The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z_s' or 'z_d', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	Determine dest addr	Determine dest addr	Write to dest reg

Example: MOVSS [05h], [06h]

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
Contents of 86h = 33h

PUSHL **Store Literal at FSR2, Decrement FSR2**

Syntax: PUSHL k

Operands: $0 \le k \le 255$

Operation: k → (FSR2),
FSR2 – 1 → FSR2

Status Affected: None

Encoding:

1111	1010	kkkk	kkkk

Description: The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.

This instruction allows users to push values onto a software stack.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read 'k'	Process data	Write to destination

Example: PUSHL 08h

Before Instruction
FSR2H:FSR2L = 01ECh
Memory (01ECh) = 00h

After Instruction
FSR2H:FSR2L = 01EBh
Memory (01ECh) = 08h

31.2 DC Characteristics: Power-Down and Supply Current PIC18F66K80 Family (Industrial/Extended) (Continued)

PIC18F66K80 Family (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) Cont.(2,3)						
	PIC18**LF**XXK80	880	1600	nA	-40°C	VDD = 1.8V(4) Regulator Disabled	FOSC = 31 kHz (**RC_IDLE** mode, LF-INTOSC)
		880	1600	nA	+25°C		
		880	1600	nA	+60°C		
		1	2	µA	+85°C		
		5	10	µA	+125°C		
	PIC18**LF**XXK80	1.6	5	µA	-40°C	VDD = 3.3V(4) Regulator Disabled	
		1.6	5	µA	+25°C		
		1.6	5	µA	+60°C		
		2	6	µA	+85°C		
		7	12	µA	+125°C		
	PIC18**F**XXK80	41	130	µA	-40°C	VDD = 3.3V(5) Regulator Enabled	
		59	130	µA	+25°C		
		64	130	µA	+60°C		
		70	150	µA	+85°C		
		80	175	µA	+125°C		
	PIC18**F**XXK80	53	160	µA	-40°C	VDD = 5V(5) Regulator Enabled	
		62	160	µA	+25°C		
		70	160	µA	+60°C		
		85	170	µA	+85°C		
		100	180	µA	+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 a high-impedance state and tied to VDD or VSS, and all features that add delta current are disabled (such as WDT, SOSC 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;
$\overline{MCLR}$ = VDD; WDT enabled/disabled as specified.

3: 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.

4: For LF devices, $\overline{RETEN}$ (CONFIG1L<0>) = 1.

5: For F devices, SRETEN (WDTCON<4>) = 1 and $\overline{RETEN}$ (CONFIG1L<0>) = 0.