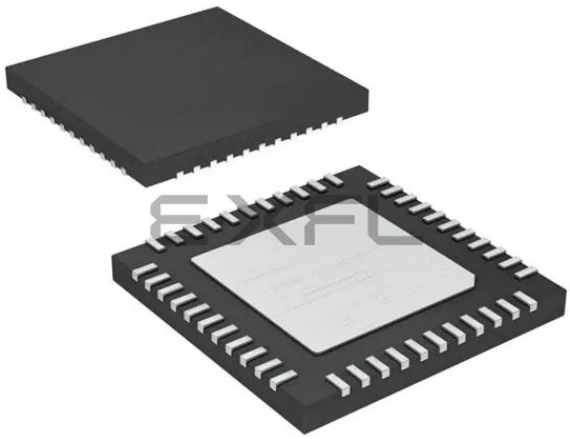


**Welcome to E-XFL.COM**

## 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 | 48MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, POR, PWM, WDT |
| Number of I/O | 34 |
| Program Memory Size | 32KB (16K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 3.8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2V ~ 3.6V |
| Data Converters | A/D 13x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-VQFN Exposed Pad |
| Supplier Device Package | 44-QFN (8x8) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf45j11-i-ml |

**REGISTER 3-2: OSCCON: OSCILLATOR CONTROL REGISTER (ACCESS FD3h)**

| R/W-0 | R/W-1 | R/W-1 | R/W-0 | R-1(1) | U-1 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| IDLEN | IRCF2 | IRCF1 | IRCF0 | OSTS | — | SCS1 | SCS0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit | |
| -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 Idle mode on SLEEP instruction
0 = Device enters Sleep mode on SLEEP instruction

bit 6-4 **IRCF<2:0>:** Internal Oscillator Frequency Select bits(4)
111 = 8 MHz (INTOSC drives clock directly)
110 = 4 MHz(2)
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz (from either INTOSC/256 or INTRC directly)(3)

bit 3 **OSTS:** Oscillator Start-up Time-out Status bit(1)
1 = Oscillator Start-up Timer time-out has expired; primary oscillator is running
0 = Oscillator Start-up Timer time-out is running; primary oscillator is not ready

bit 2 **Unimplemented:** Read as '1'

bit 1-0 **SCS<1:0>:** System Clock Select bits
11 = Postscaled internal clock (INTRC/INTOSC derived)
10 = Reserved
01 = Timer1 oscillator
00 = Primary clock source (INTOSC postscaler output when FOSC<2:0> = 001 or 000)
00 = Primary clock source (CPU divider output for other values of FOSC<2:0>)

**Note 1:** Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.
**2:** Default output frequency of INTOSC on Reset (4 MHz).
**3:** Source selected by the INTSRC bit (OSCTUNE<7>).
**4:** When using INTOSC to drive the 4x PLL, select 8 MHz or 4 MHz only to avoid operating the 4x PLL outside of specification.


© 2011 Microchip Technology Inc.

## 5.2 Master Clear ($\overline{MCLR}$)

The Master Clear Reset ($\overline{MCLR}$) pin provides a method for triggering a hard external Reset of the device. A Reset is generated by holding the pin low. PIC18 extended microcontroller devices have a noise filter in the $\overline{MCLR}$ Reset path, which detects and ignores small pulses.

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

## 5.3 Power-on Reset (POR)

A POR condition 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 $\overline{MCLR}$ pin through a resistor (1 kΩ to 10 kΩ) to VDD. This will eliminate external RC components usually needed to create a POR delay.

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 $\overline{POR}$ bit (RCON<1>). The state of the bit is set to '0' whenever a Power-on Reset occurs; it does not change for any other Reset event. $\overline{POR}$ 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.

## 5.4 Brown-out Reset (BOR)

"F" devices incorporate two types of BOR circuits: one which monitors VDDCORE and one which monitors VDD. Only one BOR circuit can be active at a time. When in normal Run mode, Idle or normal Sleep modes, the BOR circuit that monitors VDDCORE is active and will cause the device to be held in BOR if VDDCORE drops below VBOR (parameter D005). Once VDDCORE rises back above VBOR, the device will be held in Reset until the expiration of the Power-up Timer, with period, TPWRT (parameter 33).

During Deep Sleep operation, the on-chip core voltage regulator is disabled and VDDCORE is allowed to drop to ground levels. If the Deep Sleep BOR circuit is enabled by the DSBOREN Configuration bit (CONFIG3L<2> = 1), it will monitor VDD. If VDD drops below the VDSBOR threshold, the device will be held in a Reset state similar to POR. All registers will be set back to their POR Reset values and the contents of the DSGPR0 and DSGPR1 holding registers will be lost.

Additionally, if any I/O pins had been configured as outputs during Deep Sleep, these pins will be tri-stated and the device will no longer be held in Deep Sleep. Once the VDD voltage recovers back above the VDSBOR threshold, and once the core voltage regulator achieves a VDDCORE voltage above VBOR, the device will begin executing code again normally, but the DS bit in the WDTCON register will not be set. The device behavior will be similar to hard cycling all power to the device.

On "LF" devices, the VDDCORE BOR circuit is always disabled because the internal core voltage regulator is disabled. Instead of monitoring VDDCORE, PIC18LF devices in this family can use the VDD BOR circuit to monitor VDD excursions below the VDSBOR threshold. The VDD BOR circuit can be disabled by setting the DSBOREN bit = 0.

The VDD BOR circuit is enabled when DSBOREN = 1 on "LF" devices, or on "F" devices while in Deep Sleep with DSBOREN = 1. When enabled, the VDD BOR circuit is extremely low power (typ. 40 nA) during normal operation above ~2.3V on VDD. If VDD drops below this DSBOR arming level when the VDD BOR circuit is enabled, the device may begin to consume additional current (typ. 50 μA) as internal features of the circuit power up. The higher current is necessary to achieve more accurate sensing of the VDD level. However, the device will not enter Reset until VDD falls below the VDSBOR threshold.

