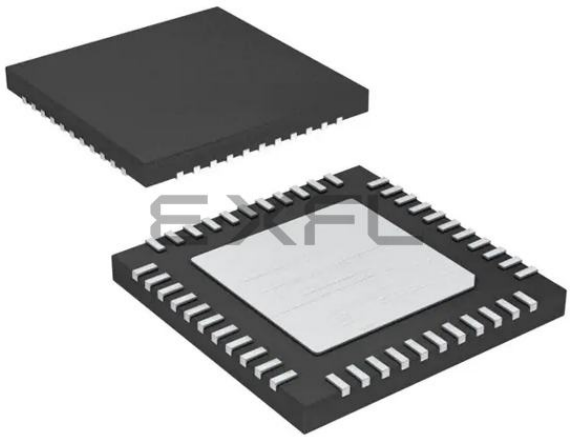


Welcome to E-XFL.COM

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	64KB (32K 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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	44-VQFN Exposed Pad
Supplier Device Package	44-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18f46k80-e-ml

3.8 Effects of Power-Managed Modes on the Various Clock Sources

When PRI_IDLE mode is selected, the designated primary oscillator continues to run without interruption. For all other power-managed modes, the oscillator using the OSC1 pin is disabled. The OSC1 pin (and OSC2 pin if used by the oscillator) will stop oscillating.

In secondary clock modes (SEC_RUN and SEC_IDLE), the SOSC oscillator is operating and providing the device clock. The SOSC oscillator may also run in all power-managed modes if required to clock SOSC.

In RC_RUN and RC_IDLE modes, the internal oscillator provides the device clock source. The 31 kHz LF-INTOSC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 28.2 "Watchdog Timer (WDT)"** through **Section 28.5 "Fail-Safe Clock Monitor"** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up).

If the Sleep mode is selected, all clock sources are stopped. Since all the transistor switching currents have been stopped, Sleep mode achieves the lowest current consumption of the device (only leakage currents).

Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The INTOSC is required to support WDT operation. The SOSC oscillator may be operating to support Timer1 or 3. Other features may be operating that do not require a device clock source (i.e., MSSP slave, INTx pins and others). Peripherals that may add significant current consumption are listed in **Section 31.2 "DC Characteristics: Power-Down and Supply Current PIC18F66K80 Family (Industrial/Extended)"**.

3.9 Power-up Delays

Power-up delays are controlled by two timers, so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 5.6.1 "Power-up Timer (PWRT)"**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up time of about 64 ms (Parameter 33, Table 31-11); it is always enabled.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (HS, XT or LP modes). The OST does this by counting 1,024 oscillator cycles before allowing the oscillator to clock the device.

There is a delay of interval, T_{CSD} (Parameter 38, Table 31-11), following POR, while the controller becomes ready to execute instructions.

TABLE 3-4: OSC1 AND OSC2 PIN STATES IN SLEEP MODE

Oscillator Mode	OSC1 Pin	OSC2 Pin
EC, ECPLL	Floating, pulled by external clock	At logic low (clock/4 output)
HS, HSPLL	Feedback inverter disabled at quiescent voltage level	Feedback inverter disabled at quiescent voltage level
INTOSC, INTPLL1/2	I/O pin, RA6, direction controlled by TRISA<6>	I/O pin, RA6, direction controlled by TRISA<7>

Note: See **Section 5.0 "Reset"** for time-outs due to Sleep and $\overline{\text{MCLR}}$ Reset.

5.4 Brown-out Reset (BOR)

The PIC18F66K80 family has four BOR Power modes:

- High-Power BOR
- Medium Power BOR
- Low-Power BOR
- Zero-Power BOR

Each power mode is selected by the BORPWR<1:0> setting (CONFIG2L<6:5>). For low, medium and high-power BOR, the module monitors the VDD depending on the BORV<1:0> setting (CONFIG1L<3:2>). The typical current draw (ΔIBOR) for zero, low and medium power BOR is 200 nA, 750 nA and 3 μA, respectively. A BOR event re-arms the Power-on Reset. It also causes a Reset, depending on which of the trip levels has been set: 1.8V, 2V, 2.7V or 3V.

BOR is enabled by BOREN<1:0> (CONFIG2L<2:1>) and the SBOREN bit (RCON<6>). The four BOR configurations are summarized in Table 5-1.

In Zero-Power BOR (ZPBORMV), the module monitors the VDD voltage and re-arms the POR at about 2V. ZPBORMV does not cause a Reset, but re-arms the POR.

The BOR accuracy varies with its power level. The lower the power setting, the less accurate the BOR trip levels are. Therefore, the high-power BOR has the highest accuracy and the low-power BOR has the lowest accuracy. The trip levels (BVDD, Parameter D005), current consumption (**Section 31.2 "DC Characteristics: Power-Down and Supply Current PIC18F66K80 Family (Industrial/Extended)"**) and time required below BVDD (TBOR, Parameter 35) can all be found in **Section 31.0 "Electrical Characteristics"**.

5.4.1 SOFTWARE ENABLED BOR

When BOREN<1:0> = 01, the BOR can be enabled or disabled by the user in software. This is done with the control bit, SBOREN (RCON<6>). Setting SBOREN enables the BOR to function as previously described. Clearing SBOREN disables the BOR entirely. The SBOREN bit operates only in this mode; otherwise it is read as '0'.

Placing the BOR under software control gives the user the additional flexibility of tailoring the application to its environment without having to reprogram the device to change BOR configuration. It also allows the user to tailor device power consumption in software by eliminating the incremental current that the BOR consumes. While the BOR current is typically very small, it may have some impact in low-power applications.

Note:	Even when BOR is under software control, the Brown-out Reset voltage level is still set by the BORV<1:0> Configuration bits; it cannot be changed in software.

5.4.2 DETECTING BOR

When Brown-out Reset is enabled, the $\overline{BOR}$ bit always resets to '0' on any Brown-out Reset or Power-on Reset event. This makes it difficult to determine if a Brown-out Reset event has occurred just by reading the state of $\overline{BOR}$ alone. A more reliable method is to simultaneously check the state of both $\overline{POR}$ and $\overline{BOR}$. This assumes that the $\overline{POR}$ bit is reset to '1' in software immediately after any Power-on Reset event. IF $\overline{BOR}$ is '0' while $\overline{POR}$ is '1', it can be reliably assumed that a Brown-out Reset event has occurred.

