



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

### Applications of "[Embedded - Microcontrollers](#)"

#### Details

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	I <sup>2</sup> C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, HLVD, POR, PWM, WDT
Number of I/O	54
Program Memory Size	48KB (24K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	4.2V ~ 5.5V
Data Converters	A/D 12x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f6527t-i-pt">https://www.e-xfl.com/product-detail/microchip-technology/pic18f6527t-i-pt</a>

# PIC18F8722 FAMILY

---

## 2.6.5.1 Compensating with the EUSART

An adjustment may be required when the EUSART begins to generate framing errors or receives data with errors while in Asynchronous mode. Framing errors indicate that the device clock frequency is too high. To adjust for this, decrement the value in OSCTUNE to reduce the clock frequency. On the other hand, errors in data may suggest that the clock speed is too low. To compensate, increment OSCTUNE to increase the clock frequency.

## 2.6.5.2 Compensating with the Timers

This technique compares device clock speed to some reference clock. Two timers may be used; one timer is clocked by the peripheral clock, while the other is clocked by a fixed reference source, such as the Timer1 oscillator.

Both timers are cleared, but the timer clocked by the reference generates interrupts. When an interrupt occurs, the internally clocked timer is read and both timers are cleared. If the internally clocked timer value is much greater than expected, then the internal oscillator block is running too fast. To adjust for this, decrement the OSCTUNE register.

## 2.6.5.3 Compensating with the CCP Module in Capture Mode

A CCP module can use free running Timer1 (or Timer3), clocked by the internal oscillator block and an external event with a known period (i.e., AC power frequency). The time of the first event is captured in the CCPRxH:CCPRxL registers and is recorded for use later. When the second event causes a capture, the time of the first event is subtracted from the time of the second event. Since the period of the external event is known, the time difference between events can be calculated.

If the measured time is much greater than the calculated time, the internal oscillator block is running too fast. To compensate, decrement the OSCTUNE register. If the measured time is much less than the calculated time, the internal oscillator block is running too slow. To compensate, increment the OSCTUNE register.

## 2.7 Clock Sources and Oscillator Switching

The PIC18F8722 family of devices includes a feature that allows the device clock source to be switched from the main oscillator to an alternate clock source. These devices also offer two alternate clock sources. When an alternate clock source is enabled, the various power-managed operating modes are available.

Essentially, there are three clock sources for these devices:

- Primary oscillators
- Secondary oscillators
- Internal oscillator block

The **primary oscillators** include the External Crystal and Resonator modes, the External RC modes, the External Clock modes and the internal oscillator block. The particular mode is defined by the FOSC<3:0> Configuration bits. The details of these modes are covered earlier in this chapter.

The **secondary oscillators** are those external sources not connected to the OSC1 or OSC2 pins. These sources may continue to operate even after the controller is placed in a power-managed mode.

The PIC18F8722 family of devices offers the Timer1 oscillator as a secondary oscillator. This oscillator, in all power-managed modes, is often the time base for functions such as a real-time clock.

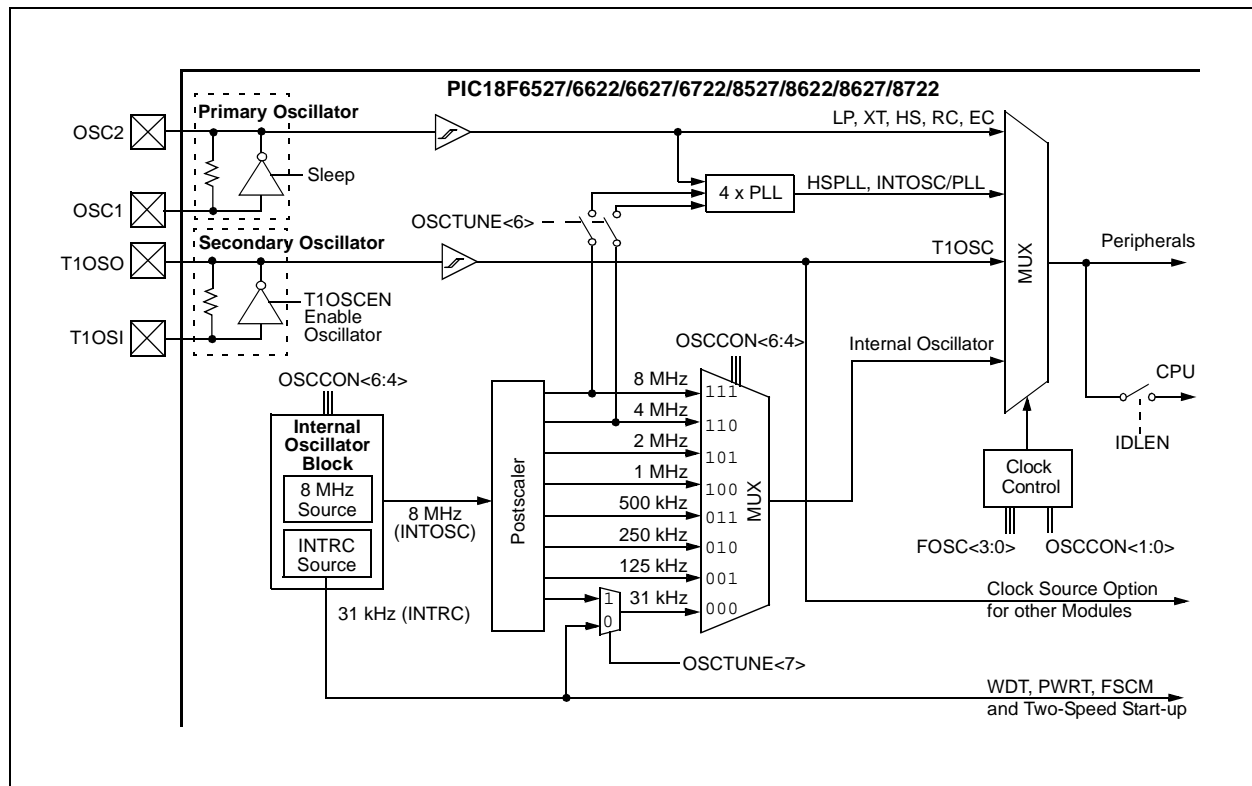
Most often, a 32.768 kHz watch crystal is connected between the RC0/T1OSO/T13CKI and RC1/T1OSI pins. Like the LP mode oscillator circuit, loading capacitors are also connected from each pin to ground.

The Timer1 oscillator is discussed in greater detail in **Section 13.3 “Timer1 Oscillator”**.

In addition to being a primary clock source, the **internal oscillator block** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor.

The clock sources for the PIC18F8722 family of devices are shown in Figure 2-11. See **Section 25.0 “Special Features of the CPU”** for Configuration register details.

**FIGURE 2-11: PIC18F8722 FAMILY CLOCK DIAGRAM**



# PIC18F8722 FAMILY

## 3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

Certain exits from power-managed modes do not invoke the OST at all. There are two cases:

- PRI\_IDLE mode, where the primary clock source is not stopped and
- the primary clock source is not any of the LP, XT, HS or HSPLL modes.