### 5.4.1 DETECTING BOR

The $\overline{BOR}$ bit always resets to '0' on any VDDCORE, BOR or POR 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.

If the voltage regulator is disabled (LF devices), the VDDCORE BOR functionality is disabled. In this case, the $\overline{BOR}$ bit cannot be used to determine a Brown-out Reset event. The $\overline{BOR}$ bit is still cleared by a Power-on Reset event.

## 5.5 Configuration Mismatch (CM)

The Configuration Mismatch (CM) Reset is designed to detect, and attempt to recover from, random memory corrupting events. These include Electrostatic Discharge (ESD) events, which can cause widespread single bit changes throughout the device and result in catastrophic failure.

In PIC18FXXJ Flash devices, the device Configuration registers (located in the configuration memory space) are continuously monitored during operation by comparing their values to complimentary shadow registers. If a mismatch is detected between the two sets of registers, a CM Reset automatically occurs. These events are captured by the $\overline{CM}$ bit (RCON<5>). The state of the bit is set to '0' whenever a CM event occurs; it does not change for any other Reset event.

A CM Reset behaves similarly to a $\overline{MCLR}$, RESET instruction, WDT time-out or Stack Event Resets. As with all hard and power Reset events, the device Configuration Words are reloaded from the Flash Configuration Words in program memory as the device restarts.

## 5.6 Power-up Timer (PWRT)

PIC18F46J11 family devices incorporate an on-chip PWRT to help regulate the POR process. The PWRT is always enabled. The main function is to ensure that the device voltage is stable before code is executed.

The Power-up Timer (PWRT) of the PIC18F46J11 family devices is a 5-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of 32 x 32 μs = 1 ms. While the PWRT is counting, the device is held in Reset.

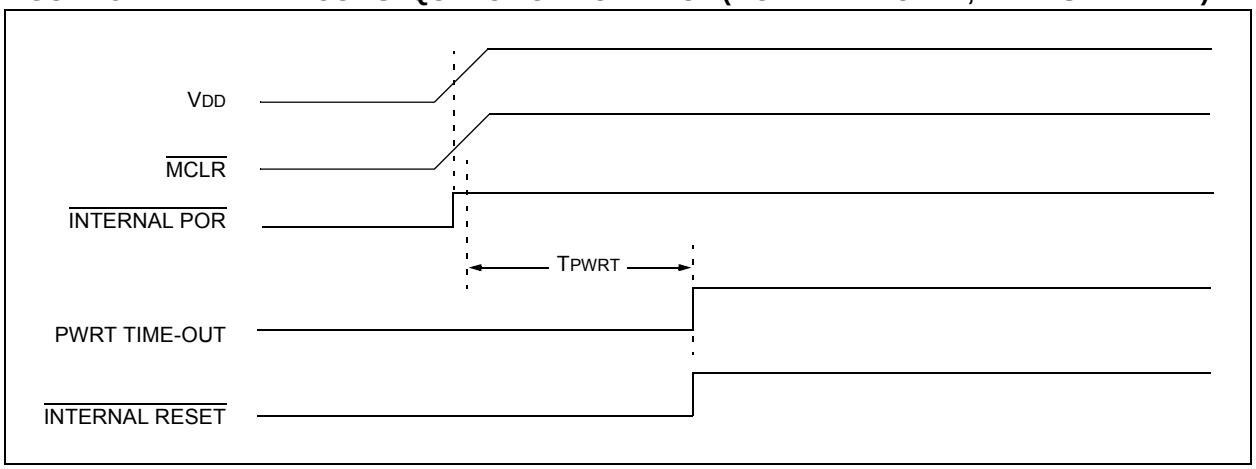
The power-up time delay depends on the INTRC clock and will vary from chip-to-chip due to temperature and process variation. See DC parameter 33 (TPWRT) for details.

### 5.6.1 TIME-OUT SEQUENCE

The PWRT time-out is invoked after the POR pulse has cleared. The total time-out will vary based on the status of the PWRT. Figure 5-2, Figure 5-3, Figure 5-4 and Figure 5-5 all depict time-out sequences on power-up with the PWRT.

Since the time-outs occur from the POR pulse, if $\overline{MCLR}$ is kept low long enough, the PWRT will expire. Bringing $\overline{MCLR}$ high will begin execution immediately if a clock source is available (Figure 5-4). This is useful for testing purposes, or to synchronize more than one PIC18FXXXX device operating in parallel.

**FIGURE 5-2: TIME-OUT SEQUENCE ON POWER-UP ($\overline{MCLR}$ TIED TO VDD, VDD RISE < TPWRT)**





© 2011 Microchip Technology Inc.

**TABLE 5-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register | Applicable Devices | | Power-on Reset, Brown-out Reset, Wake From Deep Sleep | $\overline{\text{MCLR}}$ Resets WDT Reset RESET Instruction Stack Resets CM Resets | Wake-up via WDT or Interrupt |
|---|---|---|---|---|---|
| RPINR8 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR7 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR6 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR4 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR3 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR2 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPINR1 | PIC18F2XJ11 | PIC18F4XJ11 | ---1 1111 | ---1 1111 | ---u uuuu |
| RPOR24 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR23 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR22 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR21 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR20 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR19 | — | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR18 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR17 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR16 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR15 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR14 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR13 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR12 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR11 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR10 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR9 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR8 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR7 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR6 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR5 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR4 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR3 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR2 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR1 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |
| RPOR0 | PIC18F2XJ11 | PIC18F4XJ11 | ---0 0000 | ---0 0000 | ---u uuuu |

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

**Note 1:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

**2:** When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

**3:** One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

**4:** See Table 5-1 for Reset value for specific condition.

**5:** Not implemented for PIC18F2XJ11 devices.

**6:** Not implemented on "LF" devices.

## 6.4 Data Addressing Modes

**Note:** The execution of some instructions in the core PIC18 instruction set is changed when the PIC18 extended instruction set is enabled. See **Section 6.6 "Data Memory and the Extended Instruction Set"** for more information.