5.4.3 DISABLING BOR IN SLEEP MODE

When BOREN<1:0> = 10, the BOR remains under hardware control and operates as previously described. Whenever the device enters Sleep mode, however, the BOR is automatically disabled. When the device returns to any other operating mode, BOR is automatically re-enabled.

This mode allows for applications to recover from brown-out situations, while actively executing code, when the device requires BOR protection the most. At the same time, it saves additional power in Sleep mode by eliminating the small incremental BOR current.

TABLE 5-1: BOR CONFIGURATIONS

BOR Configuration		Status of SBOREN (RCON<6>)	BOR Operation
BOREN1	BOREN0		
0	0	Unavailable	BOR is disabled; must be enabled by reprogramming the Configuration bits.
0	1	Available	BOR is enabled in software; operation is controlled by SBOREN.
1	0	Unavailable	BOR is enabled in hardware, in Run and Idle modes; disabled during Sleep mode.
1	1	Unavailable	BOR is enabled in hardware; must be disabled by reprogramming the Configuration bits.

6.1.3.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit (CONFIG4L<0>). When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

6.1.4 FAST REGISTER STACK

A Fast Register Stack is provided for the STATUS, WREG and BSR registers to provide a “fast return” option for interrupts. This stack is only one level deep and is neither readable nor writable. It is loaded with the current value of the corresponding register when the processor vectors for an interrupt. All interrupt sources will push values into the Stack registers. The values in the registers are then loaded back into the working registers if the RETFIE, FAST instruction is used to return from the interrupt.

If both low and high-priority interrupts are enabled, the Stack registers cannot be used reliably to return from low-priority interrupts. If a high-priority interrupt occurs while servicing a low-priority interrupt, the Stack register values stored by the low-priority interrupt will be overwritten. In these cases, users must save the key registers in software during a low-priority interrupt.

If interrupt priority is not used, all interrupts may use the Fast Register Stack for returns from interrupt. If no interrupts are used, the Fast Register Stack can be used to restore the STATUS, WREG and BSR registers at the end of a subroutine call. To use the Fast Register Stack for a subroutine call, a CALL label, FAST instruction must be executed to save the STATUS, WREG and BSR registers to the Fast Register Stack. A RETURN, FAST instruction is then executed to restore these registers from the Fast Register Stack.

Example 6-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

EXAMPLE 6-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST    ;STATUS, WREG, BSR
                   ;SAVED IN FAST REGISTER
                   ;STACK
       •
       •

SUB1   •
       •
    RETURN FAST    ;RESTORE VALUES SAVED
                   ;IN FAST REGISTER STACK
```

6.1.5 LOOK-UP TABLES IN PROGRAM MEMORY

There may be programming situations that require the creation of data structures, or look-up tables, in program memory. For PIC18 devices, look-up tables can be implemented in two ways:

- Computed GOTO
- Table Reads

6.1.5.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the Program Counter. An example is shown in Example 6-2.

A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW nn instructions. The W register is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW nn instructions that returns the value, ‘nn’, to the calling function.

The offset value (in WREG) specifies the number of bytes that the Program Counter should advance and should be multiples of two (LSb = 0).

In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

EXAMPLE 6-2: COMPUTED GOTO USING AN OFFSET VALUE

