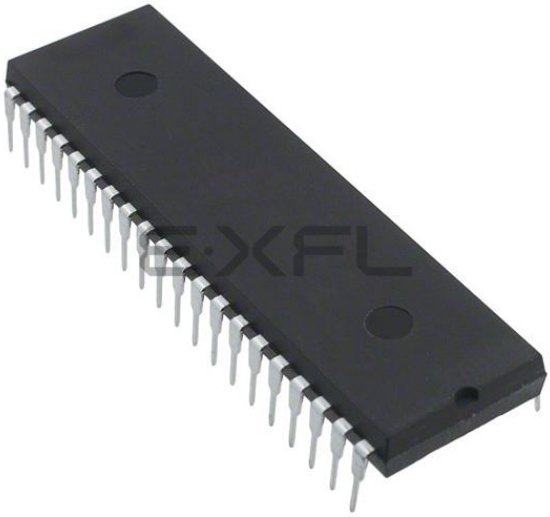


Welcome to E-XFL.COM

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 ~ 3.6V
Data Converters	A/D 11x12b
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/pic18lf45k80-i-p

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.

If the IRCFx bits and the INTSRC bit are all clear, the INTOSC output (HF-INTOSC/MF-INTOSC) is not enabled and the HFIOFS and MFIOFS bits will remain clear. There will be no indication of the current clock source. The LF-INTOSC source is providing the device clocks.

If the IRCFx bits are changed from all clear (thus, enabling the INTOSC output) or if INTSRC or MFIOSEL is set, the HFIOFS or MFIOFS bit is set after the INTOSC output becomes stable. For details, see Table 4-3.

TABLE 4-3: INTERNAL OSCILLATOR FREQUENCY STABILITY BITS

IRCF<2:0>	INTSRC	MFIOSEL	Status of MFIOFS or HFIOFS when INTOSC is Stable
000	0	x	MFIOFS = 0, HFIOFS = 0 and clock source is LF-INTOSC
000	1	0	MFIOFS = 0, HFIOFS = 1 and clock source is HF-INTOSC
000	1	1	MFIOFS = 1, HFIOFS = 0 and clock source is MF-INTOSC
Non-Zero	x	0	MFIOFS = 0, HFIOFS = 1 and clock source is HF-INTOSC
Non-Zero	x	1	MFIOFS = 1, HFIOFS = 0 and clock source is MF-INTOSC

Clocks to the device continue while the INTOSC source stabilizes after an interval of TIOBST (Parameter 39, Table 31-11).

If the IRCFx bits were previously at a non-zero value, or if INTSRC was set before setting SCS1 and the INTOSC source was already stable, the HFIOFS or MFIOFS bit will remain set.

On transitions from RC_RUN mode to PRI_RUN mode, the device continues to be clocked from the INTOSC multiplexer while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 4-4). When the clock switch is complete, the HFIOFS or MFIOFS bit is cleared, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCSx bits are not affected by the switch. The LF-INTOSC source will continue to run if either the WDT or the Fail-Safe Clock Monitor (FSCM) is enabled.

9.0 8 x 8 HARDWARE MULTIPLIER

9.1 Introduction

All PIC18 devices include an 8 x 8 hardware multiplier as part of the ALU. The multiplier performs an unsigned operation and yields a 16-bit result that is stored in the product register pair, PRODH:PRODL. The multiplier's operation does not affect any flags in the STATUS register.

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows PIC18 devices to be used in many applications previously reserved for digital-signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 9-1.

9.2 Operation

Example 9-1 shows the instruction sequence for an 8 x 8 unsigned multiplication. Only one instruction is required when one of the arguments is already loaded in the WREG register.

Example 9-2 shows the sequence to do an 8 x 8 signed multiplication. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 9-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W     ;
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
```

EXAMPLE 9-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF    ARG1, W
MULWF   ARG2        ; ARG1 * ARG2 ->
                    ; PRODH:PRODL
BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG1
MOVF    ARG2, W
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F    ; PRODH = PRODH
                    ;         - ARG2
```

TABLE 9-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time			
				@ 64 MHz	@ 48 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	4.3 μs	5.7 μs	27.6 μs	69 μs
	Hardware multiply	1	1	62.5 ns	83.3 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	5.6 μs	7.5 μs	36.4 μs	91 μs
	Hardware multiply	6	6	375 ns	500 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	15.1 μs	20.1 μs	96.8 μs	242 μs
	Hardware multiply	28	28	1.7 μs	2.3 μs	11.2 μs	28 μs
16 x 16 signed	Without hardware multiply	52	254	15.8 μs	21.2 μs	101.6 μs	254 μs
	Hardware multiply	35	40	2.5 μs	3.3 μs	16.0 μs	40 μs

10.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack.

If a fast return from interrupt is not used (see **Section 6.3 "Data Memory Organization"**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine (ISR). Depending on the user's application, other registers also may need to be saved.

