**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 | 16 |
| Program Memory Size | 16KB (8K 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 10x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 28-SSOP (0.209", 5.30mm Width) |
| Supplier Device Package | 28-SSOP |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18lf24j11t-i-ss |

**REGISTER 4-3: DSGPR0: DEEP SLEEP PERSISTENT GENERAL PURPOSE REGISTER 0 (BANKED F4Eh)**

| R/W-xxxx(1) | |
|---|---|
| Deep Sleep Persistent General Purpose bits | |
| 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 Deep Sleep Persistent General Purpose bits
Contents are retained even in Deep Sleep mode.

**Note 1:** All register bits are maintained unless: $V_{DDCORE}$ drops below the normal BOR threshold outside of Deep Sleep, or the device is in Deep Sleep and the dedicated DSBOR is enabled and $V_{DD}$ drops below the DSBOR threshold, or DSBOR is enabled or disabled, but $V_{DD}$ is hard cycled to near $V_{SS}$.

**REGISTER 4-4: DSGPR1: DEEP SLEEP PERSISTENT GENERAL PURPOSE REGISTER 1 (BANKED F4Fh)**

| R/W-xxxx(1) | |
|---|---|
| Deep Sleep Persistent General Purpose bits | |
| 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 Deep Sleep Persistent General Purpose bits
Contents are retained even in Deep Sleep mode.

**Note 1:** All register bits are maintained unless: $V_{DDCORE}$ drops below the normal BOR threshold outside of Deep Sleep, or, the device is in Deep Sleep and the dedicated DSBOR is enabled and $V_{DD}$ drops below the DSBOR threshold, or DSBOR is enabled or disabled, but $V_{DD}$ is hard cycled to near $V_{SS}$.

**EXAMPLE 4-1: ULTRA LOW-POWER WAKE-UP INITIALIZATION**

```
//*****************************************************************************
//Configure a remappable output pin with interrupt capability
//for ULPWU function (RP21 => RD4/INT1 in this example)
//*****************************************************************************
RPOR21 = 13;// ULPWU function mapped to RP21/RD4
RPINR1 = 21;// INT1 mapped to RP21 (RD4)

//***************************
//Charge the capacitor on RA0
//***************************
TRISAbits.TRISA0 = 0;
LATAbits.LATA0 = 1;
for(i = 0; i < 10000; i++) Nop();

//*********************************
//Stop Charging the capacitor on RA0
//*********************************
TRISAbits.TRISA0 = 1;

//****************************************
//Enable the Ultra Low Power Wakeup module
//and allow capacitor discharge
//****************************************
WDTCONbits.ULPEN = 1;
WDTCONbits.ULPSINK = 1;

//****************************************
//For Sleep,  Enable Interrupt for ULPW.
//****************************************
INTCON3bits.INT1IF = 0;
INTCON3bits.INT1IE = 1;

//********************
//Configure Sleep Mode
//********************
//For Sleep
OSCCONbits.IDLEN = 0;

//For Deep Sleep
OSCCONbits.IDLEN = 0;// enable deep sleep
DSCONHbits.DSEN = 1;// Note: must be set just before executing Sleep();
//*****************
//Enter Sleep Mode
//*****************
Sleep();
    // for sleep, execution will resume here
    // for deep sleep, execution will restart at reset vector (use WDTCONbits.DS to detect)
```

**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 |
|---|---|---|---|---|---|
| BAUDCON1 | PIC18F2XJ11 | PIC18F4XJ11 | 0100 0-00 | 0100 0-00 | uuuu u-uu |
| SPBRGH2 | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| BAUDCON2 | PIC18F2XJ11 | PIC18F4XJ11 | 0100 0-00 | 0100 0-00 | uuuu u-uu |
| TMR3H | PIC18F2XJ11 | PIC18F4XJ11 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| TMR3L | PIC18F2XJ11 | PIC18F4XJ11 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| T3CON | PIC18F2XJ11 | PIC18F4XJ11 | 0000 -000 | uuuu -uuu | uuuu -uuu |
| TMR4 | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | uuuu uuuu | uuuu uuuu |
| PR4 | PIC18F2XJ11 | PIC18F4XJ11 | 1111 1111 | 1111 1111 | uuuu uuuu |
| T4CON | PIC18F2XJ11 | PIC18F4XJ11 | -000 0000 | -000 0000 | -uuu uuuu |
| SSP2BUF | PIC18F2XJ11 | PIC18F4XJ11 | xxxx xxxx | uuuu uuuu | uuuu uuuu |
| SSP2ADD | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSP2MSK | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSP2STAT | PIC18F2XJ11 | PIC18F4XJ11 | 1111 1111 | 1111 1111 | uuuu uuuu |
| SSP2CON1 | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| SSP2CON2 | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| CMSTAT | PIC18F2XJ11 | PIC18F4XJ11 | ---- --11 | ---- --11 | ---- --uu |
| PMADDRH(5) | — | PIC18F4XJ11 | -000 0000 | -000 0000 | -uuu uuuu |
| PMDOUT1H(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMADDRL(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMDOUT1L(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMDIN1H(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMDIN1L(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TXADDRL | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| TXADDRH | PIC18F2XJ11 | PIC18F4XJ11 | ---- 0000 | ---- 0000 | ---- uuuu |
| RXADDRL | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| RXADDRH | PIC18F2XJ11 | PIC18F4XJ11 | ---- 0000 | ---- 0000 | ---- uuuu |
| DMABCL | PIC18F2XJ11 | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| DMABCH | PIC18F2XJ11 | PIC18F4XJ11 | ---- --00 | ---- --00 | ---- --uu |
| PMCONH(5) | — | PIC18F4XJ11 | 0--0 0000 | 0--0 0000 | u--u uuuu |
| PMCONL(5) | — | PIC18F4XJ11 | 000- 0000 | 000- 0000 | uuu- uuuu |
| PMMODEH(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu uuuu |
| PMMODEL(5) | — | PIC18F4XJ11 | 0000 0000 | 0000 0000 | uuuu 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.

