



# **PIC18F85J11 Family Data Sheet**

64/80-Pin  
High-Performance Microcontrollers  
with nanoWatt Technology

---

**Note the following details of the code protection feature on Microchip devices:**

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

---

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

#### **Trademarks**

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Amplab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICKit, PICDEM, PICDEM.net, PICLAB, PICTail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rFLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

*Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.*

**QUALITY MANAGEMENT SYSTEM**  
**CERTIFIED BY DNV**  
**== ISO/TS 16949:2002 ==**

## 64/80-Pin High-Performance Microcontrollers with nanoWatt Technology

### Peripheral Highlights:

- High-Current Sink/Source 25 mA/25 mA (PORTB and PORTC)
- Up to Four External Interrupts
- Four 8-Bit/16-Bit Timer/Counter modules
- Real-Time Clock (RTC) Software module:
  - Configurable 24-hour clock, calendar, automatic 100-year or 12800-year day of week calculator using Timer1
- Two Capture/Compare/PWM (CCP) modules:
  - Capture is 16-bit, max. resolution 6.25 ns ( $T_{CY}/16$ )
  - Compare is 16-bit, max. resolution 100 ns ( $T_{CY}$ )
  - PWM output: PWM resolution is up to 10-bit
- Master Synchronous Serial Port (MSSP) module with Two Modes of Operation:
  - 3-wire/4-wire SPI (supports all 4 SPI modes)
  - I<sup>2</sup>C™ Master and Slave mode
- One Addressable USART module
- One Enhanced Addressable USART module:
  - Supports LIN 1.2
  - Auto-wake-up on Start bit and Break character
  - Auto-Baud Detect (ABD)
- 10-Bit, up to 12-Channel A/D Converter:
  - Auto-acquisition
  - Conversion available during Sleep
- Two Analog Comparators
- Programmable Reference Voltage for Comparators

### External Memory Bus (PIC18F8XJ11 only):

- Address Capability of up to 2 Mbytes
- 8-Bit or 16-Bit Interface
- 12-Bit, 16-Bit and 20-Bit Addressing modes

### Low-Power Features:

- Power-Managed modes: Run, Idle, Sleep
- Run Current Down to 9  $\mu$ A, Typical
- Idle Current Down to 2.5  $\mu$ A, Typical
- Sleep Current Down to 100 nA, Typical
- Fast INTOSC Start-up from Sleep
- Two-Speed Oscillator Start-up Reduces Crystal Stabilization Wait Time

### Flexible Oscillator Structure:

- Two Crystal modes, 4-25 MHz
- Two External Clock modes, up to 40 MHz
- 4x Phase Lock Loop (PLL)
- Internal Oscillator Block:
  - 8 user-selectable frequencies from 31.25 kHz to 8 MHz
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if peripheral clock fails

### Special Microcontroller Features:

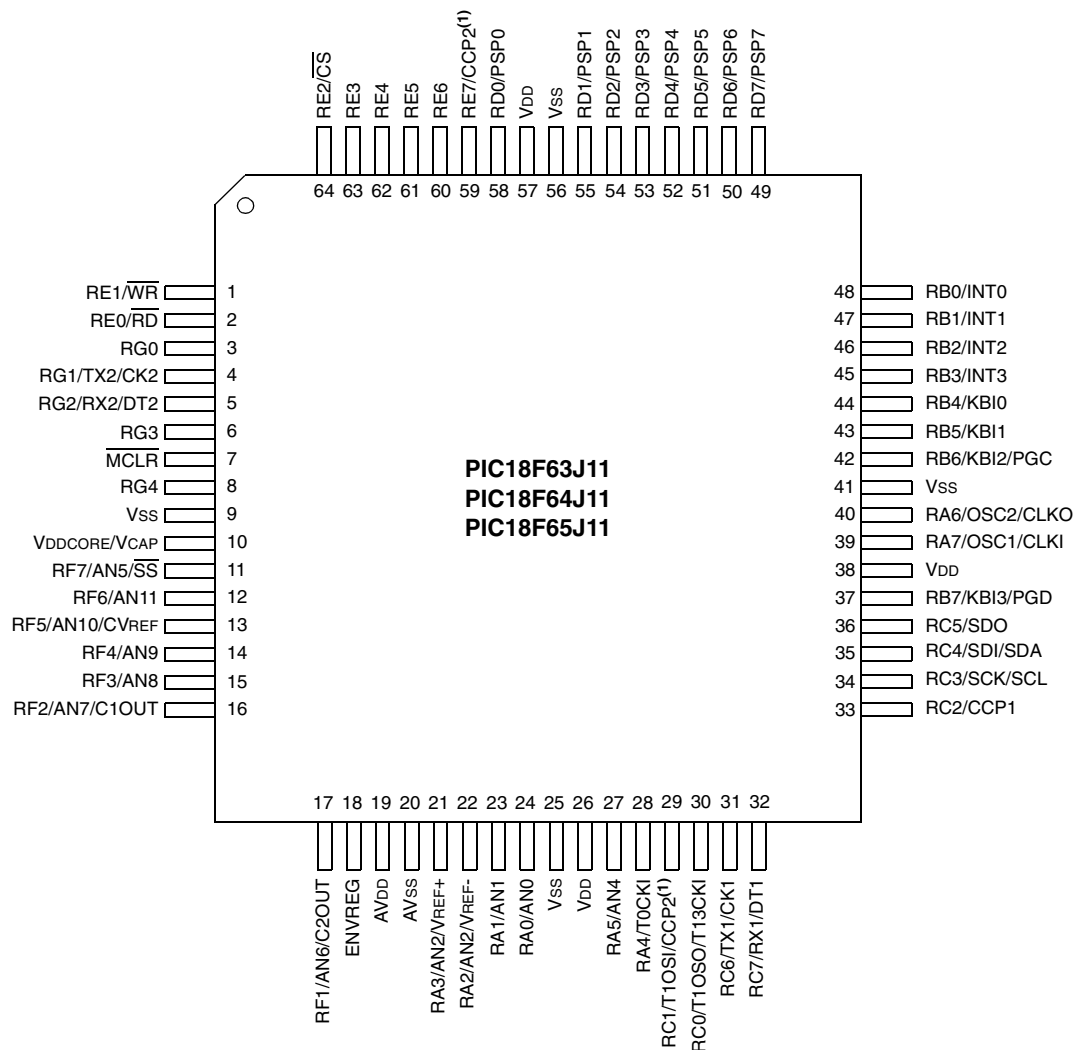
- 1,000 Erase/Write Cycle Flash Program Memory Typical
- Flash Retention 20 Years Minimum
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 4 ms to 131s
- In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug via two pins
- Operating Voltage Range: 2.0V to 3.6V
- 5.5V Tolerant Input (digital pins only)
- Selectable Open-Drain Configuration for Serial Communication and CCP pins for Driving Outputs up to 5V
- On-Chip 2.5V Regulator

| Device      | Program Memory |                            | SRAM Data Memory (bytes) | I/O | Timers 8/16-Bit | CCP | MSSP |                          | EUSART/ AUSART | 10-Bit A/D (ch) | Comparators | BOR/LVD | External Bus | PSP |
|-------------|----------------|----------------------------|--------------------------|-----|-----------------|-----|------|--------------------------|----------------|-----------------|-------------|---------|--------------|-----|
|             | Flash (bytes)  | # Single-Word Instructions |                          |     |                 |     | SPI  | Master I <sup>2</sup> C™ |                |                 |             |         |              |     |
| PIC18F63J11 | 8K             | 4096                       | 1K                       | 52  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | N            | Y   |
| PIC18F64J11 | 16K            | 8192                       | 1K                       | 52  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | N            | Y   |
| PIC18F65J11 | 32K            | 16384                      | 2K                       | 52  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | N            | Y   |
| PIC18F83J11 | 8K             | 4096                       | 1K                       | 68  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | Y            | Y   |
| PIC18F84J11 | 16K            | 8192                       | 1K                       | 68  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | Y            | Y   |
| PIC18F85J11 | 32K            | 16384                      | 2K                       | 68  | 1/3             | 2   | Y    | Y                        | 1/1            | 12              | 2           | Y       | Y            | Y   |

# PIC18F85J11 FAMILY

## Pin Diagrams

### 64-Pin TQFP

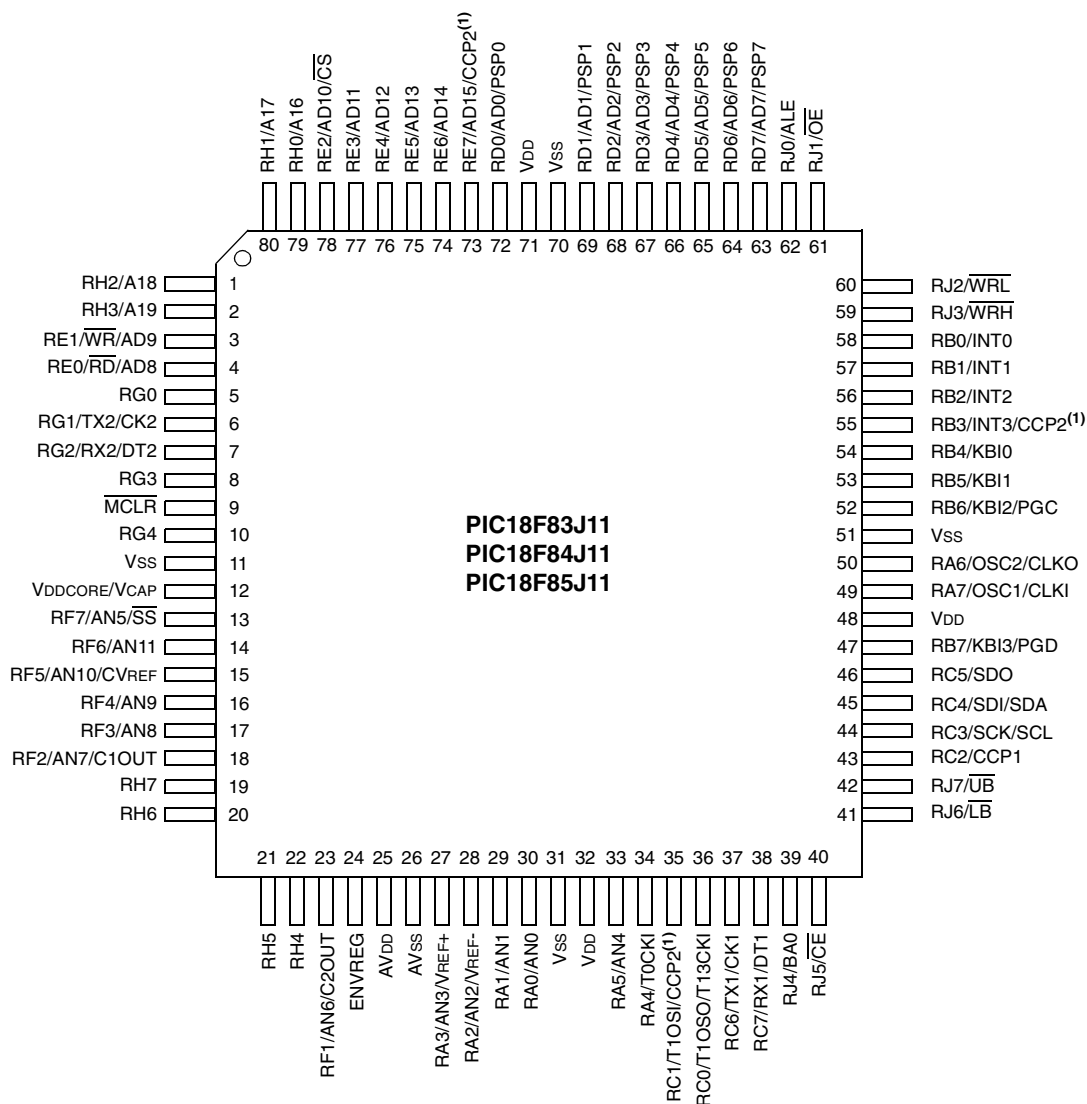


**Note 1:** The CCP2 pin placement depends on the CCP2MX Configuration bit setting.

# PIC18F85J11 FAMILY

## Pin Diagrams (Continued)

### 80-Pin TQFP



**Note 1:** The CCP2 pin placement depends on the settings of the CCP2MX and EMB1:EMB0 Configuration bits.

# PIC18F85J11 FAMILY

---

## Table of Contents

|      |                                                                                    |     |
|------|------------------------------------------------------------------------------------|-----|
| 1.0  | Device Overview .....                                                              | 7   |
| 2.0  | Oscillator Configurations .....                                                    | 29  |
| 3.0  | Power-Managed Modes .....                                                          | 37  |
| 4.0  | Reset .....                                                                        | 45  |
| 5.0  | Memory Organization .....                                                          | 57  |
| 6.0  | Flash Program Memory .....                                                         | 83  |
| 7.0  | External Memory Bus .....                                                          | 93  |
| 8.0  | 8 x 8 Hardware Multiplier .....                                                    | 105 |
| 9.0  | Interrupts .....                                                                   | 107 |
| 10.0 | I/O Ports .....                                                                    | 123 |
| 11.0 | Timer0 Module .....                                                                | 147 |
| 12.0 | Timer1 Module .....                                                                | 151 |
| 13.0 | Timer2 Module .....                                                                | 157 |
| 14.0 | Timer3 Module .....                                                                | 159 |
| 15.0 | Capture/Compare/PWM (CCP) Modules .....                                            | 163 |
| 16.0 | Master Synchronous Serial Port (MSSP) Module .....                                 | 173 |
| 17.0 | Enhanced Universal Synchronous Asynchronous Receiver Transmitter (EUSART) .....    | 217 |
| 18.0 | Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) ..... | 237 |
| 19.0 | 10-bit Analog-to-Digital Converter (A/D) Module .....                              | 251 |
| 20.0 | Comparator Module .....                                                            | 261 |
| 21.0 | Comparator Voltage Reference Module .....                                          | 267 |
| 22.0 | Special Features of the CPU .....                                                  | 271 |
| 23.0 | Instruction Set Summary .....                                                      | 285 |
| 24.0 | Development Support .....                                                          | 335 |
| 25.0 | Electrical Characteristics .....                                                   | 339 |
| 26.0 | DC and AC Characteristics Graphs and Tables .....                                  | 373 |
| 27.0 | Packaging Information .....                                                        | 375 |
|      | Appendix A: Revision History .....                                                 | 379 |
|      | Appendix B: Migration Between High-End Device Families .....                       | 379 |
|      | Index .....                                                                        | 381 |
|      | The Microchip Web Site .....                                                       | 391 |
|      | Customer Change Notification Service .....                                         | 391 |
|      | Customer Support .....                                                             | 391 |
|      | Reader Response .....                                                              | 392 |
|      | Product Identification System .....                                                | 393 |

## TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at [docerrors@microchip.com](mailto:docerrors@microchip.com) or fax the **Reader Response Form** in the back of this data sheet to (480) 792-4150. We welcome your feedback.

### Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

<http://www.microchip.com>

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000A is version A of document DS30000).

### Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; <http://www.microchip.com>
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include literature number) you are using.

### Customer Notification System

Register on our web site at [www.microchip.com](http://www.microchip.com) to receive the most current information on all of our products.

# PIC18F85J11 FAMILY

---

NOTES:



## 1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F63J11
- PIC18F64J11
- PIC18F65J11
- PIC18F83J11
- PIC18F84J11
- PIC18F85J11

This family combines the traditional advantages of all PIC18 microcontrollers – namely, high computational performance and a rich feature set – while maintaining an extremely competitive price point. These features make the PIC18F85J11 family a logical choice for many high-performance applications where price is a primary consideration.

### 1.1 Core Features

#### 1.1.1 nanoWatt TECHNOLOGY

All of the devices in the PIC18F85J11 family incorporate a range of features that can significantly reduce power consumption during operation. Key items include:

- **Alternate Run Modes:** By clocking the controller from the Timer1 source or the internal RC oscillator, power consumption during code execution can be reduced by as much as 90%.
- **Multiple Idle Modes:** The controller can also run with its CPU core disabled but the peripherals still active. In these states, power consumption can be reduced even further, to as little as 4% of normal operation requirements.
- **On-the-Fly Mode Switching:** The power-managed modes are invoked by user code during operation, allowing the user to incorporate power-saving ideas into their application's software design.

#### 1.1.2 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F85J11 family offer six different oscillator options, allowing users a range of choices in developing application hardware. These include:

- Two Crystal modes, using crystals or ceramic resonators.
- Two External Clock modes, offering the option of a divide-by-4 clock output.
- A Phase Lock Loop (PLL) frequency multiplier, available to the External Oscillator modes, which allows clock speeds of up to 40 MHz.
- An internal oscillator block which provides an 8 MHz clock ( $\pm 2\%$  accuracy) and an INTRC source (approximately 31 kHz, stable over temperature and  $V_{DD}$ ), as well as a range of six user-selectable clock frequencies, between 125 kHz to 4 MHz, for a total of eight clock frequencies. This option frees the two oscillator pins for use as additional general purpose I/O.

The internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

#### 1.1.3 MEMORY OPTIONS

The PIC18F85J11 family provides a range of program memory options, from 8 Kbytes to 32 Kbytes of code space. The Flash cells for program memory are rated to last up to 1000 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The PIC18F85J11 family also provides plenty of room for dynamic application data, with up to 2048 bytes of data RAM.

#### 1.1.4 EXTENDED INSTRUCTION SET

The PIC18F85J11 family implements the optional extension to the PIC18 instruction set, adding 8 new instructions and an Indexed Addressing mode. Enabled as a device configuration option, the extension has been specifically designed to optimize re-entrant application code originally developed in high-level languages, such as 'C'.

#### 1.1.5 EXTERNAL MEMORY BUS

In the event that 32 Kbytes of memory are inadequate for an application, the 80-pin members of the PIC18F85J11 family also implement an external memory bus. This allows the controller's internal program counter to address a memory space of up to 2 Mbytes, permitting a level of data access that few 8-bit devices can claim. This allows additional memory options, including:

- Using combinations of on-chip and external memory up to the 2-Mbyte limit
- Using external Flash memory for reprogrammable application code or large data tables
- Using external RAM devices for storing large amounts of variable data

# PIC18F85J11 FAMILY

---

## 1.1.6 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also aids in migrating to the next larger device. This is true when moving between the 64-pin members, between the 80-pin members or even jumping from 64-pin to 80-pin devices.

The PIC18F85J11 family is also largely pin compatible with other PIC18 general purpose families, such as the PIC18F8720 and PIC18F8722. This allows a new dimension to the evolution of applications, allowing developers to select different price points within Microchip's PIC18 portfolio, while maintaining a similar feature set.

## 1.2 Other Special Features

- **Communications:** The PIC18F85J11 family incorporates a range of serial communication peripherals, including an Addressable USART, a separate Enhanced USART that supports LIN specification 1.2, and one Master SSP (MSSP) module capable of both SPI and I<sup>2</sup>C™ (Master and Slave) modes of operation.
- **CCP Modules:** All devices in the family incorporate two Capture/Compare/PWM (CCP) modules. Up to four different time bases may be used to perform several different operations at once.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reducing code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit prescaler, allowing an extended time-out range that is stable across operating voltage and temperature. See **Section 25.0 “Electrical Characteristics”** for time-out periods.

## 1.3 Details on Individual Family Members

Devices in the PIC18F85J11 family are available in 64-pin and 80-pin packages. Block diagrams for the two groups are shown in Figure 1-1 and Figure 1-2.

The devices are differentiated from each other in four ways:

1. Flash program memory (three sizes, ranging from 8 Kbytes for PIC18FX3J11 devices to 32 Kbytes for PIC18FX5J11 devices).
2. Data RAM (1024 bytes for PIC18FX3J11 and PIC18FX4J11 devices, 2048 bytes for PIC18FX5J11 devices).
3. I/O ports (7 bidirectional ports on 64-pin devices, 9 bidirectional ports on 80-pin devices).
4. External Memory Bus (implemented in 80-pin devices only).

All other features for devices in this family are identical. These are summarized in Table 1-1 and Table 1-2.

The pinouts for all devices are listed in Table 1-3 and Table 1-4.

# PIC18F85J11 FAMILY

**TABLE 1-1: DEVICE FEATURES FOR THE PIC18F6XJ11 FAMILY (64-PIN DEVICES)**

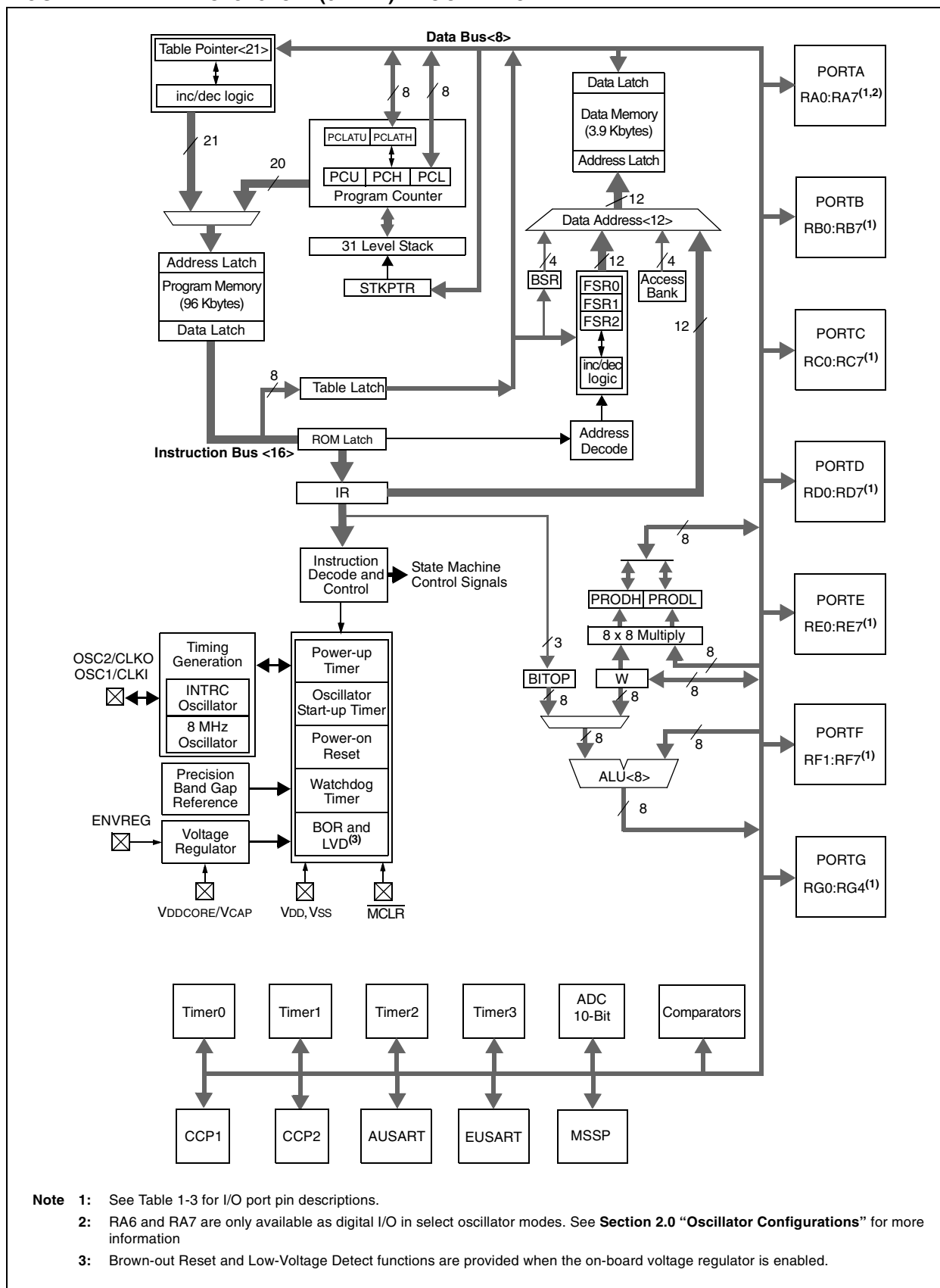
| Features                        | PIC18F63J11                                                                     | PIC18F64J11 | PIC18F65J11 |
|---------------------------------|---------------------------------------------------------------------------------|-------------|-------------|
| Operating Frequency             | DC – 40 MHz                                                                     |             |             |
| Program Memory (Bytes)          | 8K                                                                              | 16K         | 32K         |
| Program Memory (Instructions)   | 4096                                                                            | 8192        | 16384       |
| Data Memory (Bytes)             | 1024                                                                            | 1024        | 2048        |
| Interrupt Sources               | 27                                                                              |             |             |
| I/O Ports                       | Ports A, B, C, D, E, F, G                                                       |             |             |
| Timers                          | 4                                                                               |             |             |
| Capture/Compare/PWM Modules     | 2                                                                               |             |             |
| Serial Communications           | MSSP, Addressable USART, Enhanced USART                                         |             |             |
| Parallel Communications (PSP)   | Yes                                                                             |             |             |
| External Memory Bus             | No                                                                              |             |             |
| 10-Bit Analog-to-Digital Module | 12 Input Channels                                                               |             |             |
| Resets (and Delays)             | POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST) |             |             |
| Instruction Set                 | 75 Instructions, 83 with Extended Instruction Set Enabled                       |             |             |
| Packages                        | 64-Pin TQFP                                                                     |             |             |

**TABLE 1-2: DEVICE FEATURES FOR THE PIC18F8XJ11 FAMILY (80-PIN DEVICES)**

| Features                        | PIC18F83J11                                                                     | PIC18F84J11 | PIC18F85J11 |
|---------------------------------|---------------------------------------------------------------------------------|-------------|-------------|
| Operating Frequency             | DC – 40 MHz                                                                     |             |             |
| Program Memory (Bytes)          | 8K                                                                              | 16K         | 32K         |
| Program Memory (Instructions)   | 4096                                                                            | 8192        | 16384       |
| Data Memory (Bytes)             | 1024                                                                            | 1024        | 2048        |
| Interrupt Sources               | 27                                                                              |             |             |
| I/O Ports                       | Ports A, B, C, D, E, F, G, H, J                                                 |             |             |
| Timers                          | 4                                                                               |             |             |
| Capture/Compare/PWM Modules     | 2                                                                               |             |             |
| Serial Communications           | MSSP, Addressable USART, Enhanced USART                                         |             |             |
| Parallel Communications (PSP)   | Yes                                                                             |             |             |
| External Memory Bus             | Yes                                                                             |             |             |
| 10-Bit Analog-to-Digital Module | 12 Input Channels                                                               |             |             |
| Resets (and Delays)             | POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST) |             |             |
| Instruction Set                 | 75 Instructions, 83 with Extended Instruction Set Enabled                       |             |             |
| Packages                        | 80-Pin TQFP                                                                     |             |             |

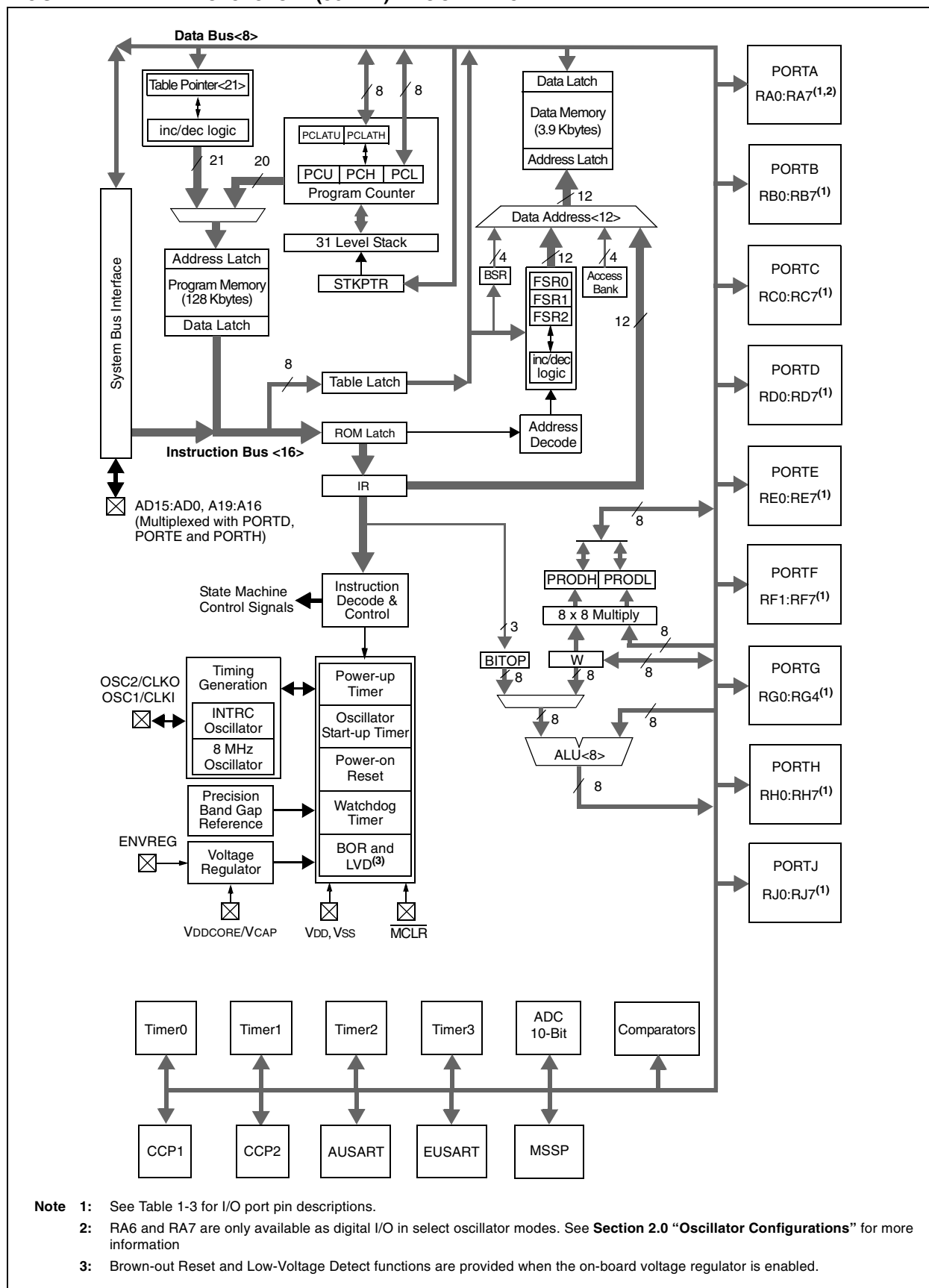
# PIC18F85J11 FAMILY

FIGURE 1-1: PIC18F6XJ11 (64-PIN) BLOCK DIAGRAM



# PIC18F85J11 FAMILY

FIGURE 1-2: PIC18F8XJ11 (80-PIN) BLOCK DIAGRAM



# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS**

| Pin Name                                 | Pin Number | Pin Type          | Buffer Type             | Description                                                                                                                                                                                                                                                                   |
|------------------------------------------|------------|-------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                          | TQFP       |                   |                         |                                                                                                                                                                                                                                                                               |
| MCLR                                     | 7          | I                 | ST                      | Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.                                                                                                                                                                           |
| RA7/OSC1/CLKI<br>RA7<br>OSC1<br>CLKI     | 39         | I/O<br>I<br>I     | TTL<br>CMOS<br>CMOS     | Oscillator crystal or external clock input.<br>General purpose I/O pin.<br>Oscillator crystal input.<br>External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)                                                       |
| RA6/OSC2/CLKO<br>RA6<br>OSC2<br><br>CLKO | 40         | I/O<br>O<br><br>O | TTL<br>—<br><br>—       | Oscillator crystal or clock output.<br>General purpose I/O pin.<br>Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.<br>In EC modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| RA0/AN0<br>RA0<br>AN0                    | 24         | I/O<br>I          | TTL<br>Analog           | PORTA is a bidirectional I/O port.<br><br>Digital I/O.<br>Analog input 0.                                                                                                                                                                                                     |
| RA1/AN1<br>RA1<br>AN1                    | 23         | I/O<br>I          | TTL<br>Analog           | Digital I/O.<br>Analog input 1.                                                                                                                                                                                                                                               |
| RA2/AN2/VREF-<br>RA2<br>AN2<br>VREF-     | 22         | I/O<br>I<br>I     | TTL<br>Analog<br>Analog | Digital I/O.<br>Analog input 2.<br>A/D reference voltage (low) input.                                                                                                                                                                                                         |
| RA3/AN3/VREF+<br>RA3<br>AN3<br>VREF+     | 21         | I/O<br>I<br>I     | TTL<br>Analog<br>Analog | Digital I/O.<br>Analog input 3.<br>A/D reference voltage (high) input.                                                                                                                                                                                                        |
| RA4/T0CKI<br>RA4<br>T0CKI                | 28         | I/O<br>I          | ST/OD<br>ST             | Digital I/O. Open-drain when configured as output.<br>Timer0 external clock input.                                                                                                                                                                                            |
| RA5/AN4<br>RA5<br>AN4                    | 27         | I/O<br>I          | TTL<br>Analog           | Digital I/O.<br>Analog input 4.                                                                                                                                                                                                                                               |
| RA6                                      |            |                   |                         | See the OSC2/CLKO/RA6 pin.                                                                                                                                                                                                                                                    |
| RA7                                      |            |                   |                         | See the OSC1/CLKI/RA7 pin.                                                                                                                                                                                                                                                    |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to V<sub>DD</sub>)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                           | Pin Number | Pin Type        | Buffer Type      | Description                                                                                                                                                |
|------------------------------------|------------|-----------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                    | TQFP       |                 |                  |                                                                                                                                                            |
| RB0/INT0<br>RB0<br>INT0            | 48         | I/O<br>I        | TTL<br>ST        | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.<br><br>Digital I/O.<br>External interrupt 0. |
| RB1/INT1<br>RB1<br>INT1            | 47         | I/O<br>I        | TTL<br>ST        | Digital I/O.<br>External interrupt 1.                                                                                                                      |
| RB2/INT2<br>RB2<br>INT2            | 46         | I/O<br>I        | TTL<br>ST        | Digital I/O.<br>External interrupt 2.                                                                                                                      |
| RB3/INT3<br>RB3<br>INT3            | 45         | I/O<br>I        | TTL<br>ST        | Digital I/O.<br>External interrupt 3.                                                                                                                      |
| RB4/KBI0<br>RB4<br>KBI0            | 44         | I/O<br>I        | TTL<br>TTL       | Digital I/O.<br>Interrupt-on-change pin.                                                                                                                   |
| RB5/KBI1<br>RB5<br>KBI1            | 43         | I/O<br>I        | TTL<br>TTL       | Digital I/O.<br>Interrupt-on-change pin.                                                                                                                   |
| RB6/KBI2/PGC<br>RB6<br>KBI2<br>PGC | 42         | I/O<br>I<br>I/O | TTL<br>TTL<br>ST | Digital I/O.<br>Interrupt-on-change pin.<br>In-Circuit Debugger and ICSP™ programming clock pin.                                                           |
| RB7/KBI3/PGD<br>RB7<br>KBI3<br>PGD | 37         | I/O<br>I<br>I/O | TTL<br>TTL<br>ST | Digital I/O.<br>Interrupt-on-change pin.<br>In-Circuit Debugger and ICSP programming data pin.                                                             |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name            | Pin Number | Pin Type | Buffer Type | Description                                                       |
|---------------------|------------|----------|-------------|-------------------------------------------------------------------|
|                     | TQFP       |          |             |                                                                   |
| RC0/T1OSO/T13CKI    | 30         |          |             | PORTC is a bidirectional I/O port.                                |
| RC0                 |            | I/O      | ST          | Digital I/O.                                                      |
| T1OSO               |            | O        | —           | Timer1 oscillator output.                                         |
| T13CKI              |            | I        | ST          | Timer1/Timer3 external clock input.                               |
| RC1/T1OSI/CCP2      | 29         |          |             |                                                                   |
| RC1                 |            | I/O      | ST          | Digital I/O.                                                      |
| T1OSI               |            | I        | CMOS        | Timer1 oscillator input.                                          |
| CCP2 <sup>(1)</sup> |            | I/O      | ST          | Capture 2 input/Compare 2 output/PWM2 output.                     |
| RC2/CCP1            | 33         |          |             |                                                                   |
| RC2                 |            | I/O      | ST          | Digital I/O.                                                      |
| CCP1                |            | I/O      | ST          | Capture 1 input/Compare 1 output/PWM1 output.                     |
| RC3/SCK/SCL         | 34         |          |             |                                                                   |
| RC3                 |            | I/O      | ST          | Digital I/O.                                                      |
| SCK                 |            | I/O      | ST          | Synchronous serial clock input/output for SPI mode.               |
| SCL                 |            | I/O      | ST          | Synchronous serial clock input/output for I <sup>2</sup> C™ mode. |
| RC4/SDI/SDA         | 35         |          |             |                                                                   |
| RC4                 |            | I/O      | ST          | Digital I/O.                                                      |
| SDI                 |            | I        | ST          | SPI data in.                                                      |
| SDA                 |            | I/O      | ST          | I <sup>2</sup> C data I/O.                                        |
| RC5/SDO             | 36         |          |             |                                                                   |
| RC5                 |            | I/O      | ST          | Digital I/O.                                                      |
| SDO                 |            | O        | —           | SPI data out.                                                     |
| RC6/TX1/CK1         | 31         |          |             |                                                                   |
| RC6                 |            | I/O      | ST          | Digital I/O.                                                      |
| TX1                 |            | O        | —           | EUSART asynchronous transmit.                                     |
| CK1                 |            | I/O      | ST          | EUSART synchronous clock (see related RX1/DT1).                   |
| RC7/RX1/DT1         | 32         |          |             |                                                                   |
| RC7                 |            | I/O      | ST          | Digital I/O.                                                      |
| RX1                 |            | I        | ST          | EUSART asynchronous receive.                                      |
| DT1                 |            | I/O      | ST          | EUSART synchronous data (see related TX1/CK1).                    |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.



# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name | Pin Number | Pin Type | Buffer Type | Description                        |
|----------|------------|----------|-------------|------------------------------------|
|          | TQFP       |          |             |                                    |
| RD0/PSP0 | 58         |          |             | PORTD is a bidirectional I/O port. |
| RD0      |            | I/O      | ST          | Digital I/O.                       |
| PSP0     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD1/PSP1 | 55         |          |             |                                    |
| RD1      |            | I/O      | ST          | Digital I/O.                       |
| PSP1     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD2/PSP2 | 54         |          |             |                                    |
| RD2      |            | I/O      | ST          | Digital I/O.                       |
| PSP2     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD3/PSP3 | 53         |          |             |                                    |
| RD3      |            | I/O      | ST          | Digital I/O.                       |
| PSP3     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD4/PSP4 | 52         |          |             |                                    |
| RD4      |            | I/O      | ST          | Digital I/O.                       |
| PSP4     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD5/PSP5 | 51         |          |             |                                    |
| RD5      |            | I/O      | ST          | Digital I/O.                       |
| PSP5     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD6/PSP6 | 50         |          |             |                                    |
| RD6      |            | I/O      | ST          | Digital I/O.                       |
| PSP6     |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD7/PSP7 | 49         |          |             |                                    |
| RD7      |            | I/O      | ST          | Digital I/O.                       |
| PSP7     |            | I/O      | TTL         | Parallel Slave Port data.          |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                                                     | Pin Number | Pin Type   | Buffer Type | Description                                                                                     |
|--------------------------------------------------------------|------------|------------|-------------|-------------------------------------------------------------------------------------------------|
|                                                              | TQFP       |            |             |                                                                                                 |
| RE0/ $\overline{\text{RD}}$<br>RE0<br>$\overline{\text{RD}}$ | 2          | I/O<br>I   | ST<br>TTL   | PORTC is a bidirectional I/O port.<br><br>Digital I/O.<br>Read control for Parallel Slave Port. |
| RE1/ $\overline{\text{WR}}$<br>RE1<br>$\overline{\text{WR}}$ | 1          | I/O<br>I   | ST<br>TTL   | Digital I/O.<br>Write control for Parallel Slave Port.                                          |
| RE2/ $\overline{\text{CS}}$<br>RE2<br>$\overline{\text{CS}}$ | 64         | I/O<br>I   | ST<br>TTL   | Digital I/O.<br>Chip select control for Parallel Slave Port.                                    |
| RE3                                                          | 63         | I/O        | ST          | Digital I/O.                                                                                    |
| RE4                                                          | 62         | I/O        | ST          | Digital I/O.                                                                                    |
| RE5                                                          | 61         | I/O        | ST          | Digital I/O.                                                                                    |
| RE6                                                          | 60         | I/O        | ST          | Digital I/O.                                                                                    |
| RE7/CCP2<br>RE7<br>CCP2 <sup>(2)</sup>                       | 59         | I/O<br>I/O | ST<br>ST    | Digital I/O.<br>Capture 2 input/Compare 2 output/PWM2 output.                                   |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to V<sub>DD</sub>)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name       | Pin Number | Pin Type | Buffer Type | Description                          |
|----------------|------------|----------|-------------|--------------------------------------|
|                | TQFP       |          |             |                                      |
| RF1/AN6/C2OUT  | 17         | I/O      | ST          | PORTF is a bidirectional I/O port.   |
| RF1            |            | I        | Analog      | Digital I/O.                         |
| AN6            |            | O        | —           | Analog input 6.                      |
| C2OUT          |            |          |             | Comparator 2 output.                 |
| RF2/AN7/C1OUT  | 16         | I/O      | ST          |                                      |
| RF2            |            | I        | Analog      | Digital I/O.                         |
| AN7            |            | O        | —           | Analog input 7.                      |
| C1OUT          |            |          |             | Comparator 1 output.                 |
| RF3/AN8        | 15         | I/O      | ST          |                                      |
| RF3            |            | I        | Analog      | Digital I/O.                         |
| AN8            |            |          |             | Analog input 8.                      |
| RF4/AN9        | 14         | I/O      | ST          |                                      |
| RF4            |            | I        | Analog      | Digital I/O.                         |
| AN9            |            |          |             | Analog input 9.                      |
| RF5/AN10/CVREF | 13         | I/O      | ST          |                                      |
| RF5            |            | I        | Analog      | Digital I/O.                         |
| AN10           |            | O        | Analog      | Analog input 10.                     |
| CVREF          |            |          |             | Comparator reference voltage output. |
| RF6/AN11       | 12         | I/O      | ST          |                                      |
| RF6            |            | I        | Analog      | Digital I/O.                         |
| AN11           |            |          |             | Analog input 11.                     |
| RF7/AN5/SS     | 11         | I/O      | ST          |                                      |
| RF7            |            | O        | Analog      | Digital I/O.                         |
| AN5            |            | I        | TTL         | Analog input 5.                      |
| SS             |            |          |             | SPI slave select input.              |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-3: PIC18F6XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name     | Pin Number    | Pin Type | Buffer Type | Description                                                          |
|--------------|---------------|----------|-------------|----------------------------------------------------------------------|
|              | TQFP          |          |             |                                                                      |
| RG0          | 3             | I/O      | ST          | PORTG is a bidirectional I/O port.<br>Digital I/O.                   |
| RG1/TX2/CK2  | 4             | I/O      | ST          | Digital I/O.                                                         |
| RG1          |               | I/O      | ST          | Digital I/O.                                                         |
| TX2          |               | O        | —           | AUSART asynchronous transmit.                                        |
| CK2          |               | I/O      | ST          | AUSART synchronous clock (see related RX2/DT2).                      |
| RG2/RX2/DT2  | 5             | I/O      | ST          | Digital I/O.                                                         |
| RG2          |               | I/O      | ST          | Digital I/O.                                                         |
| RX2          |               | I        | ST          | AUSART asynchronous receive.                                         |
| DT2          |               | I/O      | ST          | AUSART synchronous data (see related TX2/CK2).                       |
| RG3          | 6             | I/O      | ST          | Digital I/O.                                                         |
| RG4          | 8             | I/O      | ST          | Digital I/O.                                                         |
| VSS          | 9, 25, 41, 56 | P        | —           | Ground reference for logic and I/O pins.                             |
| VDD          | 26, 38, 57    | P        | —           | Positive supply for logic and I/O pins.                              |
| AVSS         | 20            | P        | —           | Ground reference for analog modules.                                 |
| AVDD         | 19            | P        | —           | Positive supply for analog modules.                                  |
| ENVREG       | 18            | I        | ST          | Enable for on-chip voltage regulator.                                |
| VDDCORE/VCAP | 10            |          |             | Core logic power or external filter capacitor connection.            |
| VDDCORE      |               | P        | —           | Positive supply for microcontroller core logic (regulator disabled). |
| VCAP         |               | P        | —           | External filter capacitor connection (regulator enabled).            |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**2:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS**

| Pin Name                                 | Pin Number | Pin Type          | Buffer Type             | Description                                                                                                                                                                                                                                                                   |
|------------------------------------------|------------|-------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                          | TQFP       |                   |                         |                                                                                                                                                                                                                                                                               |
| MCLR                                     | 9          | I                 | ST                      | Master Clear (input) or programming voltage (input). This pin is an active-low Reset to the device.                                                                                                                                                                           |
| RA7/OSC1/CLKI<br>RA7<br>OSC1<br>CLKI     | 49         | I/O<br>I<br>I     | TTL<br>CMOS<br>CMOS     | Oscillator crystal or external clock input.<br>General purpose I/O pin.<br>Oscillator crystal input.<br>External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)                                                       |
| RA6/OSC2/CLKO<br>RA6<br>OSC2<br><br>CLKO | 50         | I/O<br>O<br><br>O | TTL<br>—<br><br>—       | Oscillator crystal or clock output.<br>General purpose I/O pin.<br>Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.<br>In EC modes, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. |
| RA0/AN0<br>RA0<br>AN0                    | 30         | I/O<br>I          | TTL<br>Analog           | PORTA is a bidirectional I/O port.<br><br>Digital I/O.<br>Analog input 0.                                                                                                                                                                                                     |
| RA1/AN1<br>RA1<br>AN1                    | 29         | I/O<br>I          | TTL<br>Analog           |                                                                                                                                                                                                                                                                               |
| RA2/AN2/VREF-<br>RA2<br>AN2<br>VREF-     | 28         | I/O<br>I<br>I     | TTL<br>Analog<br>Analog |                                                                                                                                                                                                                                                                               |
| RA3/AN3/VREF+<br>RA3<br>AN3<br>VREF+     | 27         | I/O<br>I<br>I     | TTL<br>Analog<br>Analog |                                                                                                                                                                                                                                                                               |
| RA4/T0CKI<br>RA4<br>T0CKI                | 34         | I/O<br>I          | ST/OD<br>ST             |                                                                                                                                                                                                                                                                               |
| RA5/AN4<br>RA5<br>AN4                    | 33         | I/O<br>I          | TTL<br>Analog           |                                                                                                                                                                                                                                                                               |
| RA6                                      |            |                   |                         |                                                                                                                                                                                                                                                                               |
| RA7                                      |            |                   |                         |                                                                                                                                                                                                                                                                               |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to V<sub>DD</sub>)

- Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).  
**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                                            | Pin Number | Pin Type        | Buffer Type      | Description                                                                                                                                                |
|-----------------------------------------------------|------------|-----------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                     | TQFP       |                 |                  |                                                                                                                                                            |
| RB0/INT0<br>RB0<br>INT0                             | 58         | I/O<br>I        | TTL<br>ST        | PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.<br><br>Digital I/O.<br>External interrupt 0. |
| RB1/INT1<br>RB1<br>INT1                             | 57         | I/O<br>I        | TTL<br>ST        | Digital I/O.<br>External interrupt 1.                                                                                                                      |
| RB2/INT2<br>RB2<br>INT2                             | 56         | I/O<br>I        | TTL<br>ST        | Digital I/O.<br>External interrupt 2.                                                                                                                      |
| RB3/INT3/CCP2<br>RB3<br>INT3<br>CCP2 <sup>(1)</sup> | 55         | I/O<br>I<br>I/O | TTL<br>ST<br>ST  | Digital I/O.<br>External interrupt 3.<br>Capture 2 input/Compare 2 output/PWM2 output.                                                                     |
| RB4/KBI0<br>RB4<br>KBI0                             | 54         | I/O<br>I        | TTL<br>TTL       | Digital I/O.<br>Interrupt-on-change pin.                                                                                                                   |
| RB5/KBI1<br>RB5<br>KBI1                             | 53         | I/O<br>I        | TTL<br>TTL       | Digital I/O.<br>Interrupt-on-change pin.                                                                                                                   |
| RB6/KBI2/PGC<br>RB6<br>KBI2<br>PGC                  | 52         | I/O<br>I<br>I/O | TTL<br>TTL<br>ST | Digital I/O.<br>Interrupt-on-change pin.<br>In-Circuit Debugger and ICSP™ programming clock pin.                                                           |
| RB7/KBI3/PGD<br>RB7<br>KBI3<br>PGD                  | 47         | I/O<br>I<br>I/O | TTL<br>TTL<br>ST | Digital I/O.<br>Interrupt-on-change pin.<br>In-Circuit Debugger and ICSP programming data pin.                                                             |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name            | Pin Number | Pin Type | Buffer Type | Description                                                       |
|---------------------|------------|----------|-------------|-------------------------------------------------------------------|
|                     | TQFP       |          |             |                                                                   |
| RC0/T1OSO/T13CKI    | 36         |          |             | PORTC is a bidirectional I/O port.                                |
| RC0                 |            | I/O      | ST          | Digital I/O.                                                      |
| T1OSO               |            | O        | —           | Timer1 oscillator output.                                         |
| T13CKI              |            | I        | ST          | Timer1/Timer3 external clock input.                               |
| RC1/T1OSI/CCP2      | 35         |          |             |                                                                   |
| RC1                 |            | I/O      | ST          | Digital I/O.                                                      |
| T1OSI               |            | I        | CMOS        | Timer1 oscillator input.                                          |
| CCP2 <sup>(2)</sup> |            | I/O      | ST          | Capture 2 input/Compare 2 output/PWM2 output.                     |
| RC2/CCP1            | 43         |          |             |                                                                   |
| RC2                 |            | I/O      | ST          | Digital I/O.                                                      |
| CCP1                |            | I/O      | ST          | Capture 1 input/Compare 1 output/PWM1 output.                     |
| RC3/SCK/SCL         | 44         |          |             |                                                                   |
| RC3                 |            | I/O      | ST          | Digital I/O.                                                      |
| SCK                 |            | I/O      | ST          | Synchronous serial clock input/output for SPI mode.               |
| SCL                 |            | I/O      | ST          | Synchronous serial clock input/output for I <sup>2</sup> C™ mode. |
| RC4/SDI/SDA         | 45         |          |             |                                                                   |
| RC4                 |            | I/O      | ST          | Digital I/O.                                                      |
| SDI                 |            | I        | ST          | SPI data in.                                                      |
| SDA                 |            | I/O      | ST          | I <sup>2</sup> C data I/O.                                        |
| RC5/SDO             | 46         |          |             |                                                                   |
| RC5                 |            | I/O      | ST          | Digital I/O.                                                      |
| SDO                 |            | O        | —           | SPI data out.                                                     |
| RC6/TX1/CK1         | 37         |          |             |                                                                   |
| RC6                 |            | I/O      | ST          | Digital I/O.                                                      |
| TX1                 |            | O        | —           | EUSART asynchronous transmit.                                     |
| CK1                 |            | I/O      | ST          | EUSART synchronous clock (see related RX1/DT1).                   |
| RC7/RX1/DT1         | 38         |          |             |                                                                   |
| RC7                 |            | I/O      | ST          | Digital I/O.                                                      |
| RX1                 |            | I        | ST          | EUSART asynchronous receive.                                      |
| DT1                 |            | I/O      | ST          | EUSART synchronous data (see related TX1/CK1).                    |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name     | Pin Number | Pin Type | Buffer Type | Description                        |
|--------------|------------|----------|-------------|------------------------------------|
|              | TQFP       |          |             |                                    |
| RD0/AD0/PSP0 | 72         |          |             | PORTD is a bidirectional I/O port. |
| RD0          |            | I/O      | ST          | Digital I/O.                       |
| AD0          |            | I/O      | TTL         | External memory address/data 0.    |
| PSP0         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD1/AD1/PSP1 | 69         |          |             |                                    |
| RD1          |            | I/O      | ST          | Digital I/O.                       |
| AD1          |            | I/O      | TTL         | External memory address/data 1.    |
| PSP1         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD2/AD2/PSP2 | 68         |          |             |                                    |
| RD2          |            | I/O      | ST          | Digital I/O.                       |
| AD2          |            | I/O      | TTL         | External memory address/data 2.    |
| PSP2         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD3/AD3/PSP3 | 67         |          |             |                                    |
| RD3          |            | I/O      | ST          | Digital I/O.                       |
| AD3          |            | I/O      | TTL         | External memory address/data 3.    |
| PSP3         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD4/AD4/PSP4 | 66         |          |             |                                    |
| RD4          |            | I/O      | ST          | Digital I/O.                       |
| AD4          |            | I/O      | TTL         | External memory address/data 4.    |
| PSP4         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD5/AD5/PSP5 | 65         |          |             |                                    |
| RD5          |            | I/O      | ST          | Digital I/O.                       |
| AD5          |            | I/O      | TTL         | External memory address/data 5.    |
| PSP5         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD6/AD6/PSP6 | 64         |          |             |                                    |
| RD6          |            | I/O      | ST          | Digital I/O.                       |
| AD6          |            | I/O      | TTL         | External memory address/data 6.    |
| PSP6         |            | I/O      | TTL         | Parallel Slave Port data.          |
| RD7/AD7/PSP7 | 63         |          |             |                                    |
| RD7          |            | I/O      | ST          | Digital I/O.                       |
| AD7          |            | I/O      | TTL         | External memory address/data 7.    |
| PSP7         |            | I/O      | TTL         | Parallel Slave Port data.          |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.



# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                         | Pin Number | Pin Type | Buffer Type | Description                                   |
|----------------------------------|------------|----------|-------------|-----------------------------------------------|
|                                  | TQFP       |          |             |                                               |
| RE0/ $\overline{\text{RD}}$ /AD8 | 4          |          |             | PORTE is a bidirectional I/O port.            |
| RE0                              |            | I/O      | ST          | Digital I/O.                                  |
| $\overline{\text{RD}}$           |            | I        | TTL         | Read control for Parallel Slave Port.         |
| AD8                              |            | I/O      | TTL         | External memory address/data 8.               |
| RE1/ $\overline{\text{WR}}$ /AD9 | 3          |          |             |                                               |
| RE1                              |            | I/O      | ST          | Digital I/O.                                  |
| $\overline{\text{WR}}$           |            | I        | TTL         | Write control for Parallel Slave Port.        |
| AD9                              |            | I/O      | TTL         | External memory address/data 9.               |
| RE2/AD10/ $\overline{\text{CS}}$ | 78         |          |             |                                               |
| RE2                              |            | I/O      | ST          | Digital I/O.                                  |
| AD10                             |            | I/O      | TTL         | External memory address/data 10.              |
| $\overline{\text{CS}}$           |            | I        | TTL         | Chip select control for Parallel Slave Port.  |
| RE3/AD11                         | 77         |          |             |                                               |
| RE3                              |            | I/O      | ST          | Digital I/O.                                  |
| AD11                             |            | I/O      | TTL         | External memory address/data 11.              |
| RE4/AD12                         | 76         |          |             |                                               |
| RE4                              |            | I/O      | ST          | Digital I/O.                                  |
| AD12                             |            | I/O      | TTL         | External memory address/data 12.              |
| RE5/AD13                         | 75         |          |             |                                               |
| RE5                              |            | I/O      | ST          | Digital I/O.                                  |
| AD13                             |            | I/O      | TTL         | External memory address/data 13.              |
| RE6/AD14                         | 74         |          |             |                                               |
| RE6                              |            | I/O      | ST          | Digital I/O.                                  |
| AD14                             |            | I/O      | TTL         | External memory address/data 14.              |
| RE7/AD15/CCP2                    | 73         |          |             |                                               |
| RE7                              |            | I/O      | ST          | Digital I/O.                                  |
| AD15                             |            | I/O      | TTL         | External memory address/data 15.              |
| CCP2 <sup>(3)</sup>              |            | I/O      | ST          | Capture 2 input/Compare 2 output/PWM2 output. |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to V<sub>DD</sub>)

- Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).  
**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                 | Pin Number | Pin Type | Buffer Type | Description                          |
|--------------------------|------------|----------|-------------|--------------------------------------|
|                          | TQFP       |          |             |                                      |
| RF1/AN6/C2OUT            | 23         | I/O      | ST          | PORTF is a bidirectional I/O port.   |
| RF1                      |            | I        | Analog      | Digital I/O.                         |
| AN6                      |            | O        | —           | Analog input 6.                      |
| C2OUT                    |            |          |             | Comparator 2 output.                 |
| RF2/AN7/C1OUT            | 18         | I/O      | ST          | Digital I/O.                         |
| RF2                      |            | I        | Analog      | Analog input 7.                      |
| AN7                      |            | O        | —           | Comparator 1 output.                 |
| C1OUT                    |            |          |             |                                      |
| RF3/AN8                  | 17         | I/O      | ST          | Digital I/O.                         |
| RF3                      |            | I        | Analog      | Analog input 8.                      |
| AN8                      |            |          |             |                                      |
| RF4/AN9                  | 16         | I/O      | ST          | Digital I/O.                         |
| RF4                      |            | I        | Analog      | Analog input 9.                      |
| AN9                      |            |          |             |                                      |
| RF5/AN10/CVREF           | 15         | I/O      | ST          | Digital I/O.                         |
| RF5                      |            | I        | Analog      | Analog input 10.                     |
| AN10                     |            | O        | Analog      | Comparator reference voltage output. |
| CVREF                    |            |          |             |                                      |
| RF6/AN11                 | 14         | I/O      | ST          | Digital I/O.                         |
| RF6                      |            | I        | Analog      | Analog input 11.                     |
| AN11                     |            |          |             |                                      |
| RF7/AN5/ $\overline{SS}$ | 13         | I/O      | ST          | Digital I/O.                         |
| RF7                      |            | O        | Analog      | Analog input 5.                      |
| AN5                      |            | I        | TTL         | SPI slave select input.              |
| $\overline{SS}$          |            |          |             |                                      |

**Legend:** TTL = TTL compatible input CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels Analog = Analog input  
I = Input O = Output  
P = Power OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name    | Pin Number | Pin Type | Buffer Type | Description                                        |
|-------------|------------|----------|-------------|----------------------------------------------------|
|             | TQFP       |          |             |                                                    |
| RG0         | 5          | I/O      | ST          | PORTG is a bidirectional I/O port.<br>Digital I/O. |
| RG1/TX2/CK2 | 6          | I/O      | ST          | Digital I/O.                                       |
| RG1         |            | I/O      | ST          | Digital I/O.                                       |
| TX2         |            | O        | —           | AUSART asynchronous transmit.                      |
| CK2         |            | I/O      | ST          | AUSART synchronous clock (see related RX2/DT2).    |
| RG2/RX2/DT2 | 7          | I/O      | ST          | Digital I/O.                                       |
| RG2         |            | I/O      | ST          | Digital I/O.                                       |
| RX2         |            | I        | ST          | AUSART asynchronous receive.                       |
| DT2         |            | I/O      | ST          | AUSART synchronous data (see related TX2/CK2).     |
| RG3         | 8          | I/O      | ST          | Digital I/O.                                       |
| RG4         | 10         | I/O      | ST          | Digital I/O.                                       |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name              | Pin Number | Pin Type   | Buffer Type | Description                                                                                |
|-----------------------|------------|------------|-------------|--------------------------------------------------------------------------------------------|
|                       | TQFP       |            |             |                                                                                            |
| RH0/A16<br>RH0<br>A16 | 79         | I/O<br>I/O | ST<br>TTL   | PORTH is a bidirectional I/O port.<br><br>Digital I/O.<br>External memory address/data 16. |
| RH1/A17<br>RH1<br>A17 | 80         | I/O<br>I/O | ST<br>TTL   | Digital I/O.<br>External memory address/data 17.                                           |
| RH2/A18<br>RH2<br>A18 | 1          | I/O<br>I/O | ST<br>TTL   | Digital I/O.<br>External memory address/data 18.                                           |
| RH3/A19<br>RH3<br>A19 | 2          | I/O<br>I/O | ST<br>TTL   | Digital I/O.<br>External memory address/data 19.                                           |
| RH4                   | 22         | I/O        | ST          | Digital I/O.                                                                               |
| RH5                   | 21         | I/O        | ST          | Digital I/O.                                                                               |
| RH6                   | 20         | I/O        | ST          | Digital I/O.                                                                               |
| RH7                   | 19         | I/O        | ST          | Digital I/O.                                                                               |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

- Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).  
**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.  
**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

**TABLE 1-4: PIC18F8XJ11 PINOUT I/O DESCRIPTIONS (CONTINUED)**

| Pin Name                                                       | Pin Number     | Pin Type | Buffer Type | Description                                                                                                                       |
|----------------------------------------------------------------|----------------|----------|-------------|-----------------------------------------------------------------------------------------------------------------------------------|
|                                                                | TQFP           |          |             |                                                                                                                                   |
| RJ0/ALE<br>RJ0<br>ALE                                          | 62             | I/O<br>O | ST<br>—     | PORTJ is a bidirectional I/O port.<br><br>Digital I/O.<br>External memory address latch enable.                                   |
| RJ1/ $\overline{\text{OE}}$<br>RJ1<br>$\overline{\text{OE}}$   | 61             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory output enable.                                                                                    |
| RJ2/ $\overline{\text{WRL}}$<br>RJ2<br>$\overline{\text{WRL}}$ | 60             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory write low control.                                                                                |
| RJ3/ $\overline{\text{WRH}}$<br>RJ3<br>$\overline{\text{WRH}}$ | 59             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory write high control.                                                                               |
| RJ4/BA0<br>RJ4<br>BA0                                          | 39             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory byte address 0 control.                                                                           |
| RJ5/ $\overline{\text{CE}}$<br>RJ5<br>$\overline{\text{CE}}$   | 40             | I/O<br>O | ST<br>—     | Digital I/O<br>External memory chip enable control.                                                                               |
| RJ6/ $\overline{\text{LB}}$<br>RJ6<br>$\overline{\text{LB}}$   | 41             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory low byte control.                                                                                 |
| RJ7/ $\overline{\text{UB}}$<br>RJ7<br>$\overline{\text{UB}}$   | 42             | I/O<br>O | ST<br>—     | Digital I/O.<br>External memory high byte control.                                                                                |
| Vss                                                            | 11, 31, 51, 70 | P        | —           | Ground reference for logic and I/O pins.                                                                                          |
| VDD                                                            | 32, 48, 71     | P        | —           | Positive supply for logic and I/O pins.                                                                                           |
| AVss                                                           | 26             | P        | —           | Ground reference for analog modules.                                                                                              |
| AVDD                                                           | 25             | P        | —           | Positive supply for analog modules.                                                                                               |
| ENVREG                                                         | 24             | I        | ST          | Enable for on-chip voltage regulator.                                                                                             |
| VDDCORE/VCAP<br>VDDCORE                                        | 12             | P        | —           | Core logic power or external filter capacitor connection.<br>Positive supply for microcontroller core logic (regulator disabled). |
| VCAP                                                           |                | P        | —           | External filter capacitor connection (regulator enabled).                                                                         |

**Legend:** TTL = TTL compatible input      CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels      Analog = Analog input  
I = Input      O = Output  
P = Power      OD = Open-Drain (no P diode to VDD)

**Note 1:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (80-pin devices, Extended Microcontroller mode only).

**2:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

# PIC18F85J11 FAMILY

---

NOTES:

# PIC18F85J11 FAMILY

## 2.0 OSCILLATOR CONFIGURATIONS

### 2.1 Oscillator Types

The PIC18F85J11 family of devices can be operated in six different oscillator modes:

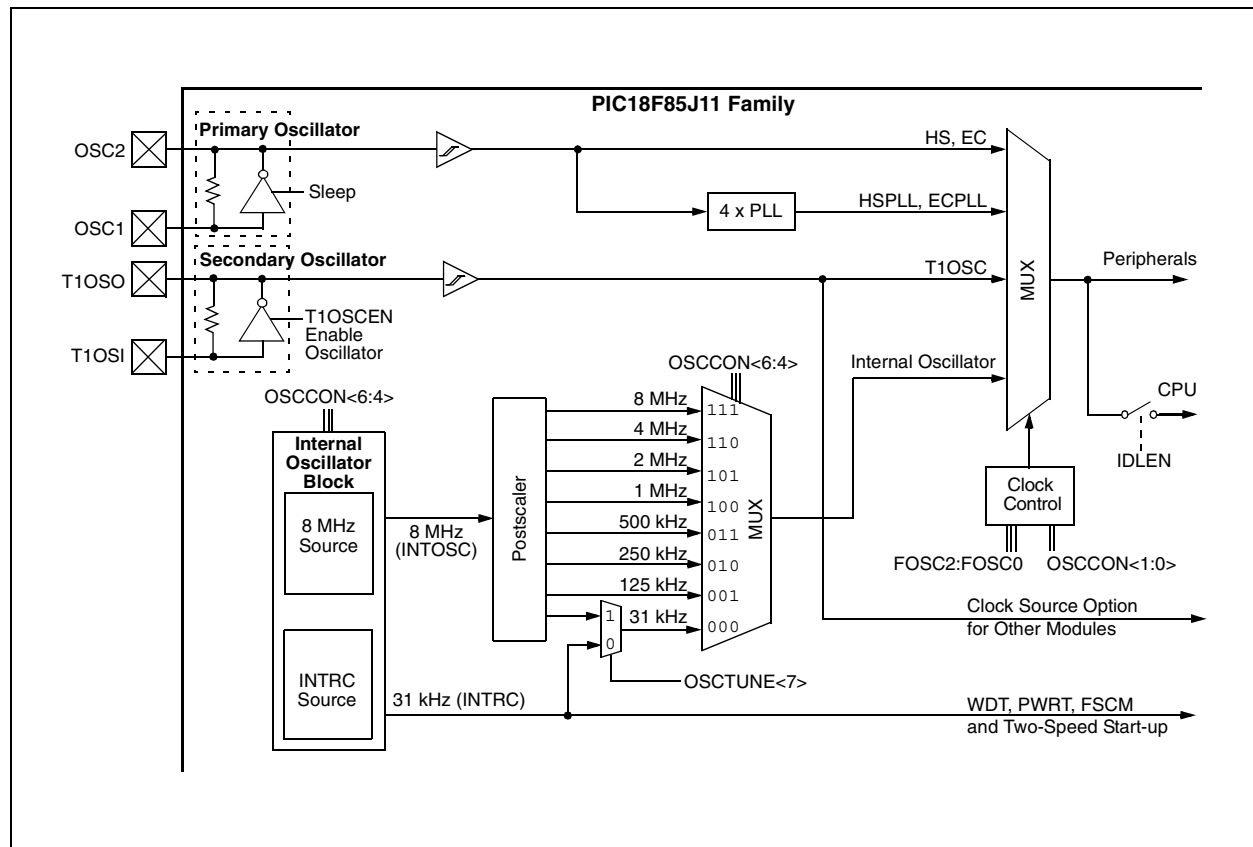
1. HS High-Speed Crystal/Resonator
2. HSPLL High-Speed Crystal/Resonator with Software PLL Control
3. EC External Clock with Fosc/4 Output
4. ECPLL External Clock with Software PLL Control
5. INTOSC Internal Fast RC (8 MHz) Oscillator
6. INTRC Internal 31 kHz Oscillator

Five of these are selected by the user by programming the FOSC2:FOSC0 Configuration bits. The sixth mode (INTRC) may be invoked under software control; it can also be configured as the default mode on device Resets.

In addition, PIC18F85J11 family devices can switch between different clock sources, either under software control or automatically under certain conditions. This allows for additional power savings by managing device clock speed in real time without resetting the application.

The clock sources for the PIC18F85J11 family of devices are shown in Figure 2-1.

**FIGURE 2-1: PIC18F85J11 FAMILY CLOCK DIAGRAM**



# PIC18F85J11 FAMILY

## 2.2 Control Registers

The OSCCON register (Register 2-1) controls the main aspects of the device clock's operation. It selects the oscillator type to be used, which of the power-managed modes to invoke and the output frequency of the INTOSC source. It also provides status on the oscillators.

The OSCTUNE register (Register 2-2) controls the tuning and operation of the internal oscillator block. It also implements the PLEN bits which control the operation of the Phase Locked Loop (PLL) in Internal Oscillator modes (see **Section 2.4.3 “PLL Frequency Multiplier”**).

### REGISTER 2-1: OSCCON: OSCILLATOR CONTROL REGISTER

| R/W-0 | R/W-1                | R/W-0                | R/W-0                | R <sup>(1)</sup> | R-0  | R/W-0               | R/W-0               |
|-------|----------------------|----------------------|----------------------|------------------|------|---------------------|---------------------|
| IDLEN | IRCF2 <sup>(2)</sup> | IRCF1 <sup>(2)</sup> | IRCF0 <sup>(2)</sup> | OSTS             | IOFS | SCS1 <sup>(4)</sup> | SCS0 <sup>(4)</sup> |
| bit 7 |                      |                      |                      |                  |      |                     | bit 0               |

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

- bit 7      **IDLEN:** Idle Enable bit  
1 = Device enters an Idle mode when a *SLEEP* instruction is executed  
0 = Device enters Sleep mode when a *SLEEP* instruction is executed
- bit 6-4    **IRCF2:IRCF0:** INTOSC Source Frequency Select bits<sup>(2)</sup>  
111 = 8 MHz (INTOSC drives clock directly)  
110 = 4 MHz  
101 = 2 MHz  
100 = 1 MHz (default)  
011 = 500 kHz  
010 = 250 kHz  
001 = 125 kHz  
000 = 31 kHz (from either INTOSC/256 or INTRC)<sup>(3)</sup>
- bit 3      **OSTS:** Oscillator Start-up Time-out Status bit<sup>(1)</sup>  
1 = Oscillator Start-up Timer (OST) time-out has expired; primary oscillator is running  
0 = Oscillator Start-up Timer (OST) time-out is running; primary oscillator is not ready
- bit 2      **IOFS:** INTOSC Frequency Stable bit  
1 = Fast RC oscillator frequency is stable  
0 = Fast RC oscillator frequency is not stable
- bit 1-0    **SCS1:SCS0:** System Clock Select bits<sup>(4)</sup>  
11 = Internal oscillator block  
10 = Primary oscillator  
01 = Timer1 oscillator  
When FOSC2 = 1:  
00 = Primary oscillator  
When FOSC2 = 0:  
00 = Internal oscillator

**Note 1:** Reset state depends on state of the IESO Configuration bit.

**2:** Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.

**3:** Source selected by the INTSRC bit (OSCTUNE<7>), see text.

**4:** Modifying these bits will cause an immediate clock source switch.



# PIC18F85J11 FAMILY

## REGISTER 2-2: OSCTUNE: OSCILLATOR TUNING REGISTER

|        |                      |       |       |       |       |       |       |
|--------|----------------------|-------|-------|-------|-------|-------|-------|
| R/W-0  | R/W-0                | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| INTSRC | PLLEN <sup>(1)</sup> | TUN5  | TUN4  | TUN3  | TUN2  | TUN1  | TUN0  |
| 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      **INTSRC:** Internal Oscillator Low-Frequency Source Select bit

1 = 31.25 kHz device clock derived from 8 MHz INTOSC source (divide-by-256 enabled)

0 = 31 kHz device clock derived from INTRC 31 kHz oscillator

bit 6      **PLEN:** Frequency Multiplier PLL Enable bit<sup>(1)</sup>

1 = PLL enabled

0 = PLL disabled

bit 5-0      **TUN5:TUN0:** Fast RC Oscillator (INTOSC) Frequency Tuning bits

011111 = Maximum frequency

•

•

●

•

000001

000000 = Center frequency. Fast RC oscillator is running at the calibrated frequency.

111111

•

•

•

•

100000 = Minimum frequency

**Note 1:** Available only for ECPLL and HSPLL oscillator configurations; otherwise, this bit is unavailable and reads as '0'.

## 2.3 Clock Sources and Oscillator Switching

Essentially, PIC18F85J11 family devices have three independent clock sources:

- Primary oscillators
- Secondary oscillators
- Internal oscillator

The **primary oscillators** can be thought of as the main device oscillators. These are any external oscillators connected to the OSC1 and OSC2 pins, and include the External Crystal and Resonator modes and the External Clock modes. In some circumstances, the internal oscillator block may be considered a primary oscillator. The particular mode is defined by the FOSC Configuration bits. The details of these modes are covered in **Section 2.4 “External Oscillator Modes”**.

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

PIC18F85J11 family devices offer the Timer1 oscillator as a secondary oscillator source. This oscillator, in all power-managed modes, is often the time base for functions such as a Real-Time Clock. The Timer1 oscillator is discussed in greater detail in **Section 12.3 “Timer1 Oscillator”**

In addition to being a primary clock source in some circumstances, the **internal oscillator** is available as a power-managed mode clock source. The INTRC source is also used as the clock source for several special features, such as the WDT and Fail-Safe Clock Monitor. The internal oscillator block is discussed in more detail in **Section 2.5 “Internal Oscillator Block”**

The PIC18F85J11 family includes features that allow the device clock source to be switched from the main oscillator, chosen by device configuration, to one of the alternate clock sources. When an alternate clock source is enabled, various power-managed operating modes are available.

# PIC18F85J11 FAMILY

## 2.3.1 CLOCK SOURCE SELECTION

The System Clock Select bits, SCS1:SCS0 (OSCCON<1:0>), select the clock source. The available clock sources are the primary clock, defined by the FOSC1:FOSC0 Configuration bits, the secondary clock (Timer1 oscillator) and the internal oscillator. The clock source changes after one or more of the bits are written to, following a brief clock transition interval.

The OSTS (OSCCON<3>) and T1RUN (T1CON<6>) bits indicate which clock source is currently providing the device clock. The OSTS bit indicates that the Oscillator Start-up Timer (OST) has timed out and the primary clock is providing the device clock in primary clock modes. The T1RUN bit indicates when the Timer1 oscillator is providing the device clock in secondary clock modes. In power-managed modes, only one of these bits will be set at any time. If neither of these bits are set, the INTRC is providing the clock, or the internal oscillator has just started and is not yet stable.

The IDLEN bit determines if the device goes into Sleep mode or one of the Idle modes when the SLEEP instruction is executed.

The use of the flag and control bits in the OSCCON register is discussed in more detail in **Section 3.0 “Power-Managed Modes”**.

**Note 1:** The Timer1 oscillator must be enabled to select the secondary clock source. The Timer1 oscillator is enabled by setting the T1OSCEN bit in the Timer1 Control register (T1CON<3>). If the Timer1 oscillator is not enabled, then any attempt to select a secondary clock source when executing a SLEEP instruction will be ignored.

**2:** It is recommended that the Timer1 oscillator be operating and stable before executing the SLEEP instruction or a very long delay may occur while the Timer1 oscillator starts.

### 2.3.1.1 System Clock Selection and the FOSC2 Configuration Bit

The SCS bits are cleared on all forms of Reset. In the device's default configuration, this means the primary oscillator defined by FOSC1:FOSC0 (that is, one of the HS or EC modes) is used as the primary clock source on device Resets.

The default clock configuration on Reset can be changed with the FOSC2 Configuration bit. This bit determines whether the external or internal oscillator will be the default clock source on subsequent device Resets. By extension, it also has the effect of determining the clock source selected when SCS1:SCS0 are in their Reset state (= 00). When FOSC2 = 1 (default), the oscillator source defined by FOSC1:FOSC0 is selected whenever SCS1:SCS0 = 00. When FOSC2 = 0, the internal oscillator block is selected whenever SCS1:SCS2 = 00.

In those cases when the internal oscillator block is the default clock on Reset, the Fast RC oscillator (INTOSC) will be used as the device clock source. It will initially start at 1 MHz; the postscaler selection that corresponds to the Reset value of the IRCF2:IRCF0 bits ('100').

Regardless of the setting of FOSC2, INTRC will always be enabled on device power-up. It serves as the clock source until the device has loaded its configuration values from memory. It is at this point that the FOSC Configuration bits are read and the oscillator selection of the operational mode is made.

Note that either the primary clock or the internal oscillator will have two bit setting options for the possible values of SCS1:SCS0, at any given time, depending on the setting of FOSC2.

## 2.3.2 OSCILLATOR TRANSITIONS

PIC18F85J11 family devices contain circuitry to prevent clock “glitches” when switching between clock sources. A short pause in the device clock occurs during the clock switch. The length of this pause is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Clock transitions are discussed in greater detail in **Section 3.1.2 “Entering Power-Managed Modes”**.

# PIC18F85J11 FAMILY

## 2.4 External Oscillator Modes

### 2.4.1 CRYSTAL OSCILLATOR/CERAMIC RESONATORS (HS MODES)

In HS or HSPLL Oscillator modes, a crystal or ceramic resonator is connected to the OSC1 and OSC2 pins to establish oscillation. Figure 2-2 shows the pin connections.

The oscillator design requires the use of a parallel cut crystal.

**Note:** Use of a series cut crystal may give a frequency out of the crystal manufacturer's specifications.

**TABLE 2-1: CAPACITOR SELECTION FOR CERAMIC RESONATORS**

| Typical Capacitor Values Used: |          |       |       |
|--------------------------------|----------|-------|-------|
| Mode                           | Freq.    | OSC1  | OSC2  |
| HS                             | 8.0 MHz  | 27 pF | 27 pF |
|                                | 16.0 MHz | 22 pF | 22 pF |

**Capacitor values are for design guidance only.**

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application. Refer to the following application notes for oscillator specific information:

- AN588, "PIC® Microcontroller Oscillator Design Guide"
- AN826, "Crystal Oscillator Basics and Crystal Selection for rPIC® and PIC® Devices"
- AN849, "Basic PIC® Oscillator Design"
- AN943, "Practical PIC® Oscillator Analysis and Design"
- AN949, "Making Your Oscillator Work"

See the notes following Table 2-2 for additional information.

**TABLE 2-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR**

| Osc Type | Crystal Freq. | Typical Capacitor Values Tested: |       |
|----------|---------------|----------------------------------|-------|
|          |               | C1                               | C2    |
| HS       | 4 MHz         | 27 pF                            | 27 pF |
|          | 8 MHz         | 22 pF                            | 22 pF |
|          | 20 MHz        | 15 pF                            | 15 pF |

**Capacitor values are for design guidance only.**

Different capacitor values may be required to produce acceptable oscillator operation. The user should test the performance of the oscillator over the expected VDD and temperature range for the application.

Refer to the Microchip application notes cited in Table 2-1 for oscillator specific information. Also see the notes following this table for additional information.

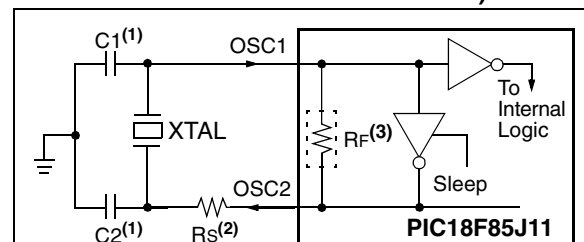
**Note 1:** Higher capacitance increases the stability of oscillator but also increases the start-up time.

**2:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

**3:** Rs may be required to avoid overdriving crystals with low drive level specification.

**4:** Always verify oscillator performance over the VDD and temperature range that is expected for the application.

**FIGURE 2-2: CRYSTAL/CERAMIC RESONATOR OPERATION (HS OR HSPLL CONFIGURATION)**



**Note 1:** See Table 2-1 and Table 2-2 for initial values of C1 and C2.

**2:** A series resistor (Rs) may be required for AT strip cut crystals.

**3:** Rf varies with the oscillator mode chosen.

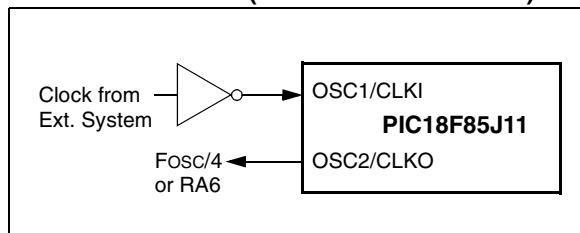
# PIC18F85J11 FAMILY

## 2.4.2 EXTERNAL CLOCK INPUT (EC MODES)

The EC and ECPLL Oscillator modes require an external clock source to be connected to the OSC1 pin. There is no oscillator start-up time required after a Power-on Reset or after an exit from Sleep mode.

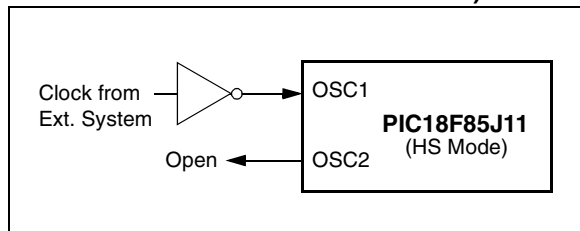
In the EC Oscillator mode, the oscillator frequency divided by 4 is available on the OSC2 pin. This signal may be used for test purposes or to synchronize other logic. Figure 2-3 shows the pin connections for the EC Oscillator mode.

**FIGURE 2-3: EXTERNAL CLOCK INPUT OPERATION (EC CONFIGURATION)**



An external clock source may also be connected to the OSC1 pin in the HS mode, as shown in Figure 2-4. In this configuration, the divide-by-4 output on OSC2 is not available.

**FIGURE 2-4: EXTERNAL CLOCK INPUT OPERATION (HS OSC CONFIGURATION)**

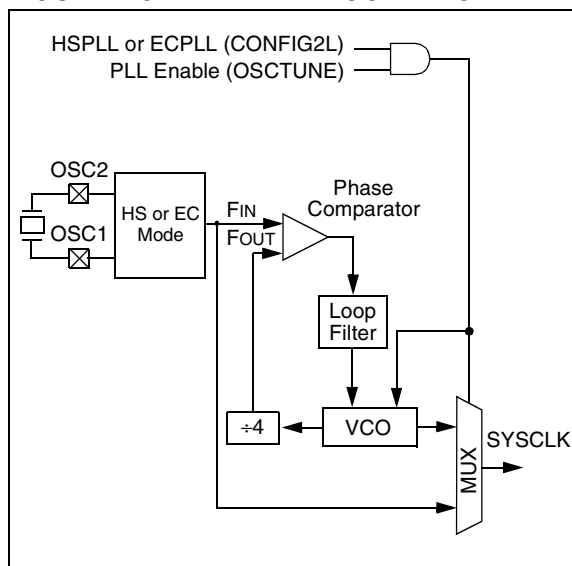


## 2.4.3 PLL FREQUENCY MULTIPLIER

A Phase Locked Loop (PLL) circuit is provided as an option for users who want to use a lower frequency oscillator circuit, or to clock the device up to its highest rated frequency from a crystal oscillator. This may be useful for customers who are concerned with EMI due to high-frequency crystals, or users who require higher clock speeds from an internal oscillator. For these reasons, the HSPLL and ECPLL modes are available.

The HSPLL and ECPLL modes provide the ability to selectively run the device at 4 times the external oscillating source to produce frequencies up to 40 MHz. The PLL is enabled by programming the FOSC2:FOSC0 Configuration bits (CONFIG2L<2:0>) to either '110' (for ECPLL) or '100' (for HSPLL). In addition, the PLEN bit (OSCTUNE<6>) must also be set. Clearing PLEN disables the PLL, regardless of the chosen oscillator configuration. It also allows additional flexibility for controlling the application's clock speed in software.

**FIGURE 2-5: PLL BLOCK DIAGRAM**



## 2.5 Internal Oscillator Block

The PIC18F85J11 family of devices includes an internal oscillator block which generates two different clock signals; either can be used as the microcontroller's clock source. This may eliminate the need for an external oscillator circuit on the OSC1 and/or OSC2 pins.

The main output is the Fast RC oscillator, or INTOSC, an 8 MHz clock source which can be used to directly drive the device clock. It also drives a postscaler which can provide a range of clock frequencies from 31 kHz to 4 MHz. INTOSC is enabled when a clock frequency from 125 kHz to 8 MHz is selected. The INTOSC output can also be enabled when 31 kHz is selected, depending on the INTSRC bit (OSCTUNE<7>).

The other clock source is the Internal RC oscillator (INTRC), which provides a nominal 31 kHz output. INTRC is enabled if it is selected as the device clock source. It is also enabled automatically when any of the following are enabled:

- Power-up Timer
- Fail-Safe Clock Monitor
- Watchdog Timer
- Two-Speed Start-up

These features are discussed in greater detail in **Section 22.0 "Special Features of the CPU"**.

The clock source frequency (INTOSC direct, INTOSC with postscaler or INTRC direct) is selected by configuring the IRCF bits of the OSCCON register. The default frequency on device Resets is 1 MHz.

### 2.5.1 OSC1 AND OSC2 PIN CONFIGURATION

Whenever the internal oscillator is configured as the default clock source (FOSC2 = 0), the OSC1 and OSC2 pins are reconfigured automatically as port pins, RA6 and RA7. In this mode, they function as general digital I/O. All oscillator functions on the pins are disabled.

### 2.5.2 INTERNAL OSCILLATOR OUTPUT FREQUENCY AND TUNING

The internal oscillator block is calibrated at the factory to produce an INTOSC output frequency of 8 MHz. It can be adjusted in the user's application by writing to TUN5:TUN0 (OSCTUNE<5:0>) in the OSCTUNE register (Register 2-2).

When the OSCTUNE register is modified, the INTOSC frequency will begin shifting to the new frequency. The oscillator will stabilize within 1 ms. Code execution continues during this shift. There is no indication that the shift has occurred.

The INTRC oscillator operates independently of the INTOSC source. Any changes in INTOSC across voltage and temperature are not necessarily reflected by changes in INTRC or vice versa. The frequency of INTRC is not affected by OSCTUNE.

### 2.5.3 INTOSC FREQUENCY DRIFT

The INTOSC frequency may drift as VDD or temperature changes, and can affect the controller operation in a variety of ways. It is possible to adjust the INTOSC frequency by modifying the value in the OSCTUNE register. This will have no effect on the INTRC clock source frequency.

Tuning INTOSC requires knowing when to make the adjustment, in which direction it should be made, and in some cases, how large a change is needed. Three compensation techniques are shown here.

#### 2.5.3.1 Compensating with the EUSART

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

#### 2.5.3.2 Compensating with the Timers

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

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

#### 2.5.3.3 Compensating with the CCP Module in Capture Mode

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

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

# PIC18F85J11 FAMILY

## 2.6 Effects of Power-Managed Modes on the Various Clock Sources

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

In secondary clock modes (SEC\_RUN and SEC\_IDLE), the Timer1 oscillator is operating and providing the device clock. The Timer1 oscillator may also run in all power-managed modes if required to clock Timer1 or Timer3.

In RC\_RUN and RC\_IDLE modes, the internal oscillator provides the device clock source. The 31 kHz INTRC output can be used directly to provide the clock and may be enabled to support various special features, regardless of the power-managed mode (see **Section 22.2 “Watchdog Timer (WDT)”** through **Section 22.5 “Fail-Safe Clock Monitor”** for more information on WDT, Fail-Safe Clock Monitor and Two-Speed Start-up).

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

Enabling any on-chip feature that will operate during Sleep will increase the current consumed during Sleep. The INTRC is required to support WDT operation. The

Timer1 oscillator may be operating to support a Real-Time Clock (RTC). Other features may be operating that do not require a device clock source (i.e., MSSP slave, PSP, INTn pins and others). Peripherals that may add significant current consumption are listed in **Section 25.2 “DC Characteristics: Power-Down and Supply Current”**.

## 2.7 Power-up Delays

Power-up delays are controlled by two timers, so that no external Reset circuitry is required for most applications. The delays ensure that the device is kept in Reset until the device power supply is stable under normal circumstances and the primary clock is operating and stable. For additional information on power-up delays, see **Section 4.6 “Power-up Timer (PWRT)”**.

The first timer is the Power-up Timer (PWRT), which provides a fixed delay on power-up (parameter 33, Table 25-10). It is always enabled.

The second timer is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable (HS modes). The OST does this by counting 1024 oscillator cycles before allowing the oscillator to clock the device.

There is a delay of interval TCSD (parameter 38, Table 25-10), following POR, while the controller becomes ready to execute instructions.

**TABLE 2-3: OSC1 AND OSC2 PIN STATES IN SLEEP MODE**

| Oscillator Mode | OSC1 Pin                                              | OSC2 Pin                                              |
|-----------------|-------------------------------------------------------|-------------------------------------------------------|
| EC, ECPLL       | Floating, pulled by external clock                    | At logic low (clock/4 output)                         |
| HS, HSPLL       | Feedback inverter disabled at quiescent voltage level | Feedback inverter disabled at quiescent voltage level |
| INTOSC          | I/O pin, RA6, direction controlled by TRISA<6>        | I/O pin, RA7, direction controlled by TRISA<7>        |

**Note:** See Table 4-2 in **Section 4.0 “Reset”** for time-outs due to Sleep and MCLR Reset.

## 3.0 POWER-MANAGED MODES

The PIC18F85J11 family devices provide the ability to manage power consumption by simply managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. For the sake of managing power in an application, there are three primary modes of operation:

- Run mode
- Idle mode
- Sleep mode

These modes define which portions of the device are clocked and at what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block); the Sleep mode does not use a clock source.

The power-managed modes include several power-saving features offered on previous PIC® MCUs. One is the clock switching feature, offered in other PIC18 devices, allowing the controller to use the Timer1 oscillator in place of the primary oscillator. Also included is the Sleep mode, offered by all PIC MCUs, where all device clocks are stopped.

### 3.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions: if the CPU is to be clocked or not and which clock source is to be used. The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS1:SCS0 bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 3-1.

### 3.1.1 CLOCK SOURCES

The SCS1:SCS0 bits allow the selection of one of three clock sources for power-managed modes. They are:

- the primary clock, as defined by the FOSC2:FOSC0 Configuration bits
- the secondary clock (Timer1 oscillator)
- the internal oscillator

### 3.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS1:SCS0 bits select the clock source and determine which Run or Idle mode is to be used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These are discussed in **Section 3.1.3 “Clock Transitions and Status Indicators”** and subsequent sections.

Entry to the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current mode and the mode being switched to, a change to a power-managed mode does not always require setting all of these bits. Many transitions may be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured correctly, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

**TABLE 3-1: POWER-MANAGED MODES**

| Mode     | OSCCON bits             |                | Module Clocking |             | Available Clock and Oscillator Source                                         |
|----------|-------------------------|----------------|-----------------|-------------|-------------------------------------------------------------------------------|
|          | IDLEN<7> <sup>(1)</sup> | SCS1:SCS0<1:0> | CPU             | Peripherals |                                                                               |
| Sleep    | 0                       | N/A            | Off             | Off         | None – All clocks are disabled                                                |
| PRI_RUN  | N/A                     | 10             | Clocked         | Clocked     | Primary – HS, EC, HSPLL, ECPLL; this is the normal, full power execution mode |
| SEC_RUN  | N/A                     | 01             | Clocked         | Clocked     | Secondary – Timer1 Oscillator                                                 |
| RC_RUN   | N/A                     | 11             | Clocked         | Clocked     | Internal Oscillator                                                           |
| PRI_IDLE | 1                       | 10             | Off             | Clocked     | Primary – HS, EC, HSPLL, ECPLL                                                |
| SEC_IDLE | 1                       | 01             | Off             | Clocked     | Secondary – Timer1 Oscillator                                                 |
| RC_IDLE  | 1                       | 11             | Off             | Clocked     | Internal Oscillator                                                           |

**Note 1:** IDLEN reflects its value when the SLEEP instruction is executed.

# PIC18F85J11 FAMILY

---

## 3.1.3 CLOCK TRANSITIONS AND STATUS INDICATORS

The length of the transition between clock sources is the sum of two cycles of the old clock source and three to four cycles of the new clock source. This formula assumes that the new clock source is stable.

Two bits indicate the current clock source and its status: OSTS (OSCCON<3>) and T1RUN (T1CON<6>). In general, only one of these bits will be set while in a given power-managed mode. When the OSTS bit is set, the primary clock is providing the device clock. When the T1RUN bit is set, the Timer1 oscillator is providing the clock. If neither of these bits is set, INTRC is clocking the device.

|              |                                                                                                                                                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> | Executing a <code>SLEEP</code> instruction does not necessarily place the device into Sleep mode. It acts as the trigger to place the controller into either the Sleep mode, or one of the Idle modes, depending on the setting of the IDLEN bit. |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 3.1.4 MULTIPLE SLEEP COMMANDS

The power-managed mode that is invoked with the `SLEEP` instruction is determined by the setting of the IDLEN bit at the time the instruction is executed. If another `SLEEP` instruction is executed, the device will enter the power-managed mode specified by IDLEN at that time. If IDLEN has changed, the device will enter the new power-managed mode specified by the new setting.

## 3.2 Run Modes

In the Run modes, clocks to both the core and peripherals are active. The difference between these modes is the clock source.

### 3.2.1 PRI\_RUN MODE

The PRI\_RUN mode is the normal, full power execution mode of the microcontroller. This is also the default mode upon a device Reset unless Two-Speed Start-up is enabled (see **Section 22.4 “Two-Speed Start-up”** for details). In this mode, the OSTS bit is set (see **Section 2.2 “Control Registers”**).

### 3.2.2 SEC\_RUN MODE

The SEC\_RUN mode is the compatible mode to the “clock switching” feature offered in other PIC18 devices. In this mode, the CPU and peripherals are clocked from the Timer1 oscillator. This gives users the option of lower power consumption while still using a high-accuracy clock source.

SEC\_RUN mode is entered by setting the SCS1:SCS0 bits to ‘01’. The device clock source is switched to the Timer1 oscillator (see Figure 3-1), the primary oscillator is shut down, the T1RUN bit (T1CON<6>) is set and the OSTS bit is cleared.



**Note:** The Timer1 oscillator should already be running prior to entering SEC\_RUN mode. If the T1OSCEN bit is not set when the SCS1:SCS0 bits are set to '01', entry to SEC\_RUN mode will not occur. If the Timer1 oscillator is enabled, but not yet running, device clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

On transitions from SEC\_RUN mode to PRI\_RUN mode, the peripherals and CPU continue to be clocked from the Timer1 oscillator while the primary clock is started. When the primary clock becomes ready, a clock switch back to the primary clock occurs (see Figure 3-2). When the clock switch is complete, the T1RUN bit is cleared, the OSTS bit is set and the primary clock is providing the clock. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run.

**FIGURE 3-1: TRANSITION TIMING FOR ENTRY TO SEC\_RUN MODE**

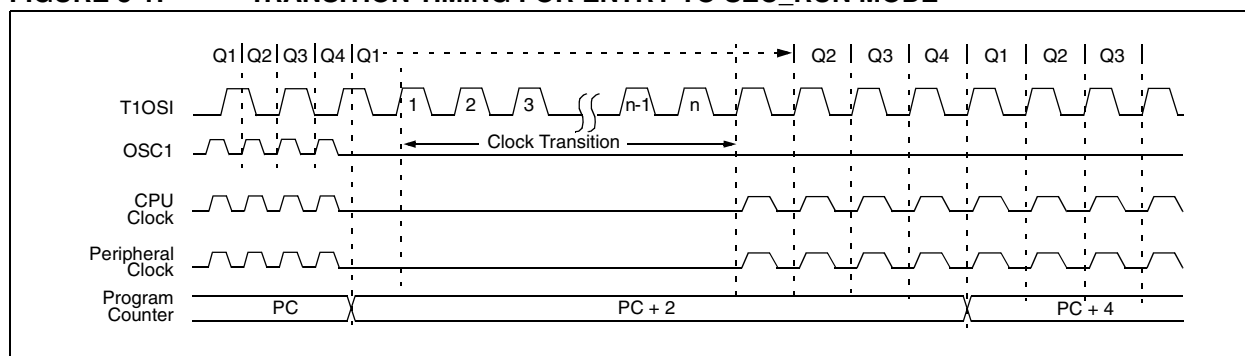
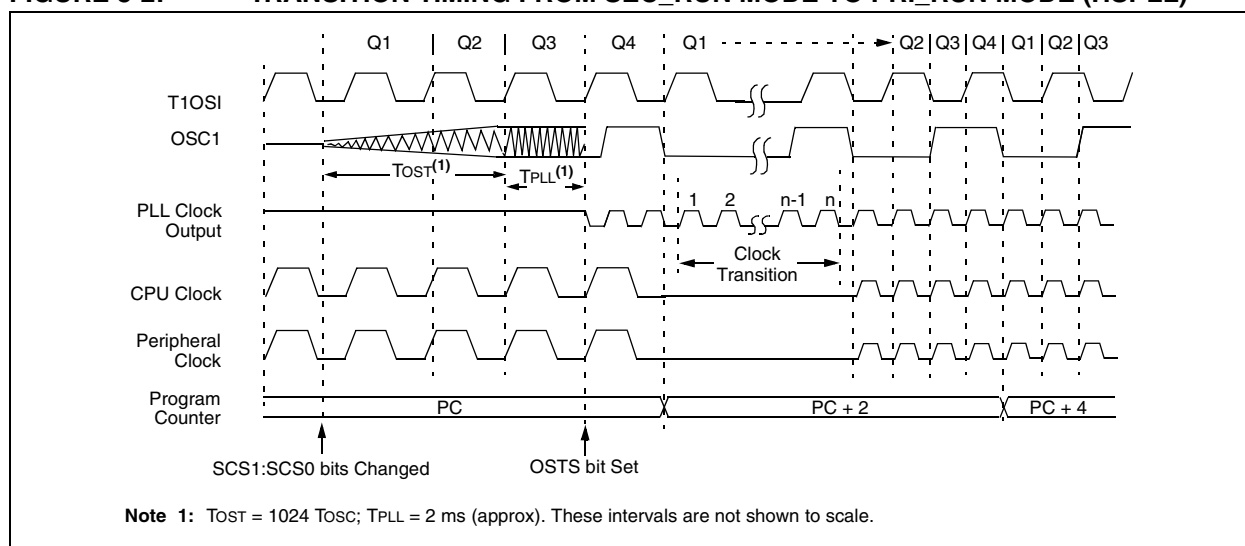


**FIGURE 3-2: TRANSITION TIMING FROM SEC\_RUN MODE TO PRI\_RUN MODE (HSPLL)**



# PIC18F85J11 FAMILY

## 3.2.3 RC\_RUN MODE

In RC\_RUN mode, the CPU and peripherals are clocked from the internal oscillator; the primary clock is shut down. This mode provides the best power conservation of all the Run modes while still executing code. It works well for user applications which are not highly timing sensitive or do not require high-speed clocks at all times.

This mode is entered by setting SCS bits to '11'. When the clock source is switched to the INTRC (see Figure 3-3), the primary oscillator is shut down and the OSTS bit is cleared.