In these instances, the primary clock source either does not require an oscillator start-up delay since it is already running (PRI\_IDLE), or normally does not require an oscillator start-up delay (RC, EC and INTIO Oscillator modes). However, a fixed delay of interval TcSD following the wake event is still required when leaving Sleep and Idle modes to allow the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

**TABLE 3-2: EXIT DELAY ON WAKE-UP BY RESET FROM SLEEP MODE OR ANY IDLE MODE (BY CLOCK SOURCES)**

Clock Source before Wake-up	Clock Source after Wake-up	Exit Delay	Clock Ready Status Bit (OSCCON)
Primary Device Clock (PRI_IDLE mode)	LP, XT, HS	TcSD <sup>(1)</sup>	OSTS
	HSPLL		
	EC, RC		IOFS
	INTOSC <sup>(2)</sup>		
T1OSC or INTRC	LP, XT, HS	TOST <sup>(3)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	TIOBST <sup>(4)</sup>	IOFS
INTOSC <sup>(2)</sup>	LP, XT, HS	TOST <sup>(4)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	None	IOFS
None (Sleep mode)	LP, XT, HS	TOST <sup>(3)</sup>	OSTS
	HSPLL	TOST + t <sub>rc</sub> <sup>(3)</sup>	
	EC, RC	TcSD <sup>(1)</sup>	
	INTOSC <sup>(2)</sup>	TIOBST <sup>(4)</sup>	IOFS

**Note 1:** TcSD (parameter 38, Table 28-12) is a required delay when waking from Sleep and all Idle modes and runs concurrently with any other required delays (see **Section 3.4 “Idle Modes”**).

**2:** Includes both the INTOSC 8 MHz source and postscaler derived frequencies. On Reset, INTOSC defaults to 1 MHz.

**3:** TOST is the Oscillator Start-up Timer (parameter 32, Table 28-12). t<sub>rc</sub> is the PLL Lock-out Timer (parameter F12, Table 28-7); it is also designated as T<sub>P</sub>LL.

**4:** Execution continues during TIOBST (parameter 39, Table 28-12), the INTOSC stabilization period.

## 4.2 Master Clear ( $\overline{\text{MCLR}}$ )

The  $\overline{\text{MCLR}}$  pin provides a method for triggering an external Reset of the device. A Reset is generated by holding the pin low. These devices have a noise filter in the  $\overline{\text{MCLR}}$  Reset path which detects and ignores small pulses.

The  $\overline{\text{MCLR}}$  pin is not driven low by any internal Resets, including the WDT.

In the PIC18F8722 family of devices, the  $\overline{\text{MCLR}}$  input can be disabled with the MCLRE Configuration bit. When  $\overline{\text{MCLR}}$  is disabled, the pin becomes a digital input. See **Section 11.5 “PORTE, TRISE and LATE Registers”** for more information.

## 4.3 Power-on Reset (POR)

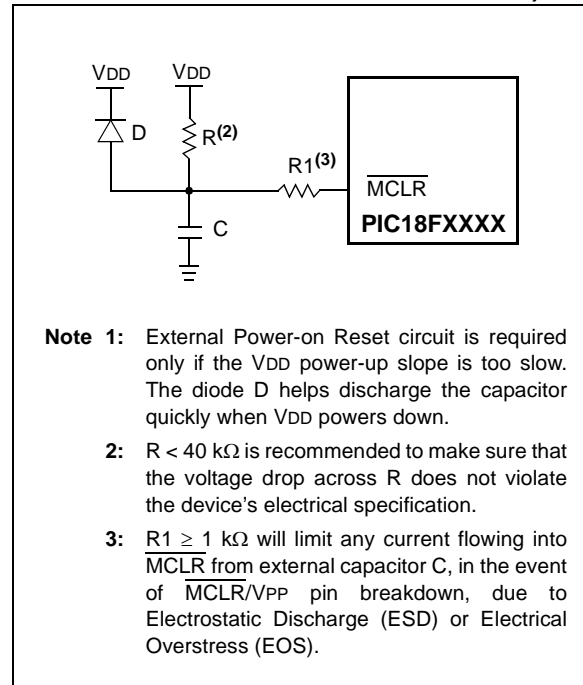
A Power-on Reset pulse is generated on-chip whenever VDD rises above a certain threshold. This allows the device to start in the initialized state when VDD is adequate for operation.

To take advantage of the POR circuitry, tie the  $\overline{\text{MCLR}}$  pin through a resistor (1 k $\Omega$  to 10 k $\Omega$ ) to VDD. This will eliminate external RC components usually needed to create a Power-on Reset delay. A minimum rise rate for VDD is specified (parameter D004, “Section 28.2 “DC Characteristics: Power-Down and Supply Current”). For a slow rise time, see Figure 4-2.

When the device starts normal operation (i.e., exits the Reset condition), device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, the device must be held in Reset until the operating conditions are met.

POR events are captured by the  $\overline{\text{POR}}$  bit (RCON<1>). The state of the bit is set to ‘0’ whenever a POR occurs; it does not change for any other Reset event.  $\overline{\text{POR}}$  is not reset to ‘1’ by any hardware event. To capture multiple events, the user manually resets the bit to ‘1’ in software following any POR.

**FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW VDD POWER-UP)<sup>(1)</sup>**



## 5.1.5.2 Table Reads and Table Writes

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 by using table reads and writes. The Table Pointer (TBLPTR) register specifies the byte address and the Table Latch (TABLAT) register contains the data that is read from or written to program memory. Data is transferred to or from program memory one byte at a time.

Table read and table write operations are discussed further in **Section 6.1 “Table Reads and Table Writes”**.

## 5.2 PIC18 Instruction Cycle

### 5.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program

memory and latched into the instruction register during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-4.

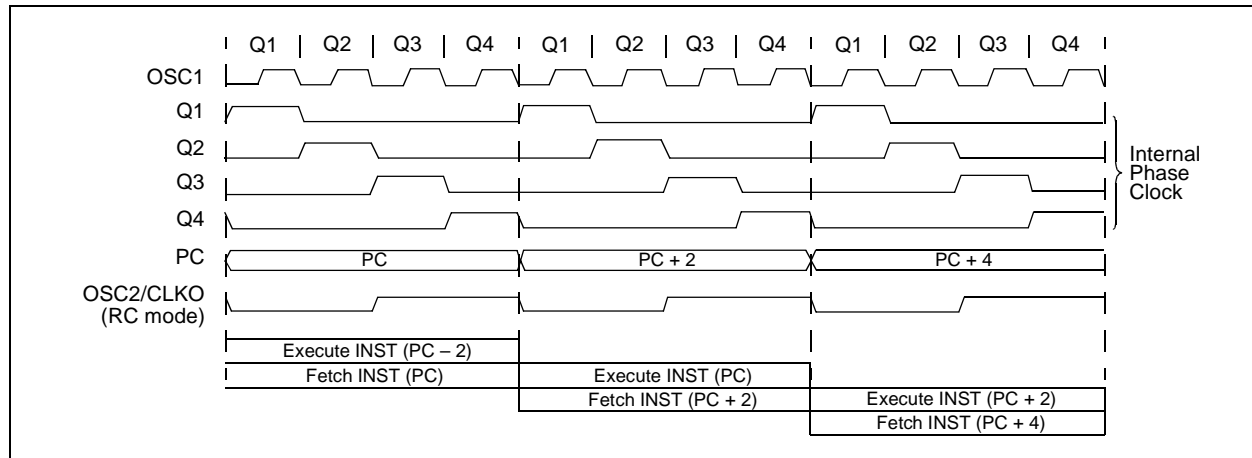
### 5.2.2 INSTRUCTION FLOW/PIPELINING

An “Instruction Cycle” consists of four Q cycles: Q1 through Q4. The instruction fetch and execute are pipelined in such a manner that a fetch takes one instruction cycle, while the decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 5-3).