## 6.2 PIC18 Instruction Cycle

### 6.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by '4' to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the PC is incremented on every Q1; the instruction is fetched from the program memory and latched into the Instruction Register (IR) during Q4. The instruction is decoded and executed during the following Q1 through Q4. Figure 6-4 illustrates the clocks and instruction execution flow.

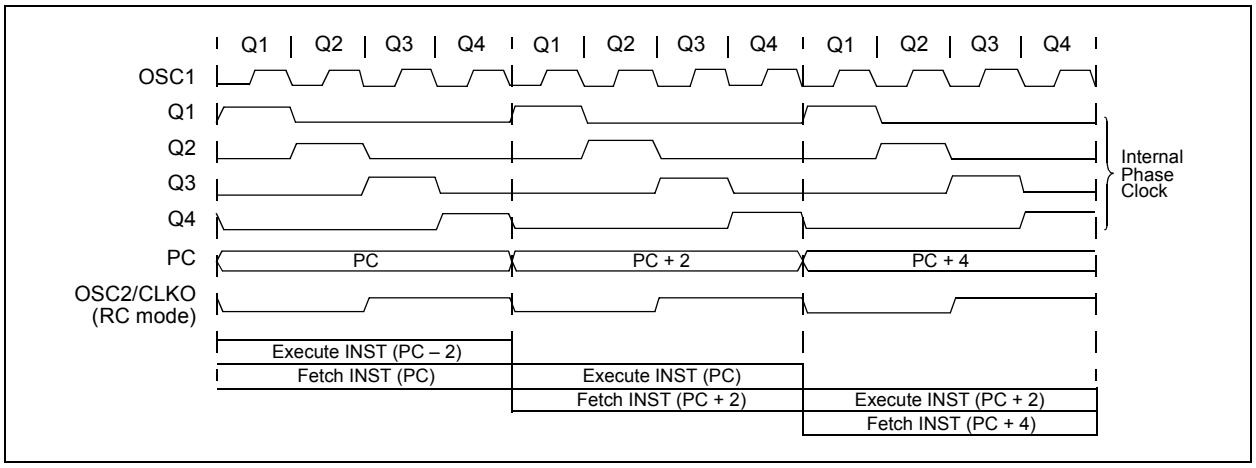
### 6.2.2 INSTRUCTION FLOW/PIPELINING

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 PC to change (e.g., GOTO), then two cycles are required to complete the instruction (Example 6-3).

A fetch cycle begins with the PC incrementing in Q1.

In the execution cycle, the fetched instruction is latched into the 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 6-4: CLOCK/INSTRUCTION CYCLE**




**EXAMPLE 6-3: INSTRUCTION PIPELINE FLOW**

| | TCY0 | TCY1 | TCY2 | TCY3 | TCY4 | TCY5 |
|---|---|---|---|---|---|---|
| 1. MOVLW 55h | Fetch 1 | Execute 1 | | | | |
| 2. MOVWF LATB | | Fetch 2 | Execute 2 | | | |
| 3. BRA SUB_1 | | | Fetch 3 | Execute 3 | | |
| 4. BSF LATA, 3 (Forced NOP) | | | | Fetch 4 | Flush (NOP) | |
| 5. Instruction @ address SUB_1 | | | | | Fetch SUB_1 | Execute SUB_1 |