```
        MOVF   OFFSET, W
        CALL   TABLE
ORG     nn00h
TABLE   ADDWF  PCL
        RETLW  nnh
        RETLW  nnh
        RETLW  nnh
        .
        .
        .
```

6.1.5.2 Table Reads

A better method of storing data in program memory allows two bytes of data to be stored in each instruction location.

Look-up table data may be stored two bytes per program word while programming. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from the program memory. Data is transferred from program memory one byte at a time.

The table read operation is discussed further in **Section 7.1 “Table Reads and Table Writes”**.

NOTES:

REGISTER 10-3: INTCON3: INTERRUPT CONTROL REGISTER 3

R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF
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 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 2 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

bit 1 **INT2IF:** INT2 External Interrupt Flag bit
1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows for software polling.

TABLE 11-1: PORTA FUNCTIONS

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RA0/CVREF/AN0/ ULPWU	RA0	0	O	DIG	LATA<0> data output; not affected by analog input.
		1	I	ST	PORTA<0> data input; disabled when analog input is enabled.
	CVREF	x	O	ANA	Comparator voltage reference output. Enabling this feature disables digital I/O.
	AN0	1	I	ANA	A/D Input Channel 0. Default input configuration on POR; does not affect digital output.
	ULPWU	1	O	DIG	Ultra Low-Power Wake-up input.
RA1/AN1/C1INC	RA1	0	O	DIG	LATA<1> data output; not affected by analog input.
		1	I	ST	PORTA<1> data input; disabled when analog input is enabled.
	AN1	1	I	ANA	A/D Input Channel 1. Default input configuration on POR; does not affect digital output.
	C1INC(1)	x	I	ANA	Comparator 1 Input C.
RA2/VREF-/AN2/ C2INC	RA2	0	O	DIG	LATA<2> data output; not affected by analog input.
		1	I	ST	PORTA<2> data input; disabled when analog functions are enabled.
	VREF-	1	I	ANA	A/D and comparator low reference voltage input.
	AN2	1	I	ANA	A/D Input Channel 2. Default input configuration on POR.
	C2INC(1)	x	I	ANA	Comparator 2 Input C.
RA3/VREF+/AN3	RA3	0	O	DIG	LATA<3> data output; not affected by analog input.
		1	I	ST	PORTA<3> data input; disabled when analog input is enabled.
	VREF+	1	I	ANA	A/D Input Channel 3. Default input configuration on POR.
	AN3	1	I	ANA	A/D and comparator high reference voltage input.
RA5/AN4/C2INB/ HLVDIN/T1CKI/ $\overline{SS}$/CTMUI	RA5	0	O	DIG	LATA<5> data output; not affected by analog input.
		1	I	ST	PORTA<5> data input; disabled when analog input is enabled.
	AN4	1	I	ANA	A/D Input Channel 4. Default configuration on POR.
	C2INB(2)	1	I	ANA	Comparator 2 Input B.
	HLVDIN	1	I	ANA	High/Low-Voltage Detect external trip point input.
	T1CKI	x	I	ST	Timer1 clock input.
	$\overline{SS}$	1	I	ST	Slave select input for MSSP module.
	CTMUI(2)	x	O	—	CTMU pulse generator charger for the C2INB comparator input.
RA6/OSC2/ CLKOUT	RA6	0	O	DIG	LATA<6> data output; disabled when FOSC2 Configuration bit is set.
		1	I	ST	PORTA<6> data input; disabled when FOSC2 Configuration bit is set.
	OSC2	x	O	ANA	Main oscillator feedback output connection (HS, XT and LP modes).
	CLKOUT	x	O	DIG	System cycle clock output (FOSC/4) (EC and INTOSC modes).
RA7/OSC1/CLKIN	RA7	0	O	DIG	LATA<7> data output; disabled when FOSC2 Configuration bit is set.
		1	I	ST	PORTA<7> data input; disabled when FOSC2 Configuration bit is set.
	OSC1	x	I	ANA	Main oscillator input connection (HS, XT, and LP modes).
	CLKIN	x	I	ANA	Main external clock source input (EC modes).

Legend: O = Output; I = Input; ANA = Analog Signal; DIG = CMOS Output; ST = Schmitt Trigger Buffer Input;
x = Don't care (TRIS bit does not affect port direction or is overridden for this option)

Note 1: This pin assignment is unavailable for 28-pin devices (PIC18F2XK80).

2: This pin assignment is only available for 28-pin devices (PIC18F2XK80).

TABLE 11-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	RA7(1)	RA6(1)	RA5	—	RA3	RA2	RA1	RA0
LATA	LATA7(1)	LATA6(1)	LATA5	—	LATA3	LATA2	LATA1	LATA0
TRISA	TRISA7(1)	TRISA6(1)	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0
ANCON0	ANSEL7	ANSEL6	ANSEL5	ANSEL4	ANSEL3	ANSEL2	ANSEL1	ANSEL0

Legend: — = unimplemented, read as '0'. Shaded cells are not used by PORTA.

Note 1: These bits are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as 'x'.

16.5 Timer3 Gates