Example 10-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

EXAMPLE 10-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

```
MOVWF   W_TEMP                  ; W_TEMP is in virtual bank
MOVFF   STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF   BSR, BSR_TEMP           ; BSR_TMEP located anywhere
;
; USER ISR CODE
;
MOVFF   BSR_TEMP, BSR           ; Restore BSR
MOVF    W_TEMP, W               ; Restore WREG
MOVFF   STATUS_TEMP, STATUS     ; Restore STATUS
```

TABLE 10-1: SUMMARY OF REGISTERS ASSOCIATED WITH INTERRUPTS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	RBIP
INTCON3	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
PIR1	PSPIP	ADIF	RC1IF	TX1IF	SSPIF	TMR1GIF	TMR2IF	TMR1IF
PIR2	OSCFIF	—	—	—	BCLIF	HLVDIF	TMR3IF	TMR3GIF
PIR3	—	—	RC2IF	TX2IF	CTMUIF	CCP2IF	CCP1IF	—
PIR4	TMR4IF	EEIF	CMP2IF	CMP1IF	—	CCP5IF	CCP4IF	CCP3IF
PIR5	IRXIF	WAKIF	ERRIF	TXB2IF	TXB1IF	TXB0IF	RXB1IF	RXB0IF
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	TMR1GIE	TMR2IE	TMR1IE
PIE2	OSCFIE	—	—	—	BCLIE	HLVDIE	TMR3IE	TMR3GIE
PIE3	—	—	RC2IE	TX2IE	CTMUIE	CCP2IE	CCP1IE	—
PIE4	TMR4IE	EEIE	CCP2IE	CMP1IE	—	CCP5IE	CCP4IE	CCP3IE
PIE5	IRXIE	WAKIE	ERRIE	TXB2IE	TXB1IE	TXB0IE	RXB1IE	RXB0IE
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	TMR1GIP	TMR2IP	TMR1IP
IPR2	OSCFIP	—	—	—	BCLIP	HLVDIP	TMR3IP	TMR3GIP
IPR3	—	—	RC2IP	TX2IP	CTMUIP	CCP2IP	CCP1IP	—
IPR4	TMR4IP	EEIP	CMP2IP	CMP1IP	—	CCP5IP	CCP4IP	CCP3IP
IPR5	IRXIP	WAKIP	ERRIP	TXB2IP	TXB1IP	TXB0IP	RXB1IP	RXB0IP
RCON	IPEN	SBOREN	$\overline{\text{CM}}$	$\overline{\text{RI}}$	$\overline{\text{TO}}$	$\overline{\text{PD}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$

Legend: Shaded cells are not used by the interrupts.

18.2 CTMU Operation

The CTMU works by using a fixed current source to charge a circuit. The type of circuit depends on the type of measurement being made.

In the case of charge measurement, the current is fixed and the amount of time the current is applied to the circuit is fixed. The amount of voltage read by the A/D becomes a measurement of the circuit's capacitance.

In the case of time measurement, the current, as well as the capacitance of the circuit, is fixed. In this case, the voltage read by the A/D is representative of the amount of time elapsed from the time the current source starts and stops charging the circuit.

If the CTMU is being used as a time delay, both capacitance and current source are fixed, as well as the voltage supplied to the comparator circuit. The delay of a signal is determined by the amount of time it takes the voltage to charge to the comparator threshold voltage.

18.2.1 THEORY OF OPERATION

The operation of the CTMU is based on the equation for charge:

$$I = C \bullet \frac{dV}{dT}$$

More simply, the amount of charge measured in coulombs in a circuit is defined as current in amperes (I) multiplied by the amount of time in seconds that the current flows (t). Charge is also defined as the capacitance in farads (C) multiplied by the voltage of the circuit (V). It follows that:

$$I \bullet t = C \bullet V$$

The CTMU module provides a constant, known current source. The A/D Converter is used to measure (V) in the equation, leaving two unknowns: capacitance (C) and time (t). The above equation can be used to calculate capacitance or time, by either the relationship using the known fixed capacitance of the circuit:

$$t = (C \bullet V)/I$$

or by:

$$C = (I \bullet t)/V$$

using a fixed time that the current source is applied to the circuit.

18.2.2 CURRENT SOURCE

At the heart of the CTMU is a precision current source, designed to provide a constant reference for measurements. The level of current is user-selectable across three ranges, or a total of two orders of magnitude, with the ability to trim the output in ±2% increments (nominal). The current range is selected by the IRNG<1:0> bits (CTMUICON<1:0>), with a value of '01' representing the lowest range.

Current trim is provided by the ITRIM<5:0> bits (CTMUICON<7:2>). These six bits allow trimming of the current source in steps of approximately 2% per step. Half of the range adjusts the current source positively and the other half reduces the current source. A value of '000000' is the neutral position (no change). A value of '100001' is the maximum negative adjustment (approximately -62%) and '011111' is the maximum positive adjustment (approximately +62%).

18.2.3 EDGE SELECTION AND CONTROL

CTMU measurements are controlled by edge events occurring on the module's two input channels. Each channel, referred to as Edge 1 and Edge 2, can be configured to receive input pulses from one of the edge input pins (CTED1 and CTED2) or CCPx Special Event Triggers (ECCP1 and CCP2). The input channels are level-sensitive, responding to the instantaneous level on the channel rather than a transition between levels. The inputs are selected using the EDG1SEL and EDG2SEL bit pairs (CTMUCONL<3:2>, 6:5>).

In addition to source, each channel can be configured for event polarity using the EDGE2POL and EDGE1POL bits (CTMUCONL<7,4>). The input channels can also be filtered for an edge event sequence (Edge 1 occurring before Edge 2) by setting the EDGSEQEN bit (CTMUCONH<2>).