**Note:** All instructions are single-cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

**TABLE 6-4: REGISTER FILE SUMMARY (PIC18F46J11 FAMILY)**

| File Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on POR, BOR | Details on Page: |
|---|---|---|---|---|---|---|---|---|---|---|
| DMACON2 | DLYCYC3 | DLYCYC2 | DLYCYC1 | DLYCYC0 | INTLVL3 | INTLVL2 | INTLVL1 | INTLVL0 | 0000 0000 | 72, 285 |
| HLVDCON | VDIRMAG | BGVST | IRVST | HLVDEN | HLVDL3 | HLVDL2 | HLVDL1 | HLVDL0 | 0000 0000 | 72 |
| PORTE | RDPU | REPU | — | — | — | RE2 | RE1 | RE0 | 00-- -xxx | 72 |
| PORTD | RD7 | RD6 | RD5 | RD4 | RD3 | RD2 | RD1 | RD0 | xxxx xxxx | 72 |
| PORTC | RC7 | RC6 | RC5 | RC4 | RC4 | RC2 | RC1 | RC0 | xxxx xxxx | 72 |
| PORTB | RB7 | RB6 | RB5 | RB4 | RB3 | RB2 | RB1 | RB0 | xxxx xxxx | 72 |
| PORTA | RA7 | RA6 | RA5 | — | RA3 | RA2 | RA1 | RA0 | xxx- xxxx | 72 |
| SPBRGH1 | EUSART1 Baud Rate Generator Register High Byte | | | | | | | | 0000 0000 | 72 |
| BAUDCON1 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 0100 0-00 | 72, 330 |
| SPBRGH2 | EUSART2 Baud Rate Generator Register High Byte | | | | | | | | 0000 0000 | 72 |
| BAUDCON2 | ABDOVF | RCIDL | RXDTP | TXCKP | BRG16 | — | WUE | ABDEN | 0100 0-00 | 72, 330 |
| TMR3H | Timer3 Register High Byte | | | | | | | | xxxx xxxx | 73 |
| TMR3L | Timer3 Register Low Byte | | | | | | | | xxxx xxxx | 73 |
| T3CON | TMR3CS1 | TMR3CS0 | T3CKPS1 | T3CKPS0 | — | $\overline{\text{T3SYNC}}$ | RD16 | TMR3ON | 0000 -000 | 73, 215 |
| TMR4 | Timer4 Register | | | | | | | | 0000 0000 | 73 |
| **PR4** | Timer4 Period Register | | | | | | | | 1111 1111 | 73 |
| T4CON | — | T4OUTPS3 | T4OUTPS2 | T4OUTPS1 | T4OUTPS0 | TMR4ON | T4CKPS1 | T4CKPS0 | -000 0000 | 73, 225 |
| SSP2BUF | MSSP2 Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | 73 |
| SSP2ADD/ SSP2MSK(4) | MSSP2 Address Register ($I^2C$™ Slave mode), MSSP2 Baud Rate Reload Register ($I^2C$ Master mode) | | | | | | | | 0000 0000 | 73, 295 |
| | MSK7 | MSK6 | MSK5 | MSK4 | MSK3 | MSK2 | MSK1 | MSK0 | 1111 1111 | 73, 295 |
| SSP2STAT | SMP | CKE | D/$\overline{\text{A}}$ | P | S | R/$\overline{\text{W}}$ | UA | BF | 0000 0000 | 73, 273 |
| SSP2CON1 | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 73, 293 |
| SSP2CON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 73, 294 |
| | GCEN | ACKSTAT | ADMSK5(4) | ADMSK4(4) | ADMSK3(4) | ADMSK2(4) | ADMSK1(4) | SEN | | |
| CMSTAT | — | — | — | — | — | — | COUT2 | COUT1 | ---- --11 | 73, 363 |
| PMADDRH/ | — | CS1 | Parallel Master Port Address High Byte | | | | | | -000 0000 | 73, 179 |
| PMDOUT1H(5) | Parallel Port Out Data High Byte (Buffer 1) | | | | | | | | 0000 0000 | 73, 179 |
| PMADDRL/ | Parallel Master Port Address Low Byte | | | | | | | | 0000 0000 | 73, 179 |
| PMDOUT1L(5) | Parallel Port Out Data Low Byte (Buffer 0) | | | | | | | | 0000 0000 | 73, 179 |
| PMDIN1H(5) | Parallel Port In Data High Byte (Buffer 1) | | | | | | | | 0000 0000 | 73 |
| PMDIN1L(5) | Parallel Port In Data Low Byte (Buffer 0) | | | | | | | | 0000 0000 | 73 |
| TXADDRL | SPI DMA Transit Data Pointer Low Byte | | | | | | | | 0000 0000 | 73 |
| TXADDRH | — | — | — | — | SPI DMA Transit Data Pointer High Byte | | | | ---- 0000 | 73 |
| RXADDRL | SPI DMA Receive Data Pointer Low Byte | | | | | | | | 0000 0000 | 73 |
| RXADDRH | — | — | — | — | SPI DMA Receive Data Pointer High Byte | | | | ---- 0000 | 73 |
| DMABCL | SPI DMA Byte Count Low Byte | | | | | | | | 0000 0000 | 73 |
| DMABCH | — | — | — | — | — | — | SPI DMA Receive Data Pointer High Byte | | ---- --00 | 73 |
| PMCONH(5) | PMPEN | — | — | ADRMUX1 | ADRMUX0 | PTBEEN | PTWREN | PTRDEN | 0--0 0000 | 73, 172 |
| PMCONL(5) | CSF1 | CSF0 | ALP | — | CS1P | BEP | WRSP | RDSP | 000- 0000 | 73, 173 |
| PMMODEH(5) | BUSY | IRQM1 | IRQM0 | INCM1 | INCM0 | MODE16 | MODE1 | MODE0 | 0000 0000 | 73, 174 |
| PMMODEL(5) | WAITB1 | WAITB0 | WAITM3 | WAITM2 | WAITM1 | WAITM0 | WAITE1 | WAITE0 | 0000 0000 | 73, 175 |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved. **Bold** indicates shared access SFRs.

**Note 1:** Bit 21 of the PC is only available in Serial Programming (SP) modes.

**2:** Reset value is '0' when Two-Speed Start-up is enabled and '1' if disabled.

**3:** The SSPxMSK registers are only accessible when SSPxCON2<3:0> = 1001.

**4:** Alternate names and definitions for these bits when the MSSP module is operating in $I^2C$™ Slave mode. See **Section 19.5.3.2 "Address Masking Modes"** for details.

**5:** These bits and/or registers are only available in 44-pin devices; otherwise, they are unimplemented and read as '0'. Reset values are shown for 44-pin devices.

**6:** The PMADDRH/PMDOUT1H and PMADDRL/PMDOUT1L register pairs share the same physical registers and addresses, but have different functions determined by the module's operating mode. See **Section 11.1.2 "Data Registers"** for more information.