Timer3 can be configured to count freely or the count can be enabled and disabled using the Timer3 gate circuitry. This is also referred to as the Timer3 gate count enable.

The Timer3 gate can also be driven by multiple selectable sources.

16.5.1 TIMER3 GATE COUNT ENABLE

The Timer3 Gate Enable mode is enabled by setting the TMR3GE bit (TxGCON<7>). The polarity of the Timer3 Gate Enable mode is configured using the T3GPOL bit (T3GCON<6>).

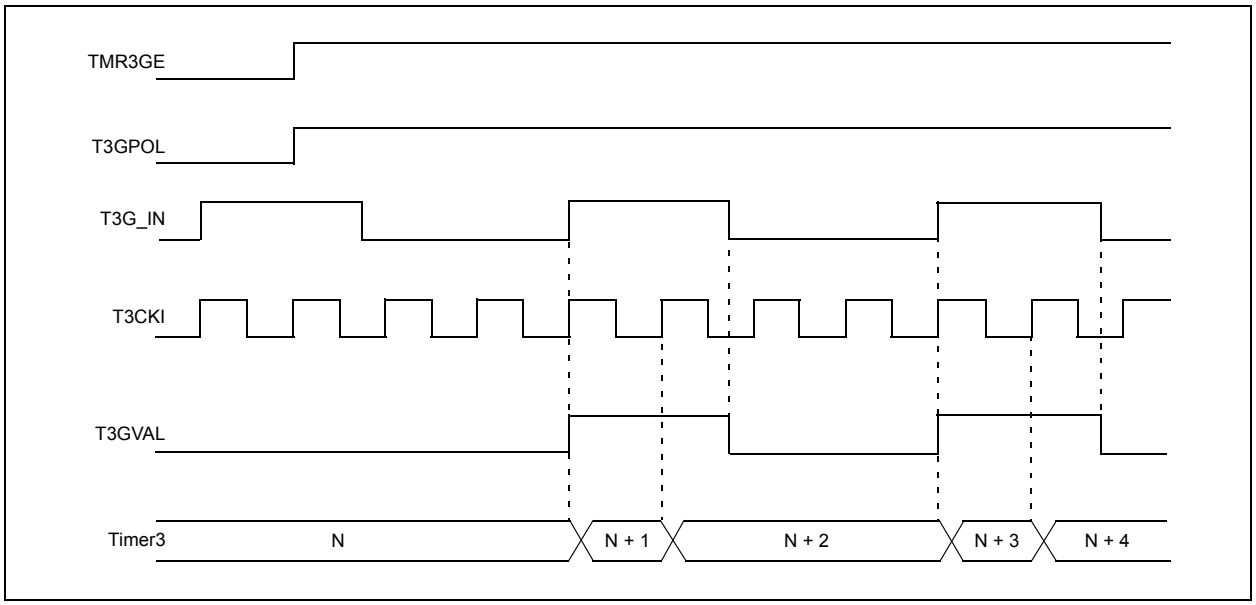
When Timer3 Gate Enable mode is enabled, Timer3 will increment on the rising edge of the Timer3 clock source. When Timer3 Gate Enable mode is disabled, no incrementing will occur and Timer3 will hold the current count. See Figure 16-2 for timing details.

TABLE 16-1: TIMER3 GATE ENABLE SELECTIONS

T3CLK(†)	T3GPOL (T3GCON<6>)	T3G Pin	Timer3 Operation
↑	0	0	Counts
↑	0	1	Holds Count
↑	1	0	Holds Count
↑	1	1	Counts

† The clock on which TMR3 is running. For more information, see T3CLK in Figure 16-1.

FIGURE 16-2: TIMER3 GATE COUNT ENABLE MODE



18.2.5 INTERRUPTS

The CTMU sets its interrupt flag (PIR3<3>) whenever the current source is enabled, then disabled. An interrupt is generated only if the corresponding interrupt enable bit (PIE3<3>) is also set. If edge sequencing is not enabled (i.e., Edge 1 must occur before Edge 2), it is necessary to monitor the edge status bits and determine which edge occurred last and caused the interrupt.

18.3 CTMU Module Initialization

The following sequence is a general guideline used to initialize the CTMU module:

1. Select the current source range using the IRNGx bits (CTMUICON<1:0>).
2. Adjust the current source trim using the ITRIMx bits (CTMUICON<7:2>).
3. Configure the edge input sources for Edge 1 and Edge 2 by setting the EDG1SEL and EDG2SEL bits (CTMUCONL<3:2> and <6:5>, respectively).
4. Configure the input polarities for the edge inputs using the EDG1POL and EDG2POL bits (CTMUCONL<4,7>).

   The default configuration is for negative edge polarity (high-to-low transitions).
5. Enable edge sequencing using the EDGSEQEN bit (CTMUCONH<2>).

   By default, edge sequencing is disabled.
6. Select the operating mode (Measurement or Time Delay) with the TGEN bit (CTMUCONH<4>).

   The default mode is Time/Capacitance Measurement.
7. Configure the module to automatically trigger an A/D conversion when the second edge event has occurred using the CTTRIG bit (CTMUCONH<0>).

   The conversion trigger is disabled by default.
8. Discharge the connected circuit by setting the IDISSEN bit (CTMUCONH<1>).
9. After waiting a sufficient time for the circuit to discharge, clear the IDISSEN bit.
10. Disable the module by clearing the CTMUEN bit (CTMUCONH<7>).
11. Clear the Edge Status bits, EDG2STAT and EDG1STAT (CTMUCONL<1:0>).

    Both bits should be cleared simultaneously, if possible, to avoid re-enabling the CTMU current source.
12. Enable both edge inputs by setting the EDGEN bit (CTMUCONH<3>).
13. Enable the module by setting the CTMUEN bit.

Depending on the type of measurement or pulse generation being performed, one or more additional modules may also need to be initialized and configured with the CTMU module:

- Edge Source Generation: In addition to the external edge input pins, ECCP1/CCP2 Special Event Triggers can be used as edge sources for the CTMU.
- Capacitance or Time Measurement: The CTMU module uses the A/D Converter to measure the voltage across a capacitor that is connected to one of the analog input channels.
- Pulse Generation: When generating system clock independent, output pulses, the CTMU module uses Comparator 2 and the associated comparator voltage reference.

18.4 Calibrating the CTMU Module

The CTMU requires calibration for precise measurements of capacitance and time, as well as for accurate time delay. If the application only requires measurement of a relative change in capacitance or time, calibration is usually not necessary. An example of a less precise application is a capacitive touch switch, in which the touch circuit has a baseline capacitance and the added capacitance of the human body changes the overall capacitance of a circuit.

If actual capacitance or time measurement is required, two hardware calibrations must take place:

- The current source needs calibration to set it to a precise current.
- The circuit being measured needs calibration to measure or nullify any capacitance other than that to be measured.