18.2.4 EDGE STATUS

The CTMUCONL register also contains two status bits, EDG2STAT and EDG1STAT (CTMUCONL<1:0>). Their primary function is to show if an edge response has occurred on the corresponding channel. The CTMU automatically sets a particular bit when an edge response is detected on its channel. The level-sensitive nature of the input channels also means that the status bits become set immediately if the channel's configuration is changed and matches the channel's current state.

The module uses the edge status bits to control the current source output to external analog modules (such as the A/D Converter). Current is only supplied to external modules when only one (not both) of the status bits is set. Current is shut off when both bits are either set or cleared. This allows the CTMU to measure current only during the interval between edges. After both status bits are set, it is necessary to clear them before another measurement is taken. Both bits should be cleared simultaneously, if possible, to avoid re-enabling the CTMU current source.

In addition to being set by the CTMU hardware, the edge status bits can also be set by software. This permits a user application to manually enable or disable the current source. Setting either (but not both) of the bits enables the current source. Setting or clearing both bits at once disables the source.

REGISTER 20-2: CCPTMRS: CCP TIMER SELECT REGISTER

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	C5TSEL	C4TSEL	C3TSEL	C2TSEL	C1TSEL
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 **Unimplemented:** Read as '0'

bit 4 **C5TSEL:** CCP5 Timer Selection bit
0 = CCP5 is based off of TMR1/TMR2
1 = CCP5 is based off of TMR3/TMR4

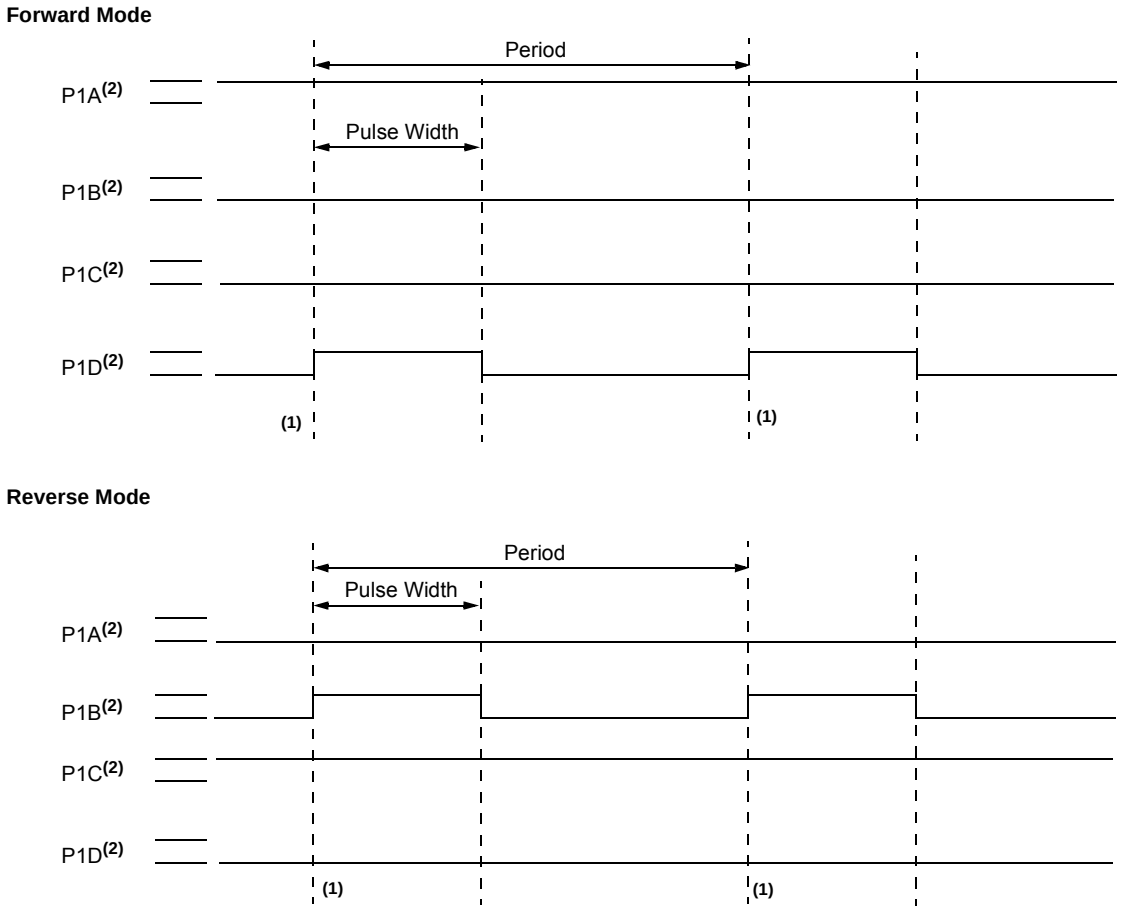
bit 3 **C4TSEL:** CCP4 Timer Selection bit
0 = CCP4 is based off of TMR1/TMR2
1 = CCP4 is based off of TMR3/TMR4

bit 2 **C3TSEL:** CCP3 Timer Selection bit
0 = CCP3 is based off of TMR1/TMR2
1 = CCP3 is based off of TMR3/TMR4

bit 1 **C2TSEL:** CCP2 Timer Selection bit
0 = CCP2 is based off of TMR1/TMR2
1 = CCP2 is based off of TMR3/TMR4

bit 0 **C1TSEL:** CCP1 Timer Selection bit
0 = ECCP1 is based off of TMR1/TMR2
1 = ECCP1 is based off of TMR3/TMR4

FIGURE 20-9: EXAMPLE OF FULL-BRIDGE PWM OUTPUT



Note 1: At this time, the TMR2 register is equal to the PR2 register.