**REGISTER 10-12: RPINR8: PERIPHERAL PIN SELECT INPUT REGISTER 8 (BANKED EEEh)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | IC2R4 | IC2R3 | IC2R2 | IC2R1 | IC2R0 |
| 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 **IC2R<4:0>:** Assign Input Capture 2 (ECCP2) to the Corresponding RPn Pin bits

**REGISTER 10-13: RPINR12: PERIPHERAL PIN SELECT INPUT REGISTER 12 (BANKED EF2h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | T1GR4 | T1GR3 | T1GR2 | T1GR1 | T1GR0 |
| 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 **T1GR<4:0>:** Timer1 Gate Input (T1G) to the Corresponding RPn Pin bits

**REGISTER 10-14: RPINR13: PERIPHERAL PIN SELECT INPUT REGISTER 13 (BANKED EF3h)**

| U-0 | U-0 | U-0 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 | R/$\overline{W}$-1 |
|---|---|---|---|---|---|---|---|
| — | — | — | T3GR4 | T3GR3 | T3GR2 | T3GR1 | T3GR0 |
| 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 **T3GR<4:0>:** Timer3 Gate Input (T3G) to the Corresponding RPn Pin bits

**REGISTER 10-21: RPOR0: PERIPHERAL PIN SELECT OUTPUT REGISTER 0 (BANKED EC6h)(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 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP0R4 | RP0R3 | RP0R2 | RP0R1 | RP0R0 |
| 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 **RP0R<4:0>:** Peripheral Output Function is Assigned to RP0 Output Pin bits (see Table 10-14 for peripheral function numbers)

**Note 1:** Register values can be changed only if PPSCON<IOLOCK> = 0.

**REGISTER 10-22: RPOR1: PERIPHERAL PIN SELECT OUTPUT REGISTER 1 (BANKED EC7h)**

| 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 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP1R4 | RP1R3 | RP1R2 | RP1R1 | RP1R0 |
| 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 **RP1R<4:0>:** Peripheral Output Function is Assigned to RP1 Output Pin bits (see Table 10-14 for peripheral function numbers)

**REGISTER 10-23: RPOR2: PERIPHERAL PIN SELECT OUTPUT REGISTER 2 (BANKED EC8h)**

| 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 |
|---|---|---|---|---|---|---|---|
| — | — | — | RP2R4 | RP2R3 | RP2R2 | RP2R1 | RP2R0 |
| 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 **RP2R<4:0>:** Peripheral Output Function is Assigned to RP2 Output Pin bits (see Table 10-14 for peripheral function numbers)

## 15.3 Timer3 16-Bit Read/Write Mode

Timer3 can be configured for 16-bit reads and writes (see **Section 15.3 "Timer3 16-Bit Read/Write Mode"**). When the RD16 control bit (T3CON<1>) is set, the address for TMR3H is mapped to a buffer register for the high byte of Timer3. A read from TMR3L will load the contents of the high byte of Timer3 into the Timer3 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer3 without having to determine whether a read of the high byte, followed by a read of the low byte, has become invalid due to a rollover between reads.

A write to the high byte of Timer3 must also take place through the TMR3H Buffer register. The Timer3 high byte is updated with the contents of TMR3H when a write occurs to TMR3L. This allows a user to write all 16 bits to both the high and low bytes of Timer3 at once.

The high byte of Timer3 is not directly readable or writable in this mode. All reads and writes must take place through the Timer3 High Byte Buffer register.

Writes to TMR3H do not clear the Timer3 prescaler. The prescaler is only cleared on writes to TMR3L.

## 15.4 Using the Timer1 Oscillator as the Timer3 Clock Source

The Timer1 internal oscillator may be used as the clock source for Timer3. The Timer1 oscillator is enabled by setting the T1OSCEN (T1CON<3>) bit. To use it as the Timer3 clock source, the TMR3CS bit must also be set. As previously noted, this also configures Timer3 to increment on every rising edge of the oscillator source.

The Timer1 oscillator is described in **Section 13.0 "Timer1 Module"**.

## 15.5 Timer3 Gate

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

Timer3 gate can also be driven by multiple selectable sources.

### 15.5.1 TIMER3 GATE COUNT ENABLE

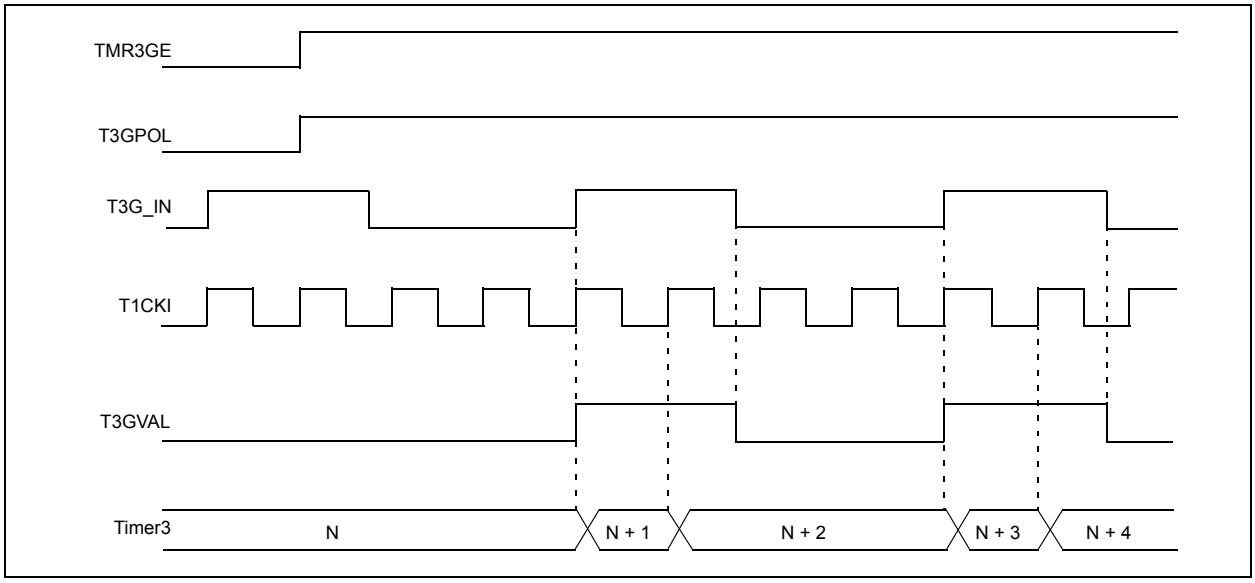
The Timer3 Gate Enable mode is enabled by setting the TMR3GE bit of the T3GCON register. The polarity of the Timer3 Gate Enable mode is configured using the T3GPOL bit of the T3GCON register.