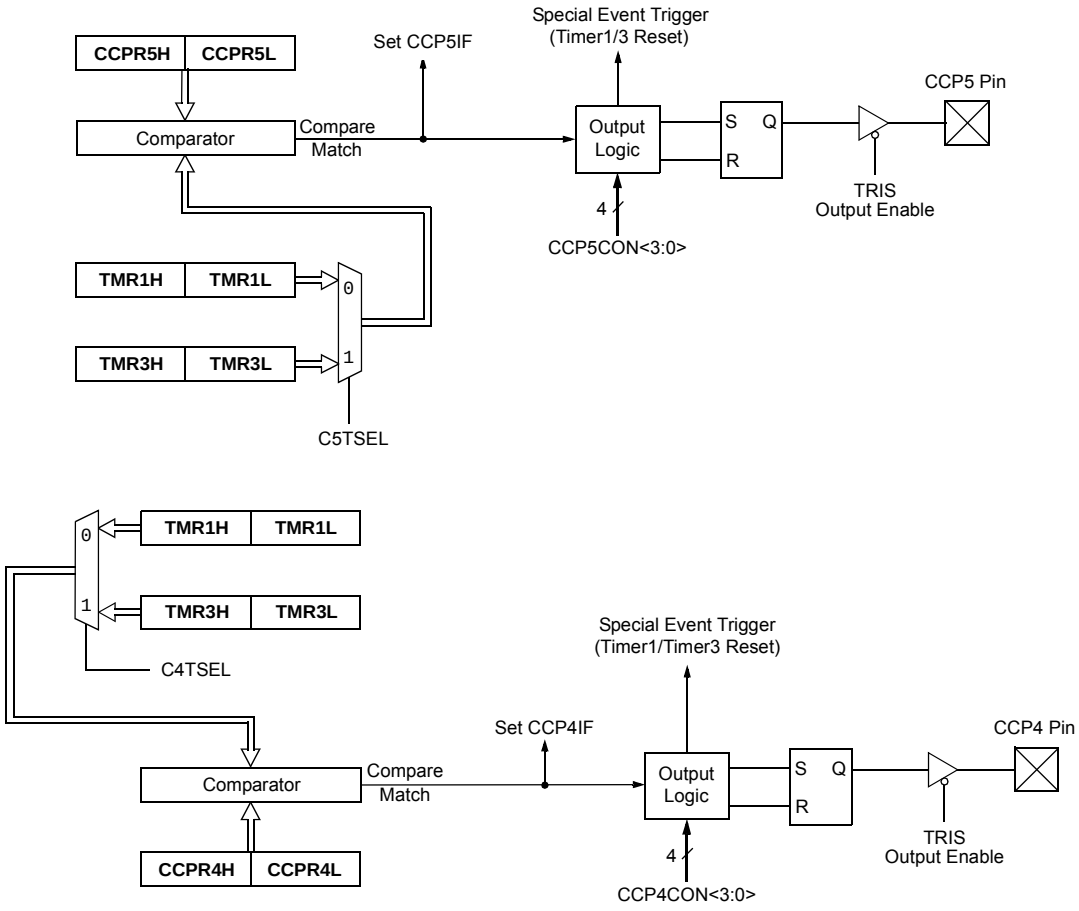
18.4.1 CURRENT SOURCE CALIBRATION

The current source on board the CTMU module has a range of ±62% nominal for each of three current ranges. For precise measurements, it is possible to measure and adjust this current source by placing a high-precision resistor, RCAL, onto an unused analog channel. An example circuit is shown in Figure 18-2.

To measure the current source:

1. Initialize the A/D Converter.
2. Initialize the CTMU.
3. Enable the current source by setting EDG1STAT (CTMUCONL<0>).
4. Issue time delay for voltage across RCAL to stabilize and A/D sample/hold capacitor to charge.
5. Perform the A/D conversion.
6. Calculate the current source current using $I = V/R_{CAL}$, where RCAL is a high-precision resistance and V is measured by performing an A/D conversion.

FIGURE 19-2: COMPARE MODE OPERATION BLOCK DIAGRAM



Note: This block diagram uses CCP4 and CCP5, and their appropriate timers as an example. For details on all of the CCP modules and their timer assignments, see Table 19-2.

21.3.9 OPERATION IN POWER-MANAGED MODES

In SPI Master mode, module clocks may be operating at a different speed than when in full-power mode; in the case of the Sleep mode, all clocks are halted.

In Idle modes, a clock is provided to the peripherals. That clock can be from the primary clock source, the secondary clock (SOSC oscillator) or the INTOSC source. See **Section 3.3 "Clock Sources and Oscillator Switching"** for additional information.

In most cases, the speed that the master clocks SPI data is not important; however, this should be evaluated for each system.

If MSSP interrupt is enabled, it can wake the controller from Sleep mode, or one of the Idle modes, when the master completes sending data. If an exit from Sleep or Idle mode is not desired, MSSP interrupts should be disabled.

If the Sleep mode is selected, all module clocks are halted and the transmission/reception will remain in that state until the device wakes. After the device returns to Run mode, the module will resume transmitting and receiving data.

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set, and if enabled, will wake the device.

21.3.10 EFFECTS OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

21.3.11 BUS MODE COMPATIBILITY

Table 21-1 shows the compatibility between the standard SPI modes, and the states of the CKP and CKE control bits.

TABLE 21-1: SPI BUS MODES

Standard SPI Mode Terminology	Control Bits State	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

There is also an SMP bit which controls when the data is sampled.

TABLE 21-2: REGISTERS ASSOCIATED WITH SPI OPERATION

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
PIR1	PSPIF	ADIF	RC1IF	TX1IF	SSPIF	TMR1GIF	TMR2IF	TMR1IF
PIE1	PSPIE	ADIE	RC1IE	TX1IE	SSPIE	TMR1GIE	TMR2IE	TMR1IE
IPR1	PSPIP	ADIP	RC1IP	TX1IP	SSPIP	TMR1GIP	TMR2IP	TMR1IP
TRISA	TRISA7	TRISA6	TRISA5	—	TRISA3	TRISA2	TRISA1	TRISA0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
SSPBUF	MSSP Receive Buffer/Transmit Register							
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
ODCON	SSPOD	CCP5OD	CCP4OD	CCP3OD	CCP2OD	CCP1OD	U2OD	U1OD
PMD0	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	UART2MD	UART1MD	SSPMD

Legend: Shaded cells are not used by the MSSP module in SPI 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

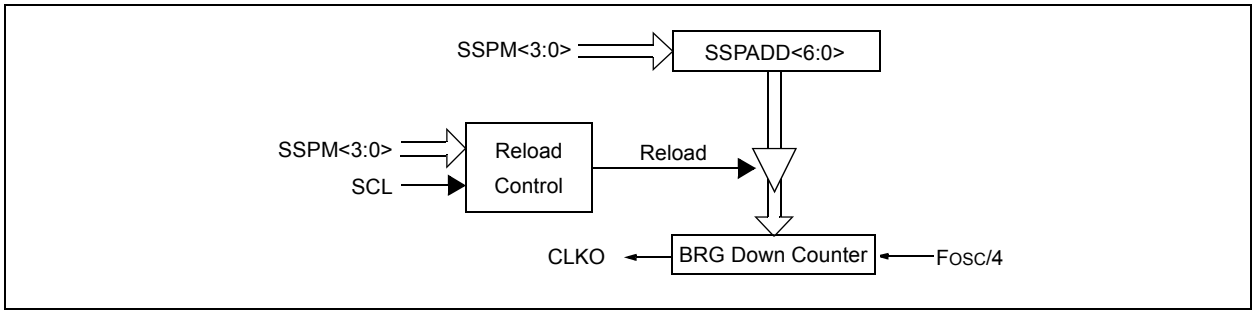


TABLE 21-3: I^2C™ CLOCK RATE w/BRG

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.

22.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

The Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) module is one of two serial I/O modules. (Generically, the EUSART is also known as a Serial Communications Interface or SCI.)

The EUSART can be configured as a full-duplex, asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

The Enhanced USARTx modules implement additional features, including automatic baud rate detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These make it ideally suited for use in Local Interconnect Network bus (LIN/J2602 bus) systems.

All members of the PIC18F66K80 family are equipped with two independent EUSART modules, referred to as EUSART1 and EUSART2. They can be configured in the following modes:

- Asynchronous (full duplex) with:
  - Auto-wake-up on character reception
  - Auto-baud calibration