On transitions from RC\_RUN mode to PRI\_RUN mode, the device continues to be clocked from the INTRC while the primary clock is started. When the primary clock becomes ready, a clock switch to the primary clock occurs (see Figure 3-4). When the clock switch is complete, the OSTS bit is set and the primary clock is providing the device clock. The IDLEN and SCS bits are not affected by the switch. The INTRC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

**FIGURE 3-3: TRANSITION TIMING TO RC\_RUN MODE**

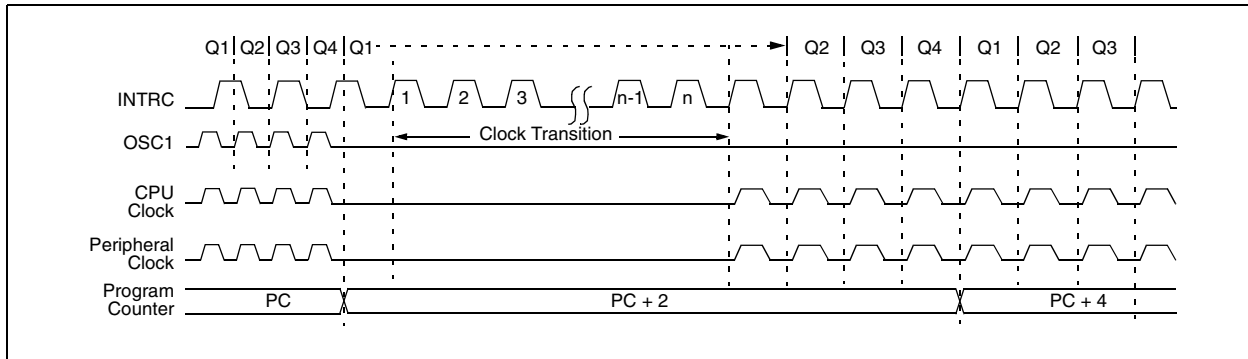
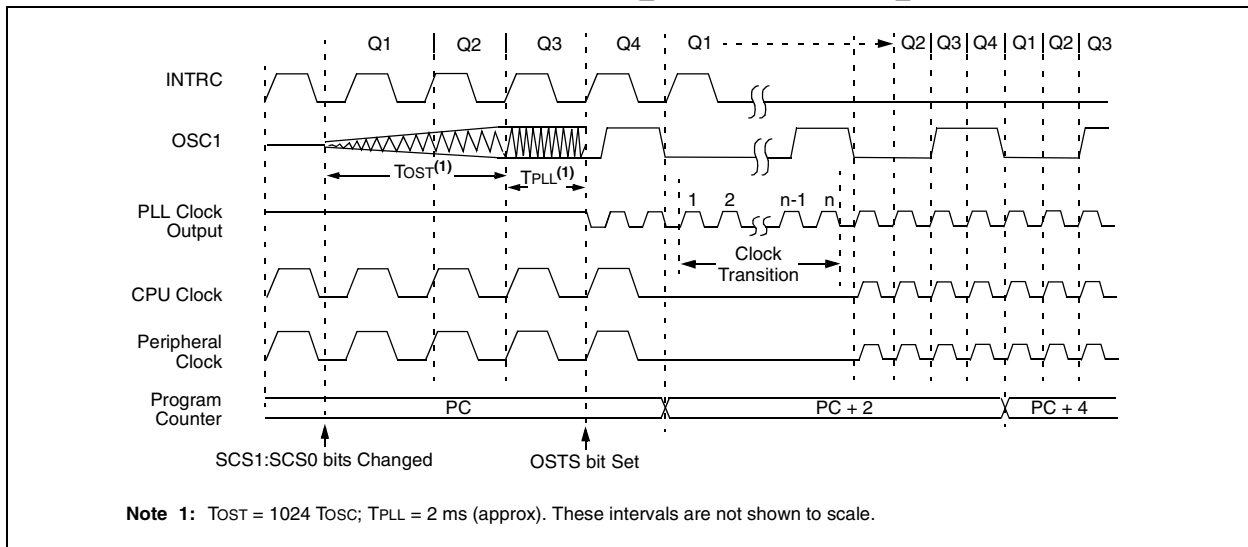


**FIGURE 3-4: TRANSITION TIMING FROM RC\_RUN MODE TO PRI\_RUN MODE**



## 3.3 Sleep Mode

The power-managed Sleep mode is identical to the legacy Sleep mode offered in all other PIC microcontrollers. It is entered by clearing the IDLEN bit (the default state on device Reset) and executing the `SLEEP` instruction. This shuts down the selected oscillator (Figure 3-5). All clock source status bits are cleared.

Entering the Sleep mode from any other mode does not require a clock switch. This is because no clocks are needed once the controller has entered Sleep. If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

When a wake event occurs in Sleep mode (by interrupt, Reset or WDT time-out), the device will not be clocked until the clock source selected by the SCS1:SCS0 bits becomes ready (see Figure 3-6), or it will be clocked from the internal oscillator if either the Two-Speed Start-up or the Fail-Safe Clock Monitor is enabled (see **Section 22.0 “Special Features of the CPU”**). In either case, the OSTS bit is set when the primary clock is providing the device clocks. The IDLEN and SCS bits are not affected by the wake-up.

## 3.4 Idle Modes

The Idle modes allow the controller's CPU to be selectively shut down while the peripherals continue to operate. Selecting a particular Idle mode allows users to further manage power consumption.

If the IDLEN bit is set to a '1' when a `SLEEP` instruction is executed, the peripherals will be clocked from the clock source selected using the SCS1:SCS0 bits; however, the CPU will not be clocked. The clock source status bits are not affected. Setting IDLEN and executing a `SLEEP` instruction provides a quick method of switching from a given Run mode to its corresponding Idle mode.

If the WDT is selected, the INTRC source will continue to operate. If the Timer1 oscillator is enabled, it will also continue to run.

Since the CPU is not executing instructions, the only exits from any of the Idle modes are by interrupt, WDT time-out or a Reset. When a wake event occurs, CPU execution is delayed by an interval of  $T_{CSD}$  (parameter 38, Table 25-10) while it becomes ready to execute code. When the CPU begins executing code, it resumes with the same clock source for the current Idle mode. For example, when waking from RC\_IDLE mode, the internal oscillator block will clock the CPU and peripherals (in other words, RC\_RUN mode). The IDLEN and SCS bits are not affected by the wake-up.

While in any Idle mode or the Sleep mode, a WDT time-out will result in a WDT wake-up to the Run mode currently specified by the SCS1:SCS0 bits.

**FIGURE 3-5: TRANSITION TIMING FOR ENTRY TO SLEEP MODE**

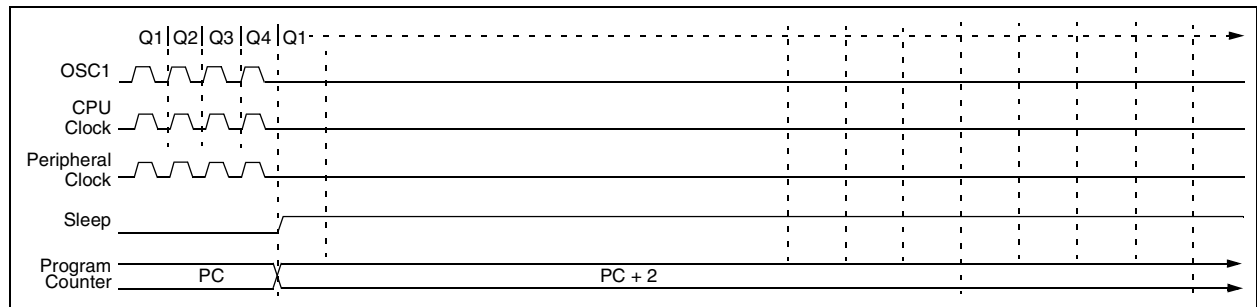
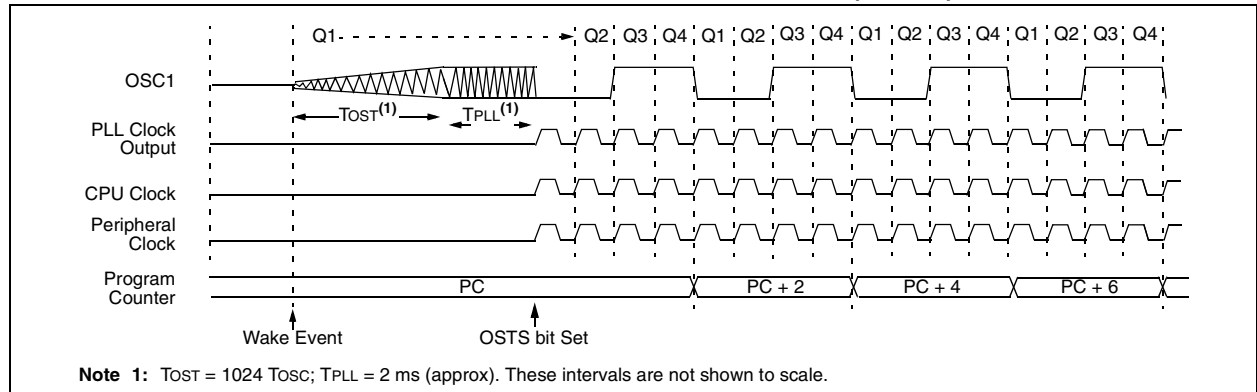


**FIGURE 3-6: TRANSITION TIMING FOR WAKE FROM SLEEP (HSPLL)**



# PIC18F85J11 FAMILY

## 3.4.1 PRI\_IDLE MODE

This mode is unique among the three low-power Idle modes, in that it does not disable the primary device clock. For timing sensitive applications, this allows for the fastest resumption of device operation with its more accurate primary clock source, since the clock source does not have to “warm up” or transition from another oscillator.

PRI\_IDLE mode is entered from PRI\_RUN mode by setting the IDLEN bit and executing a *SLEEP* instruction. If the device is in another Run mode, set IDLEN first, then set the SCS bits to ‘10’ and execute *SLEEP*. Although the CPU is disabled, the peripherals continue to be clocked from the primary clock source specified by the FOSC1:FOSC0 Configuration bits. The OSTS bit remains set (see Figure 3-7).

When a wake event occurs, the CPU is clocked from the primary clock source. A delay of interval TCSD is required between the wake event and when code execution starts. This is required to allow the CPU to become ready to execute instructions. After the wake-up, the OSTS bit remains set. The IDLEN and SCS bits are not affected by the wake-up (see Figure 3-8).

## 3.4.2 SEC\_IDLE MODE

In SEC\_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the Timer1 oscillator. This mode is entered from SEC\_RUN by setting the IDLEN bit and executing a *SLEEP* instruction. If the device is in another Run mode, set IDLEN first, then set SCS1:SCS0 to ‘01’ and execute *SLEEP*. When the clock source is switched to the Timer1 oscillator, the primary oscillator is shut down, the OSTS bit is cleared and the T1RUN bit is set.

When a wake event occurs, the peripherals continue to be clocked from the Timer1 oscillator. After an interval of TCSD following the wake event, the CPU begins executing code being clocked by the Timer1 oscillator. The IDLEN and SCS bits are not affected by the wake-up; the Timer1 oscillator continues to run (see Figure 3-8).

**Note:** The Timer1 oscillator should already be running prior to entering SEC\_IDLE mode. If the T1OSCEN bit is not set when the *SLEEP* instruction is executed, the *SLEEP* instruction will be ignored and entry to SEC\_IDLE mode will not occur. If the Timer1 oscillator is enabled, but not yet running, peripheral clocks will be delayed until the oscillator has started. In such situations, initial oscillator operation is far from stable and unpredictable operation may result.

**FIGURE 3-7: TRANSITION TIMING FOR ENTRY TO IDLE MODE**

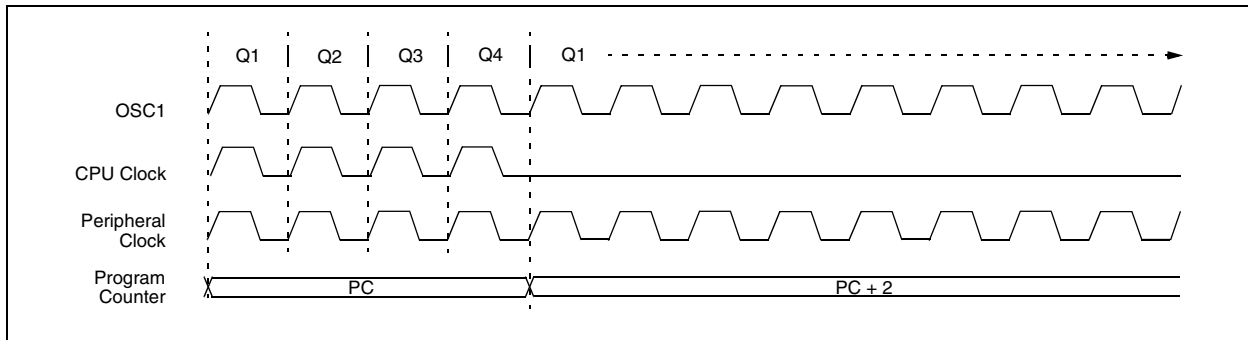
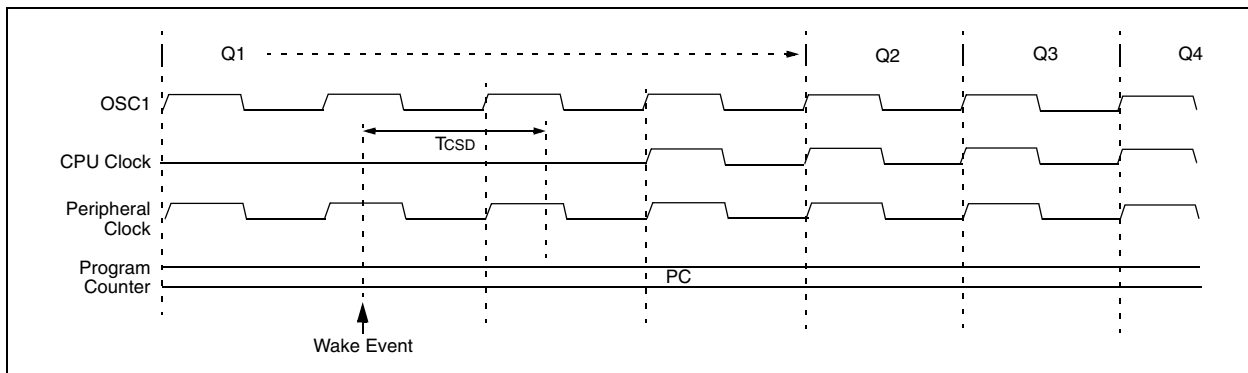


**FIGURE 3-8: TRANSITION TIMING FOR WAKE FROM IDLE TO RUN MODE**



### 3.4.3 RC\_IDLE MODE

In RC\_IDLE mode, the CPU is disabled but the peripherals continue to be clocked from the internal oscillator. This mode allows for controllable power conservation during Idle periods.

From RC\_RUN, this mode is entered by setting the IDLEN bit and executing a SLEEP instruction. If the device is in another Run mode, first set IDLEN, then clear the SCS bits and execute SLEEP. When the clock source is switched to the INTRC, the primary oscillator is shut down and the OSTS bit is cleared.

When a wake event occurs, the peripherals continue to be clocked from the INTOSC. After a delay of T<sub>CSD</sub> following the wake event, the CPU begins executing code being clocked by the INTOSC. The IDLEN and SCS bits are not affected by the wake-up. The INTOSC source will continue to run if either the WDT or the Fail-Safe Clock Monitor is enabled.

## 3.5 Exiting Idle and Sleep Modes

An exit from Sleep mode, or any of the Idle modes, is triggered by an interrupt, a Reset or a WDT time-out. This section discusses the triggers that cause exits from power-managed modes. The clocking subsystem actions are discussed in each of the power-managed mode sections (see **Section 3.2 “Run Modes”**, **Section 3.3 “Sleep Mode”** and **Section 3.4 “Idle Modes”**).

### 3.5.1 EXIT BY INTERRUPT

Any of the available interrupt sources can cause the device to exit from an Idle mode, or the Sleep mode, to a Run mode. To enable this functionality, an interrupt source must be enabled by setting its enable bit in one of the INTCON or PIE registers. The exit sequence is initiated when the corresponding interrupt flag bit is set.

On all exits from Idle or Sleep modes by interrupt, code execution branches to the interrupt vector if the GIE/GIEH bit (INTCON<7>) is set. Otherwise, code execution continues or resumes without branching (see **Section 9.0 “Interrupts”**).

A fixed delay of interval T<sub>CSD</sub> following the wake event is required when leaving Sleep and Idle modes. This delay is required for the CPU to prepare for execution. Instruction execution resumes on the first clock cycle following this delay.

### 3.5.2 EXIT BY WDT TIME-OUT

A WDT time-out will cause different actions depending on which power-managed mode the device is in when the time-out occurs.

If the device is not executing code (all Idle modes and Sleep mode), the time-out will result in an exit from the power-managed mode (see **Section 3.2 “Run Modes”** and **Section 3.3 “Sleep Mode”**). If the device is executing code (all Run modes), the time-out will result in a WDT Reset (see **Section 22.2 “Watchdog Timer (WDT)”**).

The Watchdog Timer and postscaler are cleared by one of the following events:

- executing a SLEEP or CLRWD instruction
- the loss of a currently selected clock source (if the Fail-Safe Clock Monitor is enabled)

### 3.5.3 EXIT BY RESET

Exiting an Idle or Sleep mode by Reset automatically forces the device to run from the INTRC.

### 3.5.4 EXIT WITHOUT AN OSCILLATOR START-UP DELAY

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

- PRI\_IDLE mode, where the primary clock source is not stopped; and
- the primary clock source is either the EC or ECPLL mode.

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

# PIC18F85J11 FAMILY

---

NOTES:

## 4.0 RESET

The PIC18F85J11 family of devices differentiate between various kinds of Reset:

- Power-on Reset (POR)
- $\overline{\text{MCLR}}$  Reset during normal operation
- $\overline{\text{MCLR}}$  Reset during power-managed modes
- Watchdog Timer (WDT) Reset (during execution)
- Brown-out Reset (BOR)
- RESET Instruction
- Stack Full Reset
- Stack Underflow Reset

This section discusses Resets generated by  $\overline{\text{MCLR}}$ , POR and BOR, and covers the operation of the various start-up timers. Stack Reset events are covered in **Section 5.1.6.4 “Stack Full and Underflow Resets”**. WDT Resets are covered in **Section 22.2 “Watchdog Timer (WDT)”**.

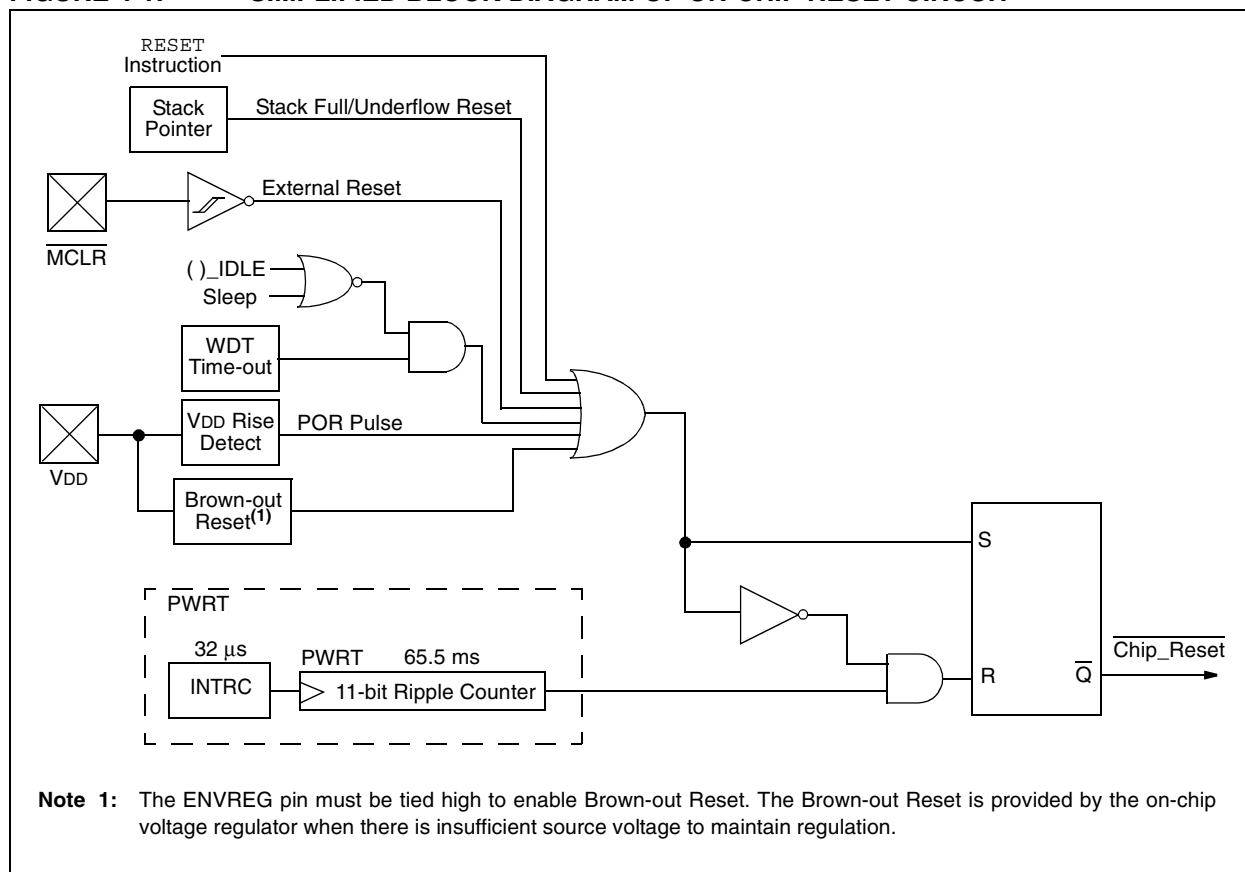
A simplified block diagram of the on-chip Reset circuit is shown in Figure 4-1.

## 4.1 RCON Register

Device Reset events are tracked through the RCON register (Register 4-1). The lower five bits of the register indicate that a specific Reset event has occurred. In most cases, these bits can only be set by the event and must be cleared by the application after the event. The state of these flag bits, taken together, can be read to indicate the type of Reset that just occurred. This is described in more detail in **Section 4.7 “Reset State of Registers”**.

The RCON register also has a control bit for setting interrupt priority (IPEN). Interrupt priority is discussed in **Section 9.0 “Interrupts”**.

**FIGURE 4-1: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT**



# PIC18F85J11 FAMILY

## REGISTER 4-1: RCON: RESET CONTROL REGISTER

| R/W-0 | U-0 | R/W-1 | R/W-1 | R-1 | R-1 | R/W-0 | R/W-0 |
|-------|-----|-------|-------|-----|-----|-------|-------|
| IPEN  | —   | CM    | RI    | TO  | PD  | POR   | BOR   |
| 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 **IPEN:** Interrupt Priority Enable bit  
1 = Enable priority levels on interrupts  
0 = Disable priority levels on interrupts (PIC16XXX Compatibility mode)
- bit 6 **Unimplemented:** Read as '0'
- bit 5 **CM:** Configuration Mismatch Flag bit  
1 = A configuration mismatch Reset has not occurred  
0 = A configuration mismatch Reset occurred. Must be set in software once the reset occurs.
- bit 4 **RI:** RESET Instruction Flag bit  
1 = The RESET instruction was not executed (set by firmware only)  
0 = The RESET instruction was executed causing a device Reset (must be set in software after a Brown-out Reset occurs)
- bit 3 **TO:** Watchdog Time-out Flag bit  
1 = Set by power-up, CLRWD instruction or SLEEP instruction  
0 = A WDT time-out occurred
- bit 2 **PD:** Power-Down Detection Flag bit  
1 = Set by power-up or by the CLRWD instruction  
0 = Set by execution of the SLEEP instruction
- bit 1 **POR:** Power-on Reset Status bit  
1 = A Power-on Reset has not occurred (set by firmware only)  
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)
- bit 0 **BOR:** Brown-out Reset Status bit  
1 = A Brown-out Reset has not occurred (set by firmware only)  
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

**Note 1:** It is recommended that the  $\overline{\text{POR}}$  bit be set after a Power-on Reset has been detected so that subsequent Power-on Resets may be detected.

**2:** If the on-chip voltage regulator is disabled,  $\overline{\text{BOR}}$  remains '0' at all times. See **Section 4.4.1 "Detecting BOR"** for more information.

**3:** Brown-out Reset is said to have occurred when  $\overline{\text{BOR}}$  is '0' and  $\overline{\text{POR}}$  is '1' (assuming that  $\overline{\text{POR}}$  was set to '1' by software immediately after a Power-on Reset).



## 4.2 Master Clear (MCLR)

The  $\overline{\text{MCLR}}$  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  $\overline{\text{MCLR}}$  Reset path which detects and ignores small pulses.

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

## 4.3 Power-on Reset (POR)

A Power-on Reset condition is generated on-chip whenever  $V_{DD}$  rises above a certain threshold. This allows the device to start in the initialized state when  $V_{DD}$  is adequate for operation.

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

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

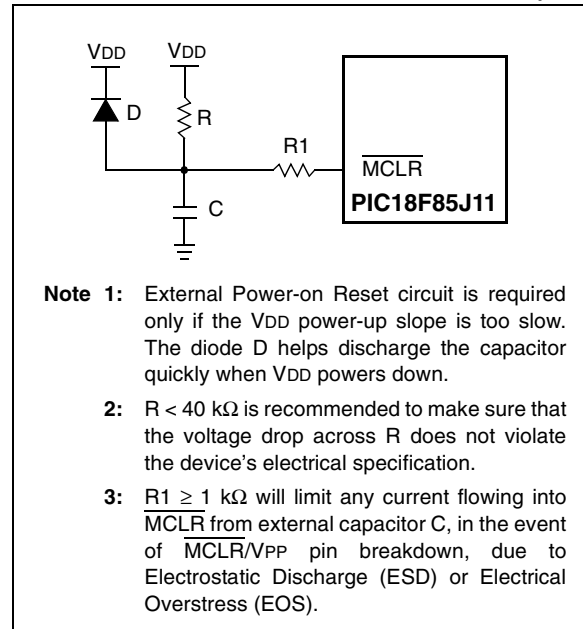
Power-on Reset events are captured by the  $\overline{\text{POR}}$  bit (RCON<1>). The state of the bit is set to '0' whenever a Power-on Reset occurs; it does not change for any other Reset event.  $\overline{\text{POR}}$  is not reset to '1' by any hardware event. To capture multiple events, the user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

## 4.4 Brown-out Reset (BOR)

The PIC18F85J11 family of devices incorporates a simple BOR function when the internal regulator is enabled (ENVREG pin is tied to  $V_{DD}$ ). The voltage regulator will trigger a Brown-out Reset when output of the regulator to the device core approaches the voltage at which the device is unable to run at full speed. The BOR circuit also keeps the device in Reset as  $V_{DD}$  rises, until the regulator's output level is sufficient for full-speed operation.

Once a BOR has occurred, the Power-up Timer will keep the chip in Reset for  $TPWRT$  (parameter 33). If  $V_{DD}$  drops below the threshold for full-speed operation while the Power-up Timer is running, the chip will go back into a Brown-out Reset and the Power-up Timer will be initialized. Once  $V_{DD}$  rises to the point where regulator output is sufficient, the Power-up Timer will execute the additional time delay.

**FIGURE 4-2: EXTERNAL POWER-ON RESET CIRCUIT (FOR SLOW  $V_{DD}$  POWER-UP)**



### 4.4.1 DETECTING BOR

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

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

# PIC18F85J11 FAMILY

## 4.5 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 PIC18FXXJXX 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 CM bit (RCON<5>) being set to '0'. This bit does not change for any other Reset event.

A CM Reset behaves similarly to a Master Clear Reset, 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.

## 4.6 Power-up Timer (PWRT)

PIC18F85J11 family devices incorporate an on-chip Power-up Timer (PWRT) to help regulate the Power-on Reset 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 PIC18F85J11 family devices is an 11-bit counter which uses the INTRC source as the clock input. This yields an approximate time interval of  $2048 \times 32 \mu\text{s} = 65.6 \text{ ms}$ . While the PWRT is counting, the device is held in Reset.

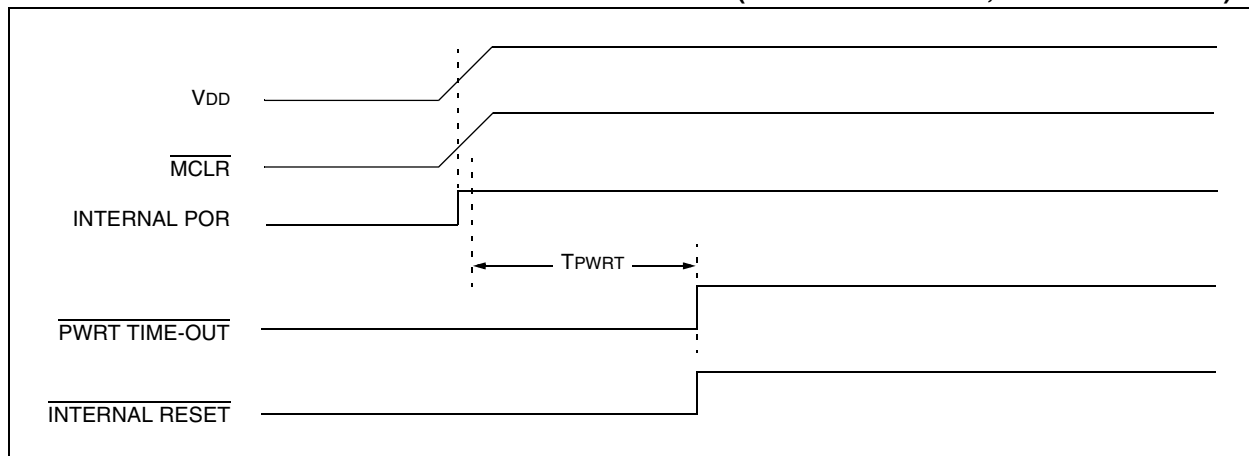
The power-up time delay depends on the INTRC clock and will vary from chip-to-chip due to temperature and process variation. See DC parameter 33 for details.

### 4.6.1 TIME-OUT SEQUENCE

If enabled, 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 4-3, Figure 4-4, Figure 4-5 and Figure 4-6 all depict time-out sequences on power-up with the Power-up Timer enabled.

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

**FIGURE 4-3: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ ,  $V_{DD} \text{ RISE} < T_{PWRT}$ )**



# PIC18F85J11 FAMILY

**FIGURE 4-4: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 1**

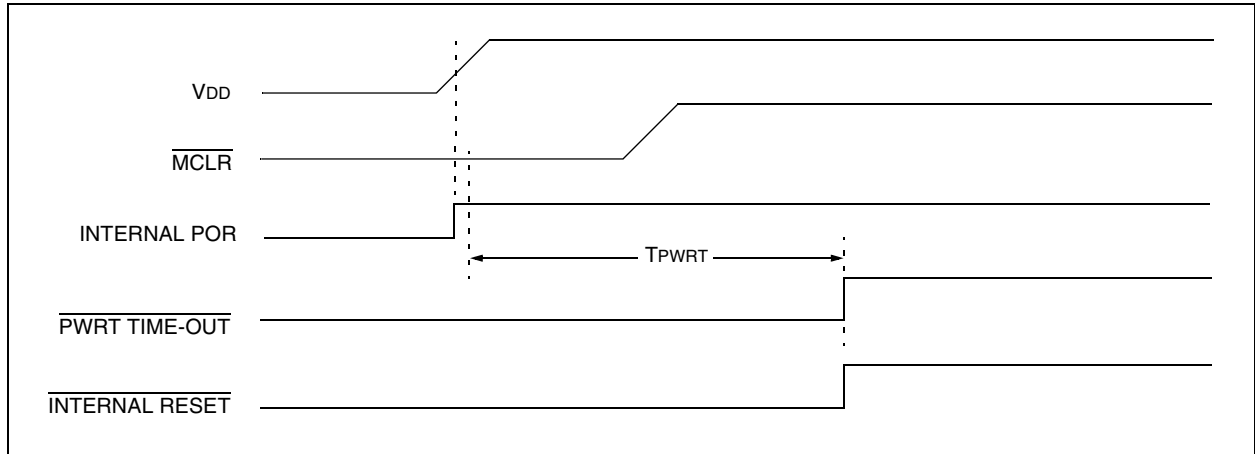


**FIGURE 4-5: TIME-OUT SEQUENCE ON POWER-UP ( $\overline{\text{MCLR}}$  NOT TIED TO  $V_{DD}$ ): CASE 2**

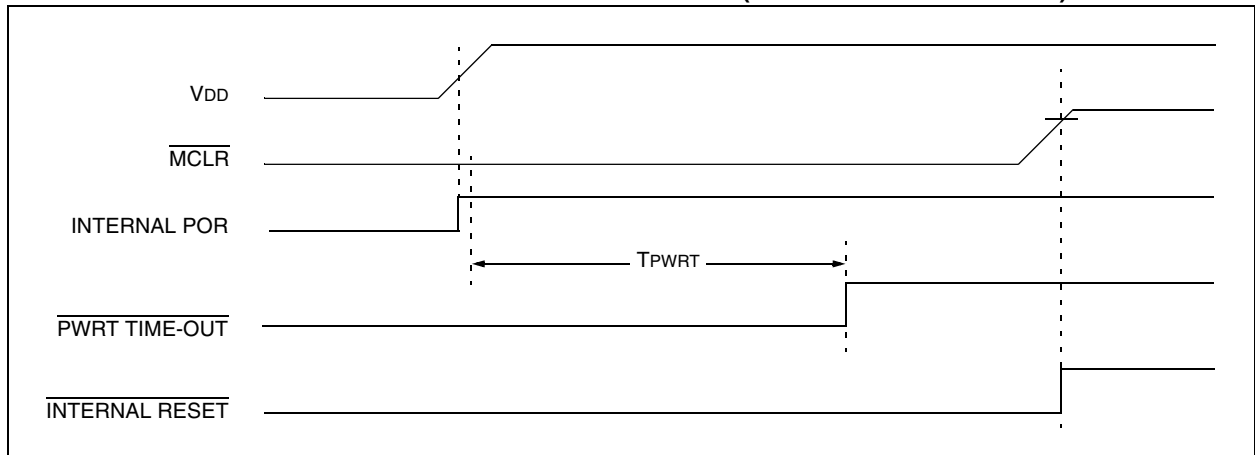
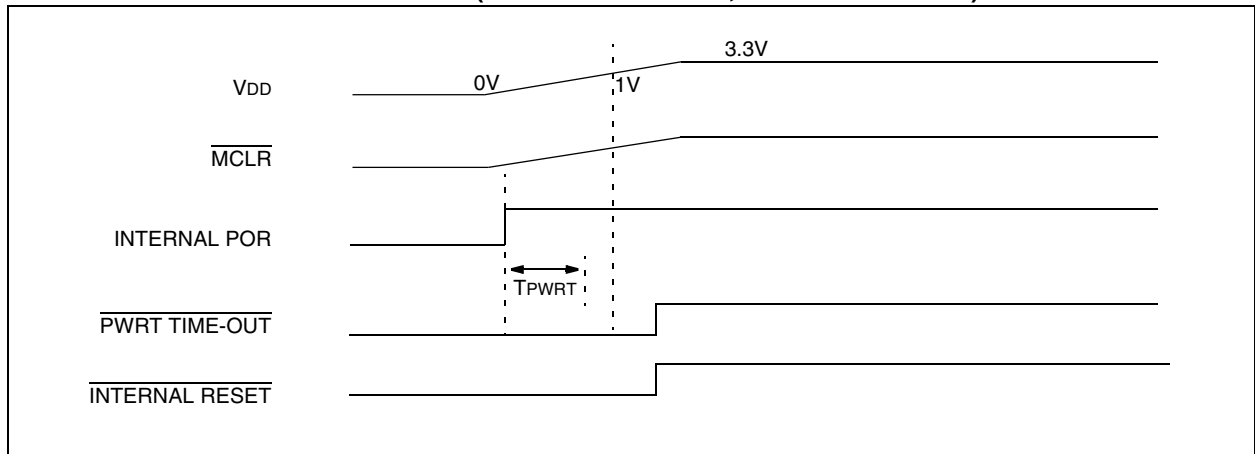


**FIGURE 4-6: SLOW RISE TIME ( $\overline{\text{MCLR}}$  TIED TO  $V_{DD}$ ,  $V_{DD}$  RISE  $>$   $T_{PWRT}$ )**



# PIC18F85J11 FAMILY

## 4.7 Reset State of Registers

Most registers are unaffected by a Reset. Their status is unknown on POR and unchanged by all other Resets. The other registers are forced to a “Reset state” depending on the type of Reset that occurred.

Most registers are not affected by a WDT wake-up, since this is viewed as the resumption of normal operation. Status bits from the RCON register,  $\overline{\text{CM}}$ ,  $\overline{\text{RI}}$ ,  $\overline{\text{TO}}$ ,  $\overline{\text{PD}}$ ,  $\overline{\text{POR}}$  and  $\overline{\text{BOR}}$ , are set or cleared differently in different Reset situations, as indicated in Table 4-1. These bits are used in software to determine the nature of the Reset.

Table 4-2 describes the Reset states for all of the Special Function Registers. These are categorized by Power-on and Brown-out Resets, Master Clear and WDT Resets and WDT wake-ups.

**TABLE 4-1: STATUS BITS, THEIR SIGNIFICANCE AND THE INITIALIZATION CONDITION FOR RCON REGISTER**

| Condition                                                 | Program Counter <sup>(1)</sup> | RCON Register          |                        |                        |                        |                         |                         | STKPTR Register |        |
|-----------------------------------------------------------|--------------------------------|------------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|-----------------|--------|
|                                                           |                                | $\overline{\text{CM}}$ | $\overline{\text{RI}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ | STKFUL          | STKUNF |
| Power-on Reset                                            | 0000h                          | 1                      | 1                      | 1                      | 1                      | 0                       | 0                       | 0               | 0      |
| RESET Instruction                                         | 0000h                          | u                      | 0                      | u                      | u                      | u                       | u                       | u               | u      |
| Brown-out Reset                                           | 0000h                          | 1                      | 1                      | 1                      | 1                      | u                       | 0                       | u               | u      |
| MCLR during power-managed Run modes                       | 0000h                          | u                      | u                      | 1                      | u                      | u                       | u                       | u               | u      |
| MCLR during power-managed Idle modes and Sleep mode       | 0000h                          | u                      | u                      | 1                      | 0                      | u                       | u                       | u               | u      |
| WDT time-out during full power or power-managed Run modes | 0000h                          | u                      | u                      | 0                      | u                      | u                       | u                       | u               | u      |
| MCLR during full power execution                          | 0000h                          | u                      | u                      | u                      | u                      | u                       | u                       | u               | u      |
| Stack Full Reset (STVREN = 1)                             | 0000h                          | u                      | u                      | u                      | u                      | u                       | u                       | 1               | u      |
| Stack Underflow Reset (STVREN = 1)                        | 0000h                          | u                      | u                      | u                      | u                      | u                       | u                       | u               | 1      |
| Stack Underflow Error (not an actual Reset, STVREN = 0)   | 0000h                          | u                      | u                      | u                      | u                      | u                       | u                       | u               | 1      |
| WDT time-out during power-managed Idle or Sleep modes     | PC + 2                         | u                      | u                      | 0                      | 0                      | u                       | u                       | u               | u      |
| Interrupt exit from power-managed modes                   | PC + 2                         | u                      | u                      | u                      | 0                      | u                       | u                       | u               | u      |

**Legend:** u = unchanged

**Note 1:** When the wake-up is due to an interrupt and the GIEH or GIEL bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

# PIC18F85J11 FAMILY

**TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS**

| Register | Applicable Devices |             | Power-on Reset,<br>Brown-out Reset | MCLR Resets<br>WDT Reset<br>RESET Instruction<br>Stack Resets<br>CM Resets | Wake-up via WDT<br>or Interrupt |
|----------|--------------------|-------------|------------------------------------|----------------------------------------------------------------------------|---------------------------------|
| TOSU     | PIC18F6XJ11        | PIC18F8XJ11 | ---0 0000                          | ---0 0000                                                                  | ---0 uuuu <sup>(1)</sup>        |
| TOSH     | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu <sup>(1)</sup>        |
| TOSL     | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu <sup>(1)</sup>        |
| STKPTR   | PIC18F6XJ11        | PIC18F8XJ11 | uu-0 0000                          | 00-0 0000                                                                  | uu-u uuuu <sup>(1)</sup>        |
| PCLATU   | PIC18F6XJ11        | PIC18F8XJ11 | ---0 0000                          | ---0 0000                                                                  | ---u uuuu                       |
| PCLATH   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| PCL      | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | PC + 2 <sup>(2)</sup>           |
| TBLPTRU  | PIC18F6XJ11        | PIC18F8XJ11 | --00 0000                          | --00 0000                                                                  | --uu uuuu                       |
| TBLPTRH  | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TBLPTRL  | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TABLAT   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| PRODH    | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PRODL    | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| INTCON   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 000x                          | 0000 000u                                                                  | uuuu uuuu <sup>(3)</sup>        |
| INTCON2  | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu <sup>(3)</sup>        |
| INTCON3  | PIC18F6XJ11        | PIC18F8XJ11 | 1100 0000                          | 1100 0000                                                                  | uuuu uuuu <sup>(3)</sup>        |
| INDF0    | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTINC0 | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTDEC0 | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PREINC0  | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PLUSW0   | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| FSR0H    | PIC18F6XJ11        | PIC18F8XJ11 | ---- xxxx                          | ---- uuuu                                                                  | ---- uuuu                       |
| FSR0L    | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| WREG     | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| INDF1    | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTINC1 | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTDEC1 | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PREINC1  | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PLUSW1   | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |

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

Shaded cells indicate conditions do not apply for the designated device.

**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 4-1 for Reset value for specific condition.

**5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

# PIC18F85J11 FAMILY

**TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register            | Applicable Devices |             | Power-on Reset,<br>Brown-out Reset | MCLR Resets<br>WDT Reset<br>RESET Instruction<br>Stack Resets<br>CM Resets | Wake-up via WDT<br>or Interrupt |
|---------------------|--------------------|-------------|------------------------------------|----------------------------------------------------------------------------|---------------------------------|
| FSR1H               | PIC18F6XJ11        | PIC18F8XJ11 | ---- xxxx                          | ---- uuuu                                                                  | ---- uuuu                       |
| FSR1L               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| BSR                 | PIC18F6XJ11        | PIC18F8XJ11 | ---- 0000                          | ---- 0000                                                                  | ---- uuuu                       |
| INDF2               | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTINC2            | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| POSTDEC2            | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PREINC2             | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| PLUSW2              | PIC18F6XJ11        | PIC18F8XJ11 | N/A                                | N/A                                                                        | N/A                             |
| FSR2H               | PIC18F6XJ11        | PIC18F8XJ11 | ---- xxxx                          | ---- uuuu                                                                  | ---- uuuu                       |
| FSR2L               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| STATUS              | PIC18F6XJ11        | PIC18F8XJ11 | ---x xxxx                          | ---u uuuu                                                                  | ---u uuuu                       |
| TMR0H               | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TMR0L               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| T0CON               | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| OSCCON              | PIC18F6XJ11        | PIC18F8XJ11 | 0100 q000                          | 0100 q000                                                                  | uuuu quuu                       |
| WDTCON              | PIC18F6XJ11        | PIC18F8XJ11 | 0--- ---0                          | 0--- ---0                                                                  | u--- ---u                       |
| RCON <sup>(4)</sup> | PIC18F6XJ11        | PIC18F8XJ11 | 0-11 11q0                          | 0-uq qqqu                                                                  | u-uu qqqu                       |
| TMR1H               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| TMR1L               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| T1CON               | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | u0uu uuuu                                                                  | uuuu uuuu                       |
| TMR2                | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| PR2                 | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | 1111 1111                       |
| T2CON               | PIC18F6XJ11        | PIC18F8XJ11 | -000 0000                          | -000 0000                                                                  | -uuu uuuu                       |
| SSPBUF              | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| SSPADD              | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| SSPSTAT             | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| SSPCON1             | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| SSPCON2             | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |

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

Shaded cells indicate conditions do not apply for the designated device.

**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 4-1 for Reset value for specific condition.

**5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

# PIC18F85J11 FAMILY

**TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register | Applicable Devices |             | Power-on Reset,<br>Brown-out Reset | MCLR Resets<br>WDT Reset<br>RESET Instruction<br>Stack Resets<br>CM Resets | Wake-up via WDT<br>or Interrupt |
|----------|--------------------|-------------|------------------------------------|----------------------------------------------------------------------------|---------------------------------|
| ADRESH   | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| ADRESL   | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| ADCON0   | PIC18F6XJ11        | PIC18F8XJ11 | 0-00 0000                          | 0-00 0000                                                                  | u-uu uuuu                       |
| ADCON1   | PIC18F6XJ11        | PIC18F8XJ11 | --00 0000                          | --00 0000                                                                  | --uu uuuu                       |
| ADCON2   | PIC18F6XJ11        | PIC18F8XJ11 | 0-00 0000                          | 0-00 0000                                                                  | u-uu uuuu                       |
| CVRCON   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| CMCON    | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0111                          | 0000 0111                                                                  | uuuu uuuu                       |
| TMR3H    | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| TMR3L    | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| T3CON    | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PSPCON   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 ----                          | 0000 ----                                                                  | uuuu ----                       |
| SPBRG1   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| RCREG1   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TXREG1   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TXSTA1   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0010                          | 0000 0010                                                                  | uuuu uuuu                       |
| RCSTA1   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 000x                          | 0000 000x                                                                  | uuuu uuuu                       |
| EECON2   | PIC18F6XJ11        | PIC18F8XJ11 | ---- ----                          | ---- ----                                                                  | ---- ----                       |
| EECON1   | PIC18F6XJ11        | PIC18F8XJ11 | ---0 x00-                          | ---0 u00-                                                                  | ---0 u00-                       |
| IPR3     | PIC18F6XJ11        | PIC18F8XJ11 | --00 -11-                          | --00 -11-                                                                  | --uu -uu-                       |
| PIR3     | PIC18F6XJ11        | PIC18F8XJ11 | --00 -00-                          | --00 -00-                                                                  | --uu -00- <sup>(3)</sup>        |
| PIE3     | PIC18F6XJ11        | PIC18F8XJ11 | --00 -00-                          | --00 -00-                                                                  | --uu -00-                       |
| IPR2     | PIC18F6XJ11        | PIC18F8XJ11 | 11-- 111-                          | 11-- 111-                                                                  | uu-- uuu-                       |
| PIR2     | PIC18F6XJ11        | PIC18F8XJ11 | 00-- 000-                          | 00-- 000-                                                                  | uu-- uuu- <sup>(3)</sup>        |
| PIE2     | PIC18F6XJ11        | PIC18F8XJ11 | 00-- 000-                          | 00-- 000-                                                                  | uu-- uuu-                       |
| IPR1     | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1-11                          | 1111 1-11                                                                  | uuuu u-uu                       |
| PIR1     | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0-00                          | 0000 0-00                                                                  | uuuu u-uu <sup>(3)</sup>        |
| PIE1     | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0-00                          | 0000 0-00                                                                  | uuuu u-uu                       |
| MEMCON   | PIC18F6XJ11        | PIC18F8XJ11 | 0-00 --00                          | 0-00 --00                                                                  | u-uu --uu                       |
| OSCTUNE  | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.  
Shaded cells indicate conditions do not apply for the designated device.

- 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 4-1 for Reset value for specific condition.
- 5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

# PIC18F85J11 FAMILY

**TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register             | Applicable Devices |             | Power-on Reset,<br>Brown-out Reset | MCLR Resets<br>WDT Reset<br>RESET Instruction<br>Stack Resets<br>CM Resets | Wake-up via WDT<br>or Interrupt |
|----------------------|--------------------|-------------|------------------------------------|----------------------------------------------------------------------------|---------------------------------|
| TRISJ                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| TRISH                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| TRISG                | PIC18F6XJ11        | PIC18F8XJ11 | 0001 1111                          | 0001 1111                                                                  | uuuu uuuu                       |
| TRISF                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 111-                          | 1111 111-                                                                  | uuuu uu-                        |
| TRISE                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1-11                          | 1111 1-11                                                                  | uuuu u-uu                       |
| TRISD                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| TRISC                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| TRISB                | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111                          | 1111 1111                                                                  | uuuu uuuu                       |
| TRISA <sup>(5)</sup> | PIC18F6XJ11        | PIC18F8XJ11 | 1111 1111 <sup>(5)</sup>           | 1111 1111 <sup>(5)</sup>                                                   | uuuu uuuu <sup>(5)</sup>        |
| LATJ                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATH                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATG                 | PIC18F6XJ11        | PIC18F8XJ11 | 00-x xxxx                          | 00-u uuuu                                                                  | uu-u uuuu                       |
| LATF                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxx-                          | uuuu uu-                                                                   | uuuu uu-                        |
| LATE                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATD                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATC                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATB                 | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| LATA <sup>(5)</sup>  | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx <sup>(5)</sup>           | uuuu uuuu <sup>(5)</sup>                                                   | uuuu uuuu <sup>(5)</sup>        |
| PORTJ                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PORTH                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PORTG                | PIC18F6XJ11        | PIC18F8XJ11 | 000x xxxx                          | 000u uuuu                                                                  | 000u uuuu                       |
| PORTF                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxx-                          | uuuu uu-                                                                   | uuuu uu-                        |
| PORTE                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx x-xx                          | uuuu u-uu                                                                  | uuuu u-uu                       |
| PORTD                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PORTC                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PORTB                | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| PORTA <sup>(5)</sup> | PIC18F6XJ11        | PIC18F8XJ11 | xx0x 0000 <sup>(5)</sup>           | uu0u 0000 <sup>(5)</sup>                                                   | uuuu uuuu <sup>(5)</sup>        |
| SPBRGH1              | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| BAUDCON1             | PIC18F6XJ11        | PIC18F8XJ11 | 01-0 0-00                          | 01-0 0-00                                                                  | uu-u u-uu                       |
| CCPR1H               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| CCPR1L               | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| CCP1CON              | PIC18F6XJ11        | PIC18F8XJ11 | --00 0000                          | --00 0000                                                                  | --uu uuuu                       |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.  
Shaded cells indicate conditions do not apply for the designated device.