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 15-2 for timing details.

**TABLE 15-1: TIMER3 GATE ENABLE SELECTIONS**

| T3CLK | T3GPOL | T3G | Timer3 Operation |
|---|---|---|---|
| ↑ | 0 | 0 | Counts |
| ↑ | 0 | 1 | Holds Count |
| ↑ | 1 | 0 | Holds Count |
| ↑ | 1 | 1 | Counts |

**FIGURE 15-2: TIMER3 GATE COUNT ENABLE MODE**



**REGISTER 17-4: ALRMCFG: ALARM CONFIGURATION REGISTER (ACCESS F91h)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| ALRMEN | CHIME | AMASK3 | AMASK2 | AMASK1 | AMASK0 | ALRMPTR1 | ALRMPTR0 |
| 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 **ALRMEN:** Alarm Enable bit

1 = Alarm is enabled (cleared automatically after an alarm event whenever ARPT<7:0> = 0000 0000 and CHIME = 0)
0 = Alarm is disabled

bit 6 **CHIME:** Chime Enable bit

1 = Chime is enabled; ALRMRPT<7:0> bits are allowed to roll over from 00h to FFh
0 = Chime is disabled; ALRMRPT<7:0> bits stop once they reach 00h

bit 5-2 **AMASK<3:0>:** Alarm Mask Configuration bits

0000 = Every half second
0001 = Every second
0010 = Every 10 seconds
0011 = Every minute
0100 = Every 10 minutes
0101 = Every hour
0110 = Once a day
0111 = Once a week
1000 = Once a month
1001 = Once a year (except when configured for February 29th, once every four years)
101x = Reserved – do not use
11xx = Reserved – do not use

bit 1-0 **ALRMPTR<1:0>:** Alarm Value Register Window Pointer bits

Points to the corresponding Alarm Value registers when reading the ALRMVALH and ALRMVALL registers. The ALRMPTR<1:0> value decrements on every read or write of ALRMVALH until it reaches '00'.

ALRMVALH<15:8>:
00 = ALRMMIN
01 = ALRMWD
10 = ALRMMNTH
11 = Unimplemented

ALRMVALL<7:0>:
00 = ALRMSEC
01 = ALRMHR
10 = ALRMDAY
11 = Unimplemented

## 18.3 Compare Mode

In Compare mode, the 16-bit CCPRx register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the ECCPx pin can be:

- Driven high
- Driven low
- Toggled (high-to-low or low-to-high)
- Remain unchanged (that is, reflects the state of the I/O latch)

The action on the pin is based on the value of the mode select bits (CCPxM<3:0>). At the same time, the interrupt flag bit, CCPxIF, is set.

### 18.3.1 ECCP PIN CONFIGURATION

Users must configure the ECCPx pin as an output by clearing the appropriate TRIS bit.

| | |
|---|---|
| **Note:** | Clearing the CCPxCON register will force the ECCPx compare output latch (depending on device configuration) to the default low level. This is not the PORTx I/O data latch. |

### 18.3.2 TIMER1/TIMER3 MODE SELECTION

Timer1 and/or Timer3 must be running in Timer mode or Synchronized Counter mode if the ECCP module is using the compare feature. In Asynchronous Counter mode, the compare operation will not work reliably.

### 18.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCPxM<3:0> = 1010), the ECCPx pin is not affected; only the CCPxIF interrupt flag is affected.

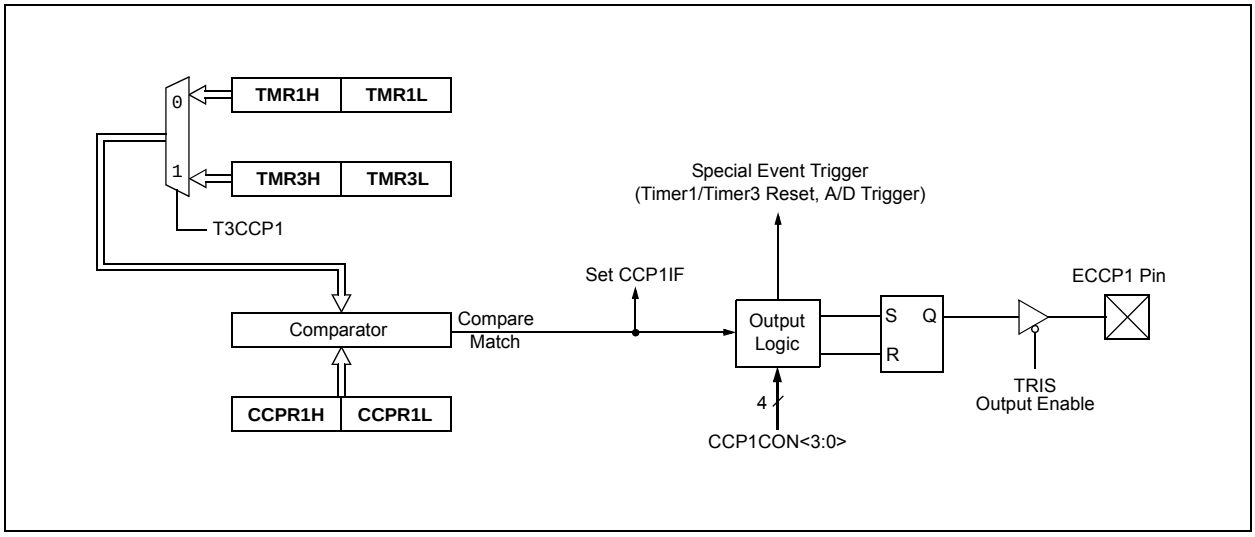
### 18.3.4 SPECIAL EVENT TRIGGER

The ECCP module is equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCPxM<3:0> = 1011).