  - 12-bit Break character transmission
- Synchronous – Master (half duplex) with selectable clock polarity
- Synchronous – Slave (half duplex) with selectable clock polarity

The pins of EUSART1 and EUSART2 are multiplexed with the functions with the following ports, depending on the device pin count. See Table 22-1.

TABLE 22-1: CONFIGURING EUSARTx PINS(1)

Pin Count	EUSART1		EUSART2	
	Port	Pins	Port	Pins
28-pin	PORTC	RC6/TX1/CK1 and RC7/RX1/DT1	PORTB	RB6/PGC/TX2/CK2/KBI2 and RB7/PGD/T3G/RX2/DT2/KBI3
40/44-pin	PORTC	RC6/TX1/CK1 and RC7/RX1/DT1	PORTD	RD6/TX2/CK2/P1C/PSP6 and RD7/RX2/DT2/P1D/PSP7
64-pin	PORTG	RG3/TX1/CK1 and RG0/RX1/DT1	PORTE	RE7/TX2/CK2 and RE6/RX2/DT2

Note 1: The EUSARTx control will automatically reconfigure the pin from input to output as needed.

In order to configure the pins as an EUSARTx:

- For EUSART1:
  - SPEN (RCSTA1<7>) must be set (= 1)
  - TRISx<x> must be set (= 1)
  - For Asynchronous and Synchronous Master modes, TRISx<x> must be cleared (= 0)
  - For Synchronous Slave mode, TRISx<x> must be set (= 1)
- For EUSART2:
  - SPEN (RCSTA2<7>) must be set (= 1)
  - TRISx<x> must be set (= 1)
  - For Asynchronous and Synchronous Master modes, TRISx<x> must be cleared (= 0)
  - For Synchronous Slave mode, TRISx<x> must be set (= 1)

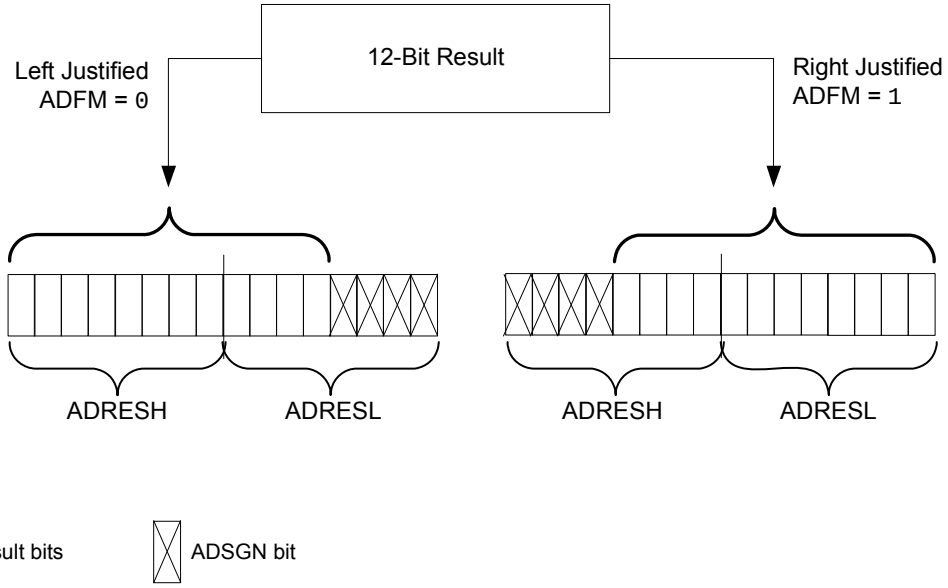
23.2.2 A/D RESULT REGISTERS

The ADRESH:ADRESL register pair is where the 12-bit A/D result and extended sign bits (ADSGNx) are loaded at the completion of a conversion. This register pair is 16 bits wide. The A/D module gives the flexibility of left or right justifying the 12-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification.

Figure 23-3 shows the operation of the A/D result justification and the location of the sign bit (ADSGNx). The extended sign bits allow for easier 16-bit math to be performed on the result. The results are represented as a two's compliment binary value. This means that when sign bits and magnitude bits are considered together in right justification, the ADRESH and ADRESL registers can be read as a single signed integer value.

When the A/D Converter is disabled, these 8-bit registers can be used as two general purpose registers.

FIGURE 23-3: A/D RESULT JUSTIFICATION



Two's Complement Example Results Number Line

Left Justified		Right Justified	
Hex	**Decimal**	**Hex**	**Decimal**
0xFFF0	4095	0x0FFF	4095
0xFFE0	4094	0x0FFE	4094
...	...	...	...
0x0020	2	0x0002	2
0x0010	1	0x0001	1
0x0000	0	0x0000	0
0xFFFF	-1	0xFFFF	-1
0xFFEF	-2	0xFFFE	-2
...	...	...	...
0x001F	-4095	0xF001	-4095
0x000F	-4096	0xF000	-4096

REGISTER 27-19: RXBnDLC: RECEIVE BUFFER 'n' DATA LENGTH CODE REGISTERS [0 ≤ n ≤ 1]

U-0	R-x	R-x	R-x	R-x	R-x	R-x	R-x
—	RXRTR	RB1	R0	DLC3	DLC2	DLC1	DLC0
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 **RXRTR:** Receiver Remote Transmission Request bit
1 = Remote transfer request
0 = No remote transfer request

bit 5 **RB1:** Reserved bit 1
Reserved by CAN Spec and read as '0'.

bit 4 **RB0:** Reserved bit 0
Reserved by CAN Spec and read as '0'.

bit 3-0 **DLC<3:0>:** Data Length Code bits
1111 = Invalid
1110 = Invalid
1101 = Invalid
1100 = Invalid
1011 = Invalid
1010 = Invalid
1001 = Invalid
1000 = Data length = 8 bytes
0111 = Data length = 7 bytes
0110 = Data length = 6 bytes
0101 = Data length = 5 bytes
0100 = Data length = 4 bytes
0011 = Data length = 3 bytes
0010 = Data length = 2 bytes
0001 = Data length = 1 byte
0000 = Data length = 0 bytes

REGISTER 27-20: RXBnDm: RECEIVE BUFFER 'n' DATA FIELD BYTE 'm' REGISTERS [0 ≤ n ≤ 1, 0 ≤ m ≤ 7]

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RXBnDm7	RXBnDm6	RXBnDm5	RXBnDm4	RXBnDm3	RXBnDm2	RXBnDm1	RXBnDm0
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 **RXBnDm<7:0>:** Receive Buffer n Data Field Byte m bits (where 0 ≤ n < 1 and 0 < m < 7)
Each receive buffer has an array of registers. For example, Receive Buffer 0 has 8 registers: RXB0D0 to RXB0D7.

REGISTER 27-21: RXERRCNT: RECEIVE ERROR COUNT REGISTER

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0
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 **REC<7:0>:** Receive Error Counter bits

This register contains the receive error value as defined by the CAN specifications. When RXERRCNT > 127, the module will go into an error-passive state. RXERRCNT does not have the ability to put the module in "bus-off" state.

EXAMPLE 27-5: READING A CAN MESSAGE