- 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 4-1 for Reset value for specific condition.
- 5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.



# PIC18F85J11 FAMILY

**TABLE 4-2: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)**

| Register | Applicable Devices |             | Power-on Reset,<br>Brown-out Reset | MCLR Resets<br>WDT Reset<br>RESET Instruction<br>Stack Resets<br>CM Resets | Wake-up via WDT<br>or Interrupt |
|----------|--------------------|-------------|------------------------------------|----------------------------------------------------------------------------|---------------------------------|
| CCPR2H   | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| CCPR2L   | PIC18F6XJ11        | PIC18F8XJ11 | xxxx xxxx                          | uuuu uuuu                                                                  | uuuu uuuu                       |
| CCP2CON  | PIC18F6XJ11        | PIC18F8XJ11 | --00 0000                          | --00 0000                                                                  | --uu uuuu                       |
| SPBRG2   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| RCREG2   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TXREG2   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 0000                          | 0000 0000                                                                  | uuuu uuuu                       |
| TXSTA2   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 -010                          | 0000 -010                                                                  | uuuu -uuu                       |
| RCSTA2   | PIC18F6XJ11        | PIC18F8XJ11 | 0000 000x                          | 0000 000x                                                                  | uuuu uuuu                       |

**Legend:** u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition.  
Shaded cells indicate conditions do not apply for the designated device.

- 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 4-1 for Reset value for specific condition.
- 5:** Bits 6 and 7 of PORTA, LATA and TRISA are enabled depending on the oscillator mode selected. When not enabled as PORTA pins, they are disabled and read as '0'.

# PIC18F85J11 FAMILY

---

NOTES:

# PIC18F85J11 FAMILY

## 5.0 MEMORY ORGANIZATION

There are two types of memory in PIC18 Flash microcontroller devices:

- Program Memory
- Data RAM

As Harvard architecture devices, the data and program memories use separate busses; this allows for concurrent access of the two memory spaces.

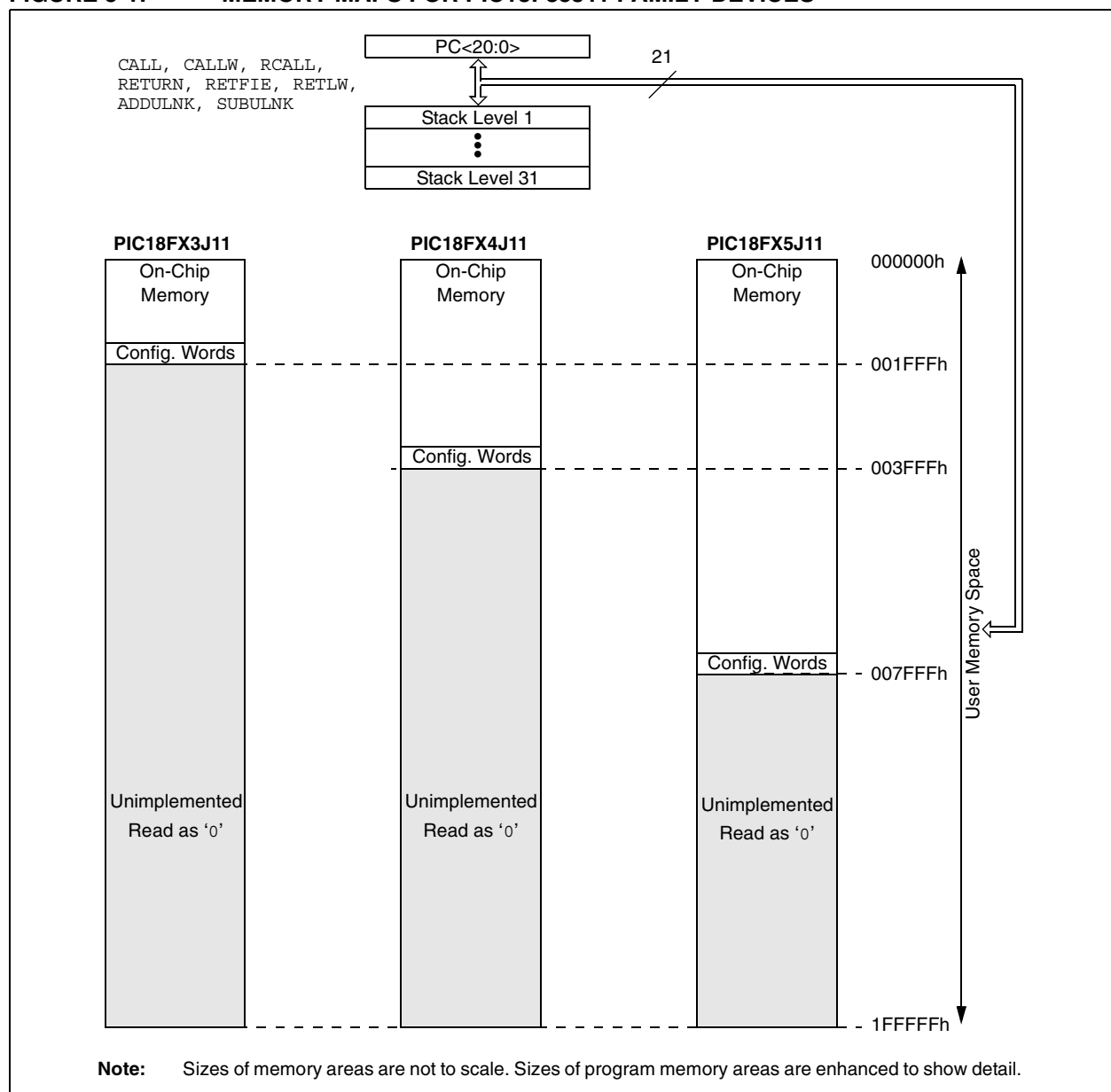
Additional detailed information on the operation of the Flash program memory is provided in **Section 6.0 “Flash Program Memory”**.

## 5.1 Program Memory Organization

PIC18 microcontrollers implement a 21-bit program counter which is capable of addressing a 2-Mbyte program memory space. Accessing a location between the upper boundary of the physically implemented memory and the 2-Mbyte address will return all ‘0’s (a NOP instruction).

The entire PIC18F85J11 family offers a range of on-chip Flash program memory sizes, from 8 Kbytes (up to 4,096 single-word instructions) to 32 Kbytes (32,768 single-word instructions). The program memory maps for individual family members are shown in Figure 5-1.

**FIGURE 5-1: MEMORY MAPS FOR PIC18F85J11 FAMILY DEVICES**



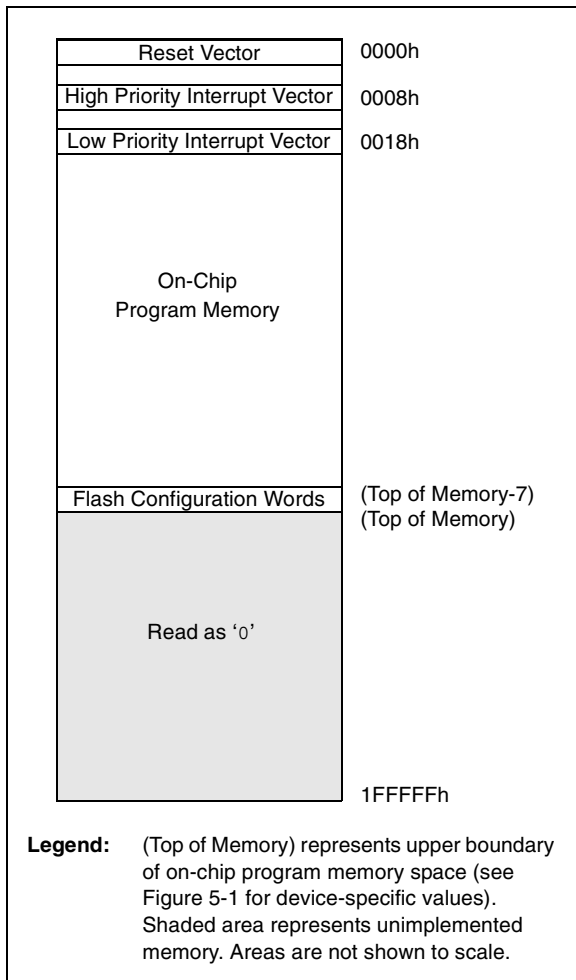
# PIC18F85J11 FAMILY

## 5.1.1 HARD MEMORY VECTORS

All PIC18 devices have a total of three hard-coded return vectors in their program memory space. The Reset vector address is the default value to which the program counter returns on all device Resets; it is located at 0000h.

PIC18 devices also have two interrupt vector addresses for the handling of high priority and low priority interrupts. The high priority interrupt vector is located at 0008h and the low priority interrupt vector is at 0018h. Their locations in relation to the program memory map are shown in Figure 5-2.

**FIGURE 5-2: HARD VECTOR AND CONFIGURATION WORD LOCATIONS FOR PIC18F85J11 FAMILY DEVICES**



## 5.1.2 FLASH CONFIGURATION WORDS

Because PIC18F85J11 family devices do not have persistent configuration memory, the top four words of on-chip program memory are reserved for configuration information. On Reset, the configuration information is copied into the Configuration registers.

The Configuration Words are stored in their program memory location in numerical order, starting with the lower byte of CONFIG1 at the lowest address and ending with the upper byte of CONFIG4. For these devices, only Configuration Words, CONFIG1 through CONFIG3, are used; CONFIG4 is reserved. The actual addresses of the Flash Configuration Word for devices in the PIC18F85J11 family are shown in Table 5-1. Their location in the memory map is shown with the other memory vectors in Figure 5-2.

Additional details on the device Configuration Words are provided in **Section 22.1 “Configuration Bits”**.

**TABLE 5-1: FLASH CONFIGURATION WORD FOR PIC18F85J11 FAMILY DEVICES**

| Device      | Program Memory (Kbytes) | Configuration Word Addresses |
|-------------|-------------------------|------------------------------|
| PIC18F63J11 | 8                       | 1FF8h to 1FFFh               |
| PIC18F83J11 |                         |                              |
| PIC18F64J11 | 16                      | 3FF8h to 3FFFh               |
| PIC18F84J11 |                         |                              |
| PIC18F65J11 | 32                      | 7FF8h to 7FFFh               |
| PIC18F85J11 |                         |                              |

# PIC18F85J11 FAMILY

## 5.1.3 PIC18F8XJ11 PROGRAM MEMORY MODES

The 80-pin devices in this family can address up to a total of 2 Mbytes of program memory. This is achieved through the external memory bus. There are two distinct operating modes available to the controllers:

- Microcontroller (MC)
- Extended Microcontroller (EMC)

The Program Memory mode is determined by setting the EMB Configuration bits (CONFIG3L<5:4>), as shown in Register 5-1. (See also **Section 22.1 “Configuration Bits”** for additional details on the device Configuration bits.)

The Program Memory modes operate as follows:

- The **Microcontroller Mode** accesses only on-chip Flash memory. Attempts to read above the top of on-chip memory causes a read of all ‘0’s (a NOP instruction).

The Microcontroller mode is also the only operating mode available to 64-pin devices.

- The **Extended Microcontroller Mode** allows access to both internal and external program memories as a single block. The device can access its entire on-chip program memory; above this, the device accesses external program memory up to the 2-Mbyte program space limit. Execution automatically switches between the two memories as required.

The setting of the EMB Configuration bits also controls the address bus width of the external memory bus. This is covered in more detail in **Section 7.0 “External Memory Bus”**.

In all modes, the microcontroller has complete access to data RAM.

Figure 5-3 compares the memory maps of the different Program Memory modes. The differences between on-chip and external memory access limitations are more fully explained in Table 5-2.

**REGISTER 5-1: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)<sup>(1)</sup>**

| R/WO-1 | R/WO-1 | R/WO-1 | R/WO-1 | R/WO-1 | U-0 | U-0 | U-0   |
|--------|--------|--------|--------|--------|-----|-----|-------|
| WAIT   | BW     | EMB1   | EMB0   | EASHFT | —   | —   | —     |
| bit 7  |        |        |        |        |     |     | bit 0 |

### Legend:

R = Readable bit                      WO = Write-Once bit                      U = Unimplemented bit, read as ‘0’  
 -n = Value when device is unprogrammed                      ‘1’ = Bit is set                      ‘0’ = Bit is cleared

- bit 7            **WAIT:** External Bus Wait Enable bit  
                  1 = Wait selections from MEMCON.WAIT<1:0> unavailable, and the device will not wait  
                  0 = Wait programmed by MEMCON.WAIT<1:0>
- bit 6            **BW:** Data Bus Width Select bit  
                  1 = 16-Bit External Bus mode  
                  0 = 8-Bit External Bus mode
- bit 5:4          **EMB1:EMB0:** External Memory Bus Configuration bits  
                  00 = Extended Microcontroller mode – 20-Bit Address mode  
                  01 = Extended Microcontroller mode – 16-Bit Address mode  
                  10 = Extended Microcontroller mode – 12-Bit Address mode  
                  11 = Microcontroller mode – external bus disabled
- bit 3            **EASHFT:** External Address Bus Shift Enable bit  
                  1 = Address shifting enabled – external address bus is shifted to start at 000000h  
                  0 = Address shifting disabled – external address bus reflects the PC value
- bit 2-0          **Unimplemented:** Read as ‘0’

**Note 1:** CONFIG3L and its associated bits are implemented only in 80-pin devices.

# PIC18F85J11 FAMILY

## 5.1.4 EXTENDED MICROCONTROLLER MODE AND ADDRESS SHIFTING

By default, devices in Extended Microcontroller mode directly present the program counter value on the external address bus for those addresses in the range of the external memory space. In practical terms, this means addresses in the external memory device below the top of on-chip memory are unavailable.

To avoid this, the Extended Microcontroller mode implements an address shifting option to enable automatic address translation. In this mode, addresses presented on the external bus are shifted down by the size of the on-chip program memory and are remapped to start at 0000h. This allows the complete use of the external memory device's memory space.

**FIGURE 5-3: MEMORY MAPS FOR PIC18F85J11 FAMILY PROGRAM MEMORY MODES**

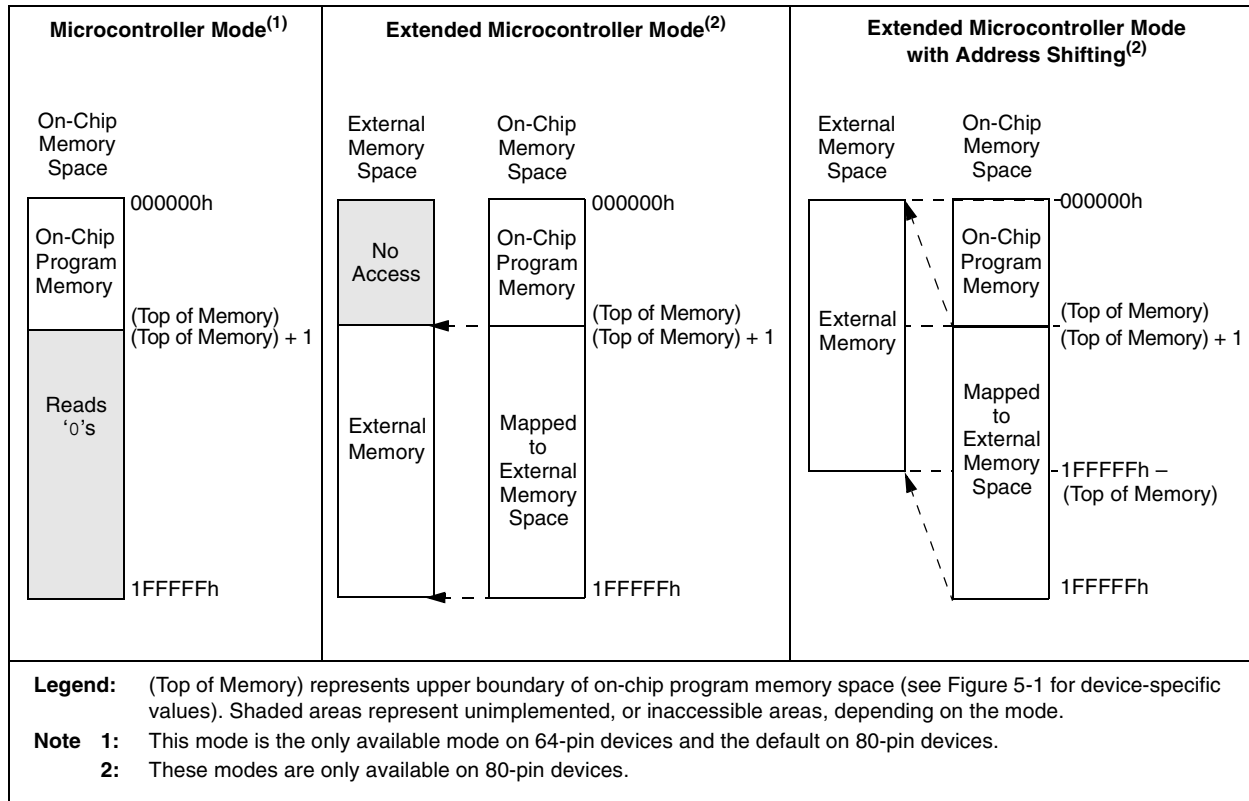


**TABLE 5-2: MEMORY ACCESS FOR PIC18F8XJ11 PROGRAM MEMORY MODES**

| Operating Mode           | Internal Program Memory |                 |                | External Program Memory |                 |                |
|--------------------------|-------------------------|-----------------|----------------|-------------------------|-----------------|----------------|
|                          | Execution From          | Table Read From | Table Write To | Execution From          | Table Read From | Table Write To |
| Microcontroller          | Yes                     | Yes             | Yes            | No Access               | No Access       | No Access      |
| Extended Microcontroller | Yes                     | Yes             | Yes            | Yes                     | Yes             | Yes            |

## 5.1.5 PROGRAM COUNTER

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide and is contained in three separate 8-bit registers. The low byte, known as the PCL register, is both readable and writable. The high byte, or PCH register, contains the PC<15:8> bits; it is not directly readable or writable. Updates to the PCH register are performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits; it is also not directly readable or writable. Updates to the PCU register are performed through the PCLATU register.

The contents of PCLATH and PCLATU are transferred to the program counter by any operation that writes PCL. Similarly, the upper two bytes of the program counter are transferred to PCLATH and PCLATU by an operation that reads PCL. This is useful for computed offsets to the PC (see **Section 5.1.8.1 “Computed GOTO”**).

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the Least Significant bit of PCL is fixed to a value of ‘0’. The PC increments by 2 to address sequential instructions in the program memory.

The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter.

## 5.1.6 RETURN ADDRESS STACK

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC is pushed onto the stack when a CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction (and on ADDULNK and SUBULNK instructions if the extended instruction set is enabled). PCLATU and PCLATH are not affected by any of the RETURN or CALL instructions.

The stack operates as a 31-word by 21-bit RAM and a 5-bit Stack Pointer, STKPTR. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the address on the top of the stack is readable and writable through the Top-of-Stack Special Function Registers. Data can also be pushed to, or popped from the stack, using these registers.

A CALL type instruction causes a push onto the stack. The Stack Pointer is first incremented and the location pointed to by the Stack Pointer is written with the contents of the PC (already pointing to the instruction following the CALL). A RETURN type instruction causes a pop from the stack. The contents of the location pointed to by the STKPTR are transferred to the PC and then the Stack Pointer is decremented.

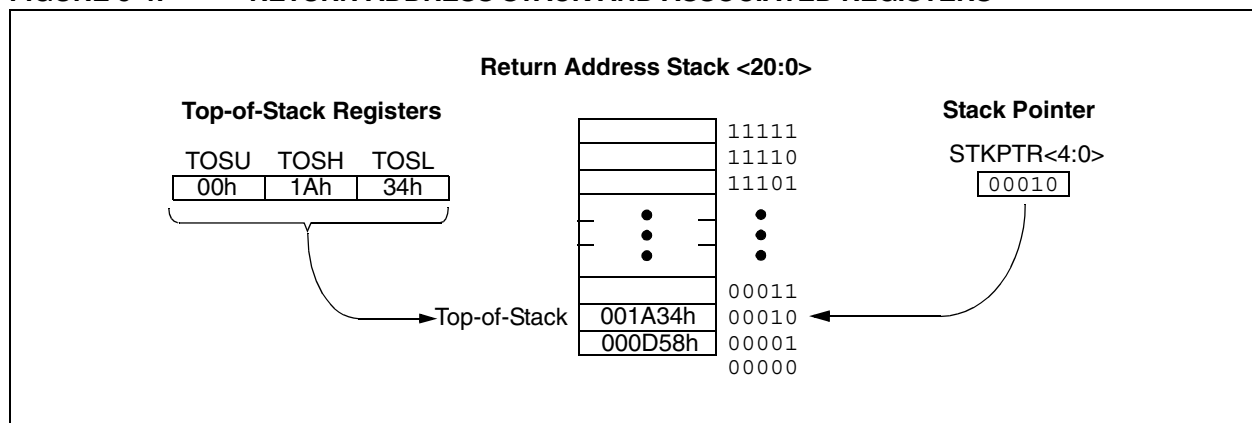
The Stack Pointer is initialized to ‘00000’ after all Resets. There is no RAM associated with the location corresponding to a Stack Pointer value of ‘00000’; this is only a Reset value. Status bits indicate if the stack is full, has overflowed or has underflowed.

### 5.1.6.1 Top-of-Stack Access

Only the top of the return address stack (TOS) is readable and writable. A set of three registers, TOSU:TOSH:TOSL, holds the contents of the stack location pointed to by the STKPTR register (Figure 5-4). This allows users to implement a software stack if necessary. After a CALL, RCALL or interrupt (and ADDULNK and SUBULNK instructions if the extended instruction set is enabled), the software can read the pushed value by reading the TOSU:TOSH:TOSL registers. These values can be placed on a user-defined software stack. At return time, the software can return these values to TOSU:TOSH:TOSL and do a return.

The user must disable the global interrupt enable bits while accessing the stack to prevent inadvertent stack corruption.

**FIGURE 5-4: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS**



# PIC18F85J11 FAMILY

## 5.1.6.2 Return Stack Pointer (STKPTR)

The STKPTR register (Register 5-2) contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments before values are pushed onto the stack and decrements after values are popped off the stack. On Reset, the Stack Pointer value will be zero. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System (RTOS) for return stack maintenance.

After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit is cleared by software or by a POR.

The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) Configuration bit. (Refer to **Section 22.1 “Configuration Bits”** for a description of the device Configuration bits.) If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to zero.

If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. Any additional pushes will not overwrite the 31st push and the STKPTR will remain at 31.

When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and set the STKUNF bit, while the Stack Pointer remains at zero. The STKUNF bit will remain set until cleared by software or until a POR occurs.

**Note:** Returning a value of zero to the PC on an underflow has the effect of vectoring the program to the Reset vector, where the stack conditions can be verified and appropriate actions can be taken. This is not the same as a Reset, as the contents of the SFRs are not affected.

## 5.1.6.3 PUSH and POP Instructions

Since the Top-of-Stack is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable feature. The PIC18 instruction set includes two instructions, **PUSH** and **POP**, that permit the TOS to be manipulated under software control. TOSU, TOSH and TOSL can be modified to place data or a return address on the stack.

The **PUSH** instruction places the current PC value onto the stack. This increments the Stack Pointer and loads the current PC value onto the stack.

The **POP** instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

## REGISTER 5-2: STKPTR: STACK POINTER REGISTER

| R/C-0                 | R/C-0                 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-----------------------|-----------------------|-----|-------|-------|-------|-------|-------|
| STKFUL <sup>(1)</sup> | STKUNF <sup>(1)</sup> | —   | SP4   | SP3   | SP2   | SP1   | SP0   |
| bit 7                 |                       |     |       |       |       |       | bit 0 |

|                   |                        |                                    |                    |
|-------------------|------------------------|------------------------------------|--------------------|
| <b>Legend:</b>    | C = Clearable-only bit |                                    |                    |
| R = Readable bit  | W = Writable bit       | U = Unimplemented bit, read as '0' |                    |
| -n = Value at POR | '1' = Bit is set       | '0' = Bit is cleared               | x = Bit is unknown |

|         |                                                                                                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------|
| bit 7   | <b>STKFUL:</b> Stack Full Flag bit <sup>(1)</sup><br>1 = Stack became full or overflowed<br>0 = Stack has not become full or overflowed |
| bit 6   | <b>STKUNF:</b> Stack Underflow Flag bit <sup>(1)</sup><br>1 = Stack underflow occurred<br>0 = Stack underflow did not occur             |
| bit 5   | <b>Unimplemented:</b> Read as '0'                                                                                                       |
| bit 4-0 | <b>SP4:SP0:</b> Stack Pointer Location bits                                                                                             |

**Note 1:** Bit 7 and bit 6 are cleared by user software or by a POR.



## 5.1.6.4 Stack Full and Underflow Resets

Device Resets on stack overflow and stack underflow conditions are enabled by setting the STVREN bit in Configuration Register 1L. When STVREN is set, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. When STVREN is cleared, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. The STKFUL or STKUNF bits are cleared by the user software or a Power-on Reset.

## 5.1.7 FAST REGISTER STACK

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

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

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

Example 5-1 shows a source code example that uses the Fast Register Stack during a subroutine call and return.

### EXAMPLE 5-1: FAST REGISTER STACK CODE EXAMPLE

```
CALL SUB1, FAST      ;STATUS, WREG, BSR
                     ;SAVED IN FAST REGISTER
                     ;STACK
    .
    .
SUB1    .
    .
    RETURN FAST      ;RESTORE VALUES SAVED
                     ;IN FAST REGISTER STACK
```

## 5.1.8 LOOK-UP TABLES IN PROGRAM MEMORY

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

- Computed GOTO
- Table Reads

### 5.1.8.1 Computed GOTO

A computed GOTO is accomplished by adding an offset to the program counter. An example is shown in Example 5-2.

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

The offset value (in WREG) specifies the number of bytes that the program counter should advance and should be multiples of 2 (LSb = 0).

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

### EXAMPLE 5-2: COMPUTED GOTO USING AN OFFSET VALUE

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

### 5.1.8.2 Table Reads

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

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

Table read operation is discussed further in Section 6.1 “Table Reads and Table Writes”.

# PIC18F85J11 FAMILY

## 5.2 PIC18 Instruction Cycle

### 5.2.1 CLOCKING SCHEME

The microcontroller clock input, whether from an internal or external source, is internally divided by four to generate four non-overlapping quadrature clocks (Q1, Q2, Q3 and Q4). Internally, the program counter is incremented on every Q1; the instruction is fetched from the program memory and latched into the Instruction Register (IR) during Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-5.

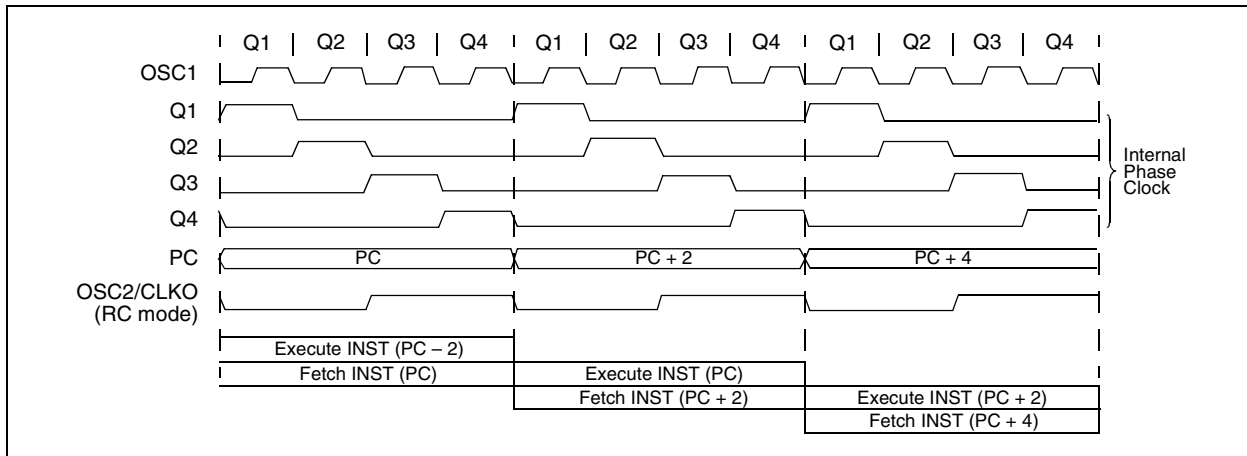
### 5.2.2 INSTRUCTION FLOW/PIPELINING

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

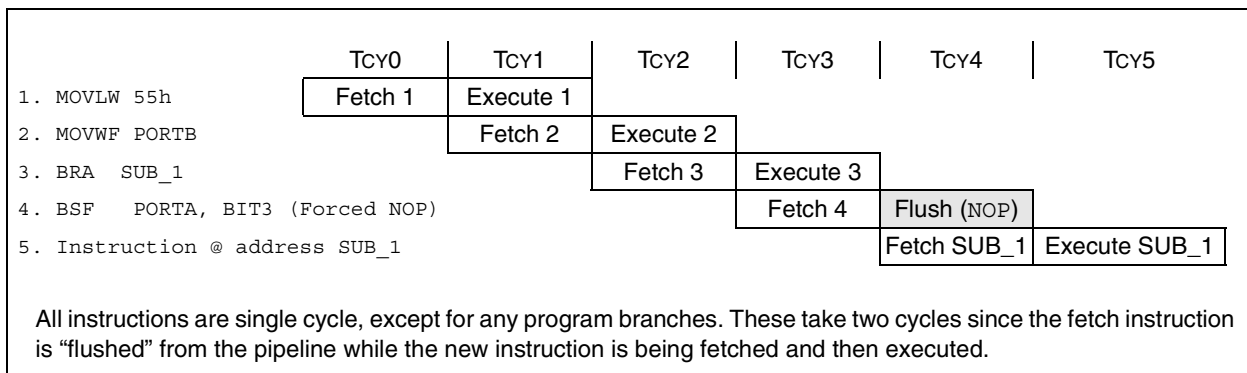
A fetch cycle begins with the Program Counter (PC) incrementing in Q1.

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

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



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



## 5.2.3 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSB = 0). To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 5.1.5 "Program Counter"**).

Figure 5-6 shows an example of how instruction words are stored in the program memory.

The **CALL** and **GOTO** instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1> which accesses the desired byte address in program memory. Instruction #2 in Figure 5-6 shows how the instruction, **GOTO 0006h**, is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 23.0 "Instruction Set Summary"** provides further details of the instruction set.

**FIGURE 5-6: INSTRUCTIONS IN PROGRAM MEMORY**

| Program Memory<br>Byte Locations → |       |            | Word Address |         |         |
|------------------------------------|-------|------------|--------------|---------|---------|
|                                    |       |            | LSB = 1      | LSB = 0 | ↓       |
| Instruction 1:                     | MOVLW | 055h       |              |         | 000000h |
|                                    |       |            |              |         | 000002h |
| Instruction 2:                     | GOTO  | 0006h      |              |         | 000004h |
|                                    |       |            |              |         | 000006h |
| Instruction 3:                     | MOVFF | 123h, 456h | 0Fh          | 55h     | 000008h |
|                                    |       |            | EFh          | 03h     | 00000Ah |
|                                    |       |            | F0h          | 00h     | 00000Ch |
|                                    |       |            | C1h          | 23h     | 00000Eh |
|                                    |       |            | F4h          | 56h     | 000010h |
|                                    |       |            |              |         | 000012h |
|                                    |       |            |              |         | 000014h |
|                                    |       |            |              |         |         |

## 5.2.4 TWO-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: **CALL**, **MOVFF**, **GOTO** and **LSFR**. In all cases, the second word of the instructions always has '1111' as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the 4 MSBs of an instruction specifies a special form of **NOP**. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed

and used by the instruction sequence. If the first word is skipped for some reason and the second word is executed by itself, a **NOP** is executed instead. This is necessary for cases when the two-word instruction is preceded by a conditional instruction that changes the PC. Example 5-4 shows how this works.

**Note:** See **Section 5.5 "Program Memory and the Extended Instruction Set"** for information on two-word instructions in the extended instruction set.

**EXAMPLE 5-4: TWO-WORD INSTRUCTIONS**

| CASE 1:             |             |                                     |
|---------------------|-------------|-------------------------------------|
| Object Code         | Source Code |                                     |
| 0110 0110 0000 0000 | TSTFSZ      | REG1 ; is RAM location 0?           |
| 1100 0001 0010 0011 | MOVFF       | REG1, REG2 ; No, skip this word     |
| 1111 0100 0101 0110 |             | ; Execute this word as a NOP        |
| 0010 0100 0000 0000 | ADDWF       | REG3 ; continue code                |
| CASE 2:             |             |                                     |
| Object Code         | Source Code |                                     |
| 0110 0110 0000 0000 | TSTFSZ      | REG1 ; is RAM location 0?           |
| 1100 0001 0010 0011 | MOVFF       | REG1, REG2 ; Yes, execute this word |
| 1111 0100 0101 0110 |             | ; 2nd word of instruction           |
| 0010 0100 0000 0000 | ADDWF       | REG3 ; continue code                |

# PIC18F85J11 FAMILY

## 5.3 Data Memory Organization

**Note:** The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each. The PIC18FX3J11/X4J11 devices, with up to 16 Kbytes of program memory, implement 4 complete banks for a total of 1024 bytes. PIC18FX5J11 devices, with 32 Kbytes of program memory, implement 8 complete banks for a total of 2048 bytes. Figure 5-7 and Figure 5-8 show the data memory organization for the devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this section.

To ensure that commonly used registers (select SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to select SFRs and the lower portion of GPR Bank 0 without using the BSR. **Section 5.3.2 “Access Bank”** provides a detailed description of the Access RAM.

### 5.3.1 BANK SELECT REGISTER

Large areas of data memory require an efficient addressing scheme to make rapid access to any address possible. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit low-order address and a 4-bit Bank Pointer.

Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the 4 Most Significant bits of a location's address; the instruction itself includes the 8 Least Significant bits. Only the four lower bits of the BSR are implemented (BSR3:BSR0). The upper four bits are unused; they will always read '0' and cannot be written to. The BSR can be loaded directly by using the `MOVLB` instruction.

The value of the BSR indicates the bank in data memory. The 8 bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 5-9.

Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh, will end up resetting the program counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 5-7 indicates which banks are implemented.

In the core PIC18 instruction set, only the `MOVFF` instruction fully specifies the 12-bit address of the source and target registers. This instruction ignores the BSR completely when it executes. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

FIGURE 5-7: DATA MEMORY MAP FOR PIC18FX3J11/X4J11 DEVICES

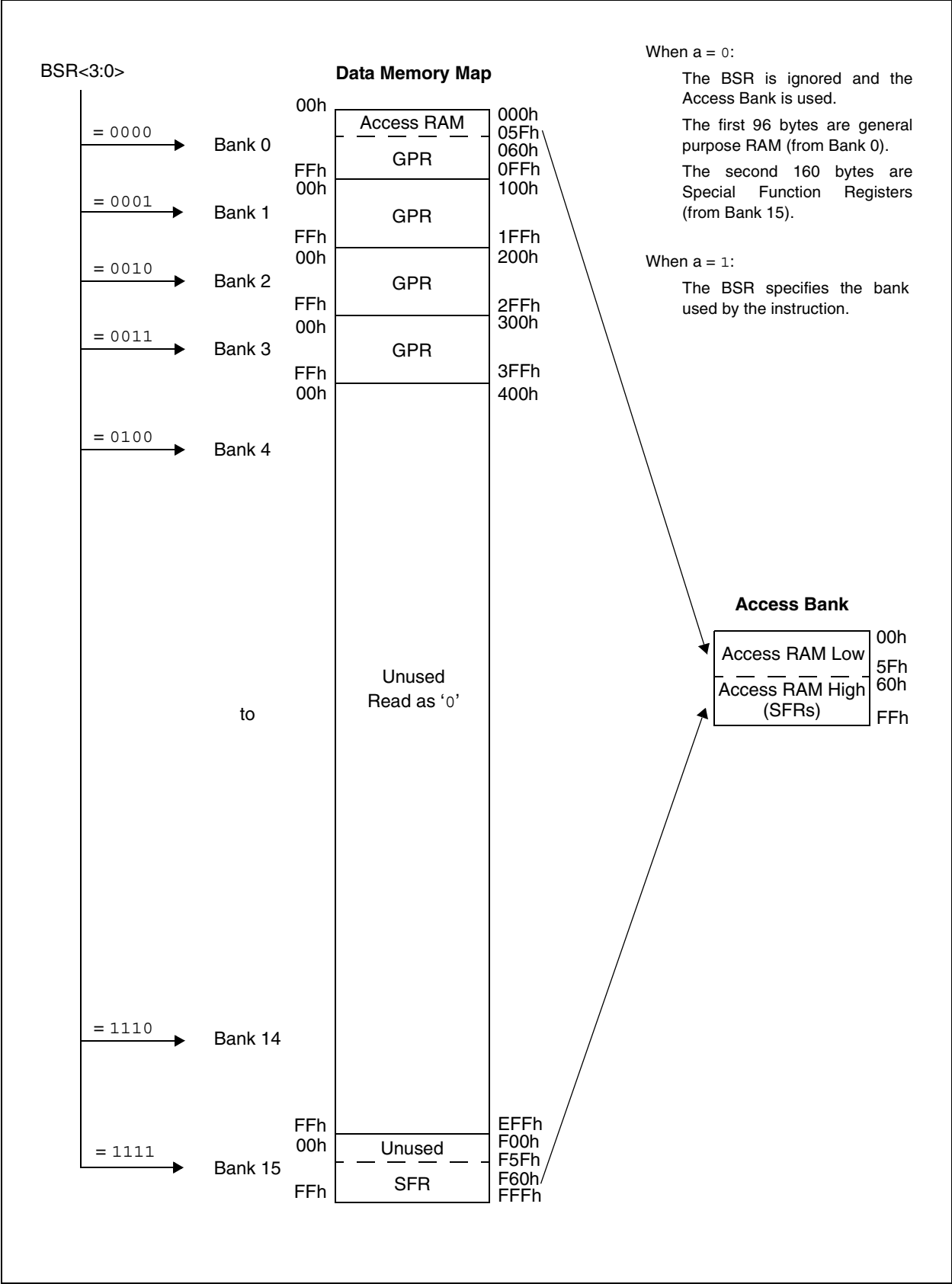
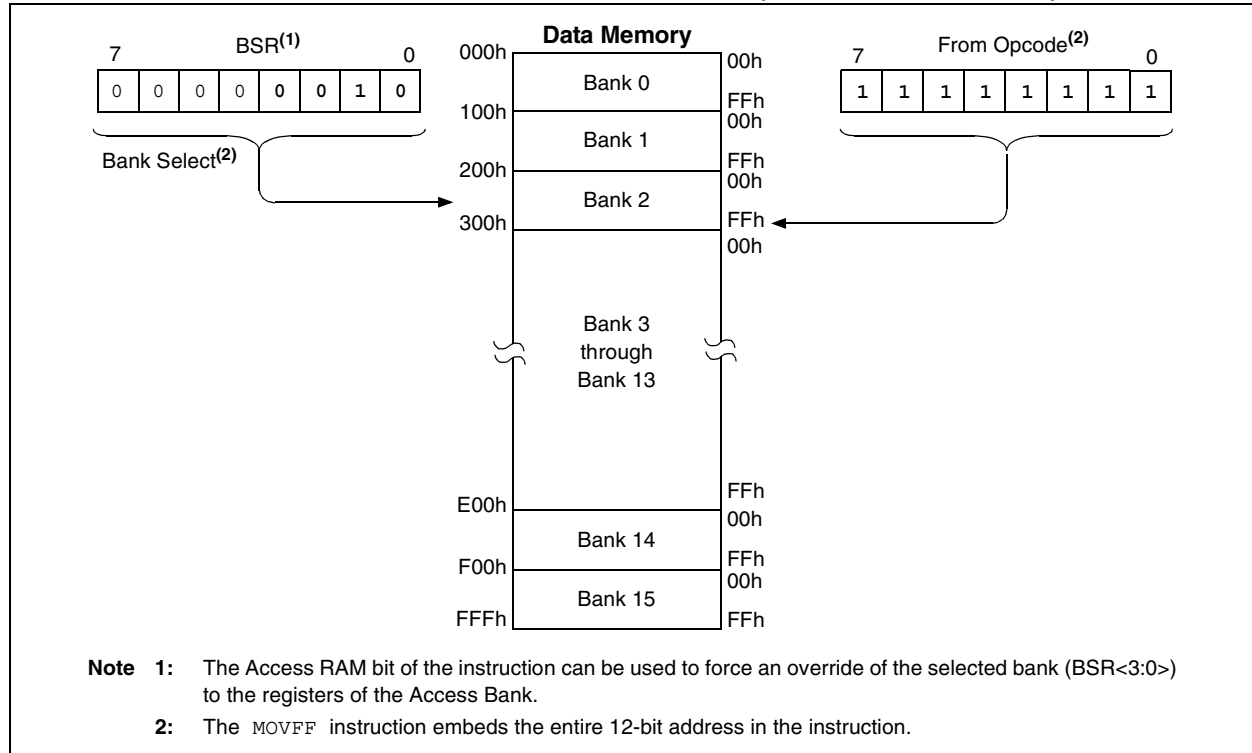




**FIGURE 5-9: USE OF THE BANK SELECT REGISTER (DIRECT ADDRESSING)**



## 5.3.2 ACCESS BANK

While the use of the BSR with an embedded 8-bit address allows users to address the entire range of data memory, it also means that the user must always ensure that the correct bank is selected. Otherwise, data may be read from, or written to, the wrong location. This can be disastrous if a GPR is the intended target of an operation, but an SFR is written to instead. Verifying and/or changing the BSR for each read or write to data memory can become very inefficient.

To streamline access for the most commonly used data memory locations, the data memory is configured with an Access Bank, which allows users to access a mapped block of memory without specifying a BSR. The Access Bank consists of the first 96 bytes of memory (00h-5Fh) in Bank 0 and the last 160 bytes of memory (60h-FFh) in Bank 15. The lower half is known as the "Access RAM" and is composed of GPRs. The upper half is where the device's SFRs are mapped. These two areas are mapped contiguously in the Access Bank and can be addressed in a linear fashion by an 8-bit address (Figure 5-7).

The Access Bank is used by core PIC18 instructions that include the Access RAM bit (the 'a' parameter in the instruction). When 'a' is equal to '1', the instruction uses the BSR and the 8-bit address included in the opcode for the data memory address. When 'a' is '0',

however, the instruction is forced to use the Access Bank address map; the current value of the BSR is ignored entirely.

Using this "forced" addressing allows the instruction to operate on a data address in a single cycle without updating the BSR first. For 8-bit addresses of 60h and above, this means that users can evaluate and operate on SFRs more efficiently. The Access RAM below 60h is a good place for data values that the user might need to access rapidly, such as immediate computational results or common program variables. Access RAM also allows for faster and more code efficient context saving and switching of variables.

The mapping of the Access Bank is slightly different when the extended instruction set is enabled (XINST Configuration bit = 1). This is discussed in more detail in **Section 5.6.3 "Mapping the Access Bank in Indexed Literal Offset Mode"**.

## 5.3.3 GENERAL PURPOSE REGISTER FILE

PIC18 devices may have banked memory in the GPR area. This is data RAM which is available for use by all instructions. GPRs start at the bottom of Bank 0 (address 000h) and grow upwards towards the bottom of the SFR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets.

# PIC18F85J11 FAMILY

## 5.3.4 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. SFRs start at the top of data memory (FFFh) and extend downward to occupy more than the top half of Bank 15 (F60h to FFFh). A list of these registers is given in Table 5-3 and Table 5-4.

The SFRs can be classified into two sets: those associated with the “core” device functionality (ALU, Resets and interrupts) and those related to the peripheral functions. The Reset and Interrupt registers are described in their respective chapters, while the ALU’s STATUS register is described later in this section. Registers related to the operation of the peripheral features are described in the chapter for that peripheral.

The SFRs are typically distributed among the peripherals whose functions they control. Unused SFR locations are unimplemented and read as ‘0’s.

**TABLE 5-3: SPECIAL FUNCTION REGISTER MAP FOR PIC18F85J11 FAMILY DEVICES**

| Address | Name                    | Address | Name                    | Address | Name             | Address | Name                  | Address | Name             |
|---------|-------------------------|---------|-------------------------|---------|------------------|---------|-----------------------|---------|------------------|
| FFFh    | TOSU                    | FDFh    | INDF2 <sup>(1)</sup>    | FBFh    | — <sup>(2)</sup> | F9Fh    | IPR1                  | F7Fh    | SPBRGH1          |
| FFEh    | TOSH                    | FDEh    | POSTINC2 <sup>(1)</sup> | FBEh    | — <sup>(2)</sup> | F9Eh    | PIR1                  | F7Eh    | BAUDCON1         |
| FFDh    | TOSL                    | FDDh    | POSTDEC2 <sup>(1)</sup> | FBDh    | — <sup>(2)</sup> | F9Dh    | PIE1                  | F7Dh    | — <sup>(2)</sup> |
| FFCh    | STKPTR                  | FDCh    | PREINC2 <sup>(1)</sup>  | FBCh    | — <sup>(2)</sup> | F9Ch    | MEMCON <sup>(3)</sup> | F7Ch    | — <sup>(2)</sup> |
| FFBh    | PCLATU                  | FDBh    | PLUSW2 <sup>(1)</sup>   | FBBh    | — <sup>(2)</sup> | F9Bh    | OSCTUNE               | F7Bh    | — <sup>(2)</sup> |
| FFAh    | PCLATH                  | FDAh    | FSR2H                   | FBAh    | — <sup>(2)</sup> | F9Ah    | TRISJ <sup>(3)</sup>  | F7Ah    | — <sup>(2)</sup> |
| FF9h    | PCL                     | FD9h    | FSR2L                   | FB9h    | — <sup>(2)</sup> | F99h    | TRISH <sup>(3)</sup>  | F79h    | — <sup>(2)</sup> |
| FF8h    | TBLPTRU                 | FD8h    | STATUS                  | FB8h    | — <sup>(2)</sup> | F98h    | TRISG                 | F78h    | — <sup>(2)</sup> |
| FF7h    | TBLPTRH                 | FD7h    | TMR0H                   | FB7h    | — <sup>(2)</sup> | F97h    | TRISF                 | F77h    | — <sup>(2)</sup> |
| FF6h    | TBLPTRL                 | FD6h    | TMR0L                   | FB6h    | — <sup>(2)</sup> | F96h    | TRISE                 | F76h    | — <sup>(2)</sup> |
| FF5h    | TABLAT                  | FD5h    | T0CON                   | FB5h    | CVRCON           | F95h    | TRISD                 | F75h    | — <sup>(2)</sup> |
| FF4h    | PRODH                   | FD4h    | — <sup>(2)</sup>        | FB4h    | CMCON            | F94h    | TRISC                 | F74h    | — <sup>(2)</sup> |
| FF3h    | PRODL                   | FD3h    | OSCCON                  | FB3h    | TMR3H            | F93h    | TRISB                 | F73h    | — <sup>(2)</sup> |
| FF2h    | INTCON                  | FD2h    | — <sup>(2)</sup>        | FB2h    | TMR3L            | F92h    | TRISA                 | F72h    | — <sup>(2)</sup> |
| FF1h    | INTCON2                 | FD1h    | WDTCON                  | FB1h    | T3CON            | F91h    | LATJ <sup>(3)</sup>   | F71h    | — <sup>(2)</sup> |
| FF0h    | INTCON3                 | FD0h    | RCON                    | FB0h    | PSPCON           | F90h    | LATH <sup>(3)</sup>   | F70h    | — <sup>(2)</sup> |
| FEFh    | INDF0 <sup>(1)</sup>    | FCFh    | TMR1H                   | FAFh    | SPBRG1           | F8Fh    | LATG                  | F6Fh    | — <sup>(2)</sup> |
| FEeh    | POSTINC0 <sup>(1)</sup> | FCEh    | TMR1L                   | FAeh    | RCREG1           | F8Eh    | LATF                  | F6Eh    | — <sup>(2)</sup> |
| FEDh    | POSTDEC0 <sup>(1)</sup> | FCDh    | T1CON                   | FADh    | TXREG1           | F8Dh    | LATE                  | F6Dh    | — <sup>(2)</sup> |
| FECh    | PREINC0 <sup>(1)</sup>  | FCCh    | TMR2                    | FACH    | TXSTA1           | F8Ch    | LATD                  | F6Ch    | — <sup>(2)</sup> |
| FEBh    | PLUSW0 <sup>(1)</sup>   | FCBh    | PR2                     | FABh    | RCSTA1           | F8Bh    | LATC                  | F6Bh    | — <sup>(2)</sup> |
| FEAh    | FSR0H                   | FCAh    | T2CON                   | FAAh    | — <sup>(2)</sup> | F8Ah    | LATB                  | F6Ah    | CCPR1H           |
| FE9h    | FSR0L                   | FC9h    | SSPBUF                  | FA9h    | — <sup>(2)</sup> | F89h    | LATA                  | F69h    | CCPR1L           |
| FE8h    | WREG                    | FC8h    | SSPADD                  | FA8h    | — <sup>(2)</sup> | F88h    | PORTJ <sup>(3)</sup>  | F68h    | CCP1CON          |
| FE7h    | INDF1 <sup>(1)</sup>    | FC7h    | SSPSTAT                 | FA7h    | EECON2           | F87h    | PORTH <sup>(3)</sup>  | F67h    | CCPR2H           |
| FE6h    | POSTINC1 <sup>(1)</sup> | FC6h    | SSPCON1                 | FA6h    | EECON1           | F86h    | PORTG                 | F66h    | CCPR2L           |
| FE5h    | POSTDEC1 <sup>(1)</sup> | FC5h    | SSPCON2                 | FA5h    | IPR3             | F85h    | PORTF                 | F65h    | CCP2CON          |
| FE4h    | PREINC1 <sup>(1)</sup>  | FC4h    | ADRESH                  | FA4h    | PIR3             | F84h    | PORTE                 | F64h    | SPBRG2           |
| FE3h    | PLUSW1 <sup>(1)</sup>   | FC3h    | ADRESL                  | FA3h    | PIE3             | F83h    | PORTD                 | F63h    | RCREG2           |
| FE2h    | FSR1H                   | FC2h    | ADCON0                  | FA2h    | IPR2             | F82h    | PORTC                 | F62h    | TXREG2           |
| FE1h    | FSR1L                   | FC1h    | ADCON1                  | FA1h    | PIR2             | F81h    | PORTB                 | F61h    | TXSTA2           |
| FE0h    | BSR                     | FC0h    | ADCON2                  | FA0h    | PIE2             | F80h    | PORTA                 | F60h    | RCSTA2           |

- Note** 1: This is not a physical register.  
2: Unimplemented registers are read as ‘0’.  
3: This register is not available on 64-pin devices.