2: The output signal is shown as active-high.

21.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPMx bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware if the TRIS bits are set.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit is set, or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I^2C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

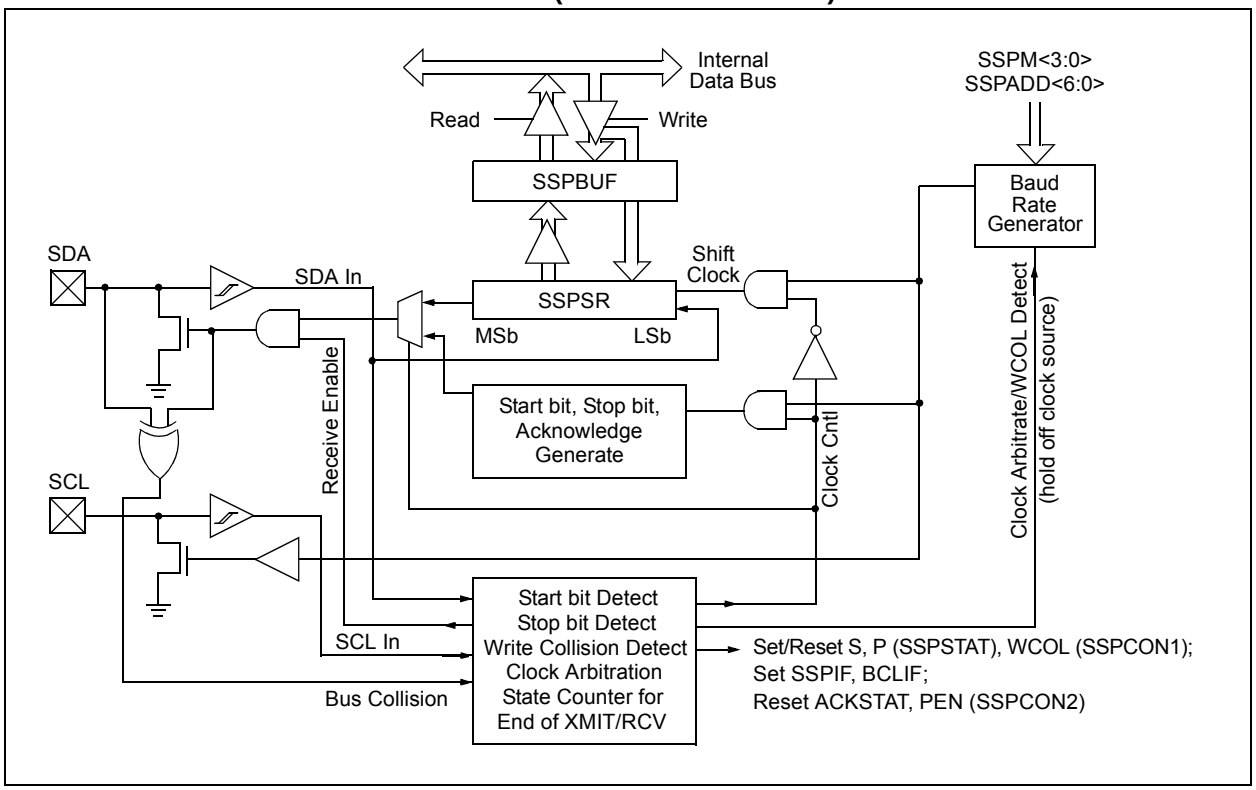
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I^2C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

Note:	The MSSP module, when configured in I^2C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

The following events will cause the MSSP Interrupt Flag bit, SSPIF, to be set (and MSSP interrupt, if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmitted
- Repeated Start

FIGURE 21-18: MSSP BLOCK DIAGRAM (I^2C™ MASTER MODE)



21.4.7 BAUD RATE

In I^2C Master mode, the Baud Rate Generator (BRG) reload value is placed in the 8 bits of the SSPADD register (Figure 21-19). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I^2C Master mode, the BRG is reloaded automatically.

Once the given operation is complete (i.e., transmission of the last data bit is followed by $\overline{ACK}$), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

Table 21-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD. The SSPADD BRG value of 00h is not supported.