```
    ; Need to read a pending message from RXB0 buffer.
    ; To receive any message, filter, mask and RXM1:RXM0 bits in RXB0CON registers must be
    ; programmed correctly.
    ;
    ; Make sure that there is a message pending in RXB0.
    BTFSS   RXB0CON, RXFUL                ; Does RXB0 contain a message?
    BRA     NoMessage                     ; No. Handle this situation...
    ; We have verified that a message is pending in RXB0 buffer.
    ; If this buffer can receive both Standard or Extended Identifier messages,
    ; identify type of message received.
    BTFSS   RXB0SIDL, EXID                ; Is this Extended Identifier?
    BRA     StandardMessage               ; No. This is Standard Identifier message.
                                          ; Yes. This is Extended Identifier message.
    ; Read all 29-bits of Extended Identifier message.
    ...
    ; Now read all data bytes
    MOVFF   RXB0DO, MY_DATA_BYTE1
    ...
    ; Once entire message is read, mark the RXB0 that it is read and no longer FULL.
    BCF     RXB0CON, RXFUL                ; This will allow CAN Module to load new messages
                                          ; into this buffer.
    ...
```

REGISTER 27-23: BnCON: TX/RX BUFFER 'n' CONTROL REGISTERS IN TRANSMIT MODE [0 ≤ n ≤ 5, TXnEN (BSEL0<n>) = 1](1)

R/W-0	R-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
TXBIF(3)	TXABT(3)	TXLARB(3)	TXERR(3)	TXREQ(2,4)	RTREN	TXPRI1(5)	TXPRI0(5)
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 **TXBIF:** Transmit Buffer Interrupt Flag bit(3)
- 1 = A message was successfully transmitted
- 0 = No message was transmitted

bit 6 **TXABT:** Transmission Aborted Status bit(3)
- 1 = Message was aborted
- 0 = Message was not aborted

bit 5 **TXLARB:** Transmission Lost Arbitration Status bit(3)
- 1 = Message lost arbitration while being sent
- 0 = Message did not lose arbitration while being sent

bit 4 **TXERR:** Transmission Error Detected Status bit(3)
- 1 = A bus error occurred while the message was being sent
- 0 = A bus error did not occur while the message was being sent

bit 3 **TXREQ:** Transmit Request Status bit(2,4)
- 1 = Requests sending a message; clears the TXABT, TXLARB and TXERR bits
- 0 = Automatically cleared when the message is successfully sent

bit 2 **RTREN:** Automatic Remote Transmission Request Enable bit
- 1 = When a remote transmission request is received, TXREQ will be automatically set
- 0 = When a remote transmission request is received, TXREQ will be unaffected

bit 1-0 **TXPRI<1:0>:** Transmit Priority bits(5)
- 11 = Priority Level 3 (highest priority)
- 10 = Priority Level 2
- 01 = Priority Level 1
- 00 = Priority Level 0 (lowest priority)

Note 1: These registers are available in Mode 1 and 2 only.
2: Clearing this bit in software while the bit is set will request a message abort.
3: This bit is automatically cleared when TXREQ is set.
4: While TXREQ is set or a transmission is in progress, Transmit Buffer registers remain read-only.
5: These bits set the order in which the Transmit Buffer register will be transferred. They do not alter the CAN message identifier.

28.0 SPECIAL FEATURES OF THE CPU

The PIC18F66K80 family of devices includes several features intended to maximize reliability and minimize cost through elimination of external components. These include:

- Oscillator Selection
- Resets:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT) and On-Chip Regulator
- Fail-Safe Clock Monitor
- Two-Speed Start-up
- Code Protection
- ID Locations
- In-Circuit Serial Programming™

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 3.0 "Oscillator Configurations"**.

A complete discussion of device Resets and interrupts is available in previous sections of this data sheet.

In addition to their Power-up and Oscillator Start-up Timers provided for Resets, the PIC18F66K80 family of devices has a Watchdog Timer, which is either permanently enabled via the Configuration bits or software controlled (if configured as disabled).

The inclusion of an internal RC oscillator (LF-INTOSC) also provides the additional benefits of a Fail-Safe Clock Monitor (FSCM) and Two-Speed Start-up. FSCM provides for background monitoring of the peripheral clock and automatic switchover in the event of its failure. Two-Speed Start-up enables code to be executed almost immediately on start-up, while the primary clock source completes its start-up delays.

All of these features are enabled and configured by setting the appropriate Configuration register bits.