While the program memory can be addressed in only one way, through the PC, information in the data memory space can be addressed in several ways. For most instructions, the addressing mode is fixed. Other instructions may use up to three modes, depending on which operands are used and whether or not the extended instruction set is enabled.

The addressing modes are:

- Inherent
- Literal
- Direct
- Indirect

An additional addressing mode, Indexed Literal Offset, is available when the extended instruction set is enabled (XINST Configuration bit = 1). Its operation is discussed in more detail in **Section 6.6.1 "Indexed Addressing with Literal Offset"**.

### 6.4.1 INHERENT AND LITERAL ADDRESSING

Many PIC18 control instructions do not need any argument at all; they either perform an operation that globally affects the device, or they operate implicitly on one register. This addressing mode is known as Inherent Addressing. Examples include SLEEP, RESET and DAW.

Other instructions work in a similar way, but require an additional explicit argument in the opcode. This is known as Literal Addressing mode, because they require some literal value as an argument. Examples include ADDLW and MOVLW, which respectively, add or move a literal value to the W register. Other examples include CALL and GOTO, which include a 20-bit program memory address.

### 6.4.2 DIRECT ADDRESSING

Direct Addressing specifies all or part of the source and/or destination address of the operation within the opcode itself. The options are specified by the arguments accompanying the instruction.

In the core PIC18 instruction set, bit-oriented and byte-oriented instructions use some version of Direct Addressing by default. All of these instructions include some 8-bit Literal Address as their LSB. This address specifies either a register address in one of the banks of data RAM (**Section 6.3.3 "General Purpose Register File"**), or a location in the Access Bank (**Section 6.3.2 "Access Bank"**) as the data source for the instruction.

The Access RAM bit, 'a', determines how the address is interpreted. When 'a' is '1', the contents of the BSR (**Section 6.3.1 "Bank Select Register"**) are used with the address to determine the complete 12-bit address of the register. When 'a' is '0', the address is interpreted as being a register in the Access Bank. Addressing that uses the Access RAM is sometimes also known as Direct Forced Addressing mode.

A few instructions, such as MOVFF, include the entire 12-bit address (either source or destination) in their opcodes. In these cases, the BSR is ignored entirely.

The destination of the operation's results is determined by the destination bit, 'd'. When 'd' is '1', the results are stored back in the source register, overwriting its original contents. When 'd' is '0', the results are stored in the W register. Instructions without the 'd' argument have a destination that is implicit in the instruction; their destination is either the target register being operated on or the W register.

### 6.4.3 INDIRECT ADDRESSING

Indirect Addressing allows the user to access a location in data memory without giving a fixed address in the instruction. This is done by using File Select Registers (FSRs) as pointers to the locations to be read or written to. Since the FSRs are themselves located in RAM as SFRs, they can also be directly manipulated under program control. This makes FSRs very useful in implementing data structures such as tables and arrays in data memory.

The registers for Indirect Addressing are also implemented with Indirect File Operands (INDFs) that permit automatic manipulation of the pointer value with auto-incrementing, auto-decrementing or offsetting with another value. This allows for efficient code using loops, such as the example of clearing an entire RAM bank in Example 6-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

**EXAMPLE 6-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING**

```
        LFSR    FSR0, 0x100 ;
NEXT    CLRF    POSTINC0    ; Clear INDF
                            ; register then
                            ; inc pointer
        BTFSS   FSR0H, 1    ; All done with
                            ; Bank1?
        BRA     NEXT        ; NO, clear next
CONTINUE                    ; YES, continue
```

Example 8-3 provides the instruction sequence for a 16 x 16 unsigned multiplication. Equation 8-1 provides the algorithm that is used. The 32-bit result is stored in four registers (RES<3:0>).

**EQUATION 8-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM**

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \cdot ARG2H:ARG2L \\
&= (ARG1H \cdot ARG2H \cdot 2^{16}) + \\
&\quad (ARG1H \cdot ARG2L \cdot 2^{8}) + \\
&\quad (ARG1L \cdot ARG2H \cdot 2^{8}) + \\
&\quad (ARG1L \cdot ARG2L)
\end{aligned}
$$

**EXAMPLE 8-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE**

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;

    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;

    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;

    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;
```

Example 8-4 provides the sequence to do a 16 x 16 signed multiply. Equation 8-2 provides the algorithm used. The 32-bit result is stored in four registers (RES<3:0>). To account for the sign bits of the arguments, the MSb for each argument pair is tested and the appropriate subtractions are done.

**EQUATION 8-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM**

$$
\begin{aligned}
RES3:RES0 &= ARG1H:ARG1L \cdot ARG2H:ARG2L \\
&= (ARG1H \cdot ARG2H \cdot 2^{16}) + \\
&\quad (ARG1H \cdot ARG2L \cdot 2^{8}) + \\
&\quad (ARG1L \cdot ARG2H \cdot 2^{8}) + \\
&\quad (ARG1L \cdot ARG2L) + \\
&\quad (-1 \cdot ARG2H\langle 7\rangle \cdot ARG1H:ARG1L \cdot 2^{16}) + \\
&\quad (-1 \cdot ARG1H\langle 7\rangle \cdot ARG2H:ARG2L \cdot 2^{16})
\end{aligned}
$$

**EXAMPLE 8-4: 16 x 16 SIGNED MULTIPLY ROUTINE**