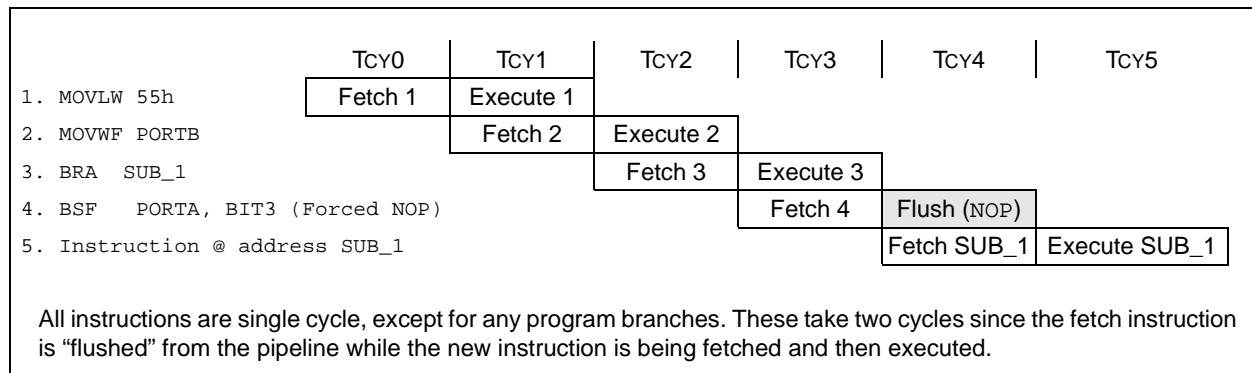
A fetch cycle begins with the program counter incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the Instruction Register (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

**FIGURE 5-4: CLOCK/INSTRUCTION CYCLE**



**EXAMPLE 5-3: INSTRUCTION PIPELINE FLOW**



# PIC18F8722 FAMILY

---

NOTES:

# PIC18F8722 FAMILY

## 7.0 EXTERNAL MEMORY BUS

**Note:** The External Memory Bus is not implemented on PIC18F6527/6622/6627/6722 (64-pin) devices.

The External Memory Bus (EMB) allows the device to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program or data memory. It supports both 8-bit and 16-bit Data Width modes and four address widths from 8 to 20 bits.

The bus is implemented with 28 pins, multiplexed across four I/O ports. Three ports (PORTD, PORTE and PORTH) are multiplexed with the address/data bus for a total of 20 available lines, while PORTJ is multiplexed with the bus control signals.

A list of the pins and their functions is provided in Table 7-1.

**TABLE 7-1: PIC18F8527/8622/8627/8722 EXTERNAL BUS – I/O PORT FUNCTIONS**

Name	Port	Bit	External Memory Bus Function
RD0/AD0	PORTD	0	Address bit 0 or Data bit 0
RD1/AD1	PORTD	1	Address bit 1 or Data bit 1
RD2/AD2	PORTD	2	Address bit 2 or Data bit 2
RD3/AD3	PORTD	3	Address bit 3 or Data bit 3
RD4/AD4	PORTD	4	Address bit 4 or Data bit 4
RD5/AD5	PORTD	5	Address bit 5 or Data bit 5
RD6/AD6	PORTD	6	Address bit 6 or Data bit 6
RD7/AD7	PORTD	7	Address bit 7 or Data bit 7
RE0/AD8	PORTE	0	Address bit 8 or Data bit 8
RE1/AD9	PORTE	1	Address bit 9 or Data bit 9
RE2/AD10	PORTE	2	Address bit 10 or Data bit 10
RE3/AD11	PORTE	3	Address bit 11 or Data bit 11
RE4/AD12	PORTE	4	Address bit 12 or Data bit 12
RE5/AD13	PORTE	5	Address bit 13 or Data bit 13
RE6/AD14	PORTE	6	Address bit 14 or Data bit 14
RE7/AD15	PORTE	7	Address bit 15 or Data bit 15
RH0/A16	PORTH	0	Address bit 16
RH1/A17	PORTH	1	Address bit 17
RH2/A18	PORTH	2	Address bit 18
RH3/A19	PORTH	3	Address bit 19
RJ0/ALE	PORTJ	0	Address Latch Enable (ALE) Control pin
RJ1/ $\overline{OE}$	PORTJ	1	Output Enable ( $\overline{OE}$ ) Control pin
RJ2/ $\overline{WRL}$	PORTJ	2	Write Low ( $\overline{WRL}$ ) Control pin
RJ3/ $\overline{WRH}$	PORTJ	3	Write High ( $\overline{WRH}$ ) Control pin
RJ4/BA0	PORTJ	4	Byte Address bit 0 (BA0)
RJ5/ $\overline{CE}$	PORTJ	5	Chip Enable ( $\overline{CE}$ ) Control pin
RJ6/ $\overline{LB}$	PORTJ	6	Lower Byte Enable ( $\overline{LB}$ ) Control pin
RJ7/ $\overline{UB}$	PORTJ	7	Upper Byte Enable ( $\overline{UB}$ ) Control pin

**Note:** For the sake of clarity, only I/O port and external bus assignments are shown here. One or more additional multiplexed features may be available on some pins.



# PIC18F8722 FAMILY

**TABLE 11-3: PORTB FUNCTIONS**

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RB0/INT0/FLT0	RB0	0	O	DIG	LATB<0> data output.
		1	I	TTL	PORTB<0> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT0	1	I	ST	External interrupt 0 input.
	FLT0	1	I	ST	ECCPx PWM Fault input, enabled in software.
RB1/INT1	RB1	0	O	DIG	LATB<1> data output.
		1	I	TTL	PORTB<1> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT1	1	I	ST	External interrupt 1 input.
RB2/INT2	RB2	0	O	DIG	LATB<2> data output.
		1	I	TTL	PORTB<2> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	INT2	1	I	ST	External interrupt 2 input.
RB3/INT3/ ECCP2/P2A	RB3	0	O	DIG	LATB<3> data output.
		1	I	TTL	PORTB<3> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared and capture input is disabled.
	INT3	1	I	ST	External interrupt 3 input.
	ECCP2 <sup>(1)</sup>	0	O	DIG	ECCP2 compare output and ECCP2 PWM output. Takes priority over port data.
		1	I	ST	ECCP2 capture input.
	P2A <sup>(1)</sup>	0	O	DIG	ECCP2 Enhanced PWM output, channel A. May be configured for tri-state during Enhanced PWM shutdown events. Takes priority over port data.
RB4/KBI0	RB4	0	O	DIG	LATB<4> data output.
		1	I	TTL	PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI0	1	I	TTL	Interrupt-on-pin change.
RB5/KBI1/PGM	RB5	0	O	DIG	LATB<5> data output
		1	I	TTL	PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI1	1	I	TTL	Interrupt-on-pin change.
	PGM	x	I	ST	Single-Supply Programming mode entry (ICSP). Enabled by LVP Configuration bit; all other pin functions disabled.
RB6/KBI2/PGC	RB6	0	O	DIG	LATB<6> data output.
		1	I	TTL	PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI2	1	I	TTL	Interrupt-on-pin change.
	PGC	x	I	ST	Serial execution (ICSP™) clock input for ICSP and ICD operation <sup>(2)</sup> .
RB7/KBI3/PGD	RB7	0	O	DIG	LATB<7> data output.
		1	I	TTL	PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared.
	KBI3	1	I	TTL	Interrupt-on-pin change.
	PGD	x	O	DIG	Serial execution data output for ICSP and ICD operation <sup>(2)</sup> .
		x	I	ST	Serial execution data input for ICSP and ICD operation <sup>(2)</sup> .