The Special Event Trigger resets the Timer register pair for whichever timer resource is currently assigned as the module's time base. This allows the CCPRx registers to serve as a programmable period register for either timer.

The Special Event Trigger can also start an A/D conversion. In order to do this, the A/D converter must already be enabled.

**FIGURE 18-2: COMPARE MODE OPERATION BLOCK DIAGRAM**



**REGISTER 20-2: RCSTAx: RECEIVE STATUS AND CONTROL REGISTER (ACCESS FACh/F9Ch)**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 | R-x |
|---|---|---|---|---|---|---|---|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
| 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 **SPEN:** Serial Port Enable bit
1 = Serial port enabled
0 = Serial port disabled (held in Reset)

bit 6 **RX9:** 9-Bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit
Asynchronous mode:
Don't care.
Synchronous mode – Master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode – Slave:
Don't care.

bit 4 **CREN:** Continuous Receive Enable bit
Asynchronous mode:
1 = Enables receiver
0 = Disables receiver
Synchronous mode:
1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)
0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit
Asynchronous mode 9-Bit (RX9 = 1):
1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit
Asynchronous mode 8-Bit (RX9 = 0):
Don't care.

bit 2 **FERR:** Framing Error bit
1 = Framing error (can be cleared by reading RCREGx register and receiving next valid byte)
0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN). UART reception will be discarded until the overun error is cleared.
0 = No overrun error

bit 0 **RX9D:** 9$^{th}$ bit of Received Data
This can be address/data bit or a parity bit and must be calculated by user firmware.

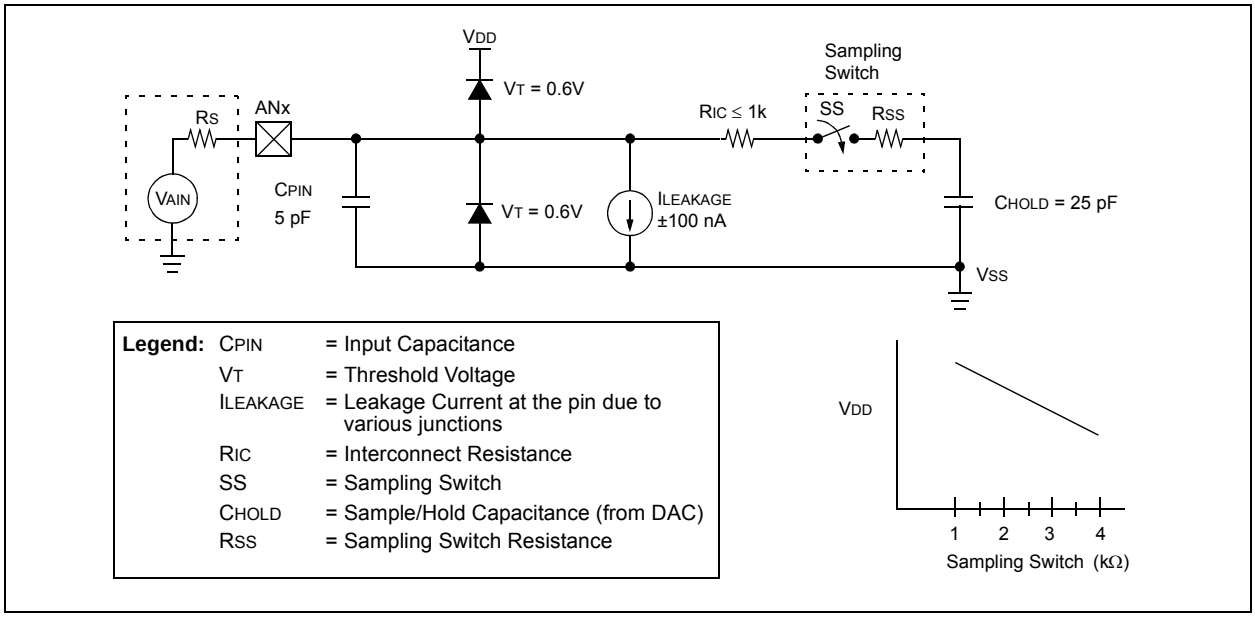
After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as an input. To determine acquisition time, see **Section 21.1 "A/D Acquisition Requirements"**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/$\overline{\text{DONE}}$ bit and the actual start of the conversion.

The following steps should be followed to do an A/D conversion:

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

   OR
   - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear bit, ADIF, if required.
7. For next conversion, go to step 1 or step 2, as required. The A/D conversion time per bit is defined as TAD. A minimum wait of 2 TAD is required before next acquisition starts.

**FIGURE 21-2: ANALOG INPUT MODEL**



## 26.0 SPECIAL FEATURES OF THE CPU

PIC18F46J11 family devices include several features intended to maximize reliability and minimize cost through elimination of external components. These are:

- Oscillator Selection
- Resets:
  - Power-on Reset (POR)
  - Power-up Timer (PWRT)
  - Oscillator Start-up Timer (OST)
  - Brown-out Reset (BOR)
- Interrupts
- Watchdog Timer (WDT)
- Fail-Safe Clock Monitor (FSCM)
- Two-Speed Start-up
- Code Protection
- In-Circuit Serial Programming (ICSP)

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 PIC18F46J11 family of devices have a configurable Watchdog Timer (WDT), which is controlled in software.

The inclusion of an internal RC oscillator 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.