```
    MOVF    ARG1L, W
    MULWF   ARG2L           ; ARG1L * ARG2L ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES1     ;
    MOVFF   PRODL, RES0     ;

    MOVF    ARG1H, W
    MULWF   ARG2H           ; ARG1H * ARG2H ->
                            ; PRODH:PRODL
    MOVFF   PRODH, RES3     ;
    MOVFF   PRODL, RES2     ;

    MOVF    ARG1L, W
    MULWF   ARG2H           ; ARG1L * ARG2H ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;

    MOVF    ARG1H, W        ;
    MULWF   ARG2L           ; ARG1H * ARG2L ->
                            ; PRODH:PRODL
    MOVF    PRODL, W        ;
    ADDWF   RES1, F         ; Add cross
    MOVF    PRODH, W        ; products
    ADDWFC  RES2, F         ;
    CLRF    WREG            ;
    ADDWFC  RES3, F         ;

    BTFSS   ARG2H, 7        ; ARG2H:ARG2L neg?
    BRA     SIGN_ARG1       ; no, check ARG1
    MOVF    ARG1L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG1H, W        ;
    SUBWFB  RES3

SIGN_ARG1
    BTFSS   ARG1H, 7        ; ARG1H:ARG1L neg?
    BRA     CONT_CODE       ; no, done
    MOVF    ARG2L, W        ;
    SUBWF   RES2            ;
    MOVF    ARG2H, W        ;
    SUBWFB  RES3