**Legend:** PWR = Power Supply, O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Buffer Input, TTL = TTL Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).

**Note 1:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared (Microprocessor, Extended Microcontroller and Microcontroller with Boot Block modes, 80-pin devices only). Default assignment is RC1.

**2:** All other pin functions are disabled when ICSP or ICD operations are enabled.

# PIC18F8722 FAMILY

## 12.1 Timer0 Operation

Timer0 can operate as either a timer or a counter; the mode is selected with the T0CS bit (T0CON<5>). In Timer mode (T0CS = 0), the module increments on every clock by default unless a different prescaler value is selected (see **Section 12.3 “Prescaler”**). If the TMR0 register is written to, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

The Counter mode is selected by setting the T0CS bit (= 1). In this mode, Timer0 increments either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit, T0SE (T0CON<4>); clearing this bit selects the rising edge. Restrictions on the external clock input are discussed below.

An external clock source can be used to drive Timer0; however, it must meet certain requirements to ensure that the external clock can be synchronized with the

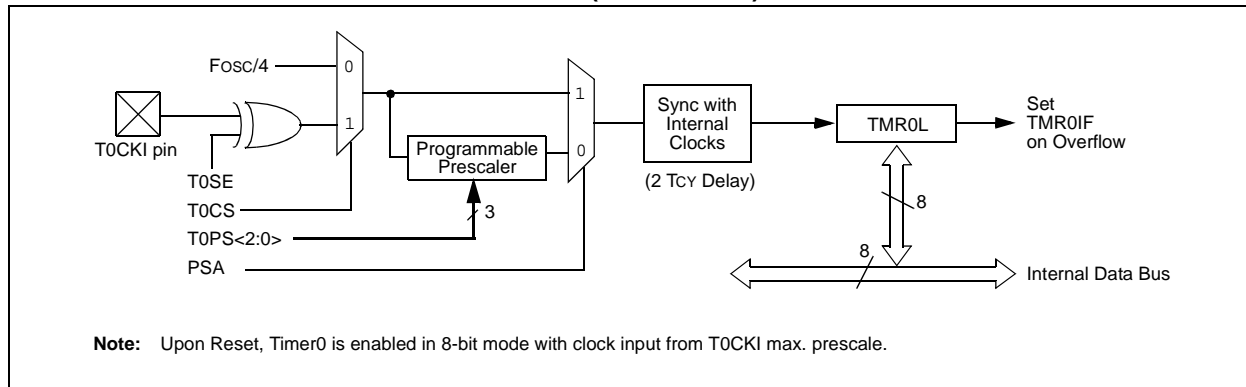
internal phase clock (Tosc). There is a delay between synchronization and the onset of incrementing the timer/counter.

## 12.2 Timer0 Reads and Writes in 16-bit Mode

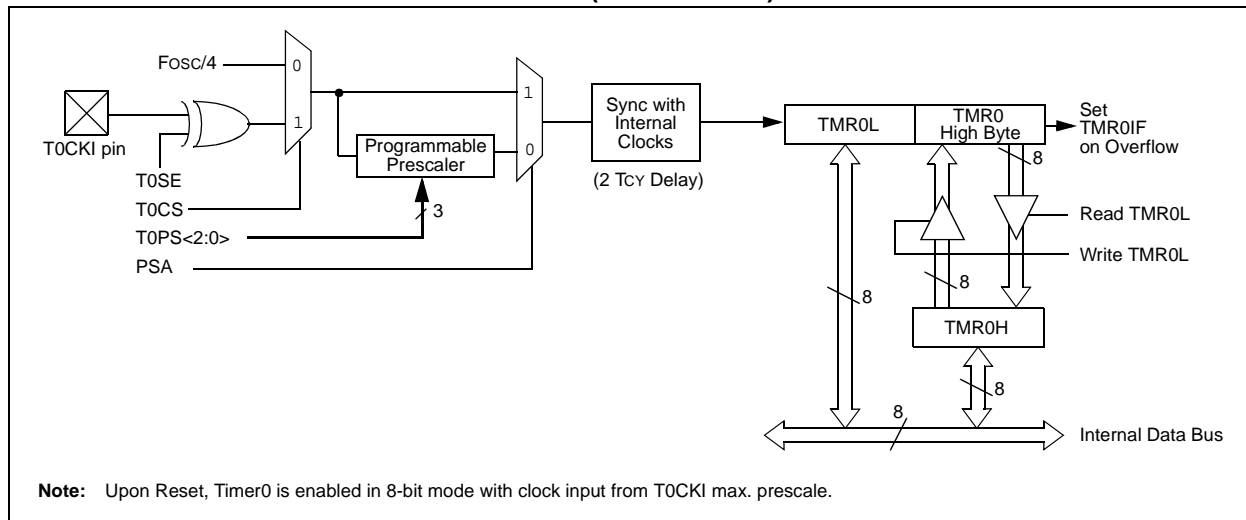
TMR0H is not the actual high byte of Timer0 in 16-bit mode; it is actually a buffered version of the real high byte of Timer0 which is not directly readable nor writable (refer to Figure 12-2). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte.

Similarly, a write to the high byte of Timer0 must also take place through the TMR0H Buffer register. The high byte is updated with the contents of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