# PIC18F85J11 FAMILY

**TABLE 5-4: PIC18F85J11 FAMILY REGISTER FILE SUMMARY**

| 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 |
|-----------|------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------------------|---------------------------------------------------------|--------------------------------------------------|--------|--------|--------|-------------------|-----------------|
| TOSU      | —                                                                                                                                  | —         | —                     | Top-of-Stack Upper Byte (TOS<20:16>)                    |                                                  |        |        |        | ---0 0000         | 51, 61          |
| TOSH      | Top-of-Stack High Byte (TOS<15:8>)                                                                                                 |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 61          |
| TOSL      | Top-of-Stack Low Byte (TOS<7:0>)                                                                                                   |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 61          |
| STKPTR    | STKFUL                                                                                                                             | STKUNF    | —                     | Return Stack Pointer                                    |                                                  |        |        |        | uu-0 0000         | 51, 62          |
| PCLATU    | —                                                                                                                                  | —         | bit 21 <sup>(1)</sup> | Holding Register for PC<20:16>                          |                                                  |        |        |        | ---0 0000         | 51, 61          |
| PCLATH    | Holding Register for PC<15:8>                                                                                                      |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 61          |
| PCL       | PC Low Byte (PC<7:0>)                                                                                                              |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 61          |
| TBLPTRU   | —                                                                                                                                  | —         | bit 21                | Program Memory Table Pointer Upper Byte (TBLPTR<20:16>) |                                                  |        |        |        | --00 0000         | 51, 86          |
| TBLPTRH   | Program Memory Table Pointer High Byte (TBLPTR<15:8>)                                                                              |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 86          |
| TBLPTRL   | Program Memory Table Pointer Low Byte (TBLPTR<7:0>)                                                                                |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 86          |
| TABLAT    | Program Memory Table Latch                                                                                                         |           |                       |                                                         |                                                  |        |        |        | 0000 0000         | 51, 86          |
| PRODH     | Product Register High Byte                                                                                                         |           |                       |                                                         |                                                  |        |        |        | xxxx xx           | 51, 105         |
| PRODL     | Product Register Low Byte                                                                                                          |           |                       |                                                         |                                                  |        |        |        | xxxx xx           | 51, 105         |
| INTCON    | GIE/GIEH                                                                                                                           | PEIE/GIEL | TMR0IE                | INT0IE                                                  | RBIE                                             | TMR0IF | INT0IF | RBIF   | 0000 000x         | 51, 109         |
| INTCON2   | RBPU                                                                                                                               | INTEDG0   | INTEDG1               | INTEDG2                                                 | INTEDG3                                          | TMR0IP | INT3IP | RBIP   | 1111 1111         | 51, 110         |
| INTCON3   | INT2IP                                                                                                                             | INT1IP    | INT3IE                | INT2IE                                                  | INT1IE                                           | INT3IF | INT2IF | INT1IF | 1100 0000         | 51, 111         |
| INDF0     | Uses contents of FSR0 to address data memory – value of FSR0 not changed (not a physical register)                                 |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 77          |
| POSTINC0  | Uses contents of FSR0 to address data memory – value of FSR0 post-incremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| POSTDEC0  | Uses contents of FSR0 to address data memory – value of FSR0 post-decremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| PREINC0   | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register)                             |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| PLUSW0    | Uses contents of FSR0 to address data memory – value of FSR0 pre-incremented (not a physical register) – value of FSR0 offset by W |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| FSR0H     | —                                                                                                                                  | —         | —                     | —                                                       | Indirect Data Memory Address Pointer 0 High Byte |        |        |        | ---- xxxx         | 51, 77          |
| FSR0L     | Indirect Data Memory Address Pointer 0 Low Byte                                                                                    |           |                       |                                                         |                                                  |        |        |        | xxxx xxxx         | 51, 77          |
| WREG      | Working Register                                                                                                                   |           |                       |                                                         |                                                  |        |        |        | xxxx xxxx         | 51              |
| INDF1     | Uses contents of FSR1 to address data memory – value of FSR1 not changed (not a physical register)                                 |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 77          |
| POSTINC1  | Uses contents of FSR1 to address data memory – value of FSR1 post-incremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| POSTDEC1  | Uses contents of FSR1 to address data memory – value of FSR1 post-decremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| PREINC1   | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register)                             |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| PLUSW1    | Uses contents of FSR1 to address data memory – value of FSR1 pre-incremented (not a physical register) – value of FSR1 offset by W |           |                       |                                                         |                                                  |        |        |        | N/A               | 51, 78          |
| FSR1H     | —                                                                                                                                  | —         | —                     | —                                                       | Indirect Data Memory Address Pointer 1 High Byte |        |        |        | ---- xxxx         | 52, 77          |
| FSR1L     | Indirect Data Memory Address Pointer 1 Low Byte                                                                                    |           |                       |                                                         |                                                  |        |        |        | xxxx xxxx         | 52, 77          |
| BSR       | —                                                                                                                                  | —         | —                     | —                                                       | Bank Select Register                             |        |        |        | ---- 0000         | 52, 66          |
| INDF2     | Uses contents of FSR2 to address data memory – value of FSR2 not changed (not a physical register)                                 |           |                       |                                                         |                                                  |        |        |        | N/A               | 52, 77          |
| POSTINC2  | Uses contents of FSR2 to address data memory – value of FSR2 post-incremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 52, 78          |
| POSTDEC2  | Uses contents of FSR2 to address data memory – value of FSR2 post-decremented (not a physical register)                            |           |                       |                                                         |                                                  |        |        |        | N/A               | 52, 78          |
| PREINC2   | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register)                             |           |                       |                                                         |                                                  |        |        |        | N/A               | 52, 78          |
| PLUSW2    | Uses contents of FSR2 to address data memory – value of FSR2 pre-incremented (not a physical register) – value of FSR2 offset by W |           |                       |                                                         |                                                  |        |        |        | N/A               | 52, 78          |
| FSR2H     | —                                                                                                                                  | —         | —                     | —                                                       | Indirect Data Memory Address Pointer 2 High Byte |        |        |        | ---- xxxx         | 52, 77          |
| FSR2L     | Indirect Data Memory Address Pointer 2 Low Byte                                                                                    |           |                       |                                                         |                                                  |        |        |        | xxxx xxxx         | 52, 77          |
| STATUS    | —                                                                                                                                  | —         | —                     | N                                                       | OV                                               | Z      | DC     | C      | --x xxxx          | 52, 75          |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Bit 21 of the PC is only available in Test mode and Serial Programming modes.

- 2: These registers and/or bits are available only on 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset states shown are for 80-pin devices.
- 3: Alternate names and definitions for these bits when the MSSP module is operating in I<sup>2</sup>C™ Slave mode. See **Section 16.4.3.2 “Address Masking”** for details.
- 4: The PLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 2.4.3 “PLL Frequency Multiplier”** for details.
- 5: RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as '0'.

# PIC18F85J11 FAMILY

**TABLE 5-4: PIC18F85J11 FAMILY REGISTER FILE SUMMARY (CONTINUED)**

| 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 |
|-----------|----------------------------------------------------------------------------------------------------------------------|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------|-------------------|-----------------|
| TMR0H     | Timer0 Register High Byte                                                                                            |          |                       |                       |                       |                       |                       |         | 0000 0000         | 52, 149         |
| TMR0L     | Timer0 Register Low Byte                                                                                             |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 52, 149         |
| T0CON     | TMR0ON                                                                                                               | T08BIT   | T0CS                  | T0SE                  | PSA                   | T0PS2                 | T0PS1                 | T0PS0   | 1111 1111         | 52, 147         |
| OSCCON    | IDLEN                                                                                                                | IRCF2    | IRCF1                 | IRCF0                 | OSTS                  | IOFS                  | SCS1                  | SCS0    | 0100 q000         | 30, 52          |
| WDTCON    | REGSLP                                                                                                               | —        | —                     | —                     | —                     | —                     | —                     | SWDTEN  | 0 --- ---0        | 52, 278         |
| RCON      | IPEN                                                                                                                 | —        | CM                    | RI                    | TO                    | PD                    | POR                   | BOR     | 0-11 11q0         | 46, 52          |
| TMR1H     | Timer1 Register High Byte                                                                                            |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 52, 155         |
| TMR1L     | Timer1 Register Low Byte                                                                                             |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 52, 155         |
| T1CON     | RD16                                                                                                                 | T1RUN    | T1CKPS1               | T1CKPS0               | T1OSEN                | T1SYN $\bar{C}$       | TMR1CS                | TMR1ON  | 0000 0000         | 52, 151         |
| TMR2      | Timer2 Register                                                                                                      |          |                       |                       |                       |                       |                       |         | 0000 0000         | 52, 158         |
| PR2       | Timer2 Period Register                                                                                               |          |                       |                       |                       |                       |                       |         | 1111 1111         | 52, 158         |
| T2CON     | —                                                                                                                    | T2OUTPS3 | T2OUTPS2              | T2OUTPS1              | T2OUTPS0              | TMR2ON                | T2CKPS1               | T2CKPS0 | -000 0000         | 52, 157         |
| SSPBUF    | MSSP Receive Buffer/Transmit Register                                                                                |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 52, 181, 216    |
| SSPAD     | MSSP Address Register (I <sup>2</sup> C™ Slave mode). MSSP Baud Rate Reload Register (I <sup>2</sup> C Master mode). |          |                       |                       |                       |                       |                       |         | 0000 0000         | 52, 216         |
| SSPSTAT   | SMP                                                                                                                  | CKE      | D/ $\bar{A}$          | P                     | S                     | R/ $\bar{W}$          | UA                    | BF      | 0000 0000         | 52, 174, 183    |
| SSPCON1   | WCOL                                                                                                                 | SSPOV    | SSPEN                 | CKP                   | SSPM3                 | SSPM2                 | SSPM1                 | SSPM0   | 0000 0000         | 52, 175, 184    |
| SSPCON2   | GCEN                                                                                                                 | ACKSTAT  | ACKDT                 | ACKEN                 | RCEN                  | PEN                   | RSEN                  | SEN     | 0000 0000         | 52, 185, 186    |
|           | GCEN                                                                                                                 | ACKSTAT  | ADMSK5 <sup>(3)</sup> | ADMSK4 <sup>(3)</sup> | ADMSK3 <sup>(3)</sup> | ADMSK2 <sup>(3)</sup> | ADMSK1 <sup>(3)</sup> | SEN     |                   |                 |
| ADRESH    | A/D Result Register High Byte                                                                                        |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 53, 259         |
| ADRESL    | A/D Result Register Low Byte                                                                                         |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 53, 259         |
| ADCON0    | ADCAL                                                                                                                | —        | CHS3                  | CHS2                  | CHS1                  | CHS0                  | GO/DONE               | ADON    | 0-00 0000         | 53, 251         |
| ADCON1    | —                                                                                                                    | —        | VCFG1                 | VCFG0                 | PCFG3                 | PCFG2                 | PCFG1                 | PCFG0   | --00 0000         | 53, 252         |
| ADCON2    | ADFM                                                                                                                 | —        | ACQT2                 | ACQT1                 | ACQT0                 | ADCS2                 | ADCS1                 | ADCS0   | 0-00 0000         | 53, 253         |
| CVRCON    | CVREN                                                                                                                | CVROE    | CVRR                  | CVRSS                 | CVR3                  | CVR2                  | CVR1                  | CVR0    | 0000 0000         | 53, 267         |
| CMCON     | C2OUT                                                                                                                | C1OUT    | C2INV                 | C1INV                 | CIS                   | CM2                   | CM1                   | CM0     | 0000 0111         | 53, 261         |
| TMR3H     | Timer3 Register High Byte                                                                                            |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 53, 161         |
| TMR3L     | Timer3 Register Low Byte                                                                                             |          |                       |                       |                       |                       |                       |         | xxxx xxxx         | 53, 161         |
| T3CON     | RD16                                                                                                                 | T3CCP2   | T3CKPS1               | T3CKPS0               | T3CCP1                | T3SYN $\bar{C}$       | TMR3CS                | TMR3ON  | 0000 0000         | 53, 159         |
| PSPCON    | IBF                                                                                                                  | OBF      | IBOV                  | PSPMODE               | —                     | —                     | —                     | —       | 0000 ----         | 53, 159         |
| SPBRG1    | EUSART Baud Rate Generator Register                                                                                  |          |                       |                       |                       |                       |                       |         | 0000 0000         | 53, 221         |
| RCREG1    | EUSART Receive Register                                                                                              |          |                       |                       |                       |                       |                       |         | 0000 0000         | 53, 229         |
| TXREG1    | EUSART Transmit Register                                                                                             |          |                       |                       |                       |                       |                       |         | 0000 0000         | 53, 227         |
| TXSTA1    | CSRC                                                                                                                 | TX9      | TXEN                  | SYNC                  | SENDB                 | BRGH                  | TRMT                  | TX9D    | 0000 0010         | 53, 218         |
| RCSTA1    | SPEN                                                                                                                 | RX9      | SREN                  | CREN                  | ADDEN                 | FERR                  | OERR                  | RX9D    | 0000 000x         | 53, 219         |
| EECON2    | EEPROM Control Register 2 (not a physical register)                                                                  |          |                       |                       |                       |                       |                       |         | ---- ----         | 53, 84          |
| EECON1    | —                                                                                                                    | —        | —                     | FREE                  | WRERR                 | WREN                  | WR                    | —       | ---0 x00-         | 53, 85          |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Bit 21 of the PC is only available in Test mode and Serial Programming modes.

- 2: These registers and/or bits are available only on 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset states shown are for 80-pin devices.
- 3: Alternate names and definitions for these bits when the MSSP module is operating in I<sup>2</sup>C™ Slave mode. See **Section 16.4.3.2 "Address Masking"** for details.
- 4: The PLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 2.4.3 "PLL Frequency Multiplier"** for details.
- 5: RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as '0'.

# PIC18F85J11 FAMILY

**TABLE 5-4: PIC18F85J11 FAMILY REGISTER FILE SUMMARY (CONTINUED)**

| 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 |
|-----------------------|-----------------------|-----------------------|---------------------|--------|--------|--------|--------|--------|-------------------|-----------------|
| IPR3                  | —                     | —                     | RC2IP               | TX2IP  | —      | CCP2IP | CCP1IP | —      | --00 -11-         | 53, 120         |
| PIR3                  | —                     | —                     | RC2IF               | TX2IF  | —      | CCP2IF | CCP1IF | —      | --00 -00-         | 53, 114         |
| PIE3                  | —                     | —                     | RC2IE               | TX2IE  | —      | CCP2IE | CCP1IE | —      | --00 -00-         | 53, 117         |
| IPR2                  | OSCFIP                | CMIP                  | —                   | —      | BCLIP  | LVDIP  | TMR3IP | —      | 11-- 111-         | 53, 119         |
| PIR2                  | OSCFIF                | CMIF                  | —                   | —      | BCLIF  | LVDIF  | TMR3IF | —      | 00-- 000-         | 53, 113         |
| PIE2                  | OSCFIE                | CMIE                  | —                   | —      | BCLIE  | LVDIE  | TMR3IE | —      | 00-- 000-         | 53, 116         |
| IPR1                  | PSPIP                 | ADIP                  | RC1IP               | TX1IP  | SSPIP  | —      | TMR2IP | TMR1IP | 1111 1-11         | 53, 118         |
| PIR1                  | PSPIF                 | ADIF                  | RC1IF               | TX1IF  | SSPIF  | —      | TMR2IF | TMR1IF | 0000 0-00         | 53, 112         |
| PIE1                  | PSPIE                 | ADIE                  | RC1IE               | TX1IE  | SSPIE  | —      | TMR2IE | TMR1IE | 0000 0-00         | 53, 115         |
| MEMCON <sup>(2)</sup> | EBDIS                 | —                     | WAIT1               | WAIT0  | —      | —      | WM1    | WM0    | 0-00 --00         | 53, 94          |
| OSCTUNE               | INTSRC                | PLLEN <sup>(4)</sup>  | TUN5                | TUN4   | TUN3   | TUN2   | TUN1   | TUN0   | 0000 0000         | 31, 53          |
| TRISJ <sup>(2)</sup>  | TRISJ7                | TRISJ6                | TRISJ5              | TRISJ4 | TRISJ3 | TRISJ2 | TRISJ1 | TRISJ0 | 1111 1111         | 54, 143         |
| TRISH <sup>(2)</sup>  | TRISH7                | TRISH6                | TRISH5              | TRISH4 | TRISH3 | TRISH2 | TRISH1 | TRISH0 | 1111 1111         | 54, 141         |
| TRISG                 | SPIOD                 | CCP2OD                | CCP1OD              | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 | 0001 1111         | 54, 140         |
| TRISF                 | TRISF7                | TRISF6                | TRISF5              | TRISF4 | TRISF3 | TRISF2 | TRISF1 | —      | 1111 111-         | 54, 138         |
| TRISE                 | TRISE7                | TRISE6                | TRISE5              | TRISE4 | TRISE3 | —      | TRISE1 | TRISE0 | 1111 1-11         | 54, 136         |
| TRISD                 | TRISD7                | TRISD6                | TRISD5              | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 1111 1111         | 54, 133         |
| TRISC                 | TRISC7                | TRISC6                | TRISC5              | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 1111 1111         | 54, 130         |
| TRISB                 | TRISB7                | TRISB6                | TRISB5              | TRISB4 | TRISB3 | TRISB2 | TRISB1 | TRISB0 | 1111 1111         | 54, 127         |
| TRISA                 | TRISA7 <sup>(5)</sup> | TRISA6 <sup>(5)</sup> | TRISA5              | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 1111 1111         | 54, 125         |
| LATJ <sup>(2)</sup>   | LATJ7                 | LATJ6                 | LATJ5               | LATJ4  | LATJ3  | LATJ2  | LATJ1  | LATJ0  | xxxx xxxx         | 54, 143         |
| LATH <sup>(2)</sup>   | LATH7                 | LATH6                 | LATH5               | LATH4  | LATH3  | LATH2  | LATH1  | LATH0  | xxxx xxxx         | 54, 141         |
| LATG                  | U2OD                  | U1OD                  | —                   | LATG4  | LATG3  | LATG2  | LATG1  | LATG0  | 00-x xxxx         | 54, 140         |
| LATF                  | LATF7                 | LATF6                 | LATF5               | LATF4  | LATF3  | LATF2  | LATF1  | —      | xxxx xxx-         | 54, 138         |
| LATE                  | LATE7                 | LATE6                 | LATE5               | LATE4  | LATE3  | LATE2  | LATE1  | LATE0  | xxxx xxxx         | 54, 136         |
| LATD                  | LATD7                 | LATD6                 | LATD5               | LATD4  | LATD3  | LATD2  | LATD1  | LATD0  | xxxx xxxx         | 54, 133         |
| LATC                  | LATC7                 | LATC6                 | LATC5               | LATC4  | LATC3  | LATC2  | LATC1  | LATC0  | xxxx xxxx         | 54, 130         |
| LATB                  | LATB7                 | LATB6                 | LATB5               | LATB4  | LATB3  | LATB2  | LATB1  | LATB0  | xxxx xxxx         | 54, 127         |
| LATA                  | LATA7 <sup>(5)</sup>  | LATA6 <sup>(5)</sup>  | LATA5               | LATA4  | LATA3  | LATA2  | LATA1  | LATA0  | xxxx xxxx         | 54, 125         |
| PORTJ <sup>(2)</sup>  | RJ7                   | RJ6                   | RJ5                 | RJ4    | RJ3    | RJ2    | RJ1    | RJ0    | xxxx xxxx         | 54, 143         |
| PORTH <sup>(2)</sup>  | RH7                   | RH6                   | RH5                 | RH4    | RH3    | RH2    | RH1    | RH0    | xxxx xxxx         | 54, 141         |
| PORTG                 | RDPU                  | REPU                  | RJPU <sup>(2)</sup> | RG4    | RG3    | RG2    | RG1    | RG0    | 000x xxxx         | 54, 140         |
| PORTF                 | RF7                   | RF6                   | RF5                 | RF4    | RF3    | RF2    | RF1    | —      | xxxx xxx-         | 54, 138         |
| PORTE                 | RE7                   | RE6                   | RE5                 | RE4    | RE3    | —      | RE1    | RE0    | xxxx x-xx         | 54, 136         |
| PORTD                 | RD7                   | RD6                   | RD5                 | RD4    | RD3    | RD2    | RD1    | RD0    | xxxx xxxx         | 54, 133         |
| PORTC                 | RC7                   | RC6                   | RC5                 | RC4    | RC3    | RC2    | RC1    | RC0    | xxxx xxxx         | 54, 130         |
| PORTB                 | RB7                   | RB6                   | RB5                 | RB4    | RB3    | RB2    | RB1    | RB0    | xxxx xxxx         | 54, 127         |
| PORTA                 | RA7 <sup>(5)</sup>    | RA6 <sup>(5)</sup>    | RA5                 | RA4    | RA3    | RA2    | RA1    | RA0    | xx0x 0000         | 54, 125         |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Bit 21 of the PC is only available in Test mode and Serial Programming modes.

- 2: These registers and/or bits are available only on 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset states shown are for 80-pin devices.
- 3: Alternate names and definitions for these bits when the MSSP module is operating in I<sup>2</sup>C™ Slave mode. See **Section 16.4.3.2 "Address Masking"** for details.
- 4: The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 2.4.3 "PLL Frequency Multiplier"** for details.
- 5: RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as '0'.

# PIC18F85J11 FAMILY

**TABLE 5-4: PIC18F85J11 FAMILY REGISTER FILE SUMMARY (CONTINUED)**

| 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 |
|-----------|------------------------------------------|-------|-------|-------|--------|--------|--------|--------|-------------------|-----------------|
| SPBRGH1   | EUSART Baud Rate Generator High Byte     |       |       |       |        |        |        |        | 0000 0000         | 54, 221         |
| BAUDCON1  | ABDOVF                                   | RCMT  | —     | SCKP  | BRG16  | —      | WUE    | ABDEN  | 01-0 0-00         | 54, 220         |
| CCPR1H    | Capture/Compare/PWM Register 1 High Byte |       |       |       |        |        |        |        | xxxx xxxx         | 54, 164         |
| CCPR1L    | Capture/Compare/PWM Register 1 Low Byte  |       |       |       |        |        |        |        | xxxx xxxx         | 54, 164         |
| CCP1CON   | —                                        | —     | DC1B1 | DC1B0 | CCP1M3 | CCP1M2 | CCP1M1 | CCP1M0 | --00 0000         | 54, 163         |
| CCPR2H    | Capture/Compare/PWM Register 2 High Byte |       |       |       |        |        |        |        | xxxx xxxx         | 55, 164         |
| CCPR2L    | Capture/Compare/PWM Register 2 Low Byte  |       |       |       |        |        |        |        | xxxx xxxx         | 55, 164         |
| CCP2CON   | —                                        | —     | DC2B1 | DC2B0 | CCP2M3 | CCP2M2 | CCP2M1 | CCP2M0 | --00 0000         | 55, 163         |
| SPBRG2    | AUSART Baud Rate Generator Register      |       |       |       |        |        |        |        | 0000 0000         | 55, 240         |
| RCREG2    | AUSART Receive Register                  |       |       |       |        |        |        |        | 0000 0000         | 55, 245         |
| TXREG2    | AUSART Transmit Register                 |       |       |       |        |        |        |        | 0000 0000         | 55, 243         |
| TXSTA2    | CSRC                                     | TX9   | TXEN  | SYNC  | —      | BRGH   | TRMT   | TX9D   | 0000 -010         | 55, 238         |
| RCSTA2    | SPEN                                     | RX9   | SREN  | CREN  | ADDEN  | FERR   | OERR   | RX9D   | 0000 000x         | 55, 239         |

**Legend:** x = unknown, u = unchanged, - = unimplemented, q = value depends on condition, r = reserved, do not modify

**Note 1:** Bit 21 of the PC is only available in Test mode and Serial Programming modes.

- 2: These registers and/or bits are available only on 80-pin devices; otherwise, they are unimplemented and read as '0'. Reset states shown are for 80-pin devices.
- 3: Alternate names and definitions for these bits when the MSSP module is operating in I<sup>2</sup>C™ Slave mode. See **Section 16.4.3.2 “Address Masking”** for details.
- 4: The PLLEN bit is only available in specific oscillator configurations; otherwise, it is disabled and reads as '0'. See **Section 2.4.3 “PLL Frequency Multiplier”** for details.
- 5: RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as '0'.

# PIC18F85J11 FAMILY

## 5.3.5 STATUS REGISTER

The STATUS register, shown in Register 5-3, contains the arithmetic status of the ALU. The STATUS register can be the operand for any instruction, as with any other register. If the STATUS register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled.

These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the STATUS register as destination may be different than intended. For example, `CLRF STATUS` will set the Z bit but leave the other bits unchanged. The STATUS

register then reads back as '000u u1uu'. It is recommended, therefore, that only `BCF`, `BSF`, `SWAPF`, `MOVFF` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C, DC, OV or N bits in the STATUS register.

For other instructions not affecting any Status bits, see the instruction set summaries in Table 23-2 and Table 23-3.

**Note:** The C and DC bits operate as a borrow and digit borrow bit respectively, in subtraction.

### REGISTER 5-3: STATUS REGISTER

| U-0   | U-0 | U-0 | R/W-x | R/W-x | R/W-x | R/W-x             | R/W-x            |
|-------|-----|-----|-------|-------|-------|-------------------|------------------|
| —     | —   | —   | N     | OV    | Z     | DC <sup>(1)</sup> | C <sup>(2)</sup> |
| bit 7 |     |     |       |       |       |                   | bit 0            |

#### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result was negative (ALU MSB = 1).

1 = Result was negative

0 = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 7-bit magnitude which causes the sign bit (bit 7) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)

0 = No overflow occurred

bit 2 **Z:** Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/Borrow bit<sup>(1)</sup>

For `ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions:

1 = A carry-out from the 4th low-order bit of the result occurred

0 = No carry-out from the 4th low-order bit of the result

bit 0 **C:** Carry/Borrow bit<sup>(2)</sup>

For `ADDWF`, `ADDLW`, `SUBLW` and `SUBWF` instructions:

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

**Note 1:** For borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

**2:** For borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (`RRF`, `RLF`) instructions, this bit is loaded with either the high or low-order bit of the source register.

# PIC18F85J11 FAMILY

## 5.4 Data Addressing Modes

**Note:** The execution of some instructions in the core PIC18 instruction set are changed when the PIC18 extended instruction set is enabled. See **Section 5.6 “Data Memory and the Extended Instruction Set”** for more information.

While the program memory can be addressed in only one way – through the program counter – 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 greater detail in **Section 5.6.1 “Indexed Addressing with Literal Offset”**.

### 5.4.1 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.

### 5.4.2 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 Least Significant Byte. This address specifies either a register address in one of the banks of data RAM (**Section 5.3.3 “General Purpose Register File”**), or a location in the Access Bank (**Section 5.3.2 “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 5.3.1 “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.

### 5.4.3 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 Special Function Registers, 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 5-5. It also enables users to perform Indexed Addressing and other Stack Pointer operations for program memory in data memory.

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

|          |       |                          |
|----------|-------|--------------------------|
|          | LFSR  | FSR0, 100h ;             |
| NEXT     | CLRF  | POSTINC0 ; Clear INDF    |
|          |       | ; register then          |
|          |       | ; inc pointer            |
|          | BTFSS | FSR0H, 1 ; All done with |
|          |       | ; Bank1?                 |
|          | BRA   | NEXT ; NO, clear next    |
| CONTINUE |       | ; YES, continue          |

## 5.4.3.1 FSR Registers and the INDF Operand

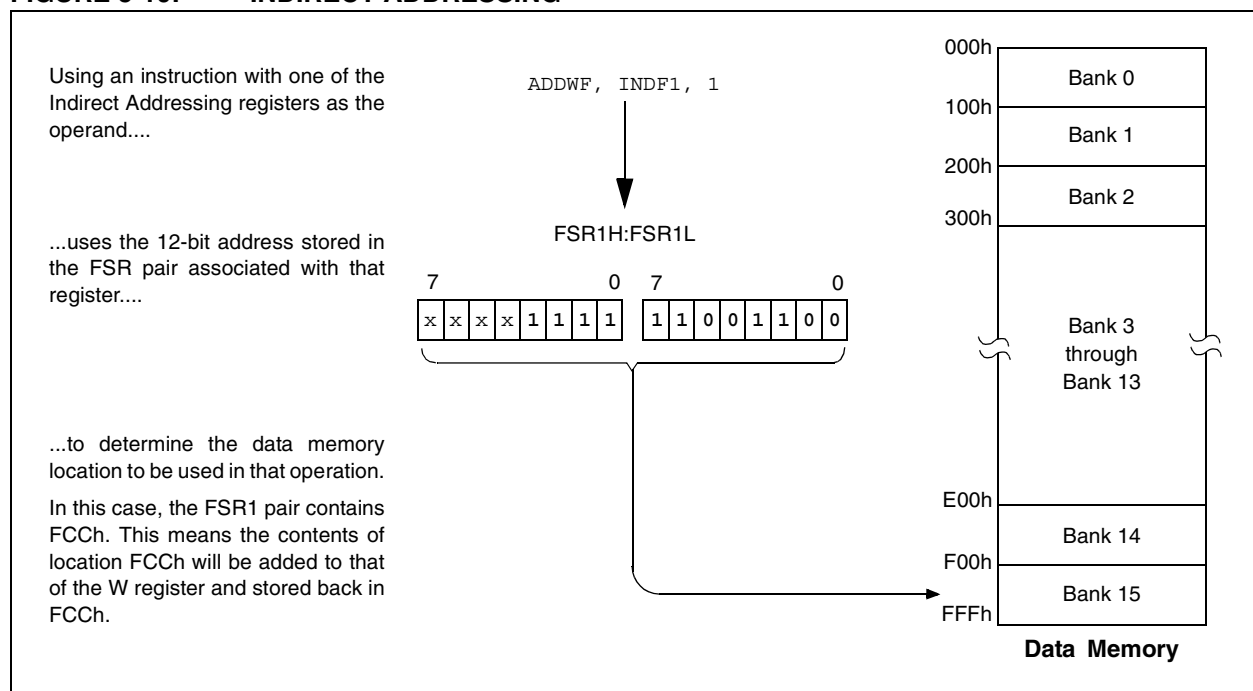
At the core of Indirect Addressing are three sets of registers: FSR0, FSR1 and FSR2. Each represents a pair of 8-bit registers, FSRnH and FSRnL. The four upper bits of the FSRnH register are not used, so each FSR pair holds a 12-bit value. This represents a value that can address the entire range of the data memory in a linear fashion. The FSR register pairs, then, serve as pointers to data memory locations.

Indirect Addressing is accomplished with a set of Indirect File Operands, INDF0 through INDF2. These can be thought of as “virtual” registers: they are mapped in

the SFR space but are not physically implemented. Reading or writing to a particular INDF register actually accesses its corresponding FSR register pair. A read from INDF1, for example, reads the data at the address indicated by FSR1H:FSR1L. Instructions that use the INDF registers as operands actually use the contents of their corresponding FSR as a pointer to the instruction’s target. The INDF operand is just a convenient way of using the pointer.

Because Indirect Addressing uses a full 12-bit address, data RAM banking is not necessary. Thus, the current contents of the BSR and the Access RAM bit have no effect on determining the target address.

**FIGURE 5-10: INDIRECT ADDRESSING**



# PIC18F85J11 FAMILY

---

## 5.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value. They are:

- **POSTDEC**: accesses the FSR value, then automatically decrements it by ‘1’ afterwards
- **POSTINC**: accesses the FSR value, then automatically increments it by ‘1’ afterwards
- **PREINC**: increments the FSR value by ‘1’, then uses it in the operation
- **PLUSW**: adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value offset by the value in the W register; neither value is actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair; that is, roll-overs of the FSRnL register from FFh to 00h carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (e.g., Z, N, OV, etc.).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

## 5.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations. As a specific case, assume that FSR0H:FSR0L contain FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution that they do not inadvertently change settings that might affect the operation of the device.

## 5.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: ADDFSR, CALLW, MOVSF, MOVSS and SUBFSR. These instructions are executed as described in **Section 5.2.4 “Two-Word Instructions”**.



## 5.6 Data Memory and the Extended Instruction Set

Enabling the PIC18 extended instruction set (XINST Configuration bit = 1) significantly changes certain aspects of data memory and its addressing. Specifically, the use of the Access Bank for many of the core PIC18 instructions is different; this is due to the introduction of a new addressing mode for the data memory space. This mode also alters the behavior of Indirect Addressing using FSR2 and its associated operands.

What does not change is just as important. The size of the data memory space is unchanged, as well as its linear addressing. The SFR map remains the same. Core PIC18 instructions can still operate in both Direct and Indirect Addressing mode; inherent and literal instructions do not change at all. Indirect Addressing with FSR0 and FSR1 also remains unchanged.

### 5.6.1 INDEXED ADDRESSING WITH LITERAL OFFSET

Enabling the PIC18 extended instruction set changes the behavior of Indirect Addressing using the FSR2 register pair and its associated file operands. Under the proper conditions, instructions that use the Access Bank – that is, most bit-oriented and byte-oriented instructions – can invoke a form of Indexed Addressing using an offset specified in the instruction. This special addressing mode is known as Indexed Addressing with Literal Offset, or Indexed Literal Offset mode.

When using the extended instruction set, this addressing mode requires the following:

- The use of the Access Bank is forced ('a' = 0); and
- The file address argument is less than or equal to 5Fh.

Under these conditions, the file address of the instruction is not interpreted as the lower byte of an address (used with the BSR in Direct Addressing) or as an 8-bit address in the Access Bank. Instead, the value is interpreted as an offset value to an Address Pointer specified by FSR2. The offset and the contents of FSR2 are added to obtain the target address of the operation.

### 5.6.2 INSTRUCTIONS AFFECTED BY INDEXED LITERAL OFFSET MODE

Any of the core PIC18 instructions that can use Direct Addressing are potentially affected by the Indexed Literal Offset Addressing mode. This includes all byte-oriented and bit-oriented instructions, or almost one-half of the standard PIC18 instruction set. Instructions that only use Inherent or Literal Addressing modes are unaffected.

Additionally, byte-oriented and bit-oriented instructions are not affected if they use the Access Bank (Access RAM bit is '1') or include a file address of 60h or above. Instructions meeting these criteria will continue to execute as before. A comparison of the different possible addressing modes when the extended instruction set is enabled is shown in Figure 5-11.

Those who desire to use byte-oriented or bit-oriented instructions in the Indexed Literal Offset mode should note the changes to assembler syntax for this mode. This is described in more detail in **Section 23.2.1 “Extended Instruction Syntax”**.

# PIC18F85J11 FAMILY

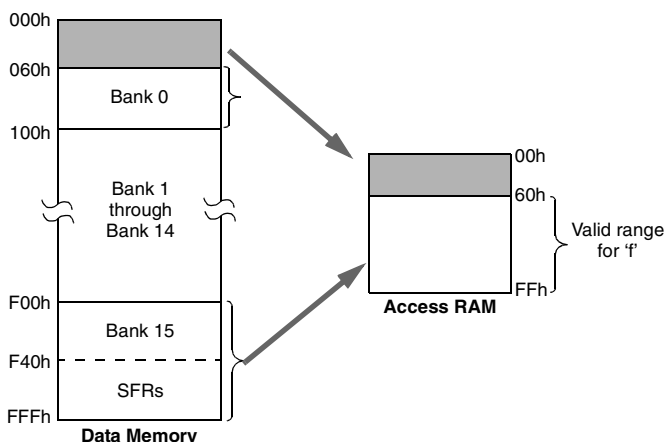
**FIGURE 5-11: COMPARING ADDRESSING OPTIONS FOR BIT-ORIENTED AND BYTE-ORIENTED INSTRUCTIONS (EXTENDED INSTRUCTION SET ENABLED)**

**EXAMPLE INSTRUCTION:** ADDWF, f, d, a (Opcode: 0010 01da ffff ffff)

**When a = 0 and f ≥ 60h:**

The instruction executes in Direct Forced mode. 'f' is interpreted as a location in the Access RAM between 060h and FFFh. This is the same as locations F60h to FFFh (Bank 15) of data memory.

Locations below 060h are not available in this addressing mode.



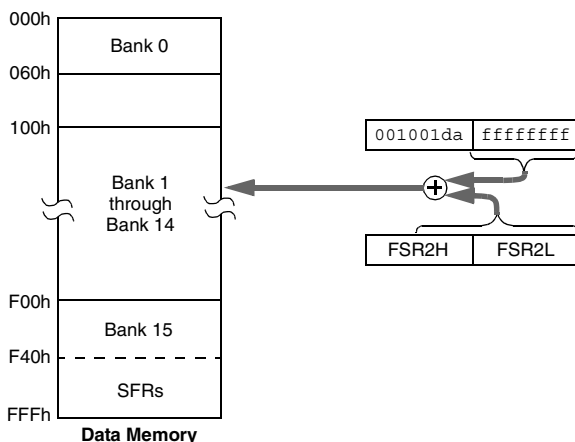
**When a = 0 and f ≤ 5Fh:**

The instruction executes in Indexed Literal Offset mode. 'f' is interpreted as an offset to the address value in FSR2. The two are added together to obtain the address of the target register for the instruction. The address can be anywhere in the data memory space.

Note that in this mode, the correct syntax is now:

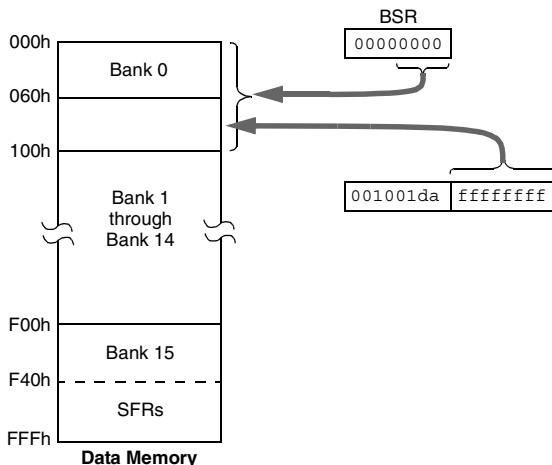
ADDWF [k], d

where 'k' is the same as 'f'.



**When a = 1 (all values of f):**

The instruction executes in Direct mode (also known as Direct Long mode). 'f' is interpreted as a location in one of the 16 banks of the data memory space. The bank is designated by the Bank Select Register (BSR). The address can be in any implemented bank in the data memory space.



## 5.6.3 MAPPING THE ACCESS BANK IN INDEXED LITERAL OFFSET MODE

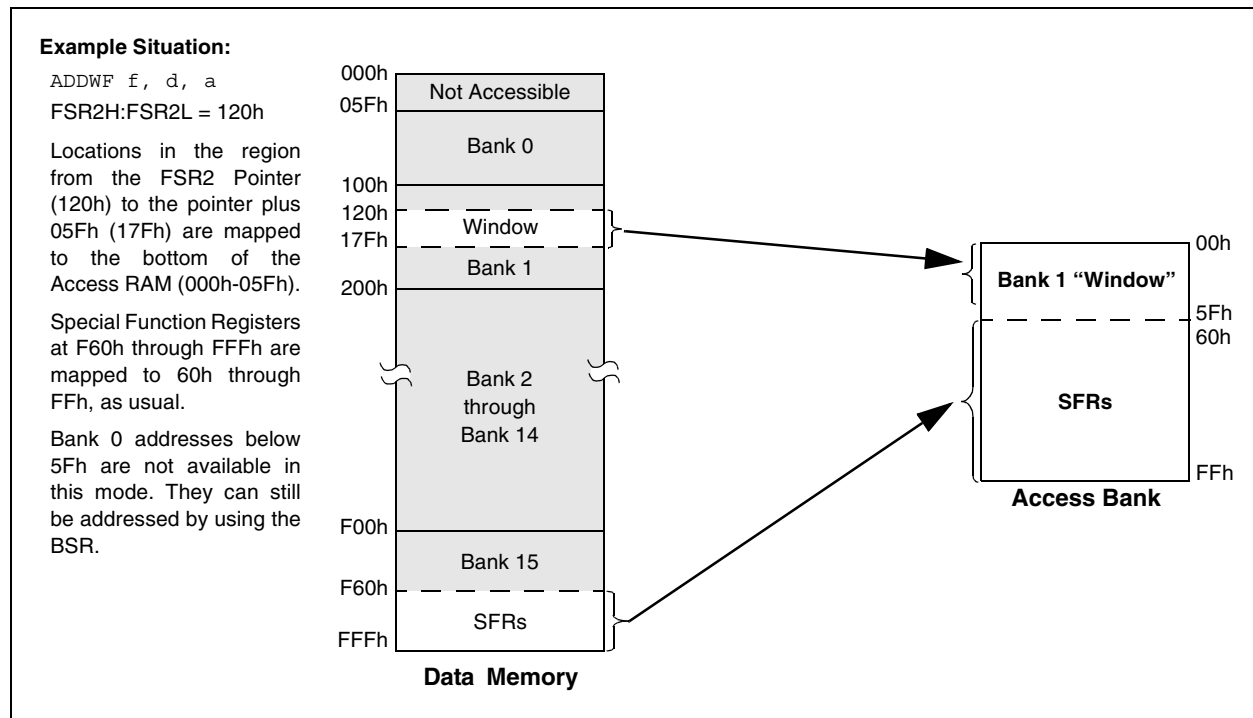
The use of Indexed Literal Offset Addressing mode effectively changes how the lower part of Access RAM (00h to 5Fh) is mapped. Rather than containing just the contents of the bottom part of Bank 0, this mode maps the contents from Bank 0 and a user-defined “window” that can be located anywhere in the data memory space. The value of FSR2 establishes the lower boundary of the addresses mapped into the window, while the upper boundary is defined by FSR2 plus 95 (5Fh). Addresses in the Access RAM above 5Fh are mapped as previously described (see **Section 5.3.2 “Access Bank”**). An example of Access Bank remapping in this addressing mode is shown in Figure 5-12.

Remapping of the Access Bank applies *only* to operations using the Indexed Literal Offset mode. Operations that use the BSR (Access RAM bit is ‘1’) will continue to use Direct Addressing as before. Any Indirect or Indexed Addressing operation that explicitly uses any of the indirect file operands (including FSR2) will continue to operate as standard Indirect Addressing. Any instruction that uses the Access Bank, but includes a register address of greater than 05Fh, will use Direct Addressing and the normal Access Bank map.

## 5.6.4 BSR IN INDEXED LITERAL OFFSET MODE

Although the Access Bank is remapped when the extended instruction set is enabled, the operation of the BSR remains unchanged. Direct Addressing, using the BSR to select the data memory bank, operates in the same manner as previously described.

**FIGURE 5-12: REMAPPING THE ACCESS BANK WITH INDEXED LITERAL OFFSET ADDRESSING**



# PIC18F85J11 FAMILY

---

NOTES:

## 6.0 FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range.

A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 64 bytes at a time. Program memory is erased in blocks of 1024 bytes at a time. A Bulk Erase operation may not be issued from user code.

Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases.

A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

## 6.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

- Table Read (TBLRD)
- Table Write (TBLWT)

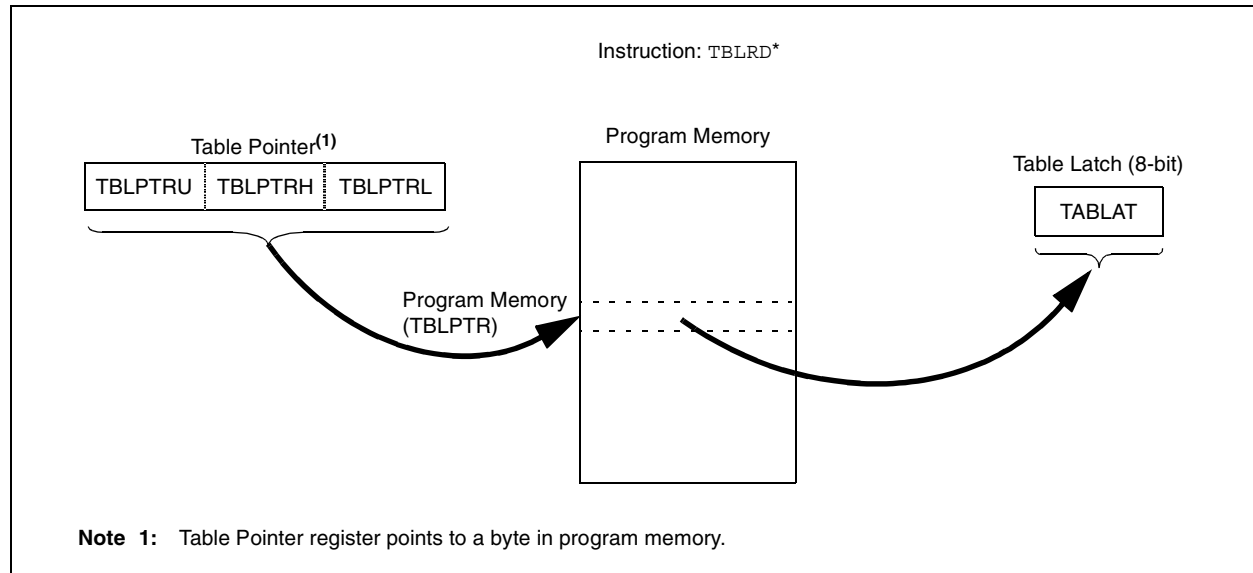
The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8-bit register (TABLAT).

Table read operations retrieve data from program memory and place it into the data RAM space. Figure 6-1 shows the operation of a table read with program memory and data RAM.

Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in **Section 6.5 “Writing to Flash Program Memory”**. Figure 6-2 shows the operation of a table write with program memory and data RAM.

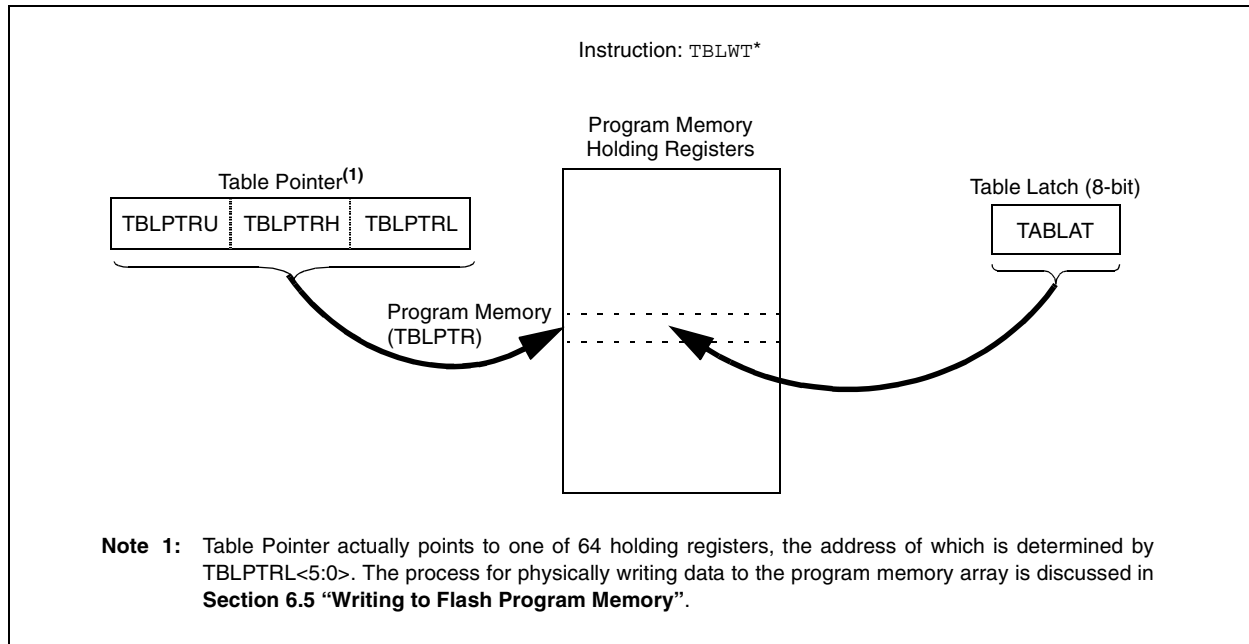
Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word-aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word-aligned.