CONT_CODE
   :
```


© 2011 Microchip Technology Inc.

**REGISTER 10-33: RPOR12: PERIPHERAL PIN SELECT OUTPUT REGISTER 12 (BANKED ED2h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP12R4 | RP12R3 | RP12R2 | RP12R1 | RP12R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP12R<4:0>:** Peripheral Output Function is Assigned to RP12 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**REGISTER 10-34: RPOR13: PERIPHERAL PIN SELECT OUTPUT REGISTER 13 (BANKED ED3h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP13R4 | RP13R3 | RP13R2 | RP13R1 | RP13R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP13R<4:0>:** Peripheral Output Function is Assigned to RP13 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**REGISTER 10-35: RPOR14: PERIPHERAL PIN SELECT OUTPUT REGISTER 14 (BANKED ED4h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP14R4 | RP14R3 | RP14R2 | RP14R1 | RP14R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP14R<4:0>:** Peripheral Output Function is Assigned to RP14 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**REGISTER 10-39: RPOR18: PERIPHERAL PIN SELECT OUTPUT REGISTER 18 (BANKED ED8h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP18R4 | RP18R3 | RP18R2 | RP18R1 | RP18R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP18R<4:0>:** Peripheral Output Function is Assigned to RP18 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**REGISTER 10-40: RPOR19: PERIPHERAL PIN SELECT OUTPUT REGISTER 19 (BANKED ED9h)(1)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP19R4 | RP19R3 | RP19R2 | RP19R1 | RP19R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP19R<4:0>:** Peripheral Output Function is Assigned to RP19 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**Note 1:** RP19 pins are not available on 28-pin devices.

**REGISTER 10-41: RPOR20: PERIPHERAL PIN SELECT OUTPUT REGISTER 20 (BANKED EDAh)(1)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 | R/$\overline{W}$-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP20R4 | RP20R3 | RP20R2 | RP20R1 | RP20R0 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | R/$\overline{W}$ = Readable, Writable if IOLOCK = 0 | | |
|---|---|---|---|
| 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 4-0 **RP20R<4:0>:** Peripheral Output Function is Assigned to RP20 Output Pin bits
(see Table 10-14 for peripheral function numbers)

**Note 1:** RP20 pins are not available on 28-pin devices.

## 15.6 Timer3 Interrupt

The TMR3 register pair (TMR3H:TMR3L) increments from 0000h to FFFFh and overflows to 0000h. The Timer3 interrupt, if enabled, is generated on overflow and is latched in interrupt flag bit, TMR3IF (PIR2<1>). This interrupt can be enabled or disabled by setting or clearing the Timer3 Interrupt Enable bit, TMR3IE (PIE2<1>).

## 15.7 Resetting Timer3 Using the ECCP Special Event Trigger

If ECCP1 or ECCP2 is configured to use Timer3 and to generate a Special Event Trigger in Compare mode (CCPxM<3:0> = 1011), this signal will reset Timer3. The trigger from ECCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 18.3.4 "Special Event Trigger"** for more information).

The module must be configured as either a timer or synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a Period register for Timer3.

If Timer3 is running in Asynchronous Counter mode, the Reset operation may not work.

In the event that a write to Timer3 coincides with a Special Event Trigger from an ECCP module, the write will take precedence.

| **Note:** | The Special Event Triggers from the ECCPx module will not set the TMR3IF interrupt flag bit (PIR1<0>). |
|---|---|

**TABLE 15-3: REGISTERS ASSOCIATED WITH TIMER3 AS A TIMER/COUNTER**

| Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page: |
|---|---|---|---|---|---|---|---|---|---|
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE | TMR0IF | INT0IF | RBIF | 90 |
| PIR2 | OSCFIF | CM2IF | CM1IF | — | BCL1IF | LVDIF | TMR3IF | CCP2IF | 92 |
| PIE2 | OSCFIE | CM2IE | CM1IE | — | BCL1IE | LVDIE | TMR3IE | CCP2IE | 92 |
| IPR2 | OSCFIP | CM2IP | CM1IP | — | BCL1IP | LVDIP | TMR3IP | CCP2IP | 92 |
| TMR3L | Timer3 Register Low Byte | | | | | | | | 93 |
| TMR3H | Timer3 Register High Byte | | | | | | | | 93 |
| T1CON | TMR1CS1 | TMR1CS0 | T1CKPS1 | T1CKPS0 | T1OSCEN | $\overline{\text{T1SYNC}}$ | RD16 | TMR1ON | 91 |
| T3CON | TMR3CS1 | TMR3CS0 | T3CKPS1 | T3CKPS0 | — | $\overline{\text{T3SYNC}}$ | RD16 | TMR3ON | 93 |
| T3GCON | TMR3GE | T3GPOL | T3GTM | T3GSPM | T3GGO/$\overline{\text{T3DONE}}$ | T3GVAL | T3GSS1 | T3GSS0 | 92 |
| TCLKCON | — | — | — | T1RUN | — | — | T3CCP2 | T3CCP1 | 94 |
| PIR3 | SSP2IF | BCL2IF | RC2IF | TX2IF | TMR4IF | CTMUIF | TMR3GIF | RTCCIF | 92 |
| PIE3 | SSP2IE | BCL2IE | RC2IE | TX2IE | TMR4IE | CTMUIE | TMR3GIE | RTCCIE | 92 |
| IPR3 | SSP2IP | BCL2IP | RC2IP | TX2IP | TMR4IP | CTMUIP | TMR3GIP | RTCCIP | 92 |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Timer3 module.

**REGISTER 17-11: HOURS: HOURS VALUE REGISTER (ACCESS F98h, PTR 01b)(1)**

| U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | — | HRTEN1 | HRTEN0 | HRONE3 | HRONE2 | HRONE1 | HRONE0 |
| 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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit bits
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit bits
Contains a value from 0 to 9.

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

**REGISTER 17-12: MINUTES: MINUTES VALUE REGISTER (ACCESS F99h, PTR 00b)**

| U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | MINTEN2 | MINTEN1 | MINTEN0 | MINONE3 | MINONE2 | MINONE1 | MINONE0 |
| 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 6-4 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits
Contains a value from 0 to 9.

**REGISTER 17-13: SECONDS: SECONDS VALUE REGISTER (ACCESS F98h, PTR 00b)**

| U-0 | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|---|---|---|---|---|---|---|---|
| — | SECTEN2 | SECTEN1 | SECTEN0 | SECONE3 | SECONE2 | SECONE1 | SECONE0 |
| 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 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits
Contains a value from 0 to 9.