### 26.1 Configuration Bits

The Configuration bits can be programmed to select various device configurations. The configuration data is stored in the last four words of Flash program memory; Figure 6-1 depicts this. The configuration data gets loaded into the volatile Configuration registers, CONFIG1L through CONFIG4H, which are readable and mapped to program memory starting at location 300000h.

Table 26-2 provides a complete list. A detailed explanation of the various bit functions is provided in Register 26-1 through Register 26-6.

#### 26.1.1 CONSIDERATIONS FOR CONFIGURING THE PIC18F46J11 FAMILY DEVICES

Unlike some previous PIC18 microcontrollers, devices of the PIC18F46J11 family do not use persistent memory registers to store configuration information. The Configuration registers, CONFIG1L through CONFIG4H, are implemented as volatile memory.

Immediately after power-up, or after a device Reset, the microcontroller hardware automatically loads the CONFIG1L through CONFIG4L registers with configuration data stored in nonvolatile Flash program memory. The last four words of Flash program memory, known as the Flash Configuration Words (FCW), are used to store the configuration data.

Table 26-1 provides the Flash program memory, which will be loaded into the corresponding Configuration register.

When creating applications for these devices, users should always specifically allocate the location of the FCW for configuration data. This is to make certain that program code is not stored in this address when the code is compiled.

The four Most Significant bits (MSb) of the FCW corresponding to CONFIG1H, CONFIG2H, CONFIG3H and CONFIG4H should always be programmed to '1111'. This makes these FCWs appear to be NOP instructions in the remote event that their locations are ever executed by accident.

To prevent inadvertent configuration changes during code execution, the Configuration registers, CONFIG1L through CONFIG4L, are loaded only once per power-up or Reset cycle. User's firmware can still change the configuration by using self-reprogramming to modify the contents of the FCW.

Modifying the FCW will not change the active contents being used in the CONFIG1L through CONFIG4H registers until after the device is reset.

**REGISTER 26-10: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F46J11 FAMILY DEVICES (BYTE ADDRESS 3FFFFFh)**

| R | R | R | R | R | R | R | R |
|---|---|---|---|---|---|---|---|
| DEV10 | DEV9 | DEV8 | DEV7 | DEV6 | DEV5 | DEV4 | DEV3 |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as ‘0’ | |
| -n = Value at Reset | ‘1’ = Bit is set | ‘0’ = Bit is cleared | x = Bit is unknown |

bit 7-0 **DEV<10:3>:** Device ID bits
These bits are used with the DEV<2:0> bits in the Device ID Register 1 to identify the part number.

| DEV<10:3> (DEVID2<7:0>) | DEV<2:0> (DEVID1<7:5>) | Device |
|---|---|---|
| 0100 1110 | 001 | PIC18F46J11 |
| 0100 1110 | 000 | PIC18F45J11 |
| 0100 1101 | 111 | PIC18F44J11 |
| 0100 1101 | 110 | PIC18F26J11 |
| 0100 1101 | 101 | PIC18F25J11 |
| 0100 1101 | 100 | PIC18F24J11 |
| 0100 1110 | 111 | PIC18LF46J11 |
| 0100 1110 | 110 | PIC18LF45J11 |
| 0100 1110 | 101 | PIC18LF44J11 |
| 0100 1110 | 100 | PIC18LF26J11 |
| 0100 1110 | 011 | PIC18LF25J11 |
| 0100 1110 | 010 | PIC18LF24J11 |

## 26.7 In-Circuit Serial Programming (ICSP)

PIC18F46J11 family microcontrollers can be serially programmed while in the end application circuit. This is simply done with two lines for clock and data and three other lines for power, ground and the programming voltage. This allows customers to manufacture boards with unprogrammed devices and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

## 26.8 In-Circuit Debugger

When the $\overline{\text{DEBUG}}$ Configuration bit is programmed to a '0', the In-Circuit Debugger functionality is enabled. This function allows simple debugging functions when used with MPLAB® IDE. When the microcontroller has this feature enabled, some resources are not available for general use.

Table 26-4 lists the resources required by the background debugger.

**TABLE 26-4: DEBUGGER RESOURCES**

| | |
|---|---|
| I/O pins: | RB6, RB7 |
| Stack: | TOSx registers reserved |

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

| 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** | | |
| | PIC18LFXXJ11 | 5.2 | 6.5 | mA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 16 MHz (**PRI_RUN** mode, 4 MHz Internal Oscillator with PLL |
| | | 5.1 | 6.4 | mA | +25°C | | |
| | | 5.1 | 6.4 | mA | +85°C | | |
| | PIC18FXXJ11 | 5.3 | 7.5 | mA | -40°C | VDD = 3.3V, VDDCORE = 10 μF Capacitor | |
| | | 5.2 | 7.4 | mA | +25°C | | |
| | | 5.2 | 7.4 | mA | +85°C | | |
| | PIC18LFXXJ11 | 9.3 | 12.0 | mA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 32 MHz, **PRI_RUN** mode, 8 MHz Internal Oscillator with PLL |
| | | 9.2 | 11.8 | mA | +25°C | | |
| | | 9.0 | 11.8 | mA | +85°C | | |
| | PIC18FXXJ11 | 9.7 | 17.5 | mA | -40°C | VDD = 3.3V, VDDCORE = 10 μF Capacitor | |
| | | 9.6 | 17.2 | mA | +25°C | | |
| | | 9.6 | 17.2 | mA | +85°C | | |
| | PIC18LFXXJ11 | 12.4 | 13.5 | mA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | FOSC = 48 MHz, **PRI_RUN** mode, 12 MHz External Oscillator with PLL |
| | | 12.2 | 13.5 | mA | +25°C | | |
| | | 12.1 | 13.9 | mA | +85°C | | |
| | PIC18FXXJ11 | 14.3 | 24.1 | mA | -40°C | VDD = 3.3V, VDDCORE = 10 μF Capacitor | |
| | | 14.2 | 23.0 | mA | +25°C | | |
| | | 14.2 | 23.0 | mA | +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.