FIGURE 21-19: BAUD RATE GENERATOR BLOCK DIAGRAM

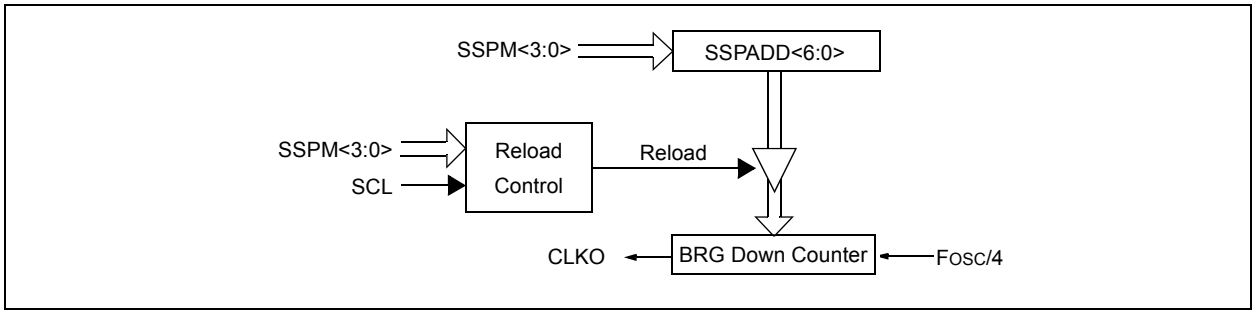


TABLE 21-3: I^2C™ CLOCK RATE w/BRG

FOSC	FCY	FCY * 2	BRG Value	FSCL (2 Rollovers of BRG)
40 MHz	10 MHz	20 MHz	18h	400 kHz(1)
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz(1)
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz(1)
4 MHz	1 MHz	2 MHz	09h	100 kHz
16 MHz(2)	4 MHz	8 MHz	03h	1 MHz(1)

Note 1: The I^2C interface does not conform to the 400 kHz I^2C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

2: A minimum 16-MHz FOSC is required for 1 MHz I^2C.

22.2.3 AUTO-BAUD RATE DETECT

The Enhanced USARTx modules support 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 22-1) 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 RXx signal, the RXx 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 Rate Detect must receive a byte with the value, 55h (ASCII "U", which is also the LIN/J2602 bus Sync character), in order to calculate the proper bit rate. The measurement is taken 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 SPBRGx begins counting up, using the preselected clock source on the first rising edge of RXx. After eight bits on the RXx pin or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGHx:SPBRGx register pair. Once the 5th edge is seen (this should correspond to the Stop bit), the ABDEN bit is automatically cleared.

If a rollover of the BRG occurs (an overflow from FFFFh to 0000h), the event is trapped by the ABDOVF status bit (BAUDCONx<7>). It is set in hardware by BRG rollovers and can be set or cleared by the user in software. ABD mode remains active after rollover events and the ABDEN bit remains set (Figure 22-2).

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. The BRG clock will be configured by the BRG16 and BRGH bits. The BRG16 bit must be set to use both SPBRG1 and SPBRGH1 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 SPBRGHx register. Refer to Table 22-5 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSARTx state machine is held in Idle. The RCxIF interrupt is set once the fifth rising edge on RXx is detected. The value in the RCREGx needs to be read to clear the RCxIF interrupt. The contents of RCREGx 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.

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 EUSARTx 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, if that feature is used it is recommended that the BRG16 bit (BAUDCONx<3>) be set.

TABLE 22-5: BRG COUNTER CLOCK RATES

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

22.2.3.1 ABD and EUSARTx Transmission

Since the BRG clock is reversed during ABD acquisition, the EUSARTx transmitter cannot be used during ABD. This means that whenever the ABDEN bit is set, TXREGx cannot be written to. Users should also ensure that ABDEN does not become set during a transmit sequence. Failing to do this may result in unpredictable EUSARTx operation.

23.8 Use of the Special Event Triggers

A/D conversion can be started by the Special Event Trigger of any of these modules:

- CCP2 – Requires CCP2M<3:0> bits (CCP2CON<3:0>) set at '1011'(†)
- ECCP1
- CTMU – Requires the setting of the CTTRIG bit (CTMUCONH<0>)
- Timer1

To start an A/D conversion:

- The A/D module must be enabled (ADON = 1)
- The appropriate analog input channel selected
- The minimum acquisition period set one of these ways:
  - Timing provided by the user
  - Selection made of an appropriate TACQ time

With these conditions met, the trigger sets the GO/$\overline{\text{DONE}}$ bit and the A/D acquisition starts.

If the A/D module is not enabled (ADON = 0), the module ignores the Special Event Trigger.

Note:	With an ECCP1 or CCP2 trigger, Timer1 or Timer3 is cleared. The timers reset to automatically repeat the A/D acquisition period with minimal software overhead (moving ADRESH:ADRESL to the desired location). If the A/D module is not enabled, the Special Event Trigger is ignored by the module, but the timer's counter resets.

23.9 Operation in Power-Managed Modes

The selection of the automatic acquisition time and A/D conversion clock is determined, in part, by the clock source and frequency while in a power-managed mode.

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT<2:0> and ADCS<2:0> bits in ADCON2 should be updated in accordance with the power-managed mode clock that will be used.

After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed. If desired, the device may be placed into the corresponding power-managed Idle mode during the conversion.

If the power-managed mode clock frequency is less than 1 MHz, the A/D RC clock source should be selected.

Operation in Sleep mode requires that the A/D RC clock be selected. If bits, ACQT<2:0>, are set to '000' and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry into Sleep mode. The IDLEN and SCS<1:0> bits in the OSCCON register must have already been cleared prior to starting the conversion.