**FIGURE 12-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)**



**FIGURE 12-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)**



# PIC18F8722 FAMILY

## 15.1 Timer3 Operation

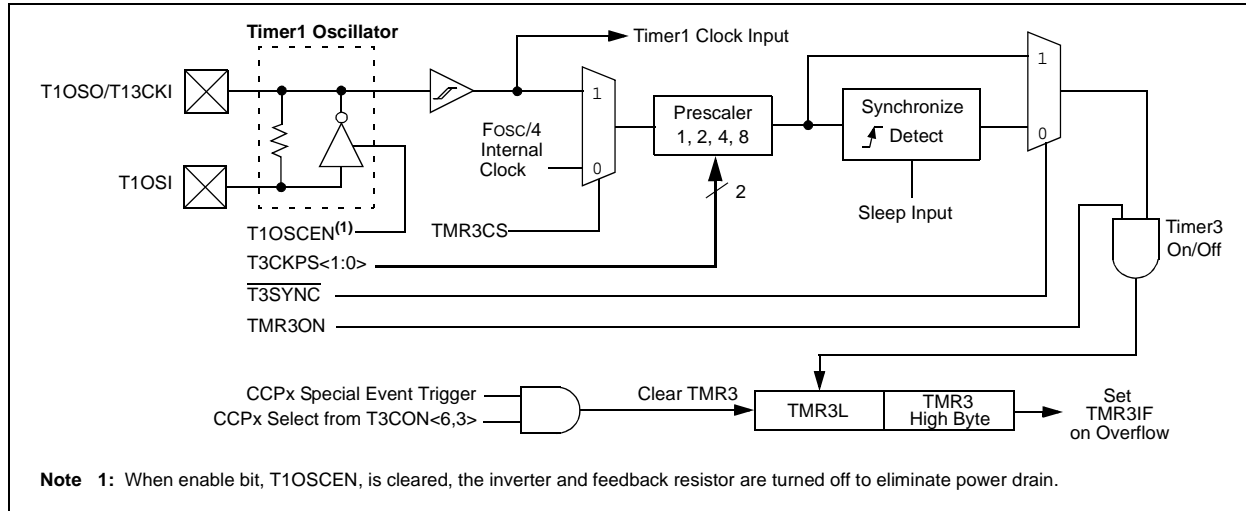
Timer3 can operate in one of three modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

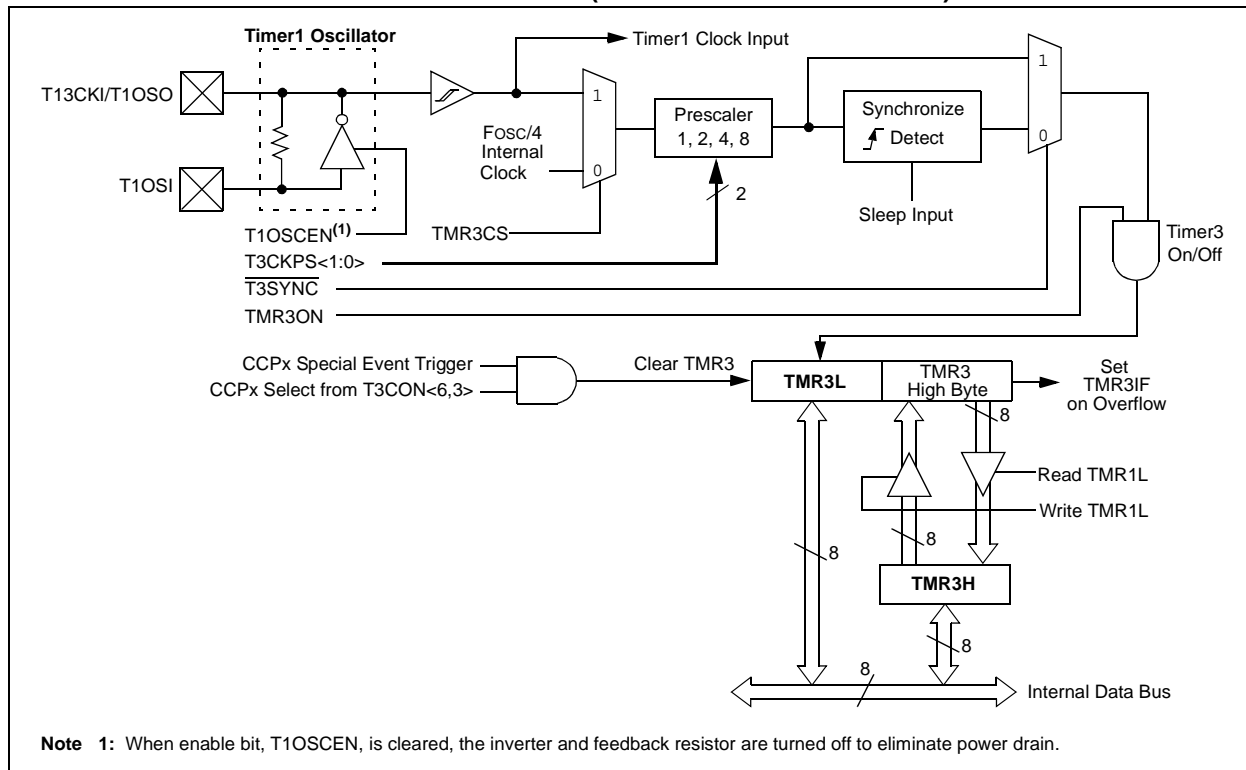
The operating mode is determined by the clock select bit, TMR3CS (T3CON<1>). When TMR3CS is cleared (= 0), Timer3 increments on every internal instruction cycle ( $F_{osc}/4$ ). When the bit is set, Timer3 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

As with Timer1, the RC1/T1OSI and RC0/T1OSO/T13CKI pins become inputs when the Timer1 oscillator is enabled. This means the values of TRISC<1:0> are ignored and the pins are read as '0'.

**FIGURE 15-1: TIMER3 BLOCK DIAGRAM**



**FIGURE 15-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)**



# PIC18F8722 FAMILY

**REGISTER 18-3: ECCPxAS: ENHANCED CCP AUTO-SHUTDOWN CONFIGURATION 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
ECCPxASE	ECCPxAS2	ECCPxAS1	ECCPxAS0	PSSxAC1	PSSxAC0	PSSxBD1	PSSxBD0
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      **ECCPxASE:** ECCP Auto-Shutdown Event Status bit  
0 = ECCP outputs are operating  
1 = A shutdown event has occurred; ECCP outputs are in shutdown state
- bit 6-4    **ECCPxAS<2:0>:** ECCP Auto-Shutdown Source Select bits  
000 = Auto-shutdown is disabled  
001 = Comparator 1 output  
010 = Comparator 2 output  
011 = Either Comparator 1 or 2  
100 = FLT0  
101 = FLT0 or Comparator 1  
110 = FLT0 or Comparator 2  
111 = FLT0 or Comparator 1 or Comparator 2
- bit 3-2    **PSSxAC<1:0>:** Pins A and C Shutdown State Control bits  
00 = Drive pins A and C to '0'  
01 = Drive pins A and C to '1'  
1x = Pins A and C tri-state
- bit 1-0    **PSSxBD<1:0>:** Pins B and D Shutdown State Control bits  
00 = Drive pins B and D to '0'  
01 = Drive pins B and D to '1'  
1x = Pins B and D tri-state