**FIGURE 6-1: TABLE READ OPERATION**



# PIC18F85J11 FAMILY

**FIGURE 6-2: TABLE WRITE OPERATION**



## 6.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

### 6.2.1 EECON1 AND EECON2 REGISTERS

The EECON1 register (Register 6-1) is the control register for memory accesses. The EECON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading EECON2 will read all '0's.

The FREE bit, when set, will allow a program memory erase operation. When FREE is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled.

The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set in hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is complete.

**Note:** During normal operation, the WRERR is read as '1'. This can indicate that a write operation was prematurely terminated by a Reset, or a write operation was attempted improperly.

The WR control bit initiates write operations. The bit cannot be cleared, only set, in software. It is cleared in hardware at the completion of the write operation.

# PIC18F85J11 FAMILY

## REGISTER 6-1: EECON1: EEPROM CONTROL REGISTER 1

| U-0   | U-0 | U-0 | R/W-0 | R/W-x | R/W-0 | R/S-0 | U-0   |
|-------|-----|-----|-------|-------|-------|-------|-------|
| —     | —   | —   | FREE  | WRERR | WREN  | WR    | —     |
| bit 7 |     |     |       |       |       |       | bit 0 |

|                   |                                    |                                                  |                    |
|-------------------|------------------------------------|--------------------------------------------------|--------------------|
| <b>Legend:</b>    | U = Unimplemented bit, read as '0' |                                                  |                    |
| R = Readable bit  | W = Writable bit                   | S = Set only bit (cannot be cleared in software) |                    |
| -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     **FREE:** Flash Row Erase Enable bit  
           1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)  
           0 = Perform write only
- bit 3     **WRERR:** Flash Program Error Flag bit  
           1 = A write operation is prematurely terminated (any Reset during self-timed programming in normal operation, or an improper write attempt)  
           0 = The write operation completed
- bit 2     **WREN:** Flash Program Write Enable bit  
           1 = Allows write cycles to Flash program memory  
           0 = Inhibits write cycles to Flash program memory
- bit 1     **WR:** Write Control bit  
           1 = Initiates a program memory erase cycle or write cycle  
               (The operation is self-timed and the bit is cleared by hardware once write is complete.  
               The WR bit can only be set (not cleared) in software.)  
           0 = Write cycle is complete
- bit 0     **Unimplemented:** Read as '0'

# PIC18F85J11 FAMILY

## 6.2.2 TABLE LATCH REGISTER (TABLAT)

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

## 6.2.3 TABLE POINTER REGISTER (TBLPTR)

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the device ID, the user ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 6-1. These operations on the TBLPTR only affect the low-order 21 bits.

## 6.2.4 TABLE POINTER BOUNDARIES

TBLPTR is used in reads, writes and erases of the Flash program memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory into TABLAT.

When a TBLWT is executed, the seven LSbs of the Table Pointer register (TBLPTR<6:0>) determine which of the 64 program memory holding registers is written to. When the timed write to program memory begins (via the WR bit), the 12 MSBs of the TBLPTR (TBLPTR<21:10>) determine which program memory block of 1024 bytes is written to. For more detail, see **Section 6.5 “Writing to Flash Program Memory”**.

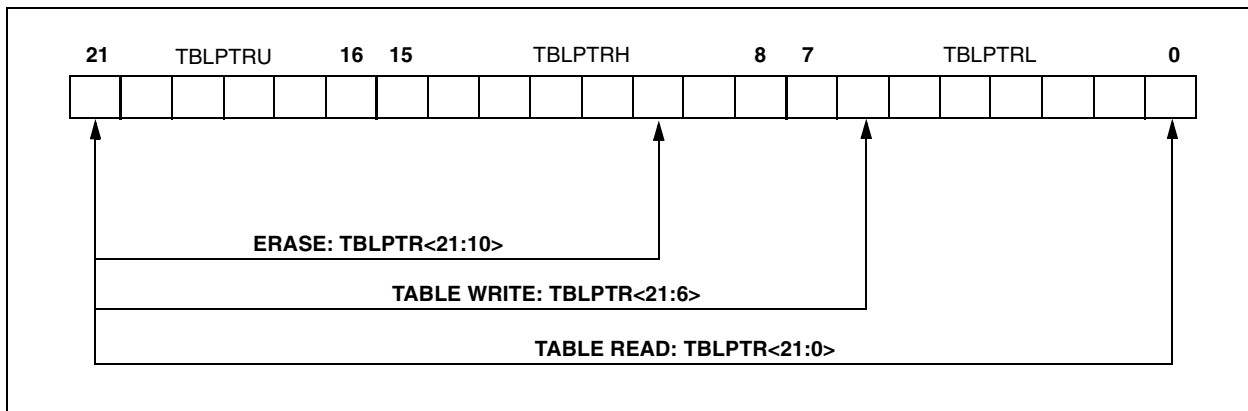
When an erase of program memory is executed, the 12 MSBs of the Table Pointer register point to the 1024-byte block that will be erased. The Least Significant bits are ignored.

Figure 6-3 describes the relevant boundaries of the TBLPTR based on Flash program memory operations.

**TABLE 6-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS**

| Example            | Operation on Table Pointer                  |
|--------------------|---------------------------------------------|
| TBLRD*<br>TBLWT*   | TBLPTR is not modified                      |
| TBLRD*+<br>TBLWT*+ | TBLPTR is incremented after the read/write  |
| TBLRD*-<br>TBLWT*- | TBLPTR is decremented after the read/write  |
| TBLRD+*<br>TBLWT+* | TBLPTR is incremented before the read/write |

**FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION**





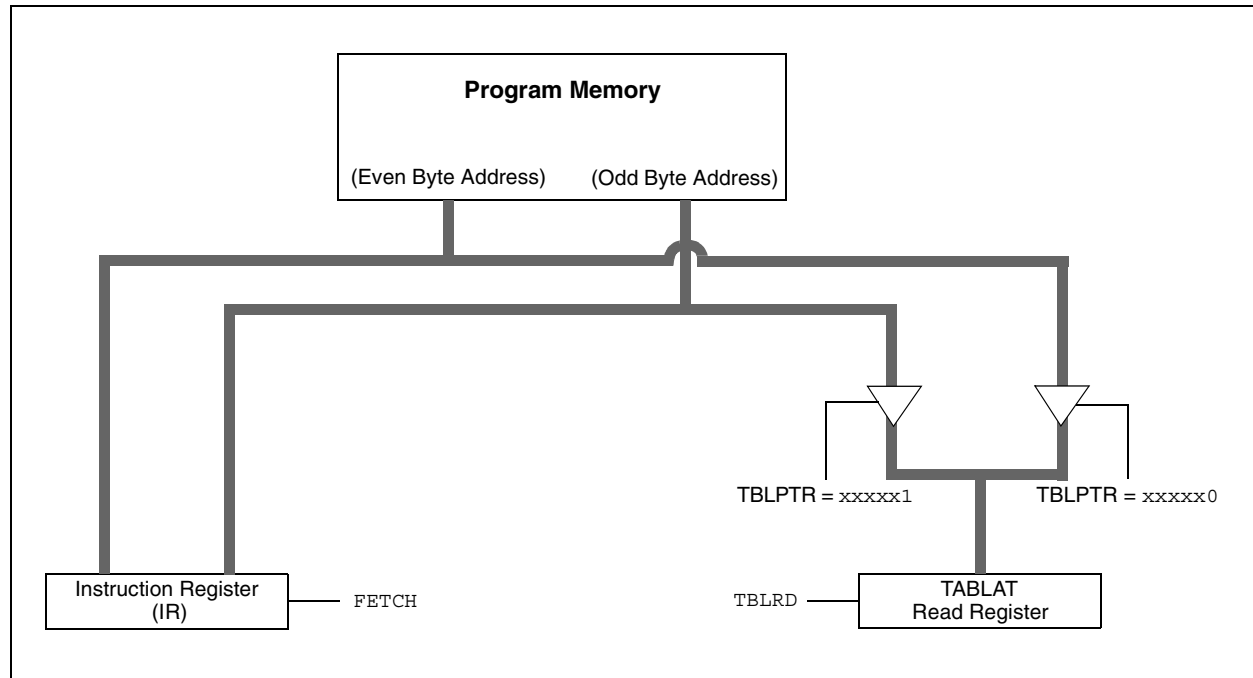
## 6.3 Reading the Flash Program Memory

The `TBLRD` instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time.

`TBLPTR` points to a byte address in program space. Executing `TBLRD` places the byte pointed to into `TABLAT`. In addition, `TBLPTR` can be modified automatically for the next table read operation.

The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 6-4 shows the interface between the internal program memory and the `TABLAT`.

**FIGURE 6-4: READS FROM FLASH PROGRAM MEMORY**



**EXAMPLE 6-1: READING A FLASH PROGRAM MEMORY WORD**

```

        MOVLW    CODE_ADDR_UPPER        ; Load TBLPTR with the base
        MOVWF    TBLPTRU                 ; address of the word
        MOVLW    CODE_ADDR_HIGH
        MOVWF    TBLPTRH
        MOVLW    CODE_ADDR_LOW
        MOVWF    TBLPTRL
READ_WORD
        TBLRD*+                          ; read into TABLAT and increment
        MOVF     TABLAT, W               ; get data
        MOVWF    WORD_EVEN
        TBLRD*+                          ; read into TABLAT and increment
        MOVF     TABLAT, W               ; get data
        MOVF     WORD_ODD
    
```

# PIC18F85J11 FAMILY

## 6.4 Erasing Flash Program Memory

The minimum erase block is 512 words or 1024 bytes. Only through the use of an external programmer, or through ICSP control, can larger blocks of program memory be Bulk Erased. Word Erase in the Flash array is not supported.

When initiating an erase sequence from the microcontroller itself, a block of 1024 bytes of program memory is erased. The Most Significant 12 bits of the TBLPTR<21:10> point to the block being erased; TBLPTR<9:0> are ignored.

The EECON1 register commands the erase operation. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for EECON2 must be used.

A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

### 6.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer register with address of row being erased.
2. Set the WREN and FREE bits (EECON1<2,4>) to enable the erase operation.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the Row Erase cycle.
7. The CPU will stall for duration of the erase for Tiw (see parameter D133A).
8. Re-enable interrupts.

#### EXAMPLE 6-2: ERASING A FLASH PROGRAM MEMORY ROW

|                   |       |                 |                               |
|-------------------|-------|-----------------|-------------------------------|
| ERASE_ROW         | MOVLW | CODE_ADDR_UPPER | ; load TBLPTR with the base   |
|                   | MOVWF | TBLPTRU         | ; address of the memory block |
|                   | MOVLW | CODE_ADDR_HIGH  |                               |
|                   | MOVWF | TBLPTRH         |                               |
|                   | MOVLW | CODE_ADDR_LOW   |                               |
|                   | MOVWF | TBLPTRL         |                               |
|                   | BSF   | EECON1, WREN    | ; enable write to memory      |
|                   | BSF   | EECON1, FREE    | ; enable Row Erase operation  |
|                   | BCF   | INTCON, GIE     | ; disable interrupts          |
|                   |       |                 |                               |
| Required Sequence | MOVLW | 55h             |                               |
|                   | MOVWF | EECON2          | ; write 55h                   |
|                   | MOVLW | 0AAh            |                               |
|                   | MOVWF | EECON2          | ; write 0AAh                  |
|                   | BSF   | EECON1, WR      | ; start erase (CPU stall)     |
|                   | BSF   | INTCON, GIE     | ; re-enable interrupts        |

## 6.5 Writing to Flash Program Memory

The minimum programming block is 32 words or 64 bytes. Word or byte programming is not supported.

Table writes are used internally to load the holding registers needed to program the Flash memory. There are 64 holding registers used by the table writes for programming.

Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction may need to be executed 64 times for each programming operation. All of the table write operations will essentially be short writes because only the holding registers are written. At the end of updating the 64 holding registers, the EECON1 register must be written to in order to start the programming operation with a long write.

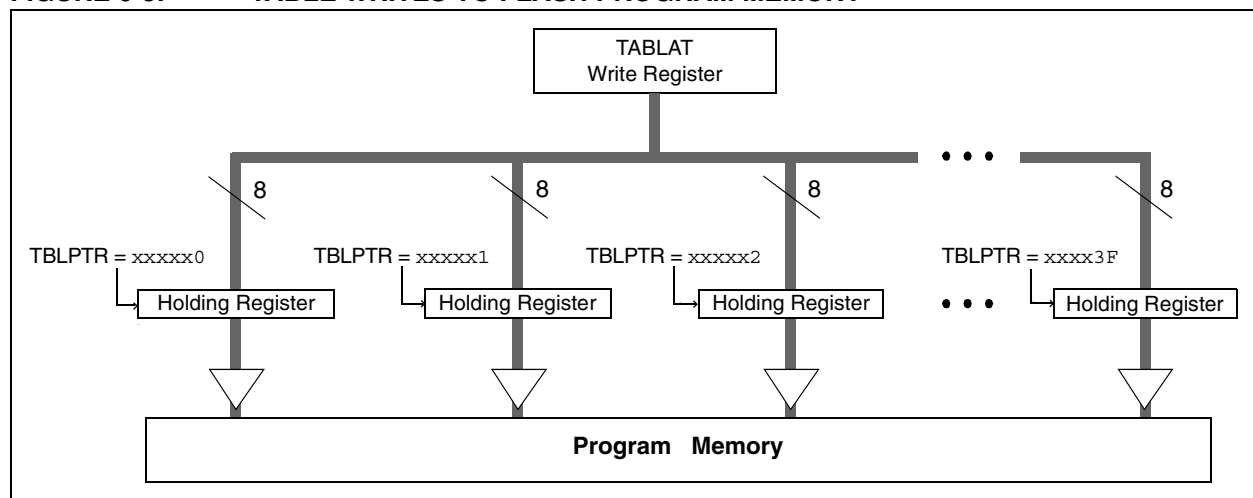
The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

The on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device.

**Note 1:** Unlike previous PIC® MCUs, members of the PIC18F85J11 family do not reset the holding registers after a write occurs. The holding registers must be cleared or overwritten before a programming sequence.

**2:** To maintain the endurance of the program memory cells, each Flash byte should not be programmed more than one time between erase operations. Before attempting to modify the contents of the target cell a second time, a Row Erase of the target row, or a Bulk Erase of the entire memory, must be performed.

**FIGURE 6-5: TABLE WRITES TO FLASH PROGRAM MEMORY**



### 6.5.1 FLASH PROGRAM MEMORY WRITE SEQUENCE

The sequence of events for programming an internal program memory location should be:

1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer register with address being erased.
4. Execute the Row Erase procedure.
5. Load Table Pointer register with address of first byte being written, minus 1.
6. Write the 64 bytes into the holding registers with auto-increment.
7. Set the WREN bit (EECON1<2>) to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write 0AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write for  $T_{iw}$  (see parameter D133A).
13. Re-enable interrupts.
14. Repeat steps 6 through 13 until all 1024 bytes are written to program memory.
15. Verify the memory (table read).

An example of the required code is shown in Example 6-3 on the following page.

**Note:** Before setting the WR bit, the Table Pointer address needs to be within the intended address range of the 64 bytes in the holding register.

\_\_\_\_\_

|                     |         |                     |  |                                          |
|---------------------|---------|---------------------|--|------------------------------------------|
|                     | MOVLW   | CODE_ADDR_UPPER     |  | ; Load TBLPTR with the base address      |
|                     | MOVWF   | TBLPTRU             |  | ; of the memory block, minus 1           |
|                     | MOVLW   | CODE_ADDR_HIGH      |  |                                          |
|                     | MOVWF   | TBLPTRH             |  |                                          |
|                     | MOVLW   | CODE_ADDR_LOW       |  |                                          |
|                     | MOVWF   | TBLPTRL             |  |                                          |
| ERASE_BLOCK         |         |                     |  |                                          |
|                     | BSF     | EECON1, WREN        |  | ; enable write to memory                 |
|                     | BSF     | EECON1, FREE        |  | ; enable Row Erase operation             |
|                     | BCF     | INTCON, GIE         |  | ; disable interrupts                     |
|                     | MOVLW   | 55h                 |  |                                          |
|                     | MOVWF   | EECON2              |  | ; write 55h                              |
|                     | MOVLW   | 0AAh                |  |                                          |
|                     | MOVWF   | EECON2              |  | ; write 0AAh                             |
|                     | BSF     | EECON1, WR          |  | ; start erase (CPU stall)                |
|                     | BSF     | INTCON, GIE         |  | ; re-enable interrupts                   |
|                     | MOVLW   | D'16'               |  |                                          |
|                     | MOVWF   | WRITE_COUNTER       |  | ; Need to write 16 blocks of 64 to write |
|                     |         |                     |  | ; one erase block of 1024                |
| RESTART_BUFFER      |         |                     |  |                                          |
|                     | MOVLW   | D'64'               |  |                                          |
|                     | MOVWF   | COUNTER             |  |                                          |
|                     | MOVLW   | BUFFER_ADDR_HIGH    |  | ; point to buffer                        |
|                     | MOVWF   | FSROH               |  |                                          |
|                     | MOVLW   | BUFFER_ADDR_LOW     |  |                                          |
|                     | MOVWF   | FSROL               |  |                                          |
| FILL_BUFFER         |         |                     |  |                                          |
|                     | ...     |                     |  | ; read the new data from I2C, SPI,       |
|                     |         |                     |  | ; PSP, USART, etc.                       |
| WRITE_BUFFER        |         |                     |  |                                          |
|                     | MOVLW   | D'64                |  | ; number of bytes in holding register    |
|                     | MOVWF   | COUNTER             |  |                                          |
| WRITE_BYTE_TO_HREGS |         |                     |  |                                          |
|                     | MOVFF   | POSTINC0, WREG      |  | ; get low byte of buffer data            |
|                     | MOVWF   | TABLAT              |  | ; present data to table latch            |
|                     | TBLWT+* |                     |  | ; write data, perform a short write      |
|                     |         |                     |  | ; to internal TBLWT holding register.    |
|                     | DECFSZ  | COUNTER             |  | ; loop until buffers are full            |
|                     | BRA     | WRITE_BYTE_TO_HREGS |  |                                          |
| PROGRAM_MEMORY      |         |                     |  |                                          |
|                     | BSF     | EECON1, WREN        |  | ; enable write to memory                 |
|                     | BCF     | INTCON, GIE         |  | ; disable interrupts                     |
|                     | MOVLW   | 55h                 |  |                                          |
| Required Sequence   | MOVWF   | EECON2              |  | ; write 55h                              |
|                     | MOVLW   | 0AAh                |  |                                          |
|                     | MOVWF   | EECON2              |  | ; write 0AAh                             |
|                     | BSF     | EECON1, WR          |  | ; start program (CPU stall)              |
|                     | BSF     | INTCON, GIE         |  | ; re-enable interrupts                   |
|                     | BCF     | EECON1, WREN        |  | ; disable write to memory                |
|                     | DECFSZ  | WRITE_COUNTER       |  | ; done with one write cycle              |
|                     | BRA     | RESTART_BUFFER      |  | ; if not done replacing the erase block  |

## 6.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

## 6.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. If the write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation, the user can check the WRERR bit and rewrite the location(s) as needed.

## 6.6 Flash Program Operation During Code Protection

See **Section 22.6 “Program Verification and Code Protection”** for details on code protection of Flash program memory.

**TABLE 6-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY**

| Name    | Bit 7                                                 | Bit 6     | Bit 5  | Bit 4                                                   | Bit 3 | Bit 2  | Bit 1  | Bit 0 | Reset Values on page |
|---------|-------------------------------------------------------|-----------|--------|---------------------------------------------------------|-------|--------|--------|-------|----------------------|
| TBLPTRU | —                                                     | —         | bit 21 | Program Memory Table Pointer Upper Byte (TBLPTR<20:16>) |       |        |        |       | 51                   |
| TBPLTRH | Program Memory Table Pointer High Byte (TBLPTR<15:8>) |           |        |                                                         |       |        |        |       | 51                   |
| TBLPTRL | Program Memory Table Pointer Low Byte (TBLPTR<7:0>)   |           |        |                                                         |       |        |        |       | 51                   |
| TABLAT  | Program Memory Table Latch                            |           |        |                                                         |       |        |        |       | 51                   |
| INTCON  | GIE/GIEH                                              | PEIE/GIEL | TMR0IE | INT0IE                                                  | RBIE  | TMR0IF | INT0IF | RBIF  | 51                   |
| EECON2  | EEPROM Control Register 2 (not a physical register)   |           |        |                                                         |       |        |        |       | 53                   |
| EECON1  | —                                                     | —         | —      | FREE                                                    | WRERR | WREN   | WR     | —     | 53                   |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used during program memory access.

# PIC18F85J11 FAMILY

---

NOTES:

# PIC18F85J11 FAMILY

## 7.0 EXTERNAL MEMORY BUS

**Note:** The external memory bus is not implemented on 64-pin devices.

The external memory bus allows the device to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program or data memory. It supports both 8 and 16-Bit Data Width modes and three address widths of up to 20 bits.

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

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

**TABLE 7-1: PIC18F85J11 FAMILY EXTERNAL BUS – I/O PORT FUNCTIONS**

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

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

# PIC18F85J11 FAMILY

## 7.1 External Memory Bus Control

The operation of the interface is controlled by the MEMCON register (Register 7-1). This register is available in all Program Memory modes except Microcontroller mode. In this mode, the register is disabled and cannot be written to.

The EBDIS bit (MEMCON<7>) controls the operation of the bus and related port functions. Clearing EBDIS enables the interface and disables the I/O functions of the ports, as well as any other functions multiplexed to those pins. Setting the bit enables the I/O ports and other functions, but allows the interface to override everything else on the pins when an external memory operation is required. By default, the external bus is always enabled and disables all other I/O.

The operation of the EBDIS bit is also influenced by the Program Memory mode being used. This is discussed in more detail in **Section 7.5 “Program Memory Modes and the External Memory Bus”**.

The WAIT bits allow for the addition of wait states to external memory operations. The use of these bits is discussed in **Section 7.3 “Wait States”**.

The WM bits select the particular operating mode used when the bus is operating in 16-Bit Data Width mode. These are discussed in more detail in **Section 7.6 “16-Bit Data Width Modes”**. These bits have no effect when an 8-Bit Data Width mode is selected.

**REGISTER 7-1: MEMCON: EXTERNAL MEMORY BUS CONTROL REGISTER**

| R/W-0  | U-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 |
|--------|-----|-------|-------|-----|-----|-------|-------|
| EBDIS  | —   | WAIT1 | WAIT0 | —   | —   | WM1   | WM0   |
| bit 15 |     |       |       |     |     |       | bit 8 |

**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      **EBDIS:** External Bus Disable bit  
1 = External bus enabled when microcontroller accesses external memory; otherwise, all external bus drivers are mapped as I/O ports  
0 = External bus always enabled, I/O ports are disabled
- bit 6      **Unimplemented:** Read as '0'
- bit 5-4    **WAIT1:WAIT0:** Table Reads and Writes Bus Cycle Wait Count bits  
11 = Table reads and writes will wait 0 Tcy  
10 = Table reads and writes will wait 1 Tcy  
01 = Table reads and writes will wait 2 Tcy  
00 = Table reads and writes will wait 3 Tcy
- bit 3-2    **Unimplemented:** Read as '0'
- bit 1-0    **WM1:WM0:** TBLWT Operation with 16-Bit Data Bus Width Select bits  
1x = Word Write mode: TABLAT0 and TABLAT1 word output,  $\overline{\text{WRH}}$  active when TABLAT1 written  
01 = Byte Select mode: TABLAT data copied on both MSB and LSB,  $\overline{\text{WRH}}$  and ( $\overline{\text{UB}}$  or  $\overline{\text{LB}}$ ) will activate  
00 = Byte Write mode: TABLAT data copied on both MSB and LSB,  $\overline{\text{WRH}}$  or  $\overline{\text{WRL}}$  will activate



## 7.2 Address and Data Width

The PIC18F85J11 family of devices can be independently configured for different address and data widths on the same memory bus. Both address and data width are set by Configuration bits in the CONFIG3L register. As Configuration bits, this means that these options can only be configured by programming the device and are not controllable in software.

The BW bit selects an 8-bit or 16-bit data bus width. Setting this bit (default) selects a data width of 16 bits.

The EMB1:EMB0 bits determine both the Program Memory mode and the address bus width. The available options are 20-bit, 16-bit and 12-bit, as well as the default Microcontroller mode (external bus disabled). Selecting a 16-bit or 12-bit width makes a corresponding number of high-order lines available for I/O functions; these pins are no longer affected by the setting of the EBDIS bit. For example, selecting a 16-Bit Address mode (EMB1:EMB0 = 01) disables A19:A16 and allows PORTH<3:0> to function without interruptions from the bus. Using the smaller address widths allows users to tailor the memory bus to the size of the external memory space for a particular design while freeing up pins for dedicated I/O operation.

Because the EMB bits have the effect of disabling pins for memory bus operations, it is important to always select an address width at least equal to the data width. If a 12-bit address width is used with a 16-bit data width, the upper four bits of data will not be available on the bus.

All combinations of address and data widths require multiplexing of address and data information on the same lines. The address and data multiplexing, as well as I/O ports made available by the use of smaller address widths, are summarized in Table 7-2.

### 7.2.1 ADDRESS SHIFTING ON THE EXTERNAL BUS

By default, the address presented on the external bus is the value of the PC. In practical terms, this means that addresses in the external memory device below the top of on-chip memory are unavailable to the microcontroller. To access these physical locations, the glue logic between the microcontroller and the external memory must somehow translate addresses.

To simplify the interface, the external bus offers an extension of Extended Microcontroller mode that automatically performs address shifting. This feature is controlled by the EASHFT Configuration bit. Setting this bit offsets addresses on the bus by the size of the microcontroller's on-chip program memory and sets the bottom address at 0000h. This allows the device to use the entire range of physical addresses of the external memory.

### 7.2.2 21-BIT ADDRESSING

As an extension of 20-bit address width operation, the external memory bus can also fully address a 2-Mbyte memory space. This is done by using the Bus Address bit 0 (BA0) control line as the Least Significant bit of the address. The  $\overline{UB}$  and  $\overline{LB}$  control signals may also be used with certain memory devices to select the upper and lower bytes within a 16-bit wide data word.

This addressing mode is available in both 8-bit and certain 16-Bit Data Width modes. Additional details are provided in **Section 7.6.3 "16-Bit Byte Select Mode"** and **Section 7.7 "8-Bit Data Width Mode"**.

**TABLE 7-2: ADDRESS AND DATA LINES FOR DIFFERENT ADDRESS AND DATA WIDTHS**

| Data Width | Address Width | Multiplexed Data and Address Lines (and Corresponding Ports) | Address-Only Lines (and Corresponding Ports)     | Ports Available for I/O     |
|------------|---------------|--------------------------------------------------------------|--------------------------------------------------|-----------------------------|
| 8-bit      | 12-bit        | AD7:AD0<br>(PORTD<7:0>)                                      | AD11:AD8<br>(PORTE<3:0>)                         | PORTE<7:4>,<br>All of PORTH |
|            | 16-bit        |                                                              | AD15:AD8<br>(PORTE<7:0>)                         | All of PORTH                |
|            | 20-bit        |                                                              | A19:A16, AD15:AD8<br>(PORTH<3:0>,<br>PORTE<7:0>) | —                           |
| 16-bit     | 16-bit        | AD15:AD0<br>(PORTD<7:0>,<br>PORTE<7:0>)                      | —                                                | All of PORTH                |
|            | 20-bit        |                                                              | A19:A16<br>(PORTH<3:0>)                          | —                           |

# PIC18F85J11 FAMILY

## 7.3 Wait States

While it may be assumed that external memory devices will operate at the microcontroller clock rate, this is often not the case. In fact, many devices require longer times to write or retrieve data than the time allowed by the execution of table read or table write operations.

To compensate for this, the external memory bus can be configured to add a fixed delay to each table operation using the bus. Wait states are enabled by setting the WAIT Configuration bit. When enabled, the amount of delay is set by the WAIT1:WAIT0 bits (MEMCON<5:4>). The delay is based on multiples of microcontroller instruction cycle time and are added following the instruction cycle when the table operation is executed. The range is from no delay to 3 Tcy (default value).

## 7.4 Port Pin Weak Pull-ups

With the exception of the upper address lines, A19:A16, the pins associated with the external memory bus are equipped with weak pull-ups. The pull-ups are controlled by the upper three bits of the PORTG register. They are named RDPU, REPU and RJPU and control pull-ups on PORTD, PORTE and PORTJ, respectively. Setting one of these bits enables the corresponding pull-ups for that port. All pull-ups are disabled by default on all device Resets.

## 7.5 Program Memory Modes and the External Memory Bus

The PIC18F85J11 family of devices are capable of operating in one of two Program Memory modes, using combinations of on-chip and external program memory. The functions of the multiplexed port pins depend on the Program Memory mode selected, as well as the setting of the EBDIS bit.

In **Microcontroller Mode**, the bus is not active and the pins have their port functions only. Writes to the MEMCOM register are not permitted. The Reset value of EBDIS ('0') is ignored and EMB pins behave as I/O ports.

In **Extended Microcontroller Mode**, the external program memory bus shares I/O port functions on the pins. When the device is fetching or doing table read/table write operations on the external program memory space, the pins will have the external bus function.

If the device is fetching and accessing internal program memory locations only, the EBDIS control bit will change the pins from external memory to I/O port functions. When EBDIS = 0, the pins function as the external bus. When EBDIS = 1, the pins function as I/O ports.

If the device fetches or accesses external memory while EBDIS = 1, the pins will switch to the external bus. If the EBDIS bit is set by a program executing from external memory, the action of setting the bit will be delayed until the program branches into the internal memory. At that time, the pins will change from external bus to I/O ports.

If the device is executing out of internal memory when EBDIS = 0, the memory bus address/data and control pins will not be active. They will go to a state where the active address/data pins are tri-state; the  $\overline{CE}$ ,  $\overline{OE}$ ,  $\overline{WRH}$ ,  $\overline{WRL}$ ,  $\overline{UB}$  and  $\overline{LB}$  signals are '1' and ALE and BA0 are '0'. Note that only those pins associated with the current address width are forced to tri-state; the other pins continue to function as I/O. In the case of 16-bit address width, for example, only AD<15:0> (PORTD and PORTE) are affected; A19:A16 (PORTH<3:0>) continue to function as I/O.

In all External Memory modes, the bus takes priority over any other peripherals that may share pins with it. This includes the Parallel Slave Port and serial communication modules which would otherwise take priority over the I/O port.

## 7.6 16-Bit Data Width Modes

In 16-Bit Data Width mode, the external memory interface can be connected to external memories in three different configurations:

- 16-Bit Byte Write
- 16-Bit Word Write
- 16-Bit Byte Select

The configuration to be used is determined by the WM1:WM0 bits in the MEMCON register (MEMCON<1:0>). These three different configurations allow the designer maximum flexibility in using both 8-bit and 16-bit devices with 16-bit data.

For all 16-Bit Data Width modes, the Address Latch Enable (ALE) pin indicates that the address bits, AD<15:0>, are available on the external memory interface bus. Following the address latch, the Output Enable signal ( $\overline{OE}$ ) will enable both bytes of program memory at once to form a 16-bit instruction word. The Chip Enable signal ( $\overline{CE}$ ) is active at any time that the microcontroller accesses external memory, whether reading or writing. It is inactive (asserted high) whenever the device is in Sleep mode.

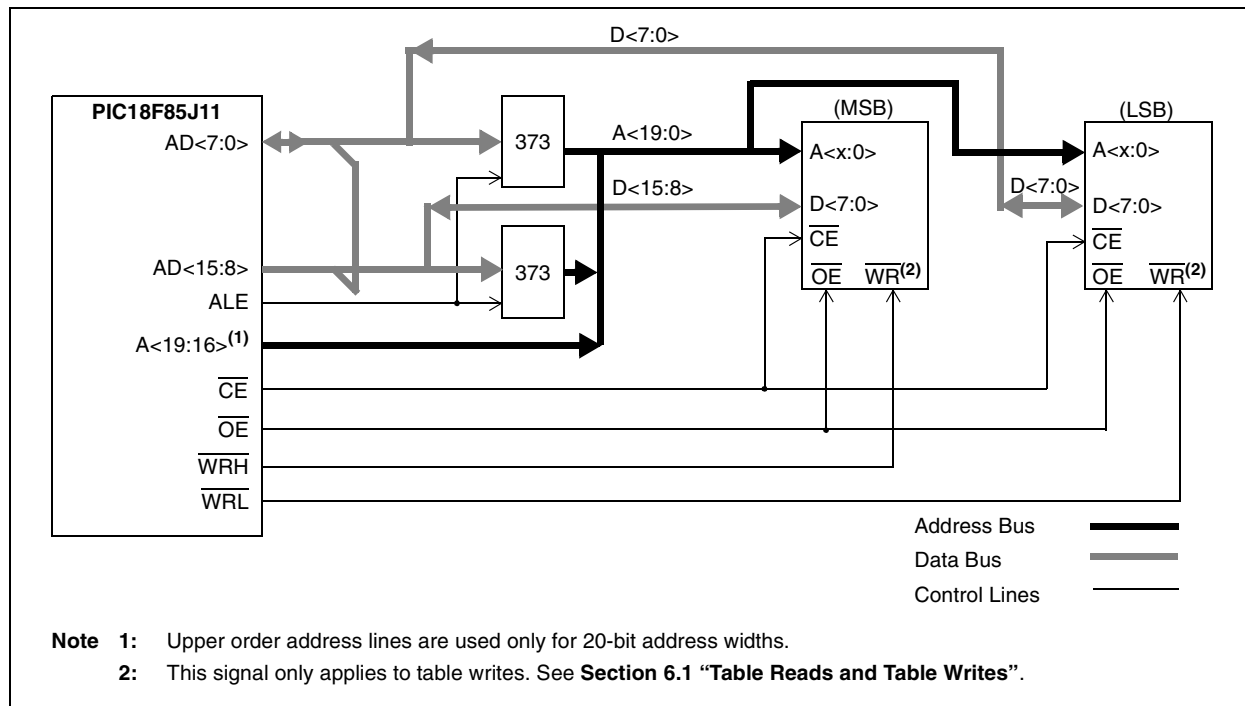
In Byte Select mode, JEDEC standard Flash memories will require BA0 for the byte address line and one I/O line to select between Byte and Word mode. The other 16-Bit Data Width modes do not need BA0. JEDEC standard static RAM memories will use the  $\overline{UB}$  or  $\overline{LB}$  signals for byte selection.

## 7.6.1 16-BIT BYTE WRITE MODE

Figure 7-1 shows an example of 16-Bit Byte Write mode for PIC18F85J11 family devices. This mode is used for two separate 8-bit memories connected for 16-bit operation. This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a TBLWT instruction cycle, the TABLAT data is presented on the upper and lower bytes of the AD15:AD0 bus. The appropriate WRH or WRL control line is strobed on the LSB of the TBLPTR.

**FIGURE 7-1: 16-BIT BYTE WRITE MODE EXAMPLE**



# PIC18F85J11 FAMILY

## 7.6.2 16-BIT WORD WRITE MODE

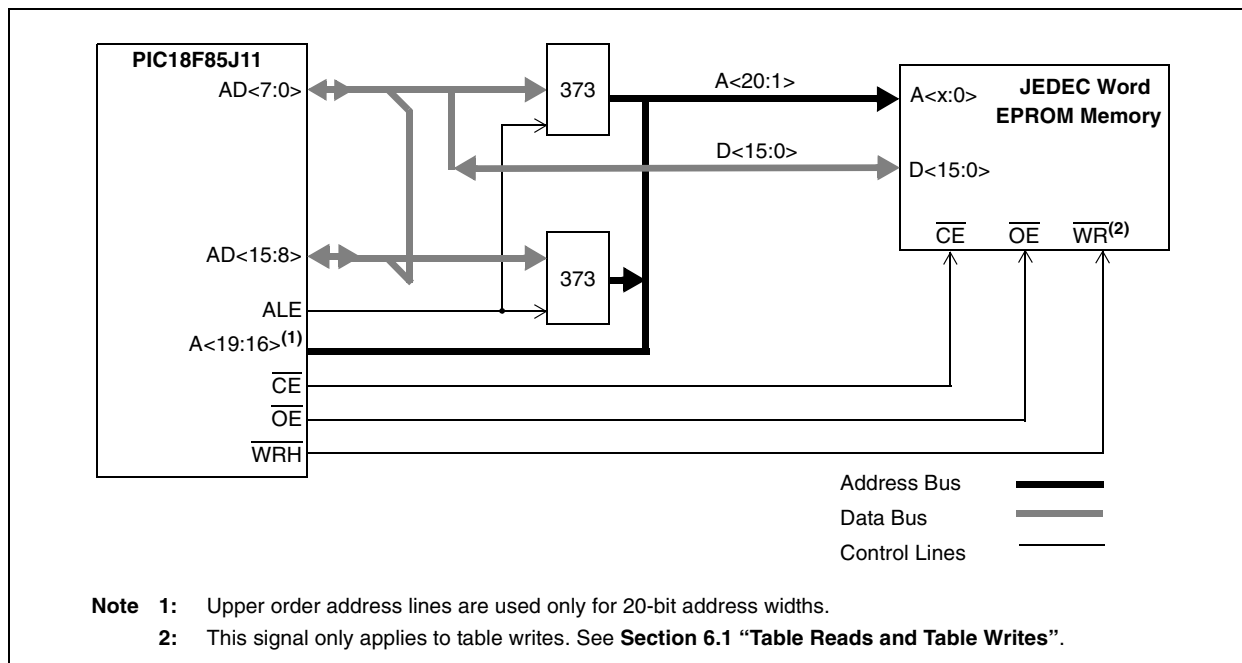
Figure 7-2 shows an example of 16-Bit Word Write mode for PIC18F85J11 family devices. This mode is used for word-wide memories which include some of the EPROM and Flash-type memories. This mode allows opcode fetches and table reads from all forms of 16-bit memory and table writes to any type of word-wide external memories. This method makes a distinction between TBLWT cycles to even or odd addresses.

During a TBLWT cycle to an even address (TBLPTR<0> = 0), the TABLAT data is transferred to a holding latch and the external address data bus is tri-stated for the data portion of the bus cycle. No write signals are activated.

During a TBLWT cycle to an odd address (TBLPTR<0> = 1), the TABLAT data is presented on the upper byte of the AD15:AD0 bus. The contents of the holding latch are presented on the lower byte of the AD15:AD0 bus.

The  $\overline{\text{WRH}}$  signal is strobed for each write cycle; the  $\overline{\text{WRL}}$  pin is unused. The signal on the BA0 pin indicates the LSB of the TBLPTR, but it is left unconnected. Instead, the  $\overline{\text{UB}}$  and  $\overline{\text{LB}}$  signals are active to select both bytes. The obvious limitation to this method is that the table write must be done in pairs on a specific word boundary to correctly write a word location.

**FIGURE 7-2: 16-BIT WORD WRITE MODE EXAMPLE**



# PIC18F85J11 FAMILY

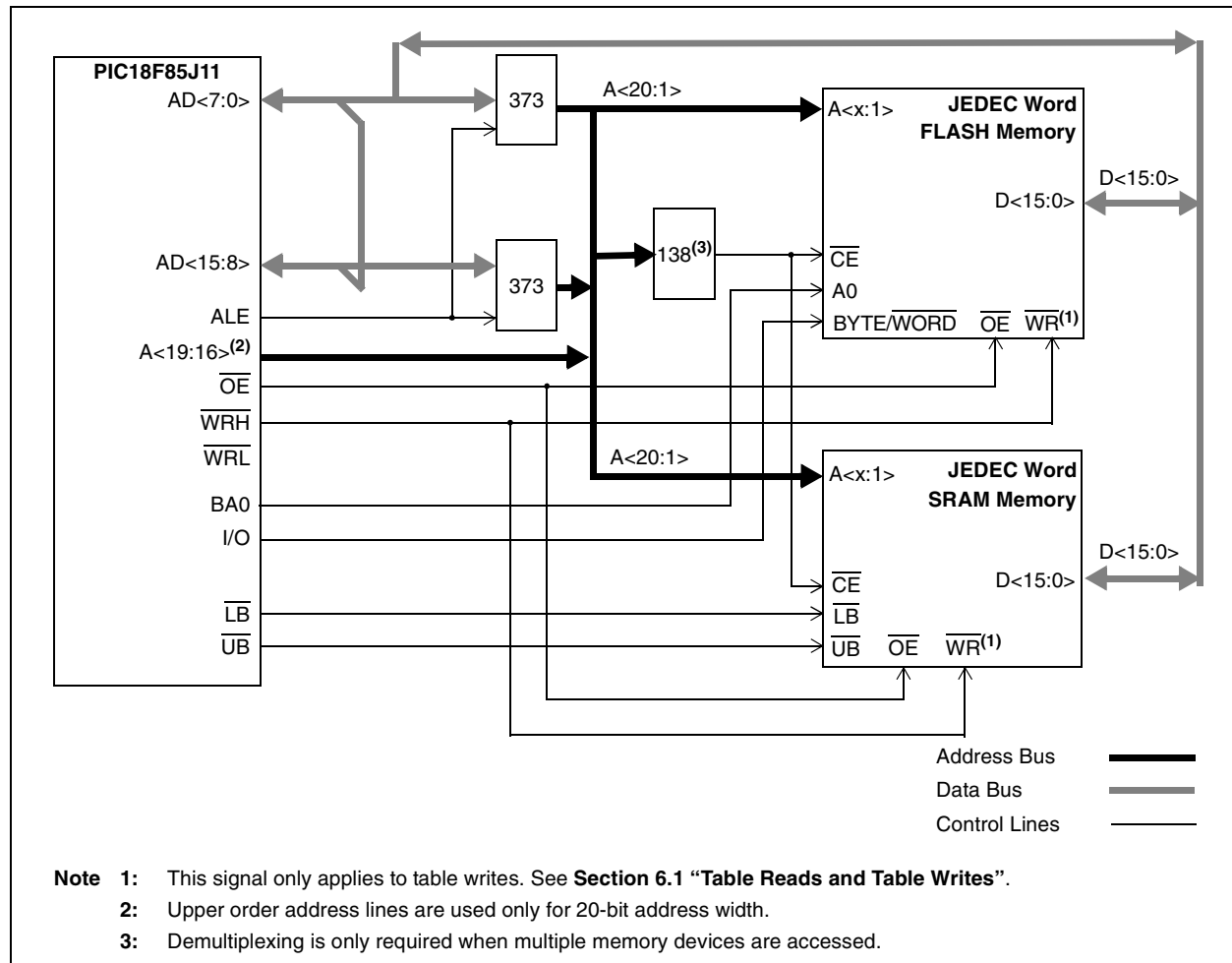
## 7.6.3 16-BIT BYTE SELECT MODE

Figure 7-3 shows an example of 16-Bit Byte Select mode. This mode allows table write operations to word-wide external memories with byte selection capability. This generally includes both word-wide Flash and SRAM devices.

During a TBLWT cycle, the TABLAT data is presented on the upper and lower byte of the AD15:AD0 bus. The  $\overline{\text{WRH}}$  signal is strobed for each write cycle; the  $\overline{\text{WRL}}$  pin is not used. The BA0 or  $\overline{\text{UB}}/\overline{\text{LB}}$  signals are used to select the byte to be written, based on the Least Significant bit of the TBLPTR register.

Flash and SRAM devices use different control signal combinations to implement Byte Select mode. JEDEC standard Flash memories require that a controller I/O port pin be connected to the memory's BYTE/WORD pin to provide the select signal. They also use the BA0 signal from the controller as a byte address. JEDEC standard static RAM memories, on the other hand, use the  $\overline{\text{UB}}$  or  $\overline{\text{LB}}$  signals to select the byte.

**FIGURE 7-3: 16-BIT BYTE SELECT MODE EXAMPLE**



# PIC18F85J11 FAMILY

## 7.6.4 16-BIT MODE TIMING

The presentation of control signals on the external memory bus is different for the various operating modes. Typical signal timing diagrams are shown in Figure 7-4 and Figure 7-5.

**FIGURE 7-4: EXTERNAL MEMORY BUS TIMING FOR TBLRD (EXTENDED MICROCONTROLLER MODE)**

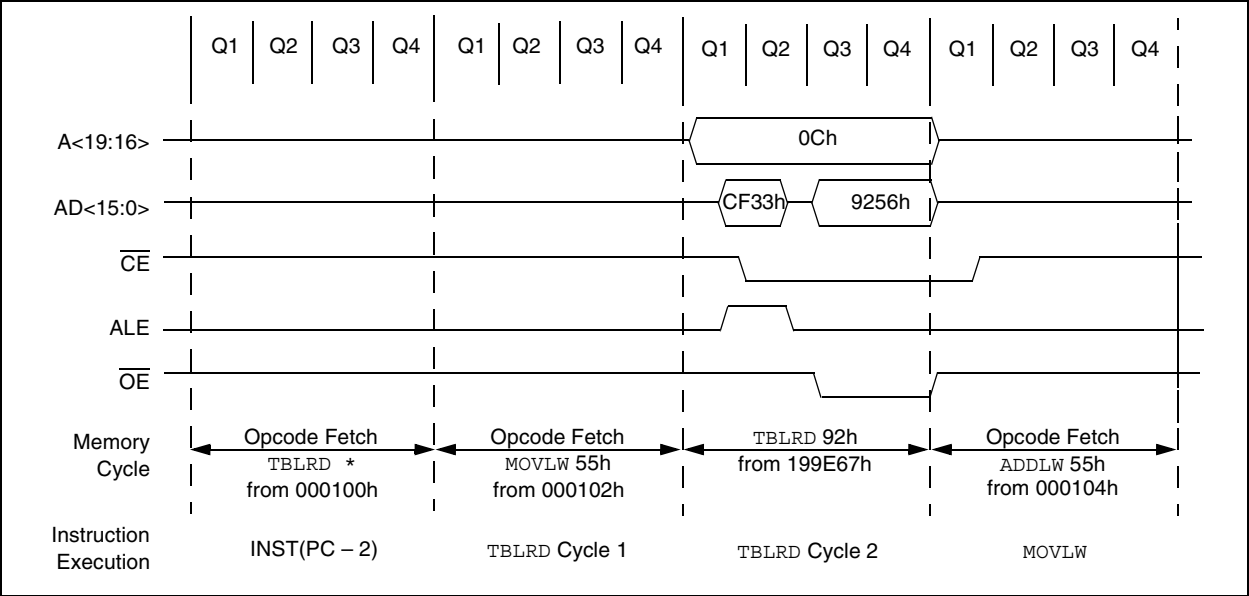
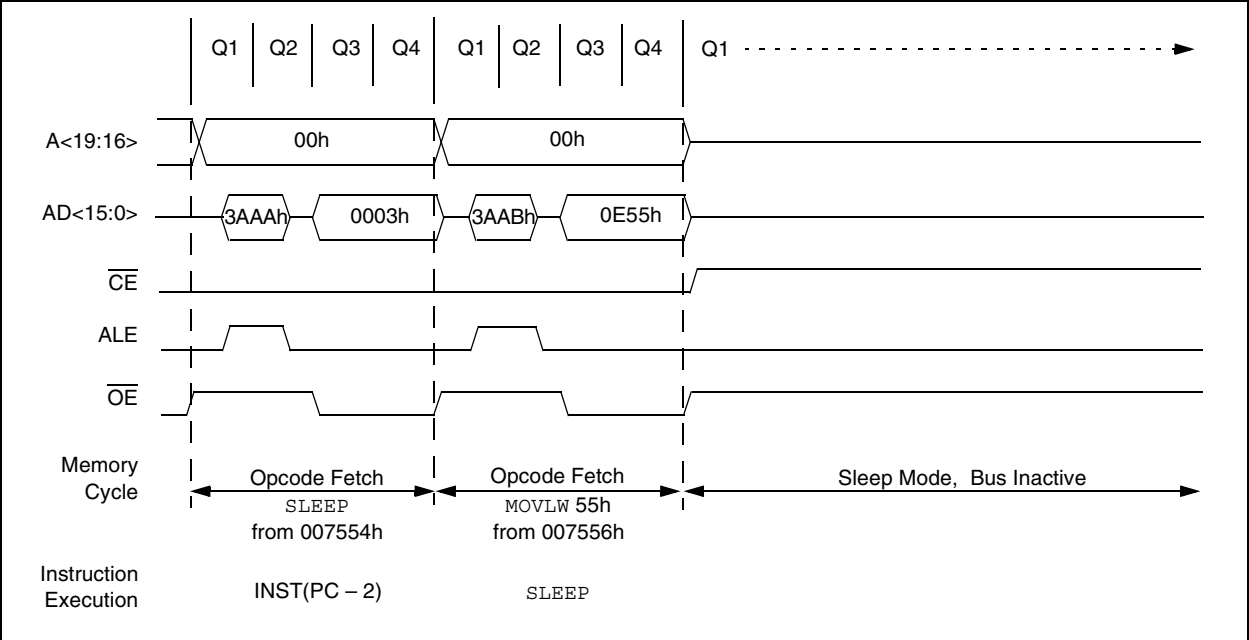


**FIGURE 7-5: EXTERNAL MEMORY BUS TIMING FOR SLEEP (EXTENDED MICROCONTROLLER MODE)**



## 7.7 8-Bit Data Width Mode

In 8-Bit Data Width mode, the external memory bus operates only in Multiplexed mode; that is, data shares the 8 Least Significant bits of the address bus.