28.1 Configuration Bits

The Configuration bits can be programmed (read as '0') or left unprogrammed (read as '1') to select various device configurations. These bits are mapped starting at program memory location, 300000h.

The user will note that address, 300000h, is beyond the user program memory space. In fact, it belongs to the configuration memory space (300000h-3FFFFFh), which can only be accessed using table reads and table writes.

Software programming the Configuration registers is done in a manner similar to programming the Flash memory. The WR bit in the EECON1 register starts a self-timed write to the Configuration register. In normal operation mode, a TBLWT instruction with the TBLPTR pointing to the Configuration register sets up the address and the data for the Configuration register write. Setting the WR bit starts a long write to the Configuration register. The Configuration registers are written a byte at a time. To write or erase a configuration cell, a TBLWT instruction can write a '1' or a '0' into the cell. For additional details on Flash programming, refer to **Section 7.5 "Writing to Flash Program Memory"**.

29.1.1 STANDARD INSTRUCTION SET

ADDLW **ADD Literal to W**

Syntax: ADDLW k

Operands: $0 \le k \le 255$

Operation: $(W) + k \rightarrow W$

Status Affected: N, OV, C, DC, Z

Encoding:

0000	1111	kkkk	kkkk

Description: The contents of W are added to the 8-bit literal ‘k’ and the result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: ADDLW 15h

Before Instruction
W = 10h
After Instruction
W = 25h

ADDWF **ADD W to f**

Syntax: ADDWF f {,d {,a}}

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

Operation: $(W) + (f) \rightarrow$ dest

Status Affected: N, OV, C, DC, Z

Encoding:

0010	01da	ffff	ffff

Description: Add W to register ‘f’. If ‘d’ is ‘0’, the result is stored in W. If ‘d’ is ‘1’, the result is stored back in register ‘f’ (default).

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 to destination

Example: ADDWF REG, 0, 0

Before Instruction
W = 17h
REG = 0C2h
After Instruction
W = 0D9h
REG = 0C2h

Note: All PIC18 instructions may take an optional label argument preceding the instruction mnemonic for use in symbolic addressing. If a label is used, the instruction format then becomes: {label} instruction argument(s).

CLRF **Clear f**

Syntax: CLRF f {,a}

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

Operation: 000h → f,
1 → Z

Status Affected: Z

Encoding:

0110	101a	ffff	ffff

Description: Clears the contents of the specified register.

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: CLRF FLAG_REG,1

Before Instruction
FLAG_REG = 5Ah
After Instruction
FLAG_REG = 00h

CLRWDT **Clear Watchdog Timer**

Syntax: CLRWDT

Operands: None

Operation: 000h → WDT,
000h → WDT postscaler,
1 → $\overline{TO}$,
1 → $\overline{PD}$

Status Affected: $\overline{TO}$, $\overline{PD}$

Encoding:

0000	0000	0000	0100

Description: CLRWDT instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, $\overline{TO}$ and $\overline{PD}$, are set.

Words: 1

Cycles: 1

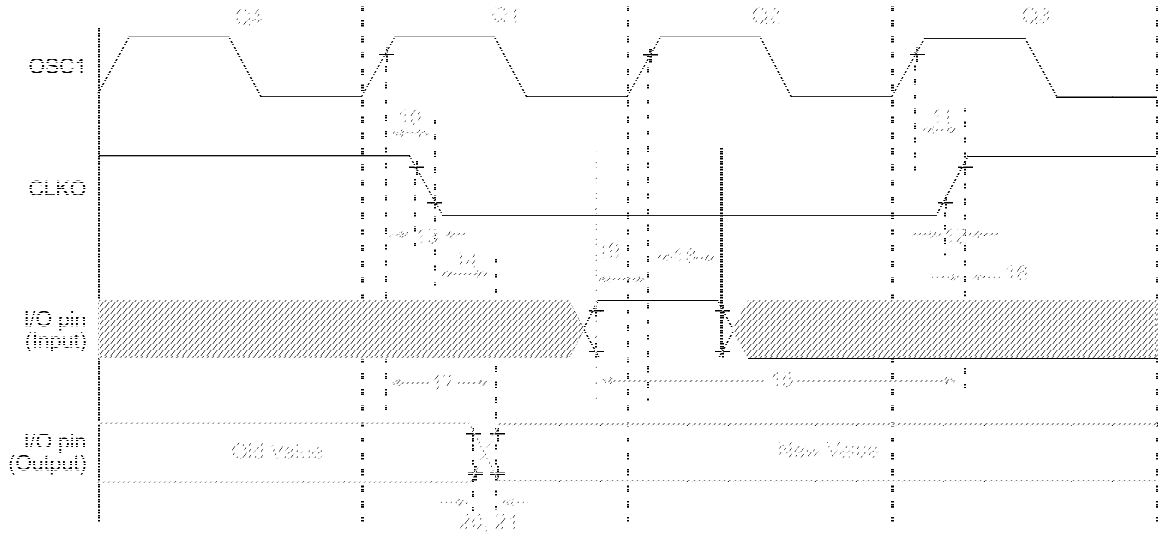
Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	No operation	Process Data	No operation

Example: CLRWDT

Before Instruction
WDT Counter = ?
After Instruction
WDT Counter = 00h
WDT Postscaler = 0
$\overline{TO}$ = 1
$\overline{PD}$ = 1

FIGURE 31-5: CLKO AND I/O TIMING



Note: Refer to Figure 31-3 for load conditions.

TABLE 31-9: CLKO AND I/O TIMING REQUIREMENTS

Param No.	Symbol	Characteristic	Min	Typ	Max	Units	Conditions
10	TOSH2CKL	OSC1 ↑ to CLKO ↓	—	75	200	ns	**(Note 1)**
11	TOSH2CKH	OSC1 ↑ to CLKO ↑	—	75	200	ns	**(Note 1)**
12	TCKR	CLKO Rise Time	—	15	30	ns	**(Note 1)**
13	TCKF	CLKO Fall Time	—	15	30	ns	**(Note 1)**
14	TCKL2IOV	CLKO ↓ to Port Out Valid	—	—	0.5 TCY + 20	ns	
15	TIOV2CKH	Port In Valid before CLKO ↑	0.25 TCY + 25	—	—	ns	
16	TCKH2IOI	Port In Hold after CLKO ↑	0	—	—	ns	
17	TOSH2IOV	OSC1 ↑ (Q1 cycle) to Port Out Valid	—	50	150	ns	
18	TOSH2IOI	OSC1 ↑ (Q2 cycle) to Port Input Invalid (I/O in hold time)	100	—	—	ns	
19	TIOV2OSH	Port Input Valid to OSC1 ↑ (I/O in setup time)	0	—	—	ns	
20	TIOR	Port Output Rise Time	—	10	25	ns	
21	TIOF	Port Output Fall Time	—	10	25	ns	
22†	TINP	INTx pin High or Low Time	20	—	—	ns	
23†	TRBP	RB<7:4> Change INTx High or Low Time	TCY	—	—	ns	

† These parameters are asynchronous events not related to any internal clock edges.

Note 1: Measurements are taken in EC mode, where CLKO output is 4 x TOSC.