#### 19.5.7.1 Baud Rate and Module Interdependence

Because MSSP1 and MSSP2 are independent, they can operate simultaneously in $I^2C$ Master mode at different baud rates. This is done by using different BRG reload values for each module.

Because this mode derives its basic clock source from the system clock, any changes to the clock will affect both modules in the same proportion. It may be possible to change one or both baud rates back to a previous value by changing the BRG reload value.

**FIGURE 19-19: BAUD RATE GENERATOR BLOCK DIAGRAM**

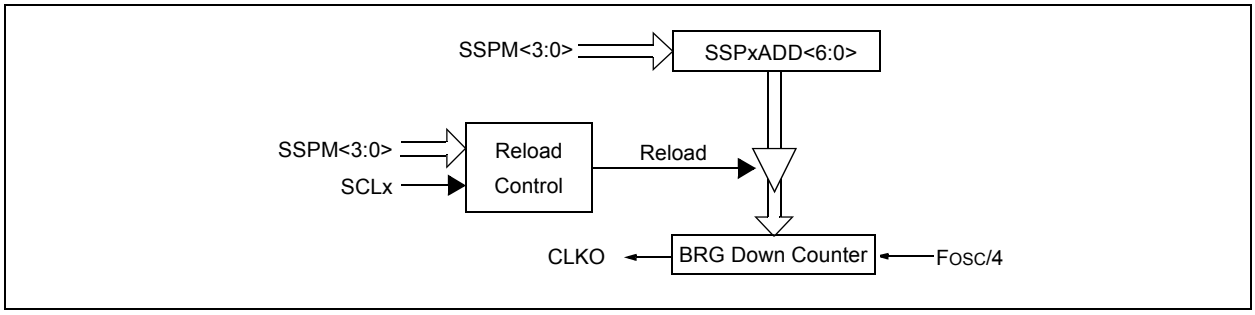



**TABLE 19-3: $I^2C$™ CLOCK RATE w/BRG**

| 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 | 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.

The ANCON0 and ANCON1 registers are used to configure the operation of the I/O pin associated with each analog channel. Setting any one of the PCFG bits configures the corresponding pin to operate as a digital only I/O. Clearing a bit configures the pin to operate as an analog input for either the A/D Converter or the comparator module; all digital peripherals are disabled and digital inputs read as '0'. As a rule, I/O pins that are multiplexed with analog inputs default to analog operation on device Resets.

In order to correctly perform A/D conversions on the VBG band gap reference (ADCON0<5:2> = 1111), the reference circuit must be powered on first. The VBGEN bit in the ANCON1 register allows the firmware to manually request that the band gap reference circuit should be enabled. For best accuracy, firmware should allow a settling time of at least 10 ms prior to performing the first acquisition on this channel after enabling the band gap reference.

The reference circuit may already have been turned on if some other hardware module (such as comparators or HLVD) has already requested it. In this case, the initial turn-on settling time may have already elapsed and firmware does not need to wait as long before measuring VBG. Once the acquisition is complete, firmware may clear the VBGEN bit, which will save a small amount of power if no other modules are still requesting the VBG reference.

**REGISTER 21-3: ANCON0: A/D PORT CONFIGURATION REGISTER 2 (BANKED F48h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| PCFG7(1) | PCFG6(1) | PCFG5(1) | PCFG4 | PCFG3 | PCFG2 | PCFG1 | PCFG0 |
| 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 **PCFG<7:0>:** Analog Port Configuration bits (AN<7:0>)
1 = Pin configured as a digital port
0 = Pin configured as an analog channel – digital input disabled and reads '0'

**Note 1:** These bits are not implemented on 28-pin devices.

**REGISTER 21-4: ANCON1: A/D PORT CONFIGURATION REGISTER 1 (BANKED F49h)**

| R/W-0 | r | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| VBGEN | — | — | PCFG12 | PCFG11 | PCFG10 | PCFG9 | PCFG8 |
| bit 7 | | | | | | | bit 0 |

| Legend: | r = Reserved | | |
|---|---|---|---|
| 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 **VBGEN:** 1.2V Band Gap Reference Enable bit
1 = 1.2V band gap reference is powered on
0 = 1.2V band gap reference is turned off to save power (if no other modules are requesting it)

bit 6 **Reserved:** Always maintain as '0' for lowest power consumption

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

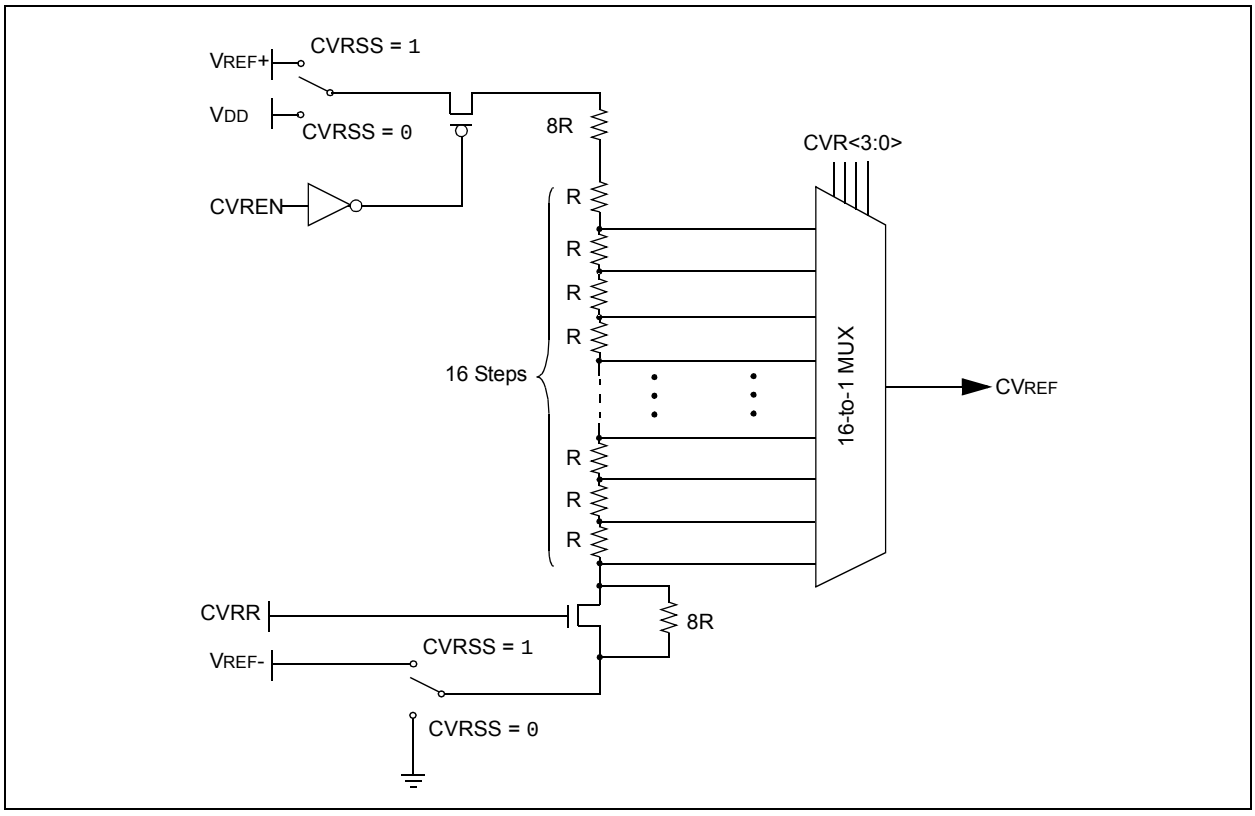
bit 4-0 **PCFG<12:8>:** Analog Port Configuration bits (AN<12:8>)
1 = Pin configured as a digital port
0 = Pin configured as an analog channel – digital input disabled and reads '0'

## 23.0 COMPARATOR VOLTAGE REFERENCE MODULE

The comparator voltage reference is a 16-tap resistor ladder network that provides a selectable reference voltage. Although its primary purpose is to provide a reference for the analog comparators, it may also be used independently of them.

Figure 23-1 provides a block diagram of the module. The resistor ladder is segmented to provide two ranges of $CV_{REF}$ values and has a power-down function to conserve power when the reference is not being used. The module's supply reference can be provided from either device $V_{DD}/V_{SS}$ or an external voltage reference.

**FIGURE 23-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



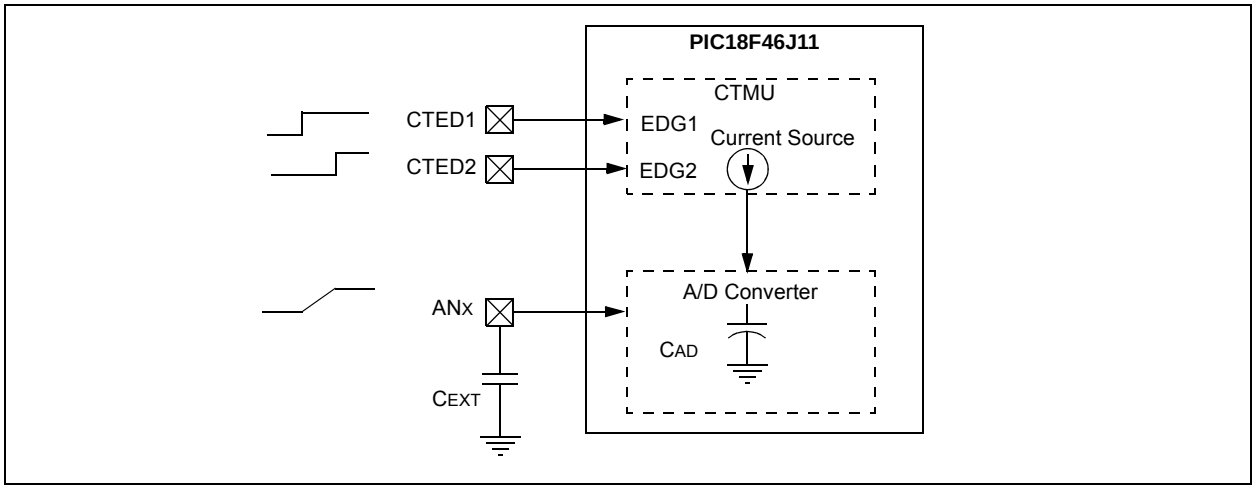
## 25.5 Measuring Time with the CTMU Module

Time can be precisely measured after the ratio ($C/I$) is measured from the current and capacitance calibration step by following these steps:

1. Initialize the A/D Converter and the CTMU.
2. Set EDG1STAT.
3. Set EDG2STAT.
4. Perform an A/D conversion.
5. Calculate the time between edges as $T = (C/I) * V$, where $I$ is calculated in the current calibration step (**Section 25.3.1 "Current Source Calibration"**), $C$ is calculated in the capacitance calibration step (**Section 25.3.2 "Capacitance Calibration"**) and $V$ is measured by performing the A/D conversion.

It is assumed that the time measured is small enough that the capacitance, $C_{AD}$ + $C_{EXT}$, provides a valid voltage to the A/D Converter. For the smallest time measurement, always set the A/D Channel Select register (AD1CHS) to an unused A/D channel; the corresponding pin for which is not connected to any circuit board trace. This minimizes added stray capacitance, keeping the total circuit capacitance close to that of the A/D Converter itself (4-5 pF). To measure longer time intervals, an external capacitor may be connected to an A/D channel and this channel selected when making a time measurement.

**FIGURE 25-3: TYPICAL CONNECTIONS AND INTERNAL CONFIGURATION FOR TIME MEASUREMENT**



**DECFSZ** **Decrement f, Skip if 0**

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

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

Operation: (f) – 1 → dest,
skip if result = 0

Status Affected: None

Encoding:

| 0010 | 11da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If the result is '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

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 $f \le 95$ (5Fh). See **Section 27.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    DECFSZ  CNT, 1, 1
        GOTO    LOOP
CONTINUE
```
Before Instruction
PC = Address (HERE)
After Instruction
CNT = CNT – 1
If CNT = 0;
PC = Address (CONTINUE)
If CNT ≠ 0;
PC = Address (HERE + 2)