REGISTER 28-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

U-0	R/P-1	U-0	R/P-1	R/P-1	R/P-1	U-0	R/P-1
—	XINST	—	SOSCSEL1	SOSCSEL0	INTOSCSEL	—	$\overline{\text{RETEN}}$
bit 7							bit 0

Legend:	P = Programmable bit		
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 **XINST:** Extended Instruction Set Enable bit

1 = Instruction set extension and Indexed Addressing mode are enabled
0 = Instruction set extension and Indexed Addressing mode are disabled (Legacy mode)

bit 5 **Unimplemented:** Read as ‘0’

bit 4-3 **SOSCSEL<1:0>:** SOSC Power Selection and Mode Configuration bits

11 = High-power SOSC circuit is selected
10 = Digital (SCLKI) mode; I/O port functionality of RC0 and RC1 is enabled
01 = Low-power SOSC circuit is selected
00 = Reserved

bit 2 **INTOSCSEL:** LF-INTOSC Low-power Enable bit

1 = LF-INTOSC in High-Power mode during Sleep
0 = LF-INTOSC in Low-Power mode during Sleep

bit 1 **Unimplemented:** Read as ‘0’

bit 0 **$\overline{\text{RETEN}}$:** VREG Sleep Enable bit

1 = Ultra low-power regulator is disabled. Regulator power in Sleep mode is controlled by REGSLP (WDTCON<7>).
0 = Ultra low-power regulator is enabled. Regulator power in Sleep mode is controlled by SRETEN (WDTCON<4>).

REGISTER 28-10: CONFIG6H: CONFIGURATION REGISTER 6 HIGH (BYTE ADDRESS 30000Bh)

R/C-1	R/C-1	R-1	U-0	U-0	U-0	U-0	U-0
WRTD	WRTB	WRTC(1)	—	—	—	—	—
bit 7							bit 0

Legend:	C = Clearable bit		
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 **WRTD:** Data EEPROM Write Protection bit
1 = Data EEPROM is not write-protected
0 = Data EEPROM is write-protected

bit 6 **WRTB:** Boot Block Write Protection bit
1 = Boot block is not write-protected(2)
0 = Boot block is write-protected(2)

bit 5 **WRTC:** Configuration Register Write Protection bit(1)
1 = Configuration registers are not write-protected(2)
0 = Configuration registers are write-protected(2)

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

Note 1: This bit is read-only in normal execution mode; it can be written only in Program mode.
2: For the memory size of the blocks, see Figure 28-6.

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.

RRNCF **Rotate Right f (No Carry)**

Syntax: 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

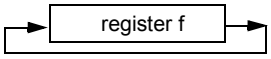
Encoding:

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' (default).

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.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