Figure 7-6 shows an example of 8-Bit Multiplexed mode for 80-pin devices. This mode is used for a single 8-bit memory connected for 16-bit operation. The instructions will be fetched as two 8-bit bytes on a shared data/address bus. The two bytes are sequentially fetched within one instruction cycle ( $T_{CY}$ ). Therefore, the designer must choose external memory devices according to timing calculations based on  $1/2 T_{CY}$  (2 times the instruction rate). For proper memory speed selection, glue logic propagation delay times must be considered, along with setup and hold times.

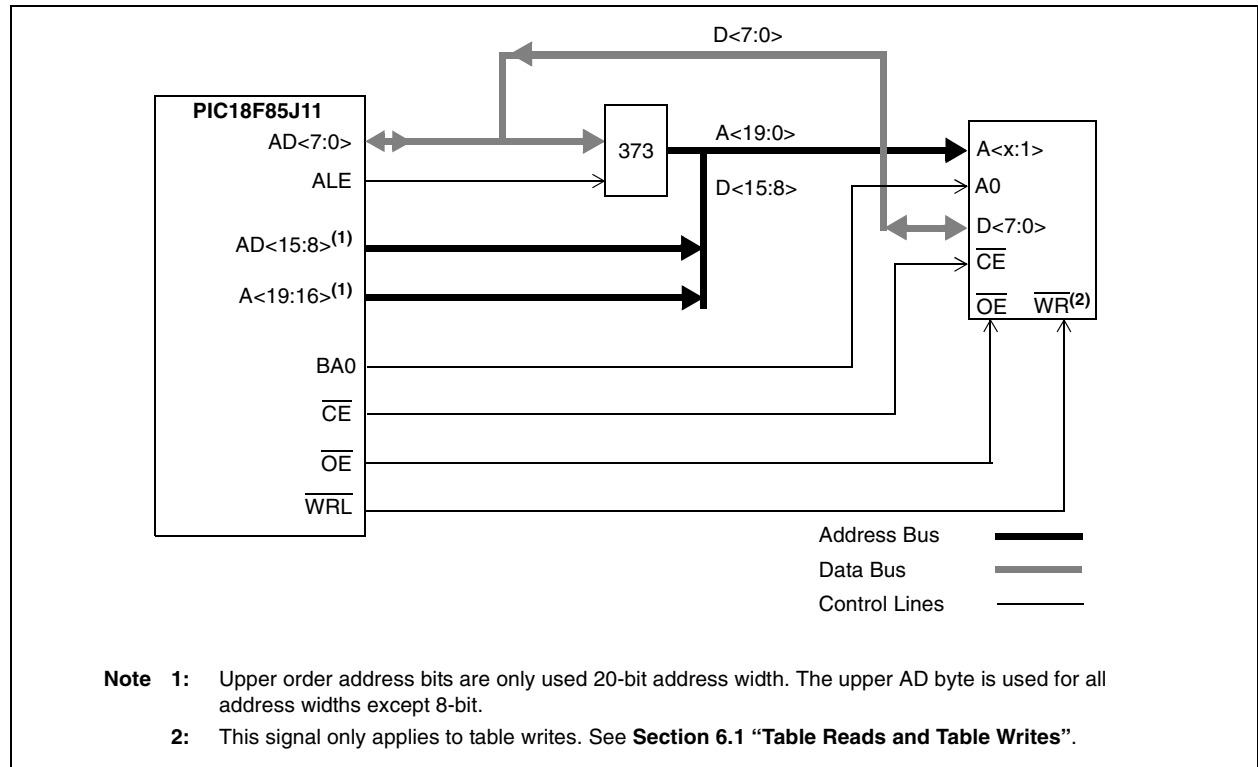
The Address Latch Enable (ALE) pin indicates that the address bits,  $AD<15:0>$ , are available on the external memory interface bus. The Output Enable signal ( $\overline{OE}$ )

will enable one byte of program memory for a portion of the instruction cycle, then  $BA0$  will change and the second byte will be enabled to form the 16-bit instruction word. The Least Significant bit of the address,  $BA0$ , must be connected to the memory devices in this mode. The Chip Enable signal ( $\overline{CE}$ ) is active at any time that the microcontroller accesses external memory, whether reading or writing. It is inactive (asserted high) whenever the device is in Sleep mode.

This generally includes basic EPROM and Flash devices. It allows table writes to byte-wide external memories.

During a  $TBLWT$  instruction cycle, the  $TBLAT$  data is presented on the upper and lower bytes of the  $AD15:AD0$  bus. The appropriate level of the  $BA0$  control line is strobed on the LSb of the  $TBLPTR$ .

**FIGURE 7-6: 8-BIT MULTIPLEXED MODE EXAMPLE**



# PIC18F85J11 FAMILY

## 7.7.1 8-BIT MODE TIMING

The presentation of control signals on the external memory bus is different for the various operating modes. Typical signal timing diagrams are shown in Figure 7-7 and Figure 7-8.

**FIGURE 7-7: EXTERNAL MEMORY BUS TIMING FOR TBLRD (EXTENDED MICROCONTROLLER MODE)**

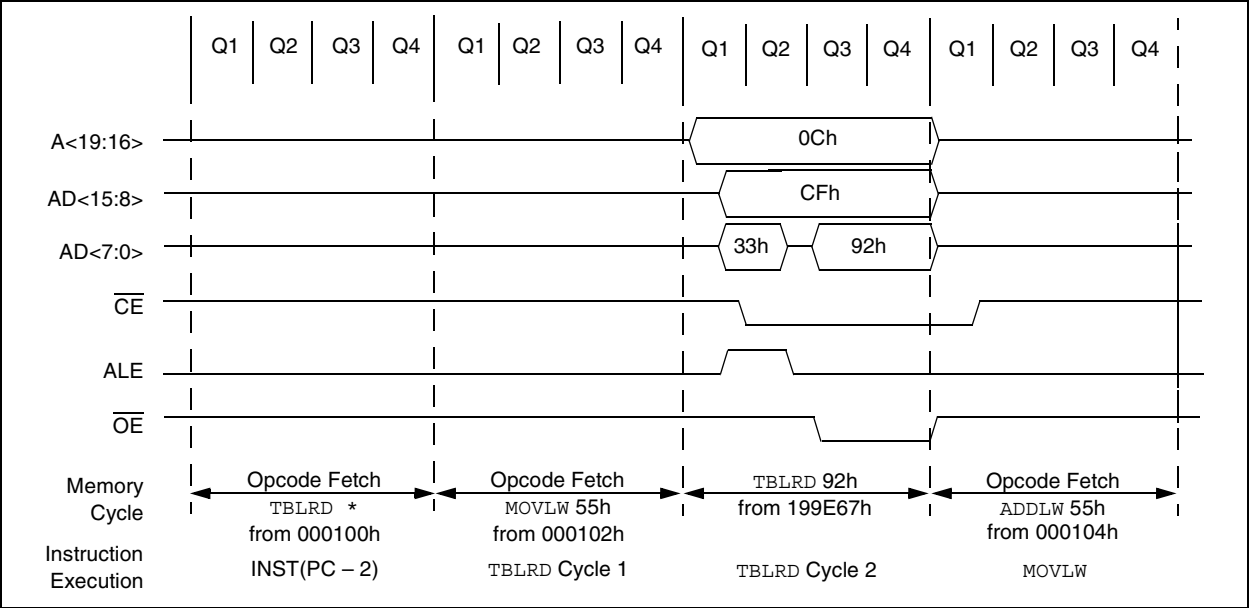
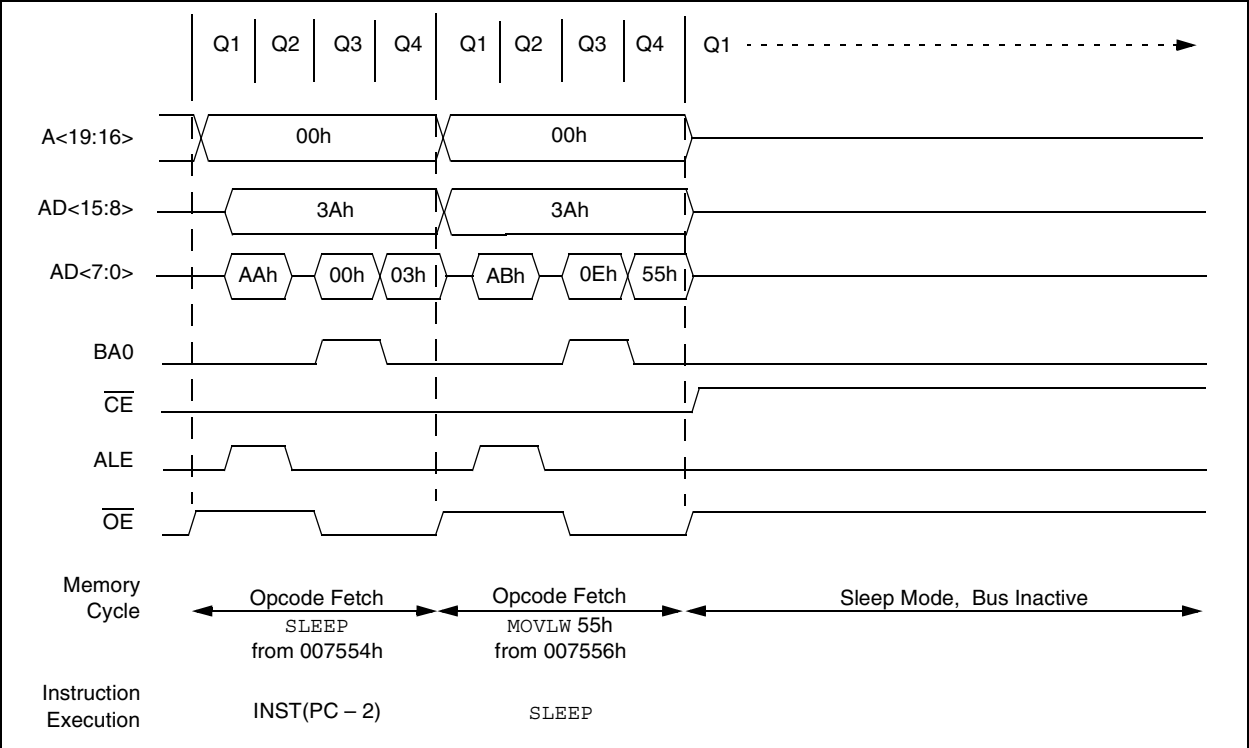


**FIGURE 7-8: EXTERNAL MEMORY BUS TIMING FOR SLEEP (EXTENDED MICROCONTROLLER MODE)**





## 7.8 Operation in Power-Managed Modes

In alternate, power-managed Run modes, the external bus continues to operate normally. If a clock source with a lower speed is selected, bus operations will run at that speed. In these cases, excessive access times for the external memory may result if wait states have been enabled and added to external memory operations. If operations in a lower power Run mode are anticipated, users should provide in their applications for adjusting memory access times at the lower clock speeds.

In Sleep and Idle modes, the microcontroller core does not need to access data; bus operations are suspended. The state of the external bus is frozen, with the address/data pins and most of the control pins holding at the same state they were in when the mode was invoked. The only potential changes are the  $\overline{CE}$ ,  $\overline{LB}$  and  $\overline{UB}$  pins, which are held at logic high.

# PIC18F85J11 FAMILY

---

NOTES:

## 8.0 8 x 8 HARDWARE MULTIPLIER

### 8.1 Introduction

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

Making multiplication a hardware operation allows it to be completed in a single instruction cycle. This has the advantages of higher computational throughput and reduced code size for multiplication algorithms and allows the PIC18 devices to be used in many applications previously reserved for digital signal processors. A comparison of various hardware and software multiply operations, along with the savings in memory and execution time, is shown in Table 8-1.

### 8.2 Operation

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

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

#### EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

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

#### EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

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

BTFSC   ARG2, SB    ; Test Sign Bit
SUBWF   PRODH, F     ; PRODH = PRODH
                        ; - ARG1

MOVF    ARG2, W
BTFSC   ARG1, SB    ; Test Sign Bit
SUBWF   PRODH, F     ; PRODH = PRODH
                        ; - ARG2
```

**TABLE 8-1: PERFORMANCE COMPARISON FOR VARIOUS MULTIPLY OPERATIONS**

| Routine          | Multiply Method           | Program Memory (Words) | Cycles (Max) | Time     |          |         |
|------------------|---------------------------|------------------------|--------------|----------|----------|---------|
|                  |                           |                        |              | @ 40 MHz | @ 10 MHz | @ 4 MHz |
| 8 x 8 unsigned   | Without hardware multiply | 13                     | 69           | 6.9 µs   | 27.6 µs  | 69 µs   |
|                  | Hardware multiply         | 1                      | 1            | 100 ns   | 400 ns   | 1 µs    |
| 8 x 8 signed     | Without hardware multiply | 33                     | 91           | 9.1 µs   | 36.4 µs  | 91 µs   |
|                  | Hardware multiply         | 6                      | 6            | 600 ns   | 2.4 µs   | 6 µs    |
| 16 x 16 unsigned | Without hardware multiply | 21                     | 242          | 24.2 µs  | 96.8 µs  | 242 µs  |
|                  | Hardware multiply         | 28                     | 28           | 2.8 µs   | 11.2 µs  | 28 µs   |
| 16 x 16 signed   | Without hardware multiply | 52                     | 254          | 25.4 µs  | 102.6 µs | 254 µs  |
|                  | Hardware multiply         | 35                     | 40           | 4.0 µs   | 16.0 µs  | 40 µs   |

# PIC18F85J11 FAMILY

Example 8-3 shows the sequence to do a 16 x 16 unsigned multiplication. Equation 8-1 shows the algorithm that is used. The 32-bit result is stored in four registers (RES3:RES0).

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

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

## 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 shows the sequence to do a 16 x 16 signed multiply. Equation 8-2 shows the algorithm used. The 32-bit result is stored in four registers (RES3:RES0). 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

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

## 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
:

```

## 9.0 INTERRUPTS

Members of the PIC18F85J11 family of devices have multiple interrupt sources and an interrupt priority feature that allows most interrupt sources to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 0008h and the low priority interrupt vector is at 0018h. High priority interrupt events will interrupt any low priority interrupts that may be in progress.

There are thirteen registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

In general, interrupt sources have three bits to control their operation. They are:

- **Flag bit** to indicate that an interrupt event occurred
- **Enable bit** that allows program execution to branch to the interrupt vector address when the flag bit is set
- **Priority bit** to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 0008h or 0018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC® MCU mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 0008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt. Low priority interrupts are not processed while high priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (0008h or 0018h). Once in the Interrupt Service Routine (ISR), the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

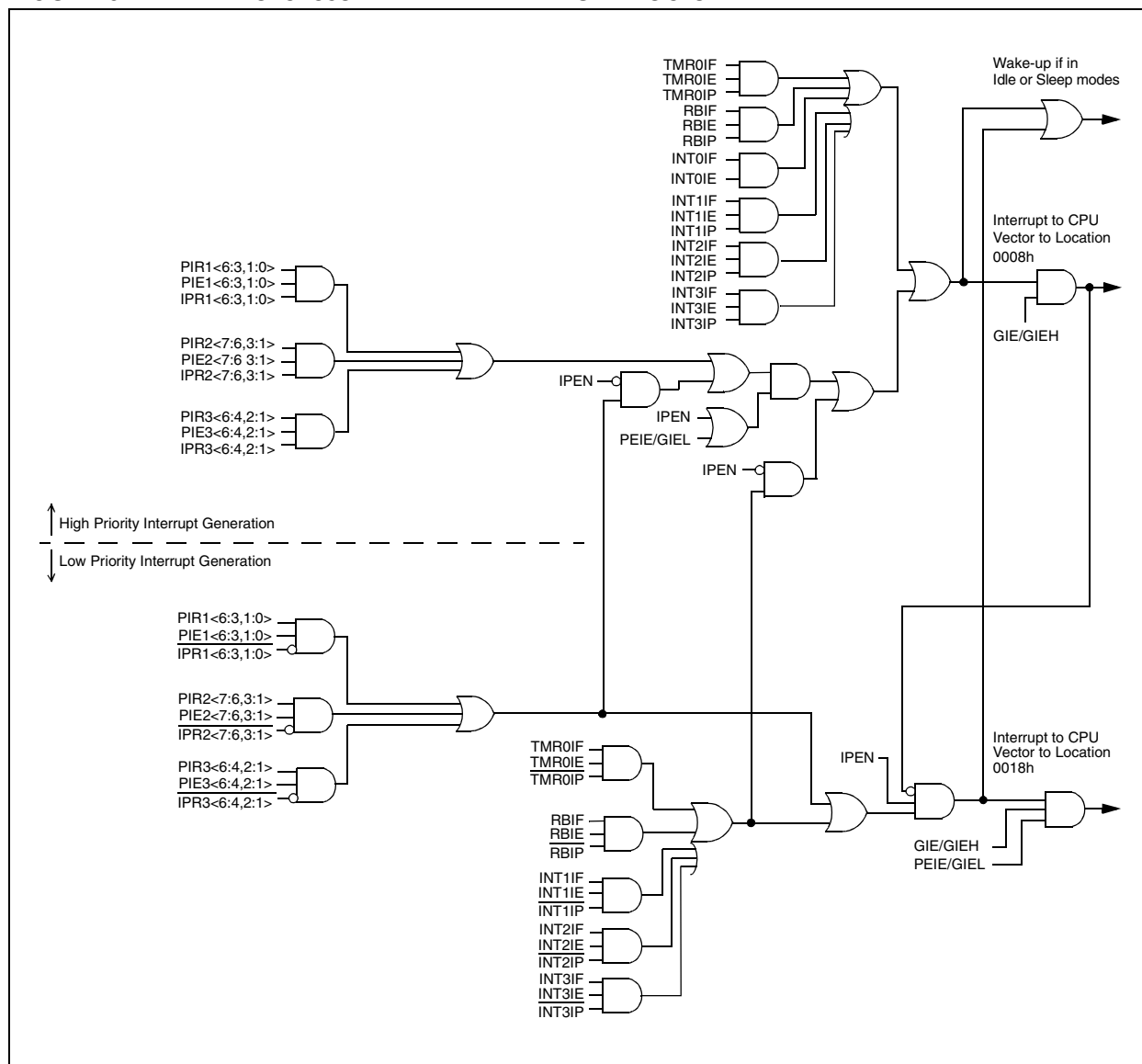
The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INTx pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

|                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Note:</b> Do not use the MOVFF instruction to modify any of the Interrupt Control registers while <b>any</b> interrupt is enabled. Doing so may cause erratic microcontroller behavior. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

# PIC18F85J11 FAMILY

FIGURE 9-1: PIC18F85J11 FAMILY INTERRUPT LOGIC



## 9.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits.

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

### REGISTER 9-1: INTCON: INTERRUPT CONTROL REGISTER

| R/W-0    | R/W-0     | R/W-0  | R/W-0  | R/W-0 | R/W-0  | R/W-0  | R/W-x               |
|----------|-----------|--------|--------|-------|--------|--------|---------------------|
| GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE | RBIE  | TMR0IF | INT0IF | RBIF <sup>(1)</sup> |
| 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            **GIE/GIEH:** Global Interrupt Enable bit  
                   When IPEN = 0:  
                   1 = Enables all unmasked interrupts  
                   0 = Disables all interrupts  
                   When IPEN = 1:  
                   1 = Enables all high priority interrupts  
                   0 = Disables all interrupts
- bit 6            **PEIE/GIEL:** Peripheral Interrupt Enable bit  
                   When IPEN = 0:  
                   1 = Enables all unmasked peripheral interrupts  
                   0 = Disables all peripheral interrupts  
                   When IPEN = 1:  
                   1 = Enables all low priority peripheral interrupts  
                   0 = Disables all low priority peripheral interrupts
- bit 5            **TMR0IE:** TMR0 Overflow Interrupt Enable bit  
                   1 = Enables the TMR0 overflow interrupt  
                   0 = Disables the TMR0 overflow interrupt
- bit 4            **INT0IE:** INT0 External Interrupt Enable bit  
                   1 = Enables the INT0 external interrupt  
                   0 = Disables the INT0 external interrupt
- bit 3            **RBIE:** RB Port Change Interrupt Enable bit  
                   1 = Enables the RB port change interrupt  
                   0 = Disables the RB port change interrupt
- bit 2            **TMR0IF:** TMR0 Overflow Interrupt Flag bit  
                   1 = TMR0 register has overflowed (must be cleared in software)  
                   0 = TMR0 register did not overflow
- bit 1            **INT0IF:** INT0 External Interrupt Flag bit  
                   1 = The INT0 external interrupt occurred (must be cleared in software)  
                   0 = The INT0 external interrupt did not occur
- bit 0            **RBIF:** RB Port Change Interrupt Flag bit<sup>(1)</sup>  
                   1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)  
                   0 = None of the RB7:RB4 pins have changed state

**Note 1:** A mismatch condition will continue to set this bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared.

# PIC18F85J11 FAMILY

## REGISTER 9-2: INTCON2: INTERRUPT CONTROL REGISTER 2

| R/W-1                           | R/W-1   | R/W-1   | R/W-1   | R/W-1   | R/W-1  | R/W-1  | R/W-1 |
|---------------------------------|---------|---------|---------|---------|--------|--------|-------|
| $\overline{\text{RBP}}\text{U}$ | INTEDG0 | INTEDG1 | INTEDG2 | INTEDG3 | TMR0IP | INT3IP | RBIP  |
| 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  **$\overline{\text{RBP}}\text{U}$** : PORTB Pull-up Enable bit  
1 = All PORTB pull-ups are disabled  
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG0**: External Interrupt 0 Edge Select bit  
1 = Interrupt on rising edge  
0 = Interrupt on falling edge
- bit 5 **INTEDG1**: External Interrupt 1 Edge Select bit  
1 = Interrupt on rising edge  
0 = Interrupt on falling edge
- bit 4 **INTEDG2**: External Interrupt 2 Edge Select bit  
1 = Interrupt on rising edge  
0 = Interrupt on falling edge
- bit 3 **INTEDG3**: External Interrupt 3 Edge Select bit  
1 = Interrupt on rising edge  
0 = Interrupt on falling edge
- bit 2 **TMR0IP**: TMR0 Overflow Interrupt Priority bit  
1 = High priority  
0 = Low priority
- bit 1 **INT3IP**: INT3 External Interrupt Priority bit  
1 = High priority  
0 = Low priority
- bit 0 **RBIP**: RB Port Change Interrupt Priority bit  
1 = High priority  
0 = Low priority

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



# PIC18F85J11 FAMILY

## REGISTER 9-3: INTCON3: INTERRUPT CONTROL REGISTER 3

|        |        |        |        |        |        |        |        |
|--------|--------|--------|--------|--------|--------|--------|--------|
| R/W-1  | R/W-1  | R/W-0  | R/W-0  | R/W-0  | R/W-0  | R/W-0  | R/W-0  |
| INT2IP | INT1IP | INT3IE | INT2IE | INT1IE | INT3IF | INT2IF | INT1IF |
| bit 7  |        |        |        |        |        |        | bit 0  |

### Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

|       |                                                                                                                                                                            |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bit 7 | <b>INT2IP:</b> INT2 External Interrupt Priority bit<br>1 = High priority<br>0 = Low priority                                                                               |
| bit 6 | <b>INT1IP:</b> INT1 External Interrupt Priority bit<br>1 = High priority<br>0 = Low priority                                                                               |
| bit 5 | <b>INT3IE:</b> INT3 External Interrupt Enable bit<br>1 = Enables the INT3 external interrupt<br>0 = Disables the INT3 external interrupt                                   |
| bit 4 | <b>INT2IE:</b> INT2 External Interrupt Enable bit<br>1 = Enables the INT2 external interrupt<br>0 = Disables the INT2 external interrupt                                   |
| bit 3 | <b>INT1IE:</b> INT1 External Interrupt Enable bit<br>1 = Enables the INT1 external interrupt<br>0 = Disables the INT1 external interrupt                                   |
| bit 2 | <b>INT3IF:</b> INT3 External Interrupt Flag bit<br>1 = The INT3 external interrupt occurred (must be cleared in software)<br>0 = The INT3 external interrupt did not occur |
| bit 1 | <b>INT2IF:</b> INT2 External Interrupt Flag bit<br>1 = The INT2 external interrupt occurred (must be cleared in software)<br>0 = The INT2 external interrupt did not occur |
| bit 0 | <b>INT1IF:</b> INT1 External Interrupt Flag bit<br>1 = The INT1 external interrupt occurred (must be cleared in software)<br>0 = The INT1 external interrupt did not occur |

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

# PIC18F85J11 FAMILY

## 9.2 PIR Registers

The PIR registers contain the individual flag bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Request (Flag) registers (PIR1, PIR2, PIR3).

**Note 1:** Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON<7>).

**2:** User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

### REGISTER 9-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

|       |       |       |       |       |     |        |        |
|-------|-------|-------|-------|-------|-----|--------|--------|
| R/W-0 | R/W-0 | R-0   | R-0   | R/W-0 | U-0 | R/W-0  | R/W-0  |
| PSPIF | ADIF  | RC1IF | TX1IF | SSPIF | —   | TMR2IF | TMR1IF |
| 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      **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit  
1 = A read or a write operation has taken place (must be cleared in software)  
0 = No read or write has occurred
- bit 6      **ADIF:** A/D Converter Interrupt Flag bit  
1 = An A/D conversion completed (must be cleared in software)  
0 = The A/D conversion is not complete
- bit 5      **RC1IF:** EUSART Receive Interrupt Flag bit  
1 = The EUSART receive buffer, RCREG1, is full (cleared when RCREG1 is read)  
0 = The EUSART receive buffer is empty
- bit 4      **TX1IF:** EUSART Transmit Interrupt Flag bit  
1 = The EUSART transmit buffer, TXREG1, is empty (cleared when TXREG1 is written)  
0 = The EUSART transmit buffer is full
- bit 3      **SSPIF:** Master Synchronous Serial Port Interrupt Flag bit  
1 = The transmission/reception is complete (must be cleared in software)  
0 = Waiting to transmit/receive
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit  
1 = TMR2 to PR2 match occurred (must be cleared in software)  
0 = No TMR2 to PR2 match occurred
- bit 0      **TMR1IF:** TMR1 Overflow Interrupt Flag bit  
1 = TMR1 register overflowed (must be cleared in software)  
0 = TMR1 register did not overflow

# PIC18F85J11 FAMILY

## REGISTER 9-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2

| R/W-0  | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0  | U-0   |
|--------|-------|-----|-----|-------|-------|--------|-------|
| OSCFIF | CMIF  | —   | —   | BCLIF | LVDIF | TMR3IF | —     |
| 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      **OSCFIF:** Oscillator Fail Interrupt Flag bit  
             1 = Device oscillator failed, clock input has changed to INTOSC (must be cleared in software)  
             0 = Device clock operating
- bit 6      **CMIF:** Comparator Interrupt Flag bit  
             1 = Comparator input has changed (must be cleared in software)  
             0 = Comparator input has not changed
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3      **BCLIF:** Bus Collision Interrupt Flag bit  
             1 = A bus collision occurred (must be cleared in software)  
             0 = No bus collision occurred
- bit 2      **LVDIF:** Low-Voltage Detect Interrupt Flag bit  
             1 = A low-voltage condition occurred (must be cleared in software)  
             0 = The device voltage is above the regulator's low-voltage trip point
- bit 1      **TMR3IF:** TMR3 Overflow Interrupt Flag bit  
             1 = TMR3 register overflowed (must be cleared in software)  
             0 = TMR3 register did not overflow
- bit 0      **Unimplemented:** Read as '0'

# PIC18F85J11 FAMILY

## REGISTER 9-6: PIR3: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 3

|       |     |       |       |     |        |        |       |
|-------|-----|-------|-------|-----|--------|--------|-------|
| U-0   | U-0 | R-0   | R-0   | U-0 | R/W-0  | R/W-0  | U-0   |
| —     | —   | RC2IF | TX2IF | —   | CCP2IF | CCP1IF | —     |
| 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 **RC2IF:** AUSART Receive Interrupt Flag bit

1 = The AUSART receive buffer, RCREG2, is full (cleared when RCREG2 is read)

0 = The AUSART receive buffer is empty

bit 4 **TX2IF:** AUSART Transmit Interrupt Flag bit

1 = The AUSART transmit buffer, TXREG2, is empty (cleared when TXREG2 is written)

0 = The AUSART transmit buffer is full

bit 3 **Unimplemented:** Read as '0'

bit 2 **CCP2IF:** CCP2 Interrupt Flag bit

#### Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)

0 = No TMR1/TMR3 register capture occurred

#### Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1/TMR3 register compare match occurred

#### PWM mode:

Unused in this mode.

bit 1 **CCP1IF:** CCP1 Interrupt Flag bit

#### Capture mode:

1 = A TMR1/TMR3 register capture occurred (must be cleared in software)

0 = No TMR1/TMR3 register capture occurred

#### Compare mode:

1 = A TMR1/TMR3 register compare match occurred (must be cleared in software)

0 = No TMR1/TMR3 register compare match occurred

#### PWM mode:

Unused in this mode.

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

## 9.3 PIE Registers

The PIE registers contain the individual enable bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2, PIE3). When IPEN = 0, the PEIE bit must be set to enable any of these peripheral interrupts.

**REGISTER 9-7: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1**

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0  | R/W-0  |
|-------|-------|-------|-------|-------|-----|--------|--------|
| PSPIE | ADIE  | RC1IE | TX1IE | SSPIE | —   | TMR2IE | TMR1IE |
| 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 **PSPIE:** Parallel Slave Port Read/Write Interrupt Enable bit

1 = Enables the PSP read/write interrupt

0 = Disables the PSP read/write interrupt

bit 6 **ADIE:** A/D Converter Interrupt Enable bit

1 = Enables the A/D interrupt

0 = Disables the A/D interrupt

bit 5 **RC1IE:** EUSART Receive Interrupt Enable bit

1 = Enables the EUSART receive interrupt

0 = Disables the EUSART receive interrupt

bit 4 **TX1IE:** EUSART Transmit Interrupt Enable bit

1 = Enables the EUSART transmit interrupt

0 = Disables the EUSART transmit interrupt

bit 3 **SSPIE:** Master Synchronous Serial Port Interrupt Enable bit

1 = Enables the MSSP interrupt

0 = Disables the MSSP interrupt

bit 2 **Unimplemented:** Read as '0'

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit

1 = Enables the TMR2 to PR2 match interrupt

0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit

1 = Enables the TMR1 overflow interrupt

0 = Disables the TMR1 overflow interrupt

# PIC18F85J11 FAMILY

## REGISTER 9-8:    **PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2**

|        |       |     |     |       |       |        |       |
|--------|-------|-----|-----|-------|-------|--------|-------|
| R/W-0  | R/W-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0  | U-0   |
| OSCFIE | CMIE  | —   | —   | BCLIE | LVDIE | TMR3IE | —     |
| 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       **OSCFIE:** Oscillator Fail Interrupt Enable bit  
            1 = Enabled  
            0 = Disabled
- bit 6       **CMIE:** Comparator Interrupt Enable bit  
            1 = Enabled  
            0 = Disabled
- bit 5-4     **Unimplemented:** Read as '0'
- bit 3       **BCLIE:** Bus Collision Interrupt Enable bit  
            1 = Enabled  
            0 = Disabled
- bit 2       **LVDIE:** Low-Voltage Detect Interrupt Enable bit  
            1 = Enabled  
            0 = Disabled
- bit 1       **TMR3IE:** TMR3 Overflow Interrupt Enable bit  
            1 = Enabled  
            0 = Disabled
- bit 0       **Unimplemented:** Read as '0'

# PIC18F85J11 FAMILY

## REGISTER 9-9:    **PIE3: PERIPHERAL INTERRUPT ENABLE REGISTER 3**

|       |     |       |       |     |        |        |       |
|-------|-----|-------|-------|-----|--------|--------|-------|
| U-0   | U-0 | R-0   | R-0   | U-0 | R/W-0  | R/W-0  | U-0   |
| —     | —   | RC2IE | TX2IE | —   | CCP2IE | CCP1IE | —     |
| 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 | <b>Unimplemented:</b> Read as '0'                                                 |
| bit 5   | <b>RC2IE:</b> AUSART Receive Interrupt Enable bit<br>1 = Enabled<br>0 = Disabled  |
| bit 4   | <b>TX2IE:</b> AUSART Transmit Interrupt Enable bit<br>1 = Enabled<br>0 = Disabled |
| bit 3   | <b>Unimplemented:</b> Read as '0'                                                 |
| bit 2   | <b>CCP2IE:</b> CCP2 Interrupt Enable bit<br>1 = Enabled<br>0 = Disabled           |
| bit 1   | <b>CCP1IE:</b> CCP1 Interrupt Enable bit<br>1 = Enabled<br>0 = Disabled           |
| bit 0   | <b>Unimplemented:</b> Read as '0'                                                 |

# PIC18F85J11 FAMILY

## 9.4 IPR Registers

The IPR registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Priority registers (IPR1, IPR2, IPR3). Using the priority bits requires that the Interrupt Priority Enable (IPEN) bit be set.

### REGISTER 9-10: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1

|       |       |       |       |       |     |        |        |
|-------|-------|-------|-------|-------|-----|--------|--------|
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | U-0 | R/W-1  | R/W-1  |
| PSPIP | ADIP  | RC1IP | TX1IP | SSPIP | —   | TMR2IP | TMR1IP |
| 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 **PSPIP:** Parallel Slave Port Read/Write Interrupt Priority bit

1 = High priority

0 = Low priority

bit 6 **ADIP:** A/D Converter Interrupt Priority bit

1 = High priority

0 = Low priority

bit 5 **RC1IP:** EUSART Receive Interrupt Priority bit

1 = High priority

0 = Low priority

bit 4 **TX1IP:** EUSART Transmit Interrupt Priority bit

1 = High priority

0 = Low priority

bit 3 **SSPIP:** Master Synchronous Serial Port Interrupt Priority bit

1 = High priority

0 = Low priority

bit 2 **Unimplemented:** Read as '0'

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit

1 = High priority

0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit

1 = High priority

0 = Low priority



# PIC18F85J11 FAMILY

## REGISTER 9-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

|        |       |     |     |       |       |        |       |
|--------|-------|-----|-----|-------|-------|--------|-------|
| R/W-1  | R/W-1 | U-0 | U-0 | R/W-1 | R/W-1 | R/W-1  | U-0   |
| OSCFIP | CMIP  | —   | —   | BCLIP | LVDIP | TMR3IP | —     |
| 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      **OSCFIP:** Oscillator Fail Interrupt Priority bit  
             1 = High priority  
             0 = Low priority
- bit 6      **CMIP:** Comparator Interrupt Priority bit  
             1 = High priority  
             0 = Low priority
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3      **BCLIP:** Bus Collision Interrupt Priority bit  
             1 = High priority  
             0 = Low priority
- bit 2      **LVDIP:** Low-Voltage Detect Interrupt Priority bit  
             1 = High priority  
             0 = Low priority
- bit 1      **TMR3IP:** TMR3 Overflow Interrupt Priority bit  
             1 = High priority  
             0 = Low priority
- bit 0      **Unimplemented:** Read as '0'

# PIC18F85J11 FAMILY

## REGISTER 9-12: IPR3: PERIPHERAL INTERRUPT PRIORITY REGISTER 3

|       |     |       |       |     |        |        |       |
|-------|-----|-------|-------|-----|--------|--------|-------|
| U-0   | U-0 | R-0   | R-0   | U-0 | R/W-1  | R/W-1  | U-0   |
| —     | —   | RC2IP | TX2IP | —   | CCP2IP | CCP1IP | —     |
| 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      **RC2IP:** AUSART Receive Priority Flag bit  
            1 = High priority  
            0 = Low priority
- bit 4      **TX2IP:** AUSART Transmit Interrupt Priority bit  
            1 = High priority  
            0 = Low priority
- bit 3      **Unimplemented:** Read as '0'
- bit 2      **CCP2IP:** CCP2 Interrupt Priority bit  
            1 = High priority  
            0 = Low priority
- bit 1      **CCP1IP:** CCP1 Interrupt Priority bit  
            1 = High priority  
            0 = Low priority
- bit 0      **Unimplemented:** Read as '0'

## 9.5 RCON Register

The RCON register contains bits used to determine the cause of the last Reset or wake-up from Idle or Sleep modes. RCON also contains the bit that enables interrupt priorities (IPEN).

**REGISTER 9-13: RCON: RESET CONTROL REGISTER**

| R/W-0 | U-0 | R/W-1                  | R/W-1                  | R-1                    | R-1                    | R/W-0                   | R/W-0                   |
|-------|-----|------------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
| IPEN  | —   | $\overline{\text{CM}}$ | $\overline{\text{RI}}$ | $\overline{\text{TO}}$ | $\overline{\text{PD}}$ | $\overline{\text{POR}}$ | $\overline{\text{BOR}}$ |
| 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      **IPEN:** Interrupt Priority Enable bit  
             1 = Enable priority levels on interrupts  
             0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)
- bit 6      **Unimplemented:** Read as '0'
- bit 5      **CM:** Configuration Mismatch Flag bit  
             For details of bit operation, see Register 4-1.
- bit 4      **RI:** RESET Instruction Flag bit  
             For details of bit operation, see Register 4-1.
- bit 3      **TO:** Watchdog Timer Time-out Flag bit  
             For details of bit operation, see Register 4-1.
- bit 2      **PD:** Power-Down Detection Flag bit  
             For details of bit operation, see Register 4-1.
- bit 1      **POR:** Power-on Reset Status bit  
             For details of bit operation, see Register 4-1.
- bit 0      **BOR:** Brown-out Reset Status bit  
             For details of bit operation, see Register 4-1.

# PIC18F85J11 FAMILY

## 9.6 INTx Pin Interrupts

External interrupts on the RB0/INT0, RB1/INT1, RB2/INT2 and RB3/INT3 pins are edge-triggered. If the corresponding INTEDGx bit in the INTCON2 register is set (= 1), the interrupt is triggered by a rising edge; if the bit is clear, the trigger is on the falling edge. When a valid edge appears on the RBx/INTx pin, the corresponding flag bit, INTxIF, is set. This interrupt can be disabled by clearing the corresponding enable bit, INTxIE. Flag bit, INTxIF, must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt.

All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from the power-managed modes if bit INTxIE was set prior to going into the power-managed modes. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up.

Interrupt priority for INT1, INT2 and INT3 is determined by the value contained in the interrupt priority bits, INT1IP (INTCON3<6>), INT2IP (INTCON3<7>) and INT3IP (INTCON2<1>). There is no priority bit associated with INT0. It is always a high priority interrupt source.

## 9.7 TMR0 Interrupt

In 8-bit mode (which is the default), an overflow in the TMR0 register (FFh → 00h) will set flag bit, TMR0IF. In 16-bit mode, an overflow in the TMR0H:TMR0L register pair (FFFFh → 0000h) will set TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit, TMR0IE (INTCON<5>). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit, TMR0IP (INTCON2<2>). See **Section 11.0 “Timer0 Module”** for further details on the Timer0 module.

## 9.8 PORTB Interrupt-on-Change

An input change on PORTB<7:4> sets flag bit, RBIF (INTCON<0>). The interrupt can be enabled/disabled by setting/clearing enable bit, RBIE (INTCON<3>). Interrupt priority for PORTB interrupt-on-change is determined by the value contained in the interrupt priority bit, RBIP (INTCON2<0>).

## 9.9 Context Saving During Interrupts

During interrupts, the return PC address is saved on the stack. Additionally, the WREG, STATUS and BSR registers are saved on the Fast Return Stack. If a fast return from interrupt is not used (see **Section 5.3 “Data Memory Organization”**), the user may need to save the WREG, STATUS and BSR registers on entry to the Interrupt Service Routine. Depending on the user's application, other registers may also need to be saved. Example 9-1 saves and restores the WREG, STATUS and BSR registers during an Interrupt Service Routine.

### EXAMPLE 9-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

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

## 10.0 I/O PORTS

Depending on the device selected and features enabled, there are up to nine ports available. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Each port has three memory mapped registers for its operation:

- TRIS register (Data Direction register)
- PORT register (reads the levels on the pins of the device)
- LAT register (Output Latch register)

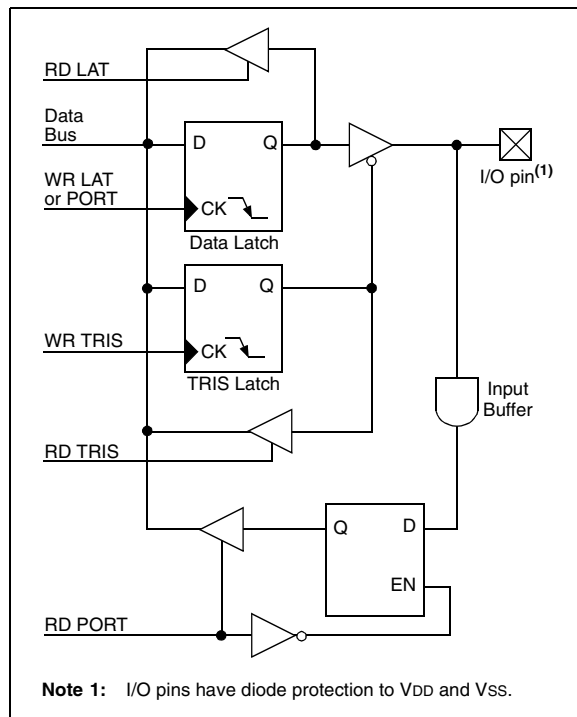
Reading the PORT register reads the current status of the pins, whereas writing to the PORT register writes to the Output Latch (LAT) register.

Setting a TRIS bit (= 1) makes the corresponding PORT pin an input (i.e., put the corresponding output driver in a High-Impedance mode). Clearing a TRIS bit (= 0) makes the corresponding PORT pin an output (i.e., put the contents of the corresponding LAT bit on the selected pin).

The Output Latch (LAT register) is useful for read-modify-write operations on the value that the I/O pins are driving. Read-modify-write operations on the LAT register read and write the latched output value for the PORT register.

A simplified model of a generic I/O port, without the interfaces to other peripherals, is shown in Figure 10-1.

**FIGURE 10-1: GENERIC I/O PORT OPERATION**



## 10.1 I/O Port Pin Capabilities

When developing an application, the capabilities of the port pins must be considered. Outputs on some pins have higher output drive strength than others. Similarly, some pins can tolerate higher than VDD input levels.

### 10.1.1 INPUT PINS AND VOLTAGE CONSIDERATIONS

The voltage tolerance of pins used as device inputs is dependent on the pin's input function. Pins that are used as digital only inputs are able to handle DC voltages up to 5.5V, a level typical for digital logic circuits. In contrast, pins that also have analog input functions of any kind can only tolerate voltages up to VDD. Voltage excursions beyond VDD on these pins should be avoided.

Table 10-1 summarizes the input voltage capabilities. Refer to **Section 25.0 "Electrical Characteristics"** for more details.

**TABLE 10-1: INPUT VOLTAGE TOLERANCE**

| Port or Pin               | Tolerated Input | Description                                                       |
|---------------------------|-----------------|-------------------------------------------------------------------|
| PORTA<7:0>                | VDD             | Only VDD input levels tolerated.                                  |
| PORTC<1:0>                |                 |                                                                   |
| PORTF<7:1>                |                 |                                                                   |
| PORTB<7:0>                | 5.5V            | Tolerates input levels above VDD, useful for most standard logic. |
| PORTC<7:2>                |                 |                                                                   |
| PORTD<7:0>                |                 |                                                                   |
| PORTE<7:0>                |                 |                                                                   |
| PORTG<4:0>                |                 |                                                                   |
| PORTH<7:0> <sup>(1)</sup> |                 |                                                                   |
| PORTJ<7:0> <sup>(1)</sup> |                 |                                                                   |

**Note 1:** Not available on 64-pin devices.

### 10.1.2 PIN OUTPUT DRIVE

When used as digital I/O, the output pin drive strengths vary for groups of pins intended to meet the needs for a variety of applications. In general, there are three classes of output pins in terms of drive capability.

PORTB and PORTC, as well as PORTA<7:6>, are designed to drive higher current loads, such as LEDs. PORTD, PORTE and PORTJ are capable of driving digital circuits associated with external memory devices. They can also drive LEDs, but only those with smaller current requirements. PORTF, PORTG and PORTH, along with PORTA<5:0>, have the lowest drive level, but are capable of driving normal digital circuit loads with a high input impedance.

Table 10-2 summarizes the output capabilities of the ports. Refer to the **"Absolute Maximum Ratings"** in **Section 25.0 "Electrical Characteristics"** for more details.

# PIC18F85J11 FAMILY

**TABLE 10-2: OUTPUT DRIVE LEVELS FOR VARIOUS PORTS**

| Low                  | Medium               | High       |
|----------------------|----------------------|------------|
| PORTA<5:0>           | PORTD                | PORTA<7:6> |
| PORTF                | PORTE                | PORTB      |
| PORTG                | PORTJ <sup>(1)</sup> | PORTC      |
| PORTH <sup>(1)</sup> |                      |            |

**Note 1:** Not available on 64-pin devices.

## 10.1.3 PULL-UP CONFIGURATION

Four of the I/O ports (PORTB, PORTD, PORTE and PORTJ) implement configurable weak pull-ups on all pins. These are internal pull-ups that allow floating digital input signals to be pulled to a consistent level without the use of external resistors.

The pull-ups are enabled with a single bit for each of the ports: RBPU (INTCON2<7>) for PORTB, and RDP, RPU and RJPU (PORTG<7:5>) for the other ports.

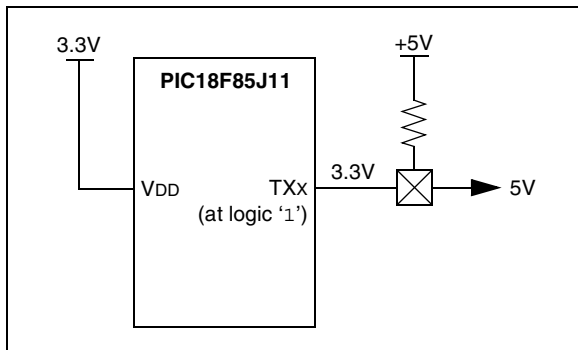
## 10.1.4 OPEN-DRAIN OUTPUTS

The output pins for several peripherals are also equipped with a configurable open-drain output option. This allows the peripherals to communicate with external digital logic, operating at a higher voltage level, without the use of level translators.

The open-drain option is implemented on port pins specifically associated with the data and clock outputs of the USARTs, the MSSP module (in SPI mode) and the CCP modules. The option is selectively enabled by setting the open-drain control bit for the corresponding module in TRISG and LATG. Their configuration is discussed in more detail in the sections for PORTC, PORTE and PORTG.

When the open-drain option is required, the output pin must also be tied through an external pull-up resistor provided by the user to a higher voltage level, up to 5V (Figure 10-2). When a digital logic high signal is output, it is pulled up to the higher voltage level.

**FIGURE 10-2: USING THE OPEN-DRAIN OUTPUT (USARTs SHOWN AS EXAMPLES)**



## 10.2 PORTA, TRISA and LATA Registers

PORTA is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISA and LATA.

RA4/T0CKI is a Schmitt Trigger input. All other PORTA pins have TTL input levels and full CMOS output drivers.

The RA4 pin is multiplexed with the Timer0 clock input. RA5 and RA3:RA0 are multiplexed with analog inputs for the A/D converter.

The operation of the analog inputs as A/D converter inputs is selected by clearing or setting the PCFG3:PCFG0 control bits in the ADCON1 register. The corresponding TRISA bits control the direction of these pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

**Note:** RA5 and RA3:RA0 are configured as analog inputs on any Reset and are read as '0'. RA4 is configured as a digital input.

RA6/OSC2/CLKO and RA7/OSC1/CLKI normally serve as the external circuit connections for the external (primary) oscillator circuit (HS Oscillator modes), or the external clock input and output (EC Oscillator modes). In these cases, RA6 and RA7 are not available as digital I/O and their corresponding TRIS and LAT bits are read as '0'. When the device is configured to use INTOSC or INTRC as the default oscillator mode (FOSC2 Configuration bit is '0'), RA6 and RA7 are automatically configured as digital I/O; the oscillator and clock in/clock out functions are disabled.