**DCFSNZ** **Decrement f, Skip if not 0**

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

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

Operation: (f) – 1 → dest,
skip if result ≠ 0

Status Affected: None

Encoding:

| 0100 | 11da | ffff | ffff |
|---|---|---|---|

Description: The contents of register 'f' are decremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If the result is not '0', the next instruction which is already fetched is discarded and a NOP is executed instead, making it a two-cycle instruction.

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 $f \le 95$ (5Fh). See **Section 27.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1(2)
**Note:** 3 cycles if skip and followed by a 2-word instruction.

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Read register 'f' | Process Data | Write to destination |

If skip:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |

If skip and followed by 2-word instruction:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| No operation | No operation | No operation | No operation |
| No operation | No operation | No operation | No operation |

Example:
```
HERE    DCFSNZ  TEMP, 1, 0
ZERO    :
NZERO   :
```
Before Instruction
TEMP = ?
After Instruction
TEMP = TEMP – 1,
If TEMP = 0;
PC = Address (ZERO)
If TEMP ≠ 0;
PC = Address (NZERO)

## 29.2 DC Characteristics: Power-Down and Supply Current PIC18F46J11 Family (Industrial)

| PIC18LFXXJ11 Family | | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
|---|---|---|---|---|---|---|---|
| **PIC18FXXJ11 Family** | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | | |
| **Param No.** | **Device** | **Typ** | **Max** | **Units** | **Conditions** | | |
| | **Power-Down Current (IPD)(1)** – Sleep mode | | | | | | |
| | PIC18LFXXJ11 | 0.011 | 1.4 | µA | -40°C | VDD = 2.0V, VDDCORE = 2.0V | **Sleep** mode, REGSLP = 1 |
| | | 0.054 | 1.4 | µA | +25°C | | |
| | | 0.51 | 6 | µA | +60°C | | |
| | | 2.0 | 10.2 | µA | +85°C | | |
| | PIC18LFXXJ11 | 0.029 | 1.5 | µA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | |
| | | 0.11 | 1.5 | µA | +25°C | | |
| | | 0.63 | 8 | µA | +60°C | | |
| | | 2.30 | 12.6 | µA | +85°C | | |
| | PIC18FXXJ11 | 2.5 | 6 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | |
| | | 3.1 | 6 | µA | +25°C | | |
| | | 3.9 | 8 | µA | +60°C | | |
| | | 5.6 | 16 | µA | +85°C | | |
| | PIC18FXXJ11 | 4.1 | 7 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | |
| | | 3.3 | 7 | µA | +25°C | | |
| | | 4.1 | 10 | µA | +60°C | | |
| | | 6.0 | 19 | µA | +85°C | | |
| | **Power-Down Current (IPD)(1)** – Deep Sleep mode | | | | | | |
| | PIC18FXXJ11 | 1 | 25 | nA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | **Deep Sleep** mode |
| | | 13 | 100 | nA | +25°C | | |
| | | 108 | 250 | nA | +60°C | | |
| | | 428 | 1000 | nA | +85°C | | |
| | PIC18FXXJ11 | 3 | 50 | nA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | |
| | | 28 | 150 | nA | +25°C | | |
| | | 170 | 389 | nA | +60°C | | |
| | | 588 | 2000 | nA | +85°C | | |