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, 0, 0
Before Instruction
    W       =   ?
    REG     =   1101 0111
After Instruction
    W       =   1110 1011
    REG     =   1101 0111
```

SETF **Set f**

Syntax: 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 is selected. If 'a' is '1', the BSR is used to select the GPR bank.

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever $f \le 95$ (5Fh). See **Section 29.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read register 'f'	Process Data	Write register 'f'

Example:
```
SETF    REG,1
Before Instruction
    REG     =   5Ah
After Instruction
    REG     =   FFh
```

30.11 PICkit 2 Development Programmer/Debugger and PICkit 2 Debug Express

The PICkit™ 2 Development Programmer/Debugger is a low-cost development tool with an easy to use interface for programming and debugging Microchip's Flash families of microcontrollers. The full featured Windows® programming interface supports baseline (PIC10F, PIC12F5xx, PIC16F5xx), midrange (PIC12F6xx, PIC16F), PIC18F, PIC24, dsPIC30, dsPIC33, and PIC32 families of 8-bit, 16-bit, and 32-bit microcontrollers, and many Microchip Serial EEPROM products. With Microchip's powerful MPLAB Integrated Development Environment (IDE) the PICkit™ 2 enables in-circuit debugging on most PIC® microcontrollers. In-Circuit-Debugging runs, halts and single steps the program while the PIC microcontroller is embedded in the application. When halted at a breakpoint, the file registers can be examined and modified.

The PICkit 2 Debug Express include the PICkit 2, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

30.12 MPLAB PM3 Device Programmer

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

30.13 Demonstration/Development Boards, Evaluation Kits, and Starter Kits

A wide variety of demonstration, development and evaluation boards for various PIC MCUs and dsPIC DSCs allows quick application development on fully functional systems. Most boards include prototyping areas for adding custom circuitry and provide application firmware and source code for examination and modification.

The boards support a variety of features, including LEDs, temperature sensors, switches, speakers, RS-232 interfaces, LCD displays, potentiometers and additional EEPROM memory.

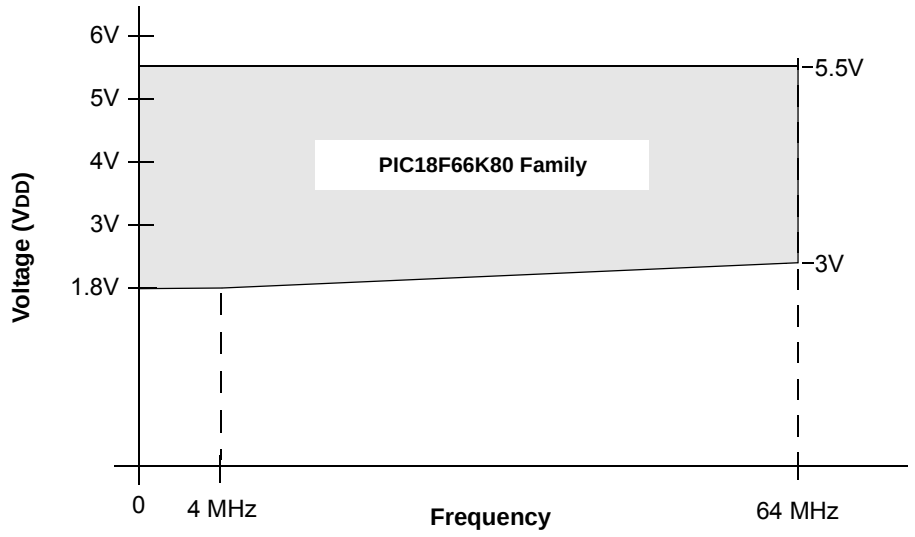
The demonstration and development boards can be used in teaching environments, for prototyping custom circuits and for learning about various microcontroller applications.

In addition to the PICDEM™ and dsPICDEM™ demonstration/development board series of circuits, Microchip has a line of evaluation kits and demonstration software for analog filter design, KEELOQ® security ICs, CAN, IrDA®, PowerSmart battery management, SEEVAL® evaluation system, Sigma-Delta ADC, flow rate sensing, plus many more.

Also available are starter kits that contain everything needed to experience the specified device. This usually includes a single application and debug capability, all on one board.

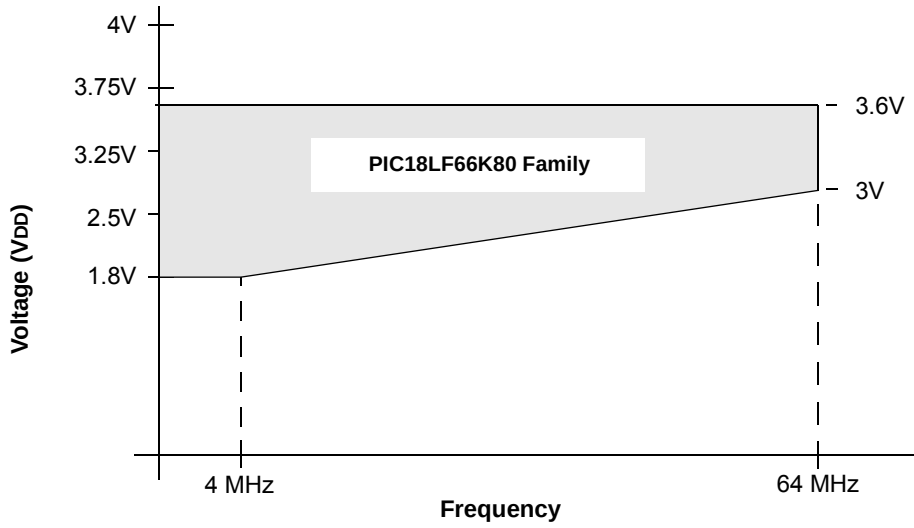
Check the Microchip web page (www.microchip.com) for the complete list of demonstration, development and evaluation kits.

FIGURE 31-1: VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL/EXTENDED)(1)



Note 1: For VDD values 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

FIGURE 31-2: VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL/EXTENDED)(1,2)