### EXAMPLE 10-1: INITIALIZING PORTA

```
CLRF    PORTA    ; Initialize PORTA by
                ; clearing output latches
CLRF    LATA      ; Alternate method to
                ; clear output data latches
MOVLW   07h      ; Configure A/D
MOVWF   ADCON1   ; for digital inputs
MOVLW   0BFh     ; Value used to initialize
                ; data direction
MOVWF   TRISA    ; Set RA<7, 5:0> as inputs,
                ; RA<6> as output
```

# PIC18F85J11 FAMILY

**TABLE 10-3: PORTA FUNCTIONS**

| Pin Name      | Function | TRIS Setting | I/O | I/O Type | Description                                                                              |
|---------------|----------|--------------|-----|----------|------------------------------------------------------------------------------------------|
| RA0/AN0       | RA0      | 0            | O   | DIG      | LATA<0> data output; not affected by analog input.                                       |
|               |          | 1            | I   | TTL      | PORTA<0> data input; disabled when analog input enabled.                                 |
|               | AN0      | 1            | I   | ANA      | A/D input channel 0. Default input configuration on POR; does not affect digital output. |
| RA1/AN1       | RA1      | 0            | O   | DIG      | LATA<1> data output; not affected by analog input.                                       |
|               |          | 1            | I   | TTL      | PORTA<1> data input; disabled when analog input enabled.                                 |
|               | AN1      | 1            | I   | ANA      | A/D input channel 1. Default input configuration on POR; does not affect digital output. |
| RA2/AN2/VREF- | RA2      | 0            | O   | DIG      | LATA<2> data output; not affected by analog input.                                       |
|               |          | 1            | I   | TTL      | PORTA<2> data input. Disabled when analog functions enabled.                             |
|               | AN2      | 1            | I   | ANA      | A/D input channel 2. Default input configuration on POR.                                 |
|               | VREF-    | 1            | I   | ANA      | A/D and comparator low reference voltage input.                                          |
| RA3/AN3/VREF+ | RA3      | 0            | O   | DIG      | LATA<3> data output; not affected by analog input.                                       |
|               |          | 1            | I   | TTL      | PORTA<3> data input; disabled when analog input enabled.                                 |
|               | AN3      | 1            | I   | ANA      | A/D input channel 3. Default input configuration on POR.                                 |
|               | VREF+    | 1            | I   | ANA      | A/D and comparator high reference voltage input.                                         |
| RA4/T0CKI     | RA4      | 0            | O   | DIG      | LATA<4> data output.                                                                     |
|               |          | 1            | I   | ST       | PORTA<4> data input; default configuration on POR.                                       |
|               | T0CKI    | x            | I   | ST       | Timer0 clock input.                                                                      |
| RA5/AN4       | RA5      | 0            | O   | DIG      | LATA<5> data output; not affected by analog input.                                       |
|               |          | 1            | I   | TTL      | PORTA<5> data input; disabled when analog input enabled.                                 |
|               | AN4      | 1            | I   | ANA      | A/D input channel 4. Default configuration on POR.                                       |
| RA6/OSC2/CLKO | RA6      | 0            | O   | DIG      | LATA<6> data output; disabled when FOSC2 Configuration bit is set.                       |
|               |          | 1            | I   | TTL      | PORTA<6> data input; disabled when FOSC2 Configuration bit is set.                       |
|               | OSC2     | x            | O   | ANA      | Main oscillator feedback output connection (HS and HSPLL modes).                         |
|               | CLKO     | x            | O   | DIG      | System cycle clock output, Fosc/4 (EC and ECPLL modes).                                  |
| RA7/OSC1/CLKI | RA7      | 0            | O   | DIG      | LATA<7> data output; disabled when FOSC2 Configuration bit is set.                       |
|               |          | 1            | I   | TTL      | PORTA<7> data input; disabled when FOSC2 Configuration bit is set.                       |
|               | OSC1     | x            | I   | ANA      | Main oscillator input connection (HS and HSPLL modes).                                   |
|               | CLKI     | x            | I   | ANA      | Main external clock source input (EC and ECPLL modes).                                   |

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

**TABLE 10-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA**

| Name   | Bit 7                 | Bit 6                 | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|--------|-----------------------|-----------------------|--------|--------|--------|--------|--------|--------|----------------------|
| PORTA  | RA7 <sup>(1)</sup>    | RA6 <sup>(1)</sup>    | RA5    | RA4    | RA3    | RA2    | RA1    | RA0    | 54                   |
| LATA   | LATA7 <sup>(1)</sup>  | LATA6 <sup>(1)</sup>  | LATA5  | LATA4  | LATA3  | LATA2  | LATA1  | LATA0  | 54                   |
| TRISA  | TRISA7 <sup>(1)</sup> | TRISA6 <sup>(1)</sup> | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 54                   |
| ADCON1 | —                     | —                     | VCFG1  | VCFG0  | PCFG3  | PCFG2  | PCFG1  | PCFG0  | 53                   |

**Legend:** — = Unimplemented, read as '0'. Shaded cells are not used by PORTA.  
x = Don't care.

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

# PIC18F85J11 FAMILY

## 10.3 PORTB, TRISB and LATB Registers

PORTB is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISB and LATB. All pins on PORTB are digital only and tolerate voltages up to 5.5V.

Each of the PORTB pins has a weak internal pull-up. A single control bit can turn on all the pull-ups. This is performed by clearing bit RBPU (INTCON2<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on a Power-on Reset.

Four of the PORTB pins (RB7:RB4) have an interrupt-on-change feature. Only pins configured as inputs can cause this interrupt to occur (i.e., any RB7:RB4 pin configured as an output is excluded from the interrupt-on-change comparison). The input pins (of RB7:RB4) are compared with the old value latched on the last read of PORTB. The “mismatch” outputs of RB7:RB4 are ORed together to generate the RB Port Change Interrupt with Flag bit, RBIF (INTCON<0>).

This interrupt can wake the device from power-managed modes. The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- a) Any read or write of PORTB (except with the MOVFF (ANY), PORTB instruction). This will end the mismatch condition.
- b) Clear flag bit, RBIF.

A mismatch condition will continue to set flag bit, RBIF. Reading PORTB will end the mismatch condition and allow flag bit, RBIF, to be cleared.

The interrupt-on-change feature is recommended for wake-up on key depression operation and operations where PORTB is only used for the interrupt-on-change feature. Polling of PORTB is not recommended while using the interrupt-on-change feature.

### EXAMPLE 10-2: INITIALIZING PORTB

|       |       |                                                                        |
|-------|-------|------------------------------------------------------------------------|
| CLRF  | PORTB | ; Initialize PORTB by<br>; clearing output<br>; data latches           |
| CLRF  | LATB  | ; Alternate method<br>; to clear output<br>; data latches              |
| MOVLW | 0CFh  | ; Value used to<br>; initialize data<br>; direction                    |
| MOVWF | TRISB | ; Set RB<3:0> as inputs<br>; RB<5:4> as outputs<br>; RB<7:6> as inputs |



# PIC18F85J11 FAMILY

**TABLE 10-5: PORTB FUNCTIONS**

| Pin Name      | Function            | TRIS Setting | I/O | I/O Type | Description                                                                     |
|---------------|---------------------|--------------|-----|----------|---------------------------------------------------------------------------------|
| RB0/INT0      | RB0                 | 0            | O   | DIG      | LATB<0> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<0> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | INT0                | 1            | I   | ST       | External interrupt 0 input.                                                     |
| RB1/INT1      | RB1                 | 0            | O   | DIG      | LATB<1> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<1> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | INT1                | 1            | I   | ST       | External interrupt 1 input.                                                     |
| RB2/INT2      | RB2                 | 0            | O   | DIG      | LATB<2> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<2> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | INT2                | 1            | I   | ST       | External interrupt 2 input.                                                     |
| RB3/INT3/CCP2 | RB3                 | 0            | O   | DIG      | LATB<3> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<3> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | INT3                | 1            | I   | ST       | External interrupt 3 input.                                                     |
|               | CCP2 <sup>(1)</sup> | 0            | O   | DIG      | CCP2 compare output and CCP2 PWM output; takes priority over port data.         |
|               |                     | 1            | I   | ST       | CCP2 capture input.                                                             |
| RB4/KBI0      | RB4                 | 0            | O   | DIG      | LATB<4> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<4> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | KBI0                |              | I   | TTL      | Interrupt-on-pin change.                                                        |
| RB5/KBI1      | RB5                 | 0            | O   | DIG      | LATB<5> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<5> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | KBI1                |              | I   | TTL      | Interrupt-on-pin change.                                                        |
| RB6/KBI2/PGC  | RB6                 | 0            | O   | DIG      | LATB<6> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<6> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | KBI2                | 1            | I   | TTL      | Interrupt-on-pin change.                                                        |
|               | PGC                 | x            | I   | ST       | Serial execution (ICSP™) clock input for ICSP and ICD operation. <sup>(2)</sup> |
| RB7/KBI3/PGD  | RB7                 | 0            | O   | DIG      | LATB<7> data output.                                                            |
|               |                     | 1            | I   | TTL      | PORTB<7> data input; weak pull-up when $\overline{\text{RBPU}}$ bit is cleared. |
|               | KBI3                | 1            | I   | TTL      | Interrupt-on-pin change.                                                        |
|               | PGD                 | x            | O   | DIG      | Serial execution data output for ICSP and ICD operation. <sup>(2)</sup>         |
|               |                     | x            | I   | ST       | Serial execution data input for ICSP and ICD operation. <sup>(2)</sup>          |

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

**Note 1:** Alternate assignment for CCP2 when the CCP2MX Configuration bit is cleared (Extended Microcontroller mode, 80-pin devices only). Default assignment is RC1.

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

**TABLE 10-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB**

| Name    | Bit 7                    | Bit 6     | Bit 5   | Bit 4   | Bit 3   | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|---------|--------------------------|-----------|---------|---------|---------|--------|--------|--------|----------------------|
| PORTB   | RB7                      | RB6       | RB5     | RB4     | RB3     | RB2    | RB1    | RB0    | 54                   |
| LATB    | LATB7                    | LATB6     | LATB5   | LATB4   | LATB3   | LATB2  | LATB1  | LATB0  | 54                   |
| TRISB   | TRISB7                   | TRISB6    | TRISB5  | TRISB4  | TRISB3  | TRISB2 | TRISB1 | TRISB0 | 54                   |
| INTCON  | GIE/GIEH                 | PEIE/GIEL | TMR0IE  | INT0IE  | RBIE    | TMR0IF | INT0IF | RBIF   | 51                   |
| INTCON2 | $\overline{\text{RBPU}}$ | INTEDG0   | INTEDG1 | INTEDG2 | INTEDG3 | TMR0IP | INT3IP | RBIP   | 51                   |
| INTCON3 | INT2IP                   | INT1IP    | INT3IE  | INT2IE  | INT1IE  | INT3IF | INT2IF | INT1IF | 51                   |

**Legend:** Shaded cells are not used by PORTB.

# PIC18F85J11 FAMILY

## 10.4 PORTC, TRISC and LATC Registers

PORTC is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISC and LATC. Only PORTC pins, RC2 through RC7, are digital only pins and can tolerate input voltages up to 5.5V.

PORTC is multiplexed with CCP, MSSP and EUSART peripheral functions (Table 10-7). The pins have Schmitt Trigger input buffers. The pins for CCP, SPI and EUSART are also configurable for open-drain output whenever these functions are active. Open-drain configuration is selected by setting the SPIOD, CCPxOD and U1OD control bits (TRISG<7:5> and LATG<6>, respectively).

RC1 is normally configured as the default peripheral pin for the CCP2 module. Assignment of CCP2 is controlled by Configuration bit, CCP2MX (default state, CCP2MX = 1).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

**Note:** These pins are configured as digital inputs on any device Reset.

The contents of the TRISC register are affected by peripheral overrides. Reading TRISC always returns the current contents, even though a peripheral device may be overriding one or more of the pins.

### EXAMPLE 10-3: INITIALIZING PORTC

```
CLRF    PORTC    ; Initialize PORTC by
                ; clearing output
                ; data latches
CLRF    LATC     ; Alternate method
                ; to clear output
                ; data latches
MOVLW   0CFh     ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISC    ; Set RC<3:0> as inputs
                ; RC<5:4> as outputs
                ; RC<7:6> as inputs
```

# PIC18F85J11 FAMILY

**TABLE 10-7: PORTC FUNCTIONS**

| Pin Name             | Function            | TRIS Setting | I/O | I/O Type | Description                                                                             |
|----------------------|---------------------|--------------|-----|----------|-----------------------------------------------------------------------------------------|
| RC0/T1OSO/<br>T13CKI | RC0                 | 0            | O   | DIG      | LATC<0> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<0> data input.                                                                    |
|                      | T1OSO               | x            | O   | ANA      | Timer1 oscillator output; enabled when Timer1 oscillator enabled. Disables digital I/O. |
|                      | T13CKI              | 1            | I   | ST       | Timer1/Timer3 counter input.                                                            |
| RC1/T1OSI/<br>CCP2   | RC1                 | 0            | O   | DIG      | LATC<1> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<1> data input.                                                                    |
|                      | T1OSI               | x            | I   | ANA      | Timer1 oscillator input; enabled when Timer1 oscillator enabled. Disables digital I/O.  |
|                      | CCP2 <sup>(1)</sup> | 0            | O   | DIG      | CCP2 compare output and CCP2 PWM output; takes priority over port data.                 |
|                      |                     | 1            | I   | ST       | CCP2 capture input.                                                                     |
| RC2/CCP1             | RC2                 | 0            | O   | DIG      | LATC<2> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<2> data input.                                                                    |
|                      | CCP1                | 0            | O   | DIG      | CCP1 compare output and CCP1 PWM output; takes priority over port data.                 |
|                      |                     | 1            | I   | ST       | CCP1 capture input.                                                                     |
| RC3/SCK/SCL          | RC3                 | 0            | O   | DIG      | LATC<3> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<3> data input.                                                                    |
|                      | SCK                 | 0            | O   | DIG      | SPI clock output (MSSP module); takes priority over port data.                          |
|                      |                     | 1            | I   | ST       | SPI clock input (MSSP module).                                                          |
|                      | SCL                 | 0            | O   | DIG      | I <sup>2</sup> C™ clock output (MSSP module); takes priority over port data.            |
|                      |                     | 1            | I   | ST       | I <sup>2</sup> C clock input (MSSP module); input type depends on module setting.       |
| RC4/SDI/SDA          | RC4                 | 0            | O   | DIG      | LATC<4> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<4> data input.                                                                    |
|                      | SDI                 | 1            | I   | ST       | SPI data input (MSSP module).                                                           |
|                      | SDA                 | 1            | O   | DIG      | I <sup>2</sup> C data output (MSSP module); takes priority over port data.              |
|                      |                     | 1            | I   | ST       | I <sup>2</sup> C data input (MSSP module); input type depends on module setting.        |
| RC5/SDO              | RC5                 | 0            | O   | DIG      | LATC<5> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<5> data input.                                                                    |
|                      | SDO                 | 0            | O   | DIG      | SPI data output (MSSP module); takes priority over port data.                           |
| RC6/TX1/CK1          | RC6                 | 0            | O   | DIG      | LATC<6> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<6> data input.                                                                    |
|                      | TX1                 | 1            | O   | DIG      | Synchronous serial data output (EUSART module); takes priority over port data.          |
|                      | CK1                 | 1            | O   | DIG      | Synchronous serial data input (EUSART module). User must configure as an input.         |
|                      |                     | 1            | I   | ST       | Synchronous serial clock input (EUSART module).                                         |
| RC7/RX1/DT1          | RC7                 | 0            | O   | DIG      | LATC<7> data output.                                                                    |
|                      |                     | 1            | I   | ST       | PORTC<7> data input.                                                                    |
|                      | RX1                 | 1            | I   | ST       | Asynchronous serial receive data input (EUSART module).                                 |
|                      | DT1                 | 1            | O   | DIG      | Synchronous serial data output (EUSART module); takes priority over port data.          |
|                      |                     | 1            | I   | ST       | Synchronous serial data input (EUSART module). User must configure as an input.         |

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

**Note 1:** Default assignment for CCP2 when CCP2MX Configuration bit is set.

# PIC18F85J11 FAMILY

**TABLE 10-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC**

| Name  | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|----------------------|
| PORTC | RC7    | RC6    | RC5    | RC4    | RC3    | RC2    | RC1    | RC0    | 54                   |
| LATC  | LATC7  | LATBC6 | LATC5  | LATCB4 | LATC3  | LATC2  | LATC1  | LATC0  | 54                   |
| TRISC | TRISC7 | TRISC6 | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 54                   |
| LATG  | U2OD   | U1OD   | —      | LATG4  | LATG3  | LATG2  | LATG1  | LATG0  | 54                   |
| TRISG | SPIOD  | CCP2OD | CCP1OD | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 | 54                   |

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

## 10.5 PORTD, TRISD and LATD Registers

PORTD is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISD and LATD. All pins on PORTD are digital only and tolerate voltages up to 5.5V.

All pins on PORTD are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

**Note:** These pins are configured as digital inputs on any device Reset.

Each of the PORTD pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is performed by setting bit RDPU (PORTG<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on all device Resets.

On 80-pin devices, PORTD is multiplexed with the system bus as part of the external memory interface. I/O port and other functions are only available when the interface is disabled, by setting the EBDIS bit (MEMCON<7>). When the interface is enabled, PORTD is the low-order byte of the multiplexed address/data bus (AD7:AD0). The TRISD bits are also overridden.

PORTD can also be configured to function as an 8-bit wide, parallel microprocessor port by setting the PSPMODE control bit (PSPCON<4>). In this mode, parallel port data takes priority over other digital I/O (but not the external memory interface). When the parallel port is active, the input buffers are TTL. For more information, refer to **Section 10.11 “Parallel Slave Port”**.

### EXAMPLE 10-4: INITIALIZING PORTD

```
CLRF    PORTD    ; Initialize PORTD by
                  ; clearing output
                  ; data latches
CLRF    LATD     ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   0CFh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISD    ; Set RD<3:0> as inputs
                  ; RD<5:4> as outputs
                  ; RD<7:6> as inputs
```

# PIC18F85J11 FAMILY

**TABLE 10-9: PORTD FUNCTIONS**

| Pin Name     | Function           | TRIS Setting | I/O | I/O Type | Description                                                          |
|--------------|--------------------|--------------|-----|----------|----------------------------------------------------------------------|
| RD0/AD0/PSP0 | RD0                | 0            | O   | DIG      | LATD<0> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<0> data input.                                                 |
|              | AD0 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 0 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 0 input. <sup>(1)</sup>          |
|              | PSP0               |              | O   | DIG      | PSP read output data (LATD<0>); takes priority over port data.       |
|              |                    |              | I   | TTL      | PSP write data input.                                                |
| RD1/AD1/PSP1 | RD1                | 0            | O   | DIG      | LATD<1> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<1> data input.                                                 |
|              | AD1 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 1 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 1 input. <sup>(1)</sup>          |
|              | PSP1               | x            | O   | DIG      | PSP read output data (LATD<1>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |
| RD2/AD2/PSP2 | RD2                | 0            | O   | DIG      | LATD<2> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<2> data input.                                                 |
|              | AD2 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 2 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 2 input. <sup>(1)</sup>          |
|              | PSP2               | x            | O   | DIG      | PSP read output data (LATD<2>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |
| RD3/AD3/PSP3 | RD3                | 0            | O   | DIG      | LATD<3> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<3> data input.                                                 |
|              | AD3 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 3 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 3 input. <sup>(1)</sup>          |
|              | PSP3               | x            | O   | DIG      | PSP read output data (LATD<3>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |
| RD4/AD4/PSP4 | RD4                | 0            | O   | DIG      | LATD<4> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<4> data input.                                                 |
|              | AD4 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 4 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 4 input. <sup>(1)</sup>          |
|              | PSP4               | x            | O   | DIG      | PSP read output data (LATD<4>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |
| RD5/AD5/PSP5 | RD5                | 0            | O   | DIG      | LATD<5> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<5> data input.                                                 |
|              | AD5 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 5 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 5 input. <sup>(1)</sup>          |
|              | PSP5               | x            | O   | DIG      | PSP read output data (LATD<5>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |
| RD6/AD6/PSP6 | RD6                | 0            | O   | DIG      | LATD<6> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<6> data input.                                                 |
|              | AD6 <sup>(2)</sup> | x            | O   | DIG-3    | External memory interface, address/data bit 6 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 6 input. <sup>(1)</sup>          |
|              | PSP6               | x            | O   | DIG      | PSP read output data (LATD<6>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |

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

**Note 1:** External memory interface I/O takes priority over all other digital and PSP I/O.

**2:** Available on 80-pin devices only.

# PIC18F85J11 FAMILY

**TABLE 10-9: PORTD FUNCTIONS (CONTINUED)**

| Pin Name     | Function           | TRIS Setting | I/O | I/O Type | Description                                                          |
|--------------|--------------------|--------------|-----|----------|----------------------------------------------------------------------|
| RD7/AD7/PSP7 | RD7                | 0            | O   | DIG      | LATD<7> data output.                                                 |
|              |                    | 1            | I   | ST       | PORTD<7> data input.                                                 |
|              | AD7 <sup>(2)</sup> | x            | O   | DIG      | External memory interface, address/data bit 7 output. <sup>(1)</sup> |
|              |                    | x            | I   | TTL      | External memory interface, data bit 7 input. <sup>(1)</sup>          |
|              | PSP7               | x            | O   | DIG      | PSP read output data (LATD<7>); takes priority over port data.       |
|              |                    | x            | I   | TTL      | PSP write data input.                                                |

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

**Note 1:** External memory interface I/O takes priority over all other digital and PSP I/O.

**2:** Available on 80-pin devices only.

**TABLE 10-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD**

| Name  | Bit 7  | Bit 6  | Bit 5               | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|--------|--------|---------------------|--------|--------|--------|--------|--------|----------------------|
| PORTD | RD7    | RD6    | RD5                 | RD4    | RD3    | RD2    | RD1    | RD0    | 54                   |
| LATD  | LATD7  | LATD6  | LATD5               | LATD4  | LATD3  | LATD2  | LATD1  | LATD0  | 54                   |
| TRISD | TRISD7 | TRISD6 | TRISD5              | TRISD4 | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 54                   |
| PORTG | RDPU   | REPU   | RJPU <sup>(1)</sup> | RG4    | RG3    | RG2    | RG1    | RG0    | 54                   |

**Legend:** Shaded cells are not used by PORTD.

**Note 1:** Unimplemented on 64-pin devices, read as '0'.

# PIC18F85J11 FAMILY

## 10.6 PORTE, TRISE and LATE Registers

PORTE is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISE and LATE. All pins on PORTE are digital only and tolerate voltages up to 5.5V.

All pins on PORTE are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output. The RE7 pin is also configurable for open-drain output when CCP2 is active on this pin. Open-drain configuration is selected by setting the CCP2OD control bit (TRISG<6>).

**Note:** These pins are configured as digital inputs on any device Reset.

Each of the PORTE pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is performed by setting bit REPU (PORTG<6>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

On 80-pin devices, PORTE is multiplexed with the system bus as part of the external memory interface. I/O port and other functions are only available when the interface is disabled by setting the EBDIS bit (MEMCON<7>). When the interface is enabled, PORTE is the high-order byte of the multiplexed address/data bus (AD15:AD8). The TRISE bits are also overridden.

When the Parallel Slave Port is active on PORTD, three of the PORTE pins (RE0, RE1 and RE2) are configured as digital control inputs for the port. The control functions are summarized in Table 10-11. The reconfiguration occurs automatically when the PSPMODE control bit (PSPCON<4>) is set. Users must still make certain the corresponding TRISE bits are set to configure these pins as digital inputs.

RE7 can also be configured as the alternate peripheral pin for the CCP2 module. This is done by clearing the CCP2MX Configuration bit.

### EXAMPLE 10-5: INITIALIZING PORTE

```
CLRF    PORTE    ; Initialize PORTE by
                ; clearing output
                ; data latches
CLRF    LATE     ; Alternate method
                ; to clear output
                ; data latches
MOVLW   03h      ; Value used to
                ; initialize data
                ; direction
MOVWF   TRISE    ; Set RE<1:0> as inputs
                ; RE<7:2> as outputs
```



# PIC18F85J11 FAMILY

**TABLE 10-11: PORTE FUNCTIONS**

| Pin Name                  | Function            | TRIS Setting | I/O | I/O Type | Description                                                             |
|---------------------------|---------------------|--------------|-----|----------|-------------------------------------------------------------------------|
| RE0/RD/AD8                | RE0                 | 0            | O   | DIG      | LATE<0> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<0> data input.                                                    |
|                           | R $\overline{D}$    | 1            | I   | TTL      | Parallel Slave Port read enable control input.                          |
|                           | AD8 <sup>(1)</sup>  | x            | O   | DIG      | External memory interface, address/data bit 8 output. <sup>(2)</sup>    |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 8 input. <sup>(2)</sup>             |
| RE1/W $\overline{R}$ /AD9 | RE1                 | 0            | O   | DIG      | LATE<1> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<1> data input.                                                    |
|                           | W $\overline{R}$    | 1            | I   | TTL      | Parallel Slave Port write enable control input.                         |
|                           | AD9 <sup>(1)</sup>  | x            | O   | DIG      | External memory interface, address/data bit 9 output. <sup>(2)</sup>    |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 9 input. <sup>(2)</sup>             |
| RE2/AD10/C $\overline{S}$ | RE2                 | 0            | O   | DIG      | LATE<2> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<2> data input.                                                    |
|                           | AD10 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 10 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 10 input. <sup>(2)</sup>            |
|                           | C $\overline{S}$    | 1            | I   | TTL      | Parallel Slave Port chip select control input.                          |
| RE3/AD11                  | RE3                 | 0            | O   | DIG      | LATE<3> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<3> data input.                                                    |
|                           | AD11 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 11 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 11 input. <sup>(2)</sup>            |
| RE4/AD12                  | RE4                 | 0            | O   | DIG      | LATE<4> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<4> data input.                                                    |
|                           | AD12 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 12 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 12 input. <sup>(2)</sup>            |
| RE5/AD13                  | RE5                 | 0            | O   | DIG      | LATE<5> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<5> data input.                                                    |
|                           | AD13 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 13 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 13 input. <sup>(2)</sup>            |
| RE6/AD14                  | RE6                 | 0            | O   | DIG      | LATE<6> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<6> data input.                                                    |
|                           | AD14 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 14 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 14 input. <sup>(2)</sup>            |
| RE7/AD15/<br>CCP2         | RE7                 | 0            | O   | DIG      | LATE<7> data output.                                                    |
|                           |                     | 1            | I   | ST       | PORTE<7> data input.                                                    |
|                           | AD15 <sup>(1)</sup> | x            | O   | DIG      | External memory interface, address/data bit 15 output. <sup>(2)</sup>   |
|                           |                     | x            | I   | TTL      | External memory interface, data bit 15 input. <sup>(2)</sup>            |
|                           | CCP2 <sup>(3)</sup> | 0            | O   | DIG      | CCP2 compare output and CCP2 PWM output; takes priority over port data. |
|                           |                     | 1            | I   | ST       | CCP2 capture input.                                                     |

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

**Note 1:** Available on 80-pin devices only.

**2:** External memory interface I/O takes priority over all other digital and PSP I/O.

**3:** Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared (all devices in Microcontroller mode).

# PIC18F85J11 FAMILY

TABLE 10-12: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

| Name  | Bit 7  | Bit 6  | Bit 5               | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|--------|--------|---------------------|--------|--------|--------|--------|--------|----------------------|
| PORTE | RE7    | RE6    | RE5                 | RE4    | RE3    | RE2    | RE1    | RE0    | 54                   |
| LATE  | LATE7  | LATE6  | LATE5               | LATE4  | LATE3  | LATE2  | LATE1  | LATE0  | 54                   |
| TRISE | TRISE7 | TRISE6 | TRISE5              | TRISE4 | TRISE3 | TRISE2 | TRISE1 | TRISE0 | 54                   |
| PORTG | RDPU   | REPU   | RJPU <sup>(1)</sup> | RG4    | RG3    | RG2    | RG1    | RG0    | 54                   |
| TRISG | SPIOD  | CCP2OD | CCP1OD              | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 | 54                   |

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

**Note 1:** Unimplemented on 64-pin devices, read as '0'.

## 10.7 PORTF, LATF and TRISF Registers

PORTF is a 7-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISF and LATF. All pins on PORTF are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

PORTF is multiplexed with analog peripheral functions. Pins RF1 through RF6 may be used as comparator inputs or outputs by setting the appropriate bits in the CMCON register. To use RF6:RF3 as digital inputs, it is also necessary to turn off the comparators.

**Note 1:** On device Resets, pins RF6:RF1 are configured as analog inputs and are read as '0'.

**2:** To configure PORTF as digital I/O, turn off comparators and set ADCON1 value.

### EXAMPLE 10-6: INITIALIZING PORTF

```
CLRF    PORTF    ; Initialize PORTF by
                  ; clearing output
                  ; data latches
CLRF    LATF     ; Alternate method
                  ; to clear output
                  ; data latches

MOVLW   07h      ;
MOVWF   CMCON    ; Turn off comparators
MOVLW   0Fh;
MOVWF   ADCON1   ; Set PORTF as digital I/O
MOVLW   0CEh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISF    ; Set RF3:RF1 as inputs
                  ; RF5:RF4 as outputs
                  ; RF7:RF6 as inputs
```

# PIC18F85J11 FAMILY

**TABLE 10-13: PORTF FUNCTIONS**

| Pin Name       | Function | TRIS Setting | I/O | I/O Type | Description                                                                                                        |
|----------------|----------|--------------|-----|----------|--------------------------------------------------------------------------------------------------------------------|
| RF1/AN6/C2OUT  | RF1      | 0            | O   | DIG      | LATF<1> data output; not affected by analog input.                                                                 |
|                |          | 1            | I   | ST       | PORTF<1> data input; disabled when analog input enabled.                                                           |
|                | AN6      | 1            | I   | ANA      | A/D input channel 6. Default configuration on POR.                                                                 |
|                | C2OUT    | 0            | O   | DIG      | Comparator 2 output; takes priority over port data.                                                                |
| RF2/AN7/C1OUT  | RF2      | 0            | O   | DIG      | LATF<2> data output; not affected by analog input.                                                                 |
|                |          | 1            | I   | ST       | PORTF<2> data input; disabled when analog input enabled.                                                           |
|                | AN7      | 1            | I   | ANA      | A/D input channel 7. Default configuration on POR.                                                                 |
|                | C1OUT    | 0            | O   | TTL      | Comparator 1 output; takes priority over port data.                                                                |
| RF3/AN8        | RF3      | 0            | O   | DIG      | LATF<3> data output; not affected by analog input.                                                                 |
|                |          | 1            | I   | ST       | PORTF<3> data input; disabled when analog input enabled.                                                           |
|                | AN8      | 1            | I   | ANA      | A/D input channel 8 and Comparator C2+ input. Default input configuration on POR; not affected by analog output.   |
| RF4/AN9        | RF4      | 0            | O   | DIG      | LATF<4> data output; not affected by analog input.                                                                 |
|                |          | 1            | I   | ST       | PORTF<4> data input; disabled when analog input enabled.                                                           |
|                | AN9      | 1            | I   | ANA      | A/D input channel 9 and Comparator C2- input. Default input configuration on POR; does not affect digital output.  |
| RF5/AN10/CVREF | RF5      | 0            | O   | DIG      | LATF<5> data output; not affected by analog input. Disabled when CVREF output enabled.                             |
|                |          | 1            | I   | ST       | PORTF<5> data input; disabled when analog input enabled. Disabled when CVREF output enabled.                       |
|                | AN10     | 1            | I   | ANA      | A/D input channel 10 and Comparator C1+ input. Default input configuration on POR.                                 |
|                | CVREF    | x            | O   | ANA      | Comparator voltage reference output. Enabling this feature disables digital I/O.                                   |
| RF6/AN11       | RF6      | 0            | O   | DIG      | LATF<6> data output; not affected by analog input.                                                                 |
|                |          | 1            | I   | ST       | PORTF<6> data input; disabled when analog input enabled.                                                           |
|                | AN11     | 1            | I   | ANA      | A/D input channel 11 and Comparator C1- input. Default input configuration on POR; does not affect digital output. |
| RF7/AN5/SS     | RF7      | 0            | O   | DIG      | LATF<7> data output.                                                                                               |
|                |          | 1            | I   | ST       | PORTF<7> data input.                                                                                               |
|                | AN5      | 1            | I   | ANA      | A/D input channel 5. Default configuration on POR.                                                                 |
|                | SS       | 1            | I   | TTL      | Slave select input for MSSP module.                                                                                |

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

**TABLE 10-14: SUMMARY OF REGISTERS ASSOCIATED WITH PORTF**

| Name   | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0 | Reset Values on page |
|--------|--------|--------|--------|--------|--------|--------|--------|-------|----------------------|
| PORTF  | RF7    | RF6    | RF5    | RF4    | RF3    | RF2    | RF1    | —     | 54                   |
| LATF   | LATF7  | LATF6  | LATF5  | LATF4  | LATF3  | LATF2  | LATF1  | —     | 54                   |
| TRISF  | TRISF7 | TRISF6 | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1 | —     | 54                   |
| ADCON1 | —      | —      | VCFG1  | VCFG0  | PCFG3  | PCFG2  | PCFG1  | PCFG0 | 53                   |
| CMCON  | C2OUT  | C1OUT  | C2INV  | C1INV  | CIS    | CM2    | CM1    | CM0   | 53                   |
| CVRCON | CVREN  | CVROE  | CVRR   | CVRSS  | CVR3   | CVR2   | CVR1   | CVR0  | 53                   |

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

## 10.8 PORTG, TRISG and LATG Registers

PORTG is a 5-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISG and LATG. All pins on PORTG are digital only and tolerate voltages up to 5.5V.

When operating as I/O, all PORTG pins have Schmitt Trigger input buffers. Pins RG1 and RG2 are multiplexed with the AUSART module. The RG1 pin is also configurable for open-drain output when the AUSART is active. Open-drain configuration is selected by setting the U2OD control bit (LATG<7>).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTG pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. The user should refer to the corresponding peripheral section for the correct TRIS bit settings. The pin override value is not loaded into the TRIS register. This allows read-modify-write of the TRIS register without concern due to peripheral overrides.

Although the port itself is only five bits wide, the PORTG<7:5> bits are still implemented to control the weak pull-ups on the I/O ports associated with PORTD, PORTE and PORTJ. Setting these bits enables the respective port pull-ups.

Most of the corresponding TRISG and LATG bits are implemented as open-drain control bits for CCP1, CCP2 and SPI (TRISG<7:5>), and the USARTs (LATG<7:6>). Setting these bits configures the output pin for the corresponding peripheral for open-drain operation. LATG<5> is not implemented.

### EXAMPLE 10-7: INITIALIZING PORTG

```
CLRF    PORTG    ; Initialize PORTG by
                  ; clearing output
                  ; data latches
CLRF    LATG      ; Alternate method
                  ; to clear output
                  ; data latches
MOVLW   04h      ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISG    ; Set RG1:RG0 as outputs
                  ; RG2 as input
                  ; RG4:RG3 as inputs
```

# PIC18F85J11 FAMILY

**TABLE 10-15: PORTG FUNCTIONS**

| Pin Name    | Function | TRIS Setting | I/O | I/O Type | Description                                                                      |
|-------------|----------|--------------|-----|----------|----------------------------------------------------------------------------------|
| RG0         | RG0      | 0            | O   | DIG      | LATG<0> data output.                                                             |
|             |          | 1            | I   | ST       | PORTG<0> data input.                                                             |
| RG1/TX2/CK2 | R21      | 0            | O   | DIG      | LATG<1> data output.                                                             |
|             |          | 1            | I   | ST       | PORTG<1> data input.                                                             |
|             | TX2      | 1            | O   | DIG      | Synchronous serial data output (AUSART2 module); takes priority over port data.  |
|             |          | 1            | O   | DIG      | Synchronous serial data input (AUSART2 module). User must configure as an input. |
|             | CK2      | 1            | O   | DIG      | Synchronous serial clock input (AUSART2 module).                                 |
|             |          | 1            | I   | ST       | Synchronous serial clock input (AUSART2 module).                                 |
| RG2/RX2/DT2 | RG2      | 0            | O   | DIG      | LATG<2> data output.                                                             |
|             |          | 1            | I   | ST       | PORTG<2> data input.                                                             |
|             | RX2      | 1            | I   | ST       | Asynchronous serial receive data input (AUSART2 module).                         |
|             |          | 1            | O   | DIG      | Synchronous serial data output (AUSART2 module); takes priority over port data.  |
|             | DT2      | 1            | O   | DIG      | Synchronous serial data output (AUSART2 module); takes priority over port data.  |
|             |          | 1            | I   | ST       | Synchronous serial data input (AUSART2 module). User must configure as an input. |
| RG3         | RG3      | 0            | O   | DIG      | LATG<3> data output.                                                             |
|             |          | 1            | I   | ST       | PORTG<3> data input.                                                             |
| RG4         | RG4      | 0            | O   | DIG      | LATG<4> data output.                                                             |
|             |          | 1            | I   | ST       | PORTG<4> data input.                                                             |

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

**TABLE 10-16: SUMMARY OF REGISTERS ASSOCIATED WITH PORTG**

| Name  | Bit 7            | Bit 6            | Bit 5                           | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|------------------|------------------|---------------------------------|--------|--------|--------|--------|--------|----------------------|
| PORTG | RDP <sub>U</sub> | REP <sub>U</sub> | RJP <sub>U</sub> <sup>(1)</sup> | RG4    | RG3    | RG2    | RG1    | RG0    | 54                   |
| LATG  | U2OD             | U1OD             | —                               | LATG4  | LATG3  | LATG2  | LATG1  | LATG0  | 54                   |
| TRISG | SPIOD            | CCP2OD           | CCP1OD                          | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 | 54                   |

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

**Note 1:** Unimplemented on 64-pin devices, read as '0'.

# PIC18F85J11 FAMILY

## 10.9 PORTH, LATH and TRISH Registers

**Note:** PORTH is available only on 80-pin devices.

PORTH is an 8-bit wide, bidirectional I/O port. The corresponding Data Direction and Output Latch registers are TRISH and LATH. All pins are digital only and tolerate voltages up to 5.5V.

All pins on PORTH are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

When the external memory interface is enabled, four of the PORTH pins function as the high-order address lines for the interface. The address output from the interface takes priority over other digital I/O. The corresponding TRISH bits are also overridden.

### EXAMPLE 10-8: INITIALIZING PORTH

```
CLRF    PORTH    ; Initialize PORTH by
                  ; clearing output
                  ; data latches

CLRF    LATH      ; Alternate method
                  ; to clear output
                  ; data latches

MOVLW   0Fh      ; Configure PORTH as
MOVWF   ADCON1    ; digital I/O
MOVLW   0CFh      ; Value used to
                  ; initialize data
                  ; direction

MOVWF   TRISH     ; Set RH3:RH0 as inputs
                  ; RH5:RH4 as outputs
                  ; RH7:RH6 as inputs
```

**TABLE 10-17: PORTH FUNCTIONS**

| Pin Name | Function | TRIS Setting | I/O | I/O Type | Description                                                                |
|----------|----------|--------------|-----|----------|----------------------------------------------------------------------------|
| RH0/A16  | RH0      | 0            | O   | DIG      | LATH<0> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<0> data input.                                                       |
|          | A16      | x            | O   | DIG      | External memory interface, address line 16. Takes priority over port data. |
| RH1/A17  | RH1      | 0            | O   | DIG      | LATH<1> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<1> data input.                                                       |
|          | A17      | x            | O   | DIG      | External memory interface, address line 17. Takes priority over port data. |
| RH2/A18  | RH2      | 0            | O   | DIG      | LATH<2> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<2> data input.                                                       |
|          | A18      | x            | O   | DIG      | External memory interface, address line 18. Takes priority over port data. |
| RH3/A19  | RH3      | 0            | O   | DIG      | LATH<3> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<3> data input.                                                       |
|          | A19      | x            | O   | DIG      | External memory interface, address line 19. Takes priority over port data. |
| RH4      | RH4      | 0            | O   | DIG      | LATH<4> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<4> data input.                                                       |
| RH5      | RH5      | 0            | O   | DIG      | LATH<5> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<5> data input.                                                       |
| RH6      | RH6      | 0            | O   | DIG      | LATH<6> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<6> data input.                                                       |
| RH7      | RH7      | 0            | O   | DIG      | LATH<7> data output.                                                       |
|          |          | 1            | I   | ST       | PORTH<7> data input.                                                       |

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

**TABLE 10-18: SUMMARY OF REGISTERS ASSOCIATED WITH PORTH**

| Name  | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|----------------------|
| PORTH | RH7    | RH6    | RH5    | RH4    | RH3    | RH2    | RH1    | RH0    | 54                   |
| LATH  | LATH7  | LATH6  | LATH5  | LATH4  | LATH3  | LATH2  | LATH1  | LATH0  | 54                   |
| TRISH | TRISH7 | TRISH6 | TRISH5 | TRISH4 | TRISH3 | TRISH2 | TRISH1 | TRISH0 | 54                   |

# PIC18F85J11 FAMILY

## 10.10 PORTJ, TRISJ and LATJ Registers

**Note:** PORTJ is available only on 80-pin devices.

PORTJ is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISJ and LATJ. All pins on PORTJ are digital only and tolerate voltages up to 5.5V.

All pins on PORTJ are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

**Note:** These pins are configured as digital inputs on any device Reset.

Each of the PORTJ pins has a weak internal pull-up. The pull-ups are provided to keep the inputs at a known state for the external memory interface while powering up. A single control bit can turn off all the pull-ups. This is performed by clearing bit, RJPU (PORTG<5>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

When the external memory interface is enabled, all of the PORTJ pins function as control outputs for the interface. This occurs automatically when the interface is enabled by clearing the EBDIS control bit (MEMCON<7>). The TRISJ bits are also overridden.

### EXAMPLE 10-9: INITIALIZING PORTJ

```
CLRF    PORTJ    ; Initialize PORTJ by
                  ; clearing output latches
CLRF    LATJ      ; Alternate method
                  ; to clear output latches
MOVLW   0CFh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISJ    ; Set RJ3:RJ0 as inputs
                  ; RJ5:RJ4 as output
                  ; RJ7:RJ6 as inputs
```



# PIC18F85J11 FAMILY

**TABLE 10-19: PORTJ FUNCTIONS**

| Pin Name                     | Function                | TRIS Setting | I/O | I/O Type | Description                                                                                     |
|------------------------------|-------------------------|--------------|-----|----------|-------------------------------------------------------------------------------------------------|
| RJ0/ALE                      | RJ0                     | 0            | O   | DIG      | LATJ<0> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<0> data input.                                                                            |
|                              | ALE                     | x            | O   | DIG      | External memory interface address latch enable control output; takes priority over digital I/O. |
| RJ1/ $\overline{\text{OE}}$  | RJ1                     | 0            | O   | DIG      | LATJ<1> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<1> data input.                                                                            |
|                              | $\overline{\text{OE}}$  | x            | O   | DIG      | External memory interface output enable control output; takes priority over digital I/O.        |
| RJ2/ $\overline{\text{WRL}}$ | RJ2                     | 0            | O   | DIG      | LATJ<2> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<2> data input.                                                                            |
|                              | $\overline{\text{WRL}}$ | x            | O   | DIG      | External memory bus write low byte control; takes priority over digital I/O.                    |
| RJ3/ $\overline{\text{WRH}}$ | RJ3                     | 0            | O   | DIG      | LATJ<3> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<3> data input.                                                                            |
|                              | $\overline{\text{WRH}}$ | x            | O   | DIG      | External memory interface write high byte control output; takes priority over digital I/O.      |
| RJ4/BA0                      | RJ4                     | 0            | O   | DIG      | LATJ<4> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<4> data input.                                                                            |
|                              | BA0                     | x            | O   | DIG      | External memory interface byte address 0 control output; takes priority over digital I/O.       |
| RJ5/ $\overline{\text{CE}}$  | RJ5                     | 0            | O   | DIG      | LATJ<5> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<5> data input.                                                                            |
|                              | $\overline{\text{CE}}$  | x            | O   | DIG      | External memory interface chip enable control output; takes priority over digital I/O.          |
| RJ6/ $\overline{\text{LB}}$  | RJ6                     | 0            | O   | DIG      | LATJ<6> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<6> data input.                                                                            |
|                              | $\overline{\text{LB}}$  | x            | O   | DIG      | External memory interface lower byte enable control output; takes priority over digital I/O.    |
| RJ7/ $\overline{\text{UB}}$  | RJ7                     | 0            | O   | DIG      | LATJ<7> data output.                                                                            |
|                              |                         | 1            | I   | ST       | PORTJ<7> data input.                                                                            |
|                              | $\overline{\text{UB}}$  | x            | O   | DIG      | External memory interface upper byte enable control output; takes priority over digital I/O.    |

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

**TABLE 10-20: SUMMARY OF REGISTERS ASSOCIATED WITH PORTJ**

| Name  | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|----------------------|
| PORTJ | RJ7    | RJ6    | RJ5    | RJ4    | RJ3    | RJ2    | RJ1    | RJ0    | 54                   |
| LATJ  | LATJ7  | LATJ6  | LATJ5  | LATJ4  | LATJ3  | LATJ2  | LATJ1  | LATJ0  | 54                   |
| TRISJ | TRISJ7 | TRISJ6 | TRISJ5 | TRISJ4 | TRISJ3 | TRISJ2 | TRISJ1 | TRISJ0 | 54                   |
| PORTG | RDPU   | REPU   | RJPU   | RG4    | RG3    | RG2    | RG1    | RG0    | 54                   |

**Legend:** Shaded cells are not used by PORTJ.

# PIC18F85J11 FAMILY

## 10.11 Parallel Slave Port

PORTD can also function as an 8-bit wide Parallel Slave Port, or microprocessor port, when control bit PSPMODE (PSPCON<4>) is set. It is asynchronously readable and writable by the external world through RD control input pin, RE0/RD, and WR control input pin, RE1/WR.

**Note:** For 80-pin devices, the Parallel Slave Port is available only in Microcontroller mode.

The PSP can directly interface to an 8-bit microprocessor data bus. The external microprocessor can read or write the PORTD latch as an 8-bit latch. Setting bit PSPMODE enables port pin, RE0/RD, to be the RD input, RE1/WR to be the WR input and RE2/CS to be the CS (Chip Select) input. For this functionality, the corresponding data direction bits of the TRISE register (TRISE<2:0>) must be configured as inputs (set).

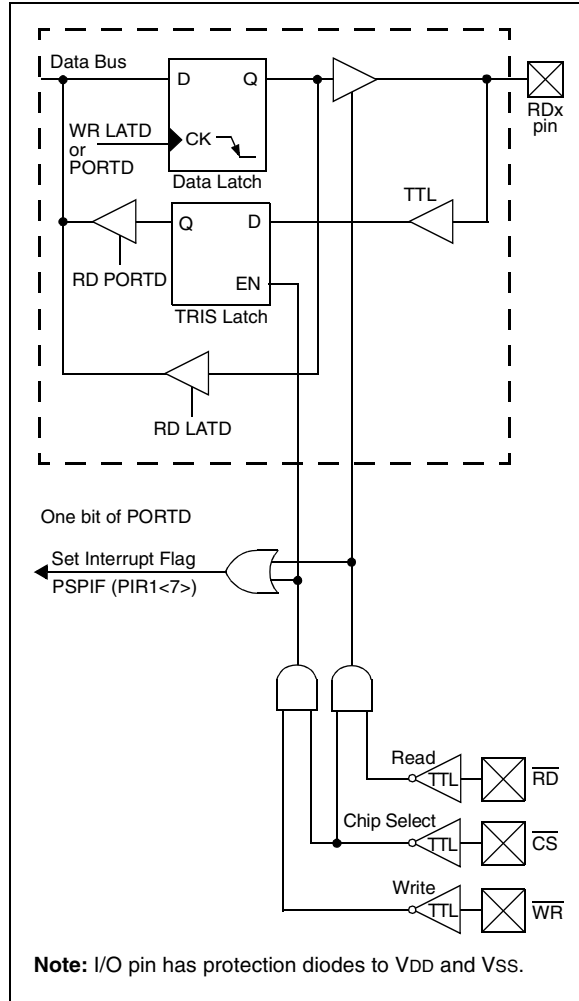
A write to the PSP occurs when both the  $\overline{\text{CS}}$  and  $\overline{\text{WR}}$  lines are first detected low and ends when either are detected high. The PSPIF and IBF flag bits are both set when the write ends.

A read from the PSP occurs when both the  $\overline{CS}$  and  $RD$  lines are first detected low. The data in PORTD is read out and the OBF bit is set. If the user writes new data to PORTD to set OBF, the data is immediately read out; however, the OBF bit is not set.

When either the  $\overline{\text{CS}}$  or  $\overline{\text{RD}}$  lines are detected high, the PORTD pins return to the input state and the PSPIF bit is set. User applications should wait for PSPIF to be set before servicing the PSP. When this happens, the IBF and OBF bits can be polled and the appropriate action taken.

The timing for the control signals in Write and Read modes is shown in Figure 10-4 and Figure 10-5, respectively.

**FIGURE 10-3: PORTD AND PORTE BLOCK DIAGRAM (PARALLEL SLAVE PORT)**



# PIC18F85J11 FAMILY

**REGISTER 10-1: PSPCON: PARALLEL SLAVE PORT CONTROL REGISTER**

| R-0   | R-0 | R/W-0 | R/W-0   | U-0 | U-0 | U-0 | U-0   |
|-------|-----|-------|---------|-----|-----|-----|-------|
| IBF   | OBF | IBOV  | PSPMODE | —   | —   | —   | —     |
| 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 **IBF:** Input Buffer Full Status bit

1 = A word has been received and is waiting to be read by the CPU

0 = No word has been received

bit 6 **OBF:** Output Buffer Full Status bit

1 = The output buffer still holds a previously written word

0 = The output buffer has been read

bit 5 **IBOV:** Input Buffer Overflow Detect bit

1 = A write occurred when a previously input word has not been read (must be cleared in software)

0 = No overflow occurred