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

| 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** | | |
| | **Module Differential Currents (ΔIWDT, ΔIOSCB, ΔIAD)** | | | | | | |
| D022 (ΔIWDT) | **Watchdog Timer** | 0.86 | 8 | µA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | PIC18LFXXJ11 |
| | | 0.97 | 8 | µA | +25°C | | |
| | | 0.98 | 10.4 | µA | +85°C | | |
| | | 0.71 | 7 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | PIC18FXXJ11 |
| | | 0.82 | 7 | µA | +25°C | | |
| | | 0.65 | 10 | µA | +85°C | | |
| | | 1.54 | 12.1 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | PIC18FXXJ11 |
| | | 1.33 | 12.1 | µA | +25°C | | |
| | | 1.16 | 13.6 | µA | +85°C | | |
| D022B (ΔIHLVD) | **High/Low-Voltage Detect** | 3.9 | 8 | µA | -40°C | VDD = 2.5V, VDDCORE = 2.5V | PIC18LFXXJ11 |
| | | 4.7 | 8 | µA | +25°C | | |
| | | 5.4 | 9 | µA | +85°C | | |
| | | 2.7 | 6 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | PIC18FXXJ11 |
| | | 3.2 | 6 | µA | +25°C | | |
| | | 3.6 | 8 | µA | +85°C | | |
| | | 3.5 | 9 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | PIC18FXXJ11 |
| | | 4.1 | 9 | µA | +25°C | | |
| | | 4.5 | 12 | µA | +85°C | | |
| D025 (ΔIOSCB) | **Real-Time Clock/Calendar with Low-Power Timer1 Oscillator** | 0.67 | 4.0 | µA | -40°C | VDD = 2.15V, VDDCORE = 10 µF Capacitor | PIC18FXXJ11<br>32.768 kHz, T1OSCEN = 1, LPT1OSC = 0 |
| | | 0.83 | 4.5 | µA | +25°C | | |
| | | 0.95 | 4.5 | µA | +60°C | | |
| | | 1.10 | 4.5 | µA | +85°C | | |
| | | 0.75 | 4.5 | µA | -40°C | VDD = 2.5V, VDDCORE = 10 µF Capacitor | |
| | | 0.92 | 5.0 | µA | +25°C | | |
| | | 1.04 | 5.0 | µA | +60°C | | |
| | | 1.21 | 5.0 | µA | +85°C | | |
| | | 0.94 | 6.5 | µA | -40°C | VDD = 3.3V, VDDCORE = 10 µF Capacitor | |
| | | 1.11 | 6.5 | µA | +25°C | | |
| | | 1.24 | 8.0 | µA | +60°C | | |
| | | 1.43 | 8.0 | µA | +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.).

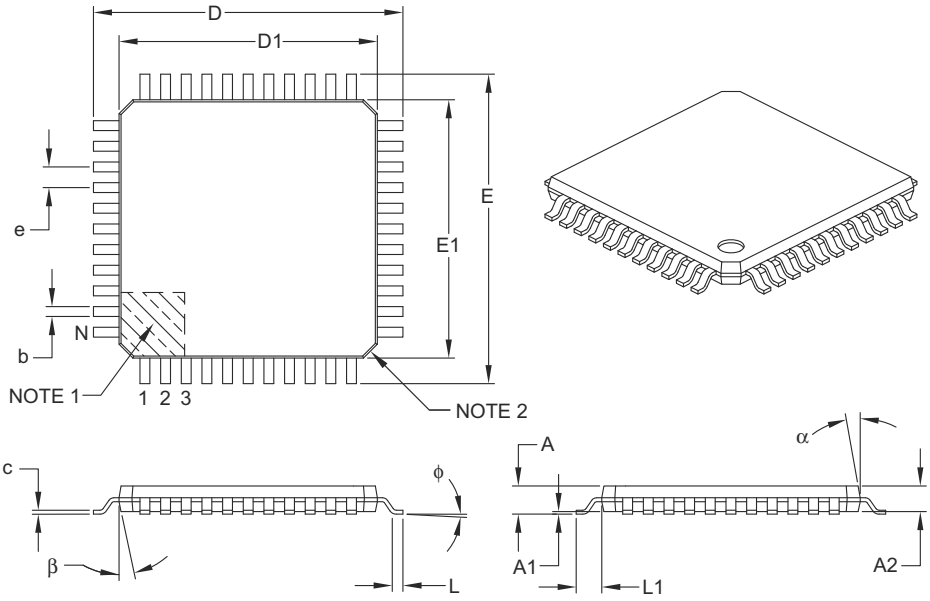
**2:** The supply current is mainly a function of operating voltage, frequency and mode. Other factors, such as I/O pin loading and switching rate, oscillator type and circuit, internal code execution pattern and temperature, also have an impact on the current consumption. All features that add delta 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.

### 44-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 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 | | 44 | |
| Lead Pitch | e | | 0.80 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 | | 12.00 BSC | |
| Overall Length | D | | 12.00 BSC | |
| Molded Package Width | E1 | | 10.00 BSC | |
| Molded Package Length | D1 | | 10.00 BSC | |
| Lead Thickness | c | 0.09 | – | 0.20 |
| Lead Width | b | 0.30 | 0.37 | 0.45 |
| Mold Draft Angle Top | α | 11° | 12° | 13° |
| Mold Draft Angle Bottom | β | 11° | 12° | 13° |

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. 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 C04-076B

**NOTES:**

## U



## V



## W



## X