# PIC18F8722 FAMILY

FIGURE 20-1: AUTOMATIC BAUD RATE CALCULATION

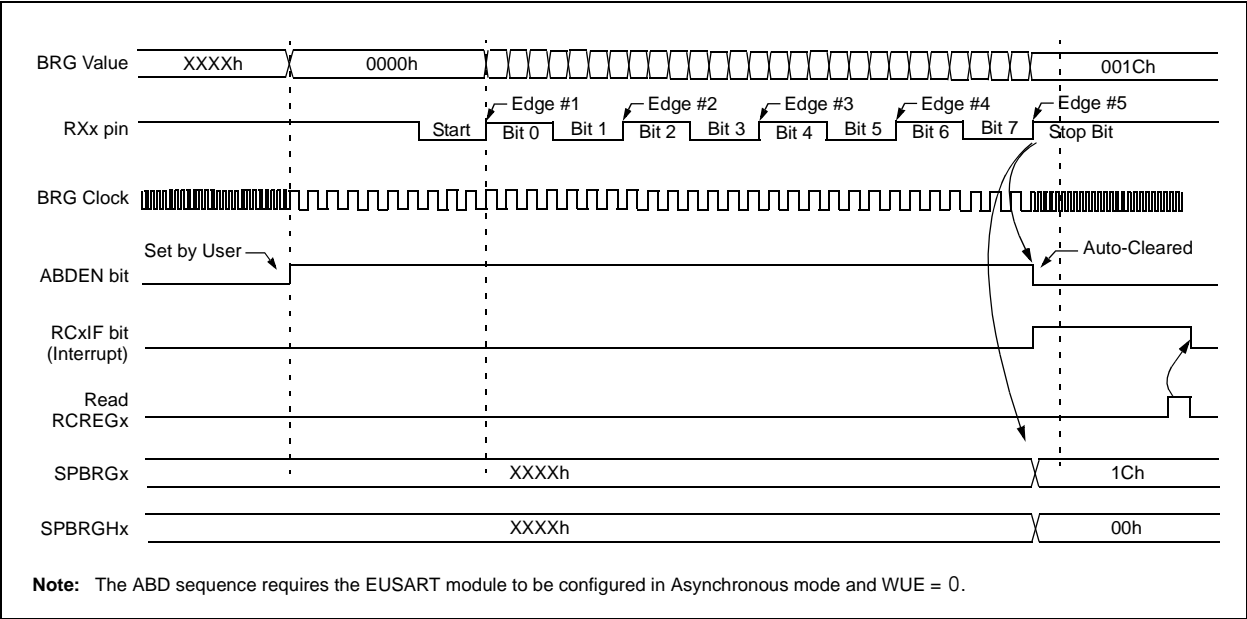
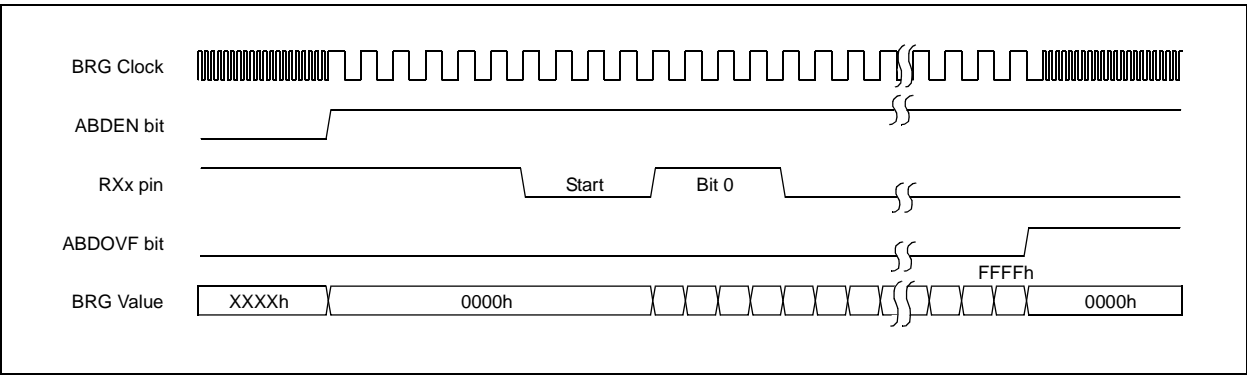


FIGURE 20-2: BRG OVERFLOW SEQUENCE



# PIC18F8722 FAMILY

## 20.2.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 20-6. The data is received on the RXx pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at Fosc. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

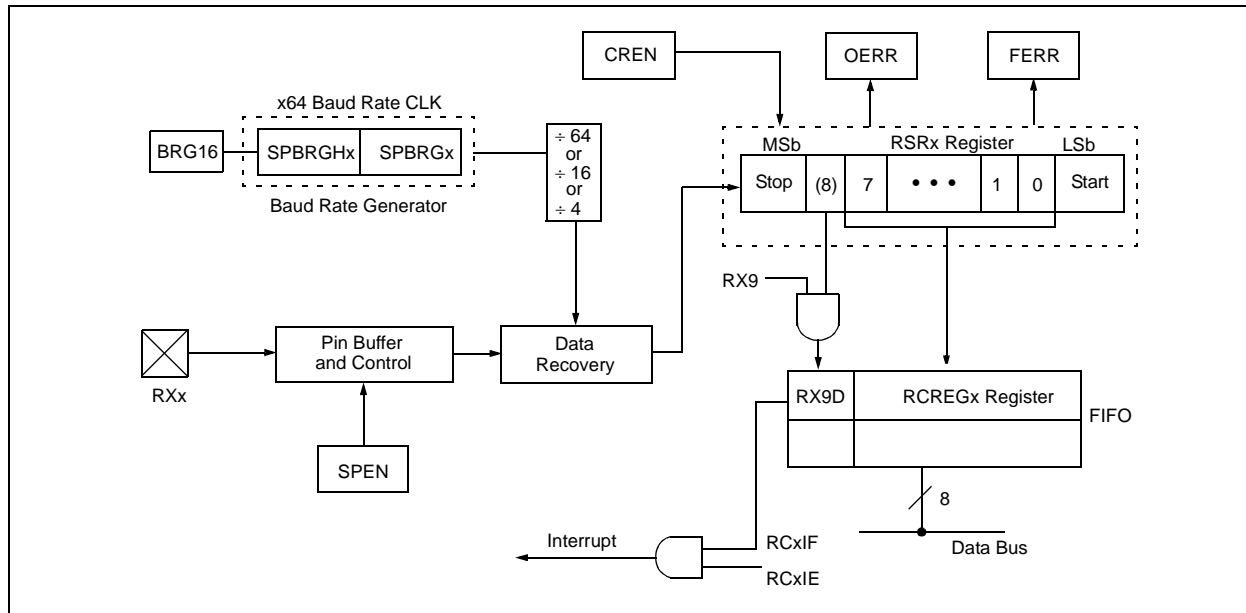
1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RCxIE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RCxIF, will be set when reception is complete and an interrupt will be generated if enable bit, RCxIE, was set.
7. Read the RCSTAx register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREGx register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

## 20.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. To set up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRGHx:SPBRGx registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCxIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCxIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCxIE and GIE bits are set.
8. Read the RCSTAx register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREGx to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

**FIGURE 20-6: EUSART RECEIVE BLOCK DIAGRAM**



# PIC18F8722 FAMILY

## BNC Branch if Not Carry

Syntax: BNC n

Operands:  $-128 \leq n \leq 127$

Operation: if Carry bit is '0'  
 $(PC) + 2 + 2n \rightarrow PC$

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:  $-128 \leq n \leq 127$

Operation: if Negative bit is '0'  
 $(PC) + 2 + 2n \rightarrow PC$

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)

# PIC18F8722 FAMILY

## SUBWFB Subtract W from f with Borrow

**Syntax:** SUBWFB f {,d {,a}}

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

**Operation:**  $(f) - (W) - (\overline{C}) \rightarrow \text{dest}$

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

**Encoding:**

0101	10da	ffff	ffff
------	------	------	------

**Description:** Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). 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 (default).  
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 26.2.3 "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:** SUBWFB REG, 1, 0