**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 high-impedance state and tied to VDD or VSS and all features that add delta current disabled (such as WDT, Timer1 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. All features that add delta current are disabled (WDT, etc.). 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/VSS;
$\overline{MCLR}$ = VDD; WDT disabled unless otherwise specified.

**3:** Low-Power Timer1 with 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.


© 2011 Microchip Technology Inc.

**TABLE 29-5: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS**

| **Operating Conditions:** -40°C < TA < +85°C (unless otherwise stated) | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristics** | **Min** | **Typ** | **Max** | **Units** | **Comments** |
| | VRGOUT | Regulator Output Voltage | 2.35 | 2.5 | 2.7 | V | Regulator enabled, VDD = 3.0V |
| | CEFC | External Filter Capacitor Value(1) | 5.4 | 10 | 18 | µF | ESR < 3Ω recommended<br>ESR < 5Ω required |

**Note 1:** CEFC applies for PIC18F devices in the family. For PIC18LF devices in the family, there is no specific minimum or maximum capacitance for VDDCORE, although proper supply rail bypassing should still be used.

**TABLE 29-6: ULPWU SPECIFICATIONS**

| **DC CHARACTERISTICS** | | | **Standard Operating Conditions (unless otherwise stated)**<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial | | | | |
|---|---|---|---|---|---|---|---|
| **Param No.** | **Sym** | **Characteristic** | **Min** | **Typ†** | **Max** | **Units** | **Conditions** |
| D100 | IULP | **Ultra Low-Power Wake-up Current** | — | 60 | — | nA | Net of I/O leakage and current sink at 1.6V on pin,<br>VDD = 3.3V<br>See Application Note *AN879,* "*Using the Microchip Ultra Low-Power Wake-up Module*" (DS00879) |

† Data in "Typ" column is at 3.3V, 25°C unless otherwise stated.

**TABLE 29-30: A/D CONVERTER CHARACTERISTICS: PIC18F46J11 FAMILY (INDUSTRIAL)**

| Param No. | Symbol | Characteristic | Min | Typ | Max | Units | Conditions |
|---|---|---|---|---|---|---|---|
| A01 | NR | Resolution | — | — | 10 | bit | $\Delta V_{REF} \geq 3.0V$ |
| A03 | EIL | Integral Linearity Error | — | — | <±1 | LSb | $\Delta V_{REF} \geq 3.0V$ |
| A04 | EDL | Differential Linearity Error | — | — | <±1 | LSb | $\Delta V_{REF} \geq 3.0V$ |
| A06 | EOFF | Offset Error | — | — | <±3 | LSb | $\Delta V_{REF} \geq 3.0V$ |
| A07 | EGN | Gain Error | — | — | <±3.5 | LSb | $\Delta V_{REF} \geq 3.0V$ |
| A10 | | Monotonicity | Guaranteed(1) | | | — | $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A20 | ΔVREF | Reference Voltage Range (VREFH – VREFL) | 2.0<br>3 | —<br>— | —<br>— | V<br>V | $V_{DD} < 3.0V$<br>$V_{DD} \geq 3.0V$ |
| A21 | VREFH | Reference Voltage High | VREFL | — | VDD + 0.3V | V | |
| A22 | VREFL | Reference Voltage Low | VSS – 0.3V | — | VREFH | V | |
| A25 | VAIN | Analog Input Voltage | VREFL | — | VREFH | V | |
| A30 | ZAIN | Recommended Impedance of Analog Voltage Source | — | — | 2.5 | kΩ | |
| A50 | IREF | VREF Input Current(2) | —<br>— | —<br>— | 5<br>150 | μA<br>μA | During VAIN acquisition.<br>During A/D conversion cycle. |

**Note 1:** The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

**2:** VREFH current is from RA3/AN3/VREF+/C1INB pin or VDD, whichever is selected as the VREFH source. VREFL current is from RA2/AN2/VREF-/CVREF/C2INB pin or VSS, whichever is selected as the VREFL source.

## PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

| PART NO. | X | /XX | XXX |
|---|---|---|---|
| Device | Temperature Range | Package | Pattern |

| | |
|---|---|
| Device[1] | PIC18F24J11<br>PIC18F25J11<br>PIC18F26J11<br>PIC18F44J11<br>PIC18F45J11<br>PIC18F46J11<br>PIC18LF24J11<br>PIC18LF25J11<br>PIC18LF26J11<br>PIC18LF44J11<br>PIC18LF45J11<br>PIC18LF46J11 |
| Temperature Range | I = -40°C to +85°C (Industrial) |
| Package | SP = Skinny PDIP<br>SS = SSOP<br>SO = SOIC<br>ML = QFN<br>PT = TQFP (Thin Quad Flatpack) |
| Pattern | QTP, SQTP, Code or Special Requirements (blank otherwise) |

**Examples:**

a) PIC18F46J11-I/PT 301 = Industrial temp., TQFP package, QTP pattern #301.

b) PIC18F46J11T-I/PT = Tape and reel, Industrial temp., TQFP package.

**Note 1:** F = Standard Voltage Range
LF = Extended Voltage Range

**2:** T = In tape and reel