Note 1: When the on-chip voltage regulator is disabled, VDD must be maintained so that VDD ≤ 3.6V.
2: For VDD values 1.8V to 3V, FMAX = (VDD – 1.72)/0.02 MHz.

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	2	5	mA	-40°C to +125°C	VDD = 3.3V(4) Regulator Disabled	FOSC = 16 MHz (**PRI_RUN** mode, 4 MHz EC oscillator with PLL)
	PIC18**F**XXK80	2	5	mA	-40°C to +125°C	VDD = 3.3V(5) Regulator Enabled	
	PIC18**F**XXK80	2	6	mA	-40°C to +125°C	VDD = 5.5V(5) Regulator Enabled	
	PIC18L**F**XXK80	7	11	mA	-40°C to +125°C	VDD = 3.3V(4) Regulator Disabled	FOSC = 64 MHz (**PRI_RUN** mode, 16 MHz EC oscillator with PLL)
	PIC18**F**XXK80	7	11	mA	-40°C to +125°C	VDD = 3.3V(5) Regulator Enabled	
	PIC18**F**XXK80	8	12	mA	-40°C to +125°C	VDD = 5.5V(5) Regulator Enabled	
	PIC18L**F**XXK80	7	11	mA	-40°C to +125°C	VDD = 3.3V(4) Regulator Disabled	FOSC = 64 MHz (**PRI_RUN** mode, EC oscillator)
	PIC18**F**XXK80	7	11	mA	-40°C to +125°C	VDD = 3.3V(4) Regulator Enabled	
	PIC18**F**XXK80	8	12	mA	-40°C to +125°C	VDD = 5.5V(5) Regulator Enabled	
	PIC18L**F**XXK80	2.3	5	mA	-40°C to +125°C	VDD = 3.3V(4) Regulator Disabled	FOSC = 64 MHz (**PRI_IDLE** mode, EC oscillator)
	PIC18**F**XXK80	2.3	5	mA	-40°C to +125°C	VDD = 3.3V(5) Regulator Enabled	
	PIC18**F**XXK80	2.5	6	mA	-40°C to +125°C	VDD = 5.5V(5) Regulator Enabled	

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.

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	75	160	µA	-40°C	VDD = 1.8V(4) Regulator Disabled	FOSC = 4 MHz (**PRI_IDLE** mode, EC oscillator)
		75	160	µA	+25°C		
		75	160	µA	+60°C		
		76	170	µA	+85°C		
		82	180	µA	+125°C		
	PIC18**LF**XXK80	148	300	µA	-40°C	VDD = 3.3V(4) Regulator Disabled	
		148	300	µA	+25°C		
		148	300	µA	+60°C		
		150	400	µA	+85°C		
		157	460	µA	+125°C		
	PIC18**F**XXK80	187	320	µA	-40°C	VDD = 3.3V(5) Regulator Enabled	
		204	320	µA	+25°C		
		212	320	µA	+60°C		
		218	420	µA	+85°C		
		230	480	µA	+125°C		
	PIC18**F**XXK80	230	500	µA	-40°C	VDD = 5V(5) Regulator Enabled	
		230	500	µA	+25°C		
		230	500	µA	+60°C		
		240	600	µA	+85°C		
		250	700	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.

FIGURE 31-13: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)

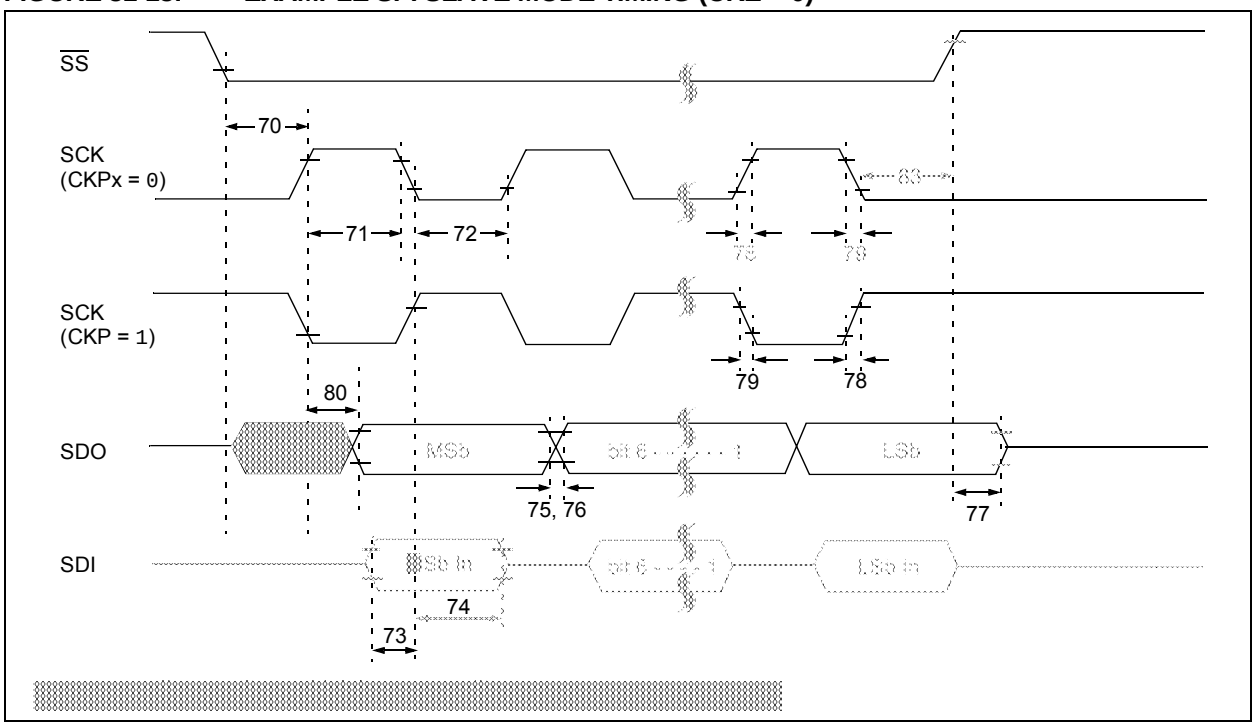


TABLE 31-17: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
70	TssL2scH, TssL2scL	$\overline{SS}$ ↓ to SCK ↓ or SCK ↑ Input		3 Tcy	—	ns	
70A	TssL2WB	$\overline{SS}$ to write to SSPBUF		3 Tcy	—	ns	
71	TscH	SCK Input High Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
71A			Single Byte	40	—	ns	**(Note 1)**
72	TscL	SCK Input Low Time (Slave mode)	Continuous	1.25 Tcy + 30	—	ns	
72A			Single Byte	40	—	ns	**(Note 1)**
73	TdiV2scH, TdiV2scL	Setup Time of SDI Data Input to SCK Edge		20	—	ns	
73A	Tb2b	Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2		1.5 Tcy + 40	—	ns	**(Note 2)**
74	TscH2diL, TscL2diL	Hold Time of SDI Data Input to SCK Edge		40	—	ns	
75	TdoR	SDO Data Output Rise Time		—	25	ns	
76	TdoF	SDO Data Output Fall Time		—	25	ns	
77	TssH2doZ	$\overline{SS}$ ↑ to SDO Output High-impedance		10	50	ns	
78	TscR	SCK Output Rise Time (Master mode)		—	25	ns	
79	TscF	SCK Output Fall Time (Master mode)		—	25	ns	
80	TscH2doV, TscL2doV	SDO Data Output Valid after SCK Edge		—	50	ns	
83	TscH2ssH, TscL2ssH	$\overline{SS}$ ↑ after SCK Edge		1.5 Tcy + 40	—	ns	

Note 1: Requires the use of Parameter 73A.

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