**Before Instruction**

REG = 19h (0001 1001)  
 W = 0Dh (0000 1101)  
 C = 1

**After Instruction**

REG = 0Ch (0000 1011)  
 W = 0Dh (0000 1101)  
 C = 1  
 Z = 0  
 N = 0 ; result is positive

**Example 2:** SUBWFB REG, 0, 0

**Before Instruction**

REG = 1Bh (0001 1011)  
 W = 1Ah (0001 1010)  
 C = 0

**After Instruction**

REG = 1Bh (0001 1011)  
 W = 00h  
 C = 1  
 Z = 1 ; result is zero  
 N = 0

**Example 3:** SUBWFB REG, 1, 0

**Before Instruction**

REG = 03h (0000 0011)  
 W = 0Eh (0000 1101)  
 C = 1

**After Instruction**

REG = F5h (1111 0100)  
 ; [2's comp]  
 W = 0Eh (0000 1101)  
 C = 0  
 Z = 0  
 N = 1 ; result is negative

## SWAPF Swap f

**Syntax:** SWAPF f {,d {,a}}

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

**Operation:**  $(f<3:0>) \rightarrow \text{dest}<7:4>$ ,  
 $(f<7:4>) \rightarrow \text{dest}<3:0>$

**Status Affected:** None

**Encoding:**

0011	10da	ffff	ffff
------	------	------	------

**Description:** The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed 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 (default).  
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 26.2.3 "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:** SWAPF REG, 1, 0

**Before Instruction**

REG = 53h

**After Instruction**

REG = 35h



# PIC18F8722 FAMILY

---

## 27.11 PICSTART® Plus Development Programmer

The PICSTART® Plus Development Programmer is an easy-to-use, low-cost, prototype programmer. It connects to the PC via a COM (RS-232) port. MPLAB Integrated Development Environment software makes using the programmer simple and efficient. The PICSTART Plus Development Programmer supports most PIC devices in DIP packages up to 40 pins. Larger pin count devices, such as the PIC16C92X and PIC17C76X, may be supported with an adapter socket. The PICSTART Plus Development Programmer is CE compliant.

## 27.12 PICkit™ 2 Development Programmer

The PICkit™ 2 Development Programmer is a low-cost programmer and selected Flash device debugger with an easy-to-use interface for programming many of Microchip's baseline, mid-range and PIC18F families of Flash memory microcontrollers. The PICkit 2 Starter Kit includes a prototyping development board, twelve sequential lessons, software and HI-TECH's PICC™ Lite C compiler, and is designed to help get up to speed quickly using PIC® microcontrollers. The kit provides everything needed to program, evaluate and develop applications using Microchip's powerful, mid-range Flash memory family of microcontrollers.

## 27.13 Demonstration, Development and Evaluation Boards

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.

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

# PIC18F8722 FAMILY

**TABLE 28-1: MEMORY PROGRAMMING REQUIREMENTS**

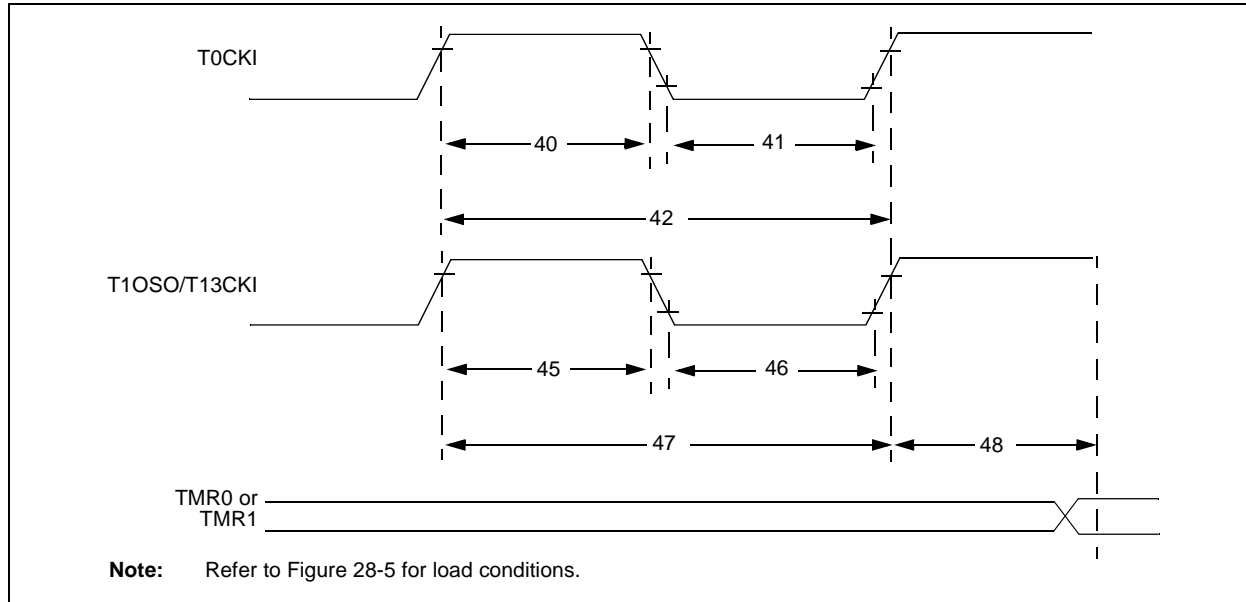
DC CHARACTERISTICS			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial				
Param No.	Sym	Characteristic	Min	Typ†	Max	Units	Conditions
<b>Data EEPROM Memory</b>							
D120	ED	Byte Endurance	100K	1M	—	E/W	-40°C to +85°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	4	—	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh <sup>(1)</sup>	1M	10M	—	E/W	
D125	IDDP	Supply Current during Programming	—	10	—	mA	
<b>Program Flash Memory</b>							
D130	EP	Cell Endurance	10K	100K	—	E/W	-40°C to +85°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132B	VPEW	VDD for Self-Timed Write and Row Erase	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	Provided no other specifications are violated
D134	TRETD	Characteristic Retention	40	100	—	Year	
D135	IDDP	Supply Current during Programming	—	10	—	mA	

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** Refer to **Section 8.8 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

# PIC18F8722 FAMILY

**FIGURE 28-12: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**



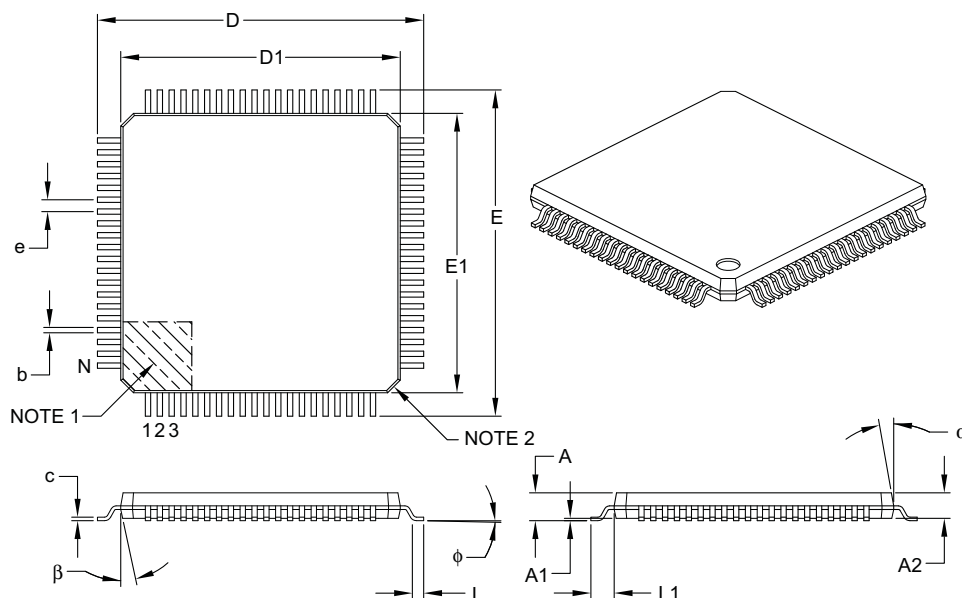
**TABLE 28-13: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

Param No.	Symbol	Characteristic		Min	Max	Units	Conditions
40	T <sub>T0H</sub>	T0CKI High Pulse Width	No prescaler	$0.5 T_{CY} + 20$	—	ns	
			With prescaler	10	—	ns	
41	T <sub>T0L</sub>	T0CKI Low Pulse Width	No prescaler	$0.5 T_{CY} + 20$	—	ns	
			With prescaler	10	—	ns	
42	T <sub>T0P</sub>	T0CKI Period	No prescaler	$T_{CY} + 10$	—	ns	
			With prescaler	Greater of: 20 ns or $(T_{CY} + 40)/N$	—	ns	
45	T <sub>T1H</sub>	T13CKI High Time	Synchronous, no prescaler	$0.5 T_{CY} + 20$	—	ns	V <sub>DD</sub> = 2.0V
			Synchronous, with prescaler	PIC18FXXXX	10	—	
				PIC18LFXXXX	25	—	
			Asynchronous	PIC18FXXXX	30	—	
				PIC18LFXXXX	50	—	
46	T <sub>T1L</sub>	T13CKI Low Time	Synchronous, no prescaler	$0.5 T_{CY} + 5$	—	ns	V <sub>DD</sub> = 2.0V
			Synchronous, with prescaler	PIC18FXXXX	10	—	
				PIC18LFXXXX	25	—	
			Asynchronous	PIC18FXXXX	30	—	
				PIC18LFXXXX	50	—	
47	T <sub>T1P</sub>	T13CKI Input Period	Synchronous	Greater of: 20 ns or $(T_{CY} + 40)/N$	—	ns	N = prescale value (1, 2, 4, 8)
			Asynchronous	60	—	ns	
	F <sub>T1</sub>	T13CKI Oscillator Input Frequency Range		DC	50	kHz	
48	T <sub>CKE2TMR1</sub>	Delay from External T13CKI Clock Edge to Timer Increment		2 T <sub>OSC</sub>	7 T <sub>OSC</sub>	—	

# PIC18F8722 FAMILY

## 80-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at <http://www.microchip.com/packaging>



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N	80		
Lead Pitch	e	0.50 BSC		
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	φ	0°	3.5°	7°
Overall Width	E	14.00 BSC		
Overall Length	D	14.00 BSC		
Molded Package Width	E1	12.00 BSC		
Molded Package Length	D1	12.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-092B

# PIC18F8722 FAMILY

PORTA		
Associated Registers .....	136	
Functions .....	136	
LATA Register .....	135	
PORTA Register .....	135	
TRISA Register .....	135	
PORTB		
Associated Registers .....	139	
Functions .....	138	
LATB Register .....	137	
PORTB Register .....	137	
RB7:RB4 Interrupt-on-Change Flag (RBIF Bit) .....	137	
TRISB Register .....	137	
PORTC		
Associated Registers .....	142	
Functions .....	141	
LATC Register .....	140	
PORTC Register .....	140	
RC3/SCKx/SCLx Pin .....	220	
TRISC Register .....	140	
PORTD		
Associated Registers .....	145	
Functions .....	144	
LATD Register .....	143	
PORTD Register .....	143	
TRISD Register .....	143	
PORTE		
Analog Port Pins .....	158	
Associated Registers .....	148	
Functions .....	147	
LATE Register .....	146	
PORTE Register .....	146	
PSP Mode Select (PSPMODE Bit) .....	158	
RE0/RD Pin .....	158	
RE1/WR Pin .....	158	
RE2/CS Pin .....	158	
TRISE Register .....	146	
PORTF		
Associated Registers .....	150	
Functions .....	150	
LATF Register .....	149	
PORTF Register .....	149	
TRISF Register .....	149	
PORTG		
Associated Registers .....	153	
Functions .....	152	
LATG Register .....	151	
PORTG Register .....	151	
TRISG Register .....	151	
PORTH		
Associated Registers .....	155	
Functions .....	155	
LATH Register .....	154	
PORTH Register .....	154	
TRISH Register .....	154	
PORTJ		
Associated Registers .....	157	
Functions .....	157	
LATJ Register .....	156	
PORTJ Register .....	156	
TRISJ Register .....	156	
Power-Managed Modes .....	41	
and A/D Operation .....	278	
and EUSART Operation .....	251	
and Multiple Sleep Commands .....	42	
and PWM Operation .....	203	
and SPI Operation .....	213	
Associated Registers .....	109	
Clock Transitions and Status Indicators .....	42	
Effects on Clock Sources .....	40	
Entering .....	41	
Exiting Idle and Sleep Modes .....	47	
by Interrupt .....	47	
by Reset .....	47	
by WDT Time-out .....	47	
Without a Start-up Delay .....	48	
Idle Modes .....	45	
PRI_IDLE .....	46	
RC_IDLE .....	47	
SEC_IDLE .....	46	
Run Modes .....	42	
PRI_RUN .....	42	
RC_RUN .....	43	
SEC_RUN .....	42	
Selecting .....	41	
Sleep Mode .....	45	
Summary (table) .....	41	
Power-on Reset (POR) .....	51	
Power-up Timer (PWRT) .....	53	
Time-out Sequence .....	53	
Power-up Delays .....	40	
Power-up Timer (PWRT) .....	40	
Prescaler		
Timer2 .....	193	
Prescaler, Timer0 .....	163	
Prescaler, Timer2 .....	185	
PRI_IDLE Mode .....	46	
PRI_RUN Mode .....	42	
Program Counter .....	66	
PCL, PCH and PCU Registers .....	66	
PCLATH and PCLATU Registers .....	66	
Program Memory		
and Extended Instruction Set .....	85	
Code Protection .....	318	
Extended Microcontroller Mode .....	63	
Instructions .....	70	
Two-Word .....	71	
Interrupt Vector .....	63	
Look-up Tables .....	68	
Map and Stack (diagram) .....	64	
Microcontroller Mode .....	63	
Microprocessor Mode .....	63	
Microprocessor with Boot Block Mode .....	63	
Reset Vector .....	63	
Program Verification and Code Protection .....	317	
Associated Registers .....	318	
Programming, Device Instructions .....	321	
PSP. See Parallel Slave Port.		
Pulse-Width Modulation. See PWM (CCP Module) and PWM (ECCP Module).		
PUSH .....	350	
PUSH and POP Instructions .....	67	
PUSHL .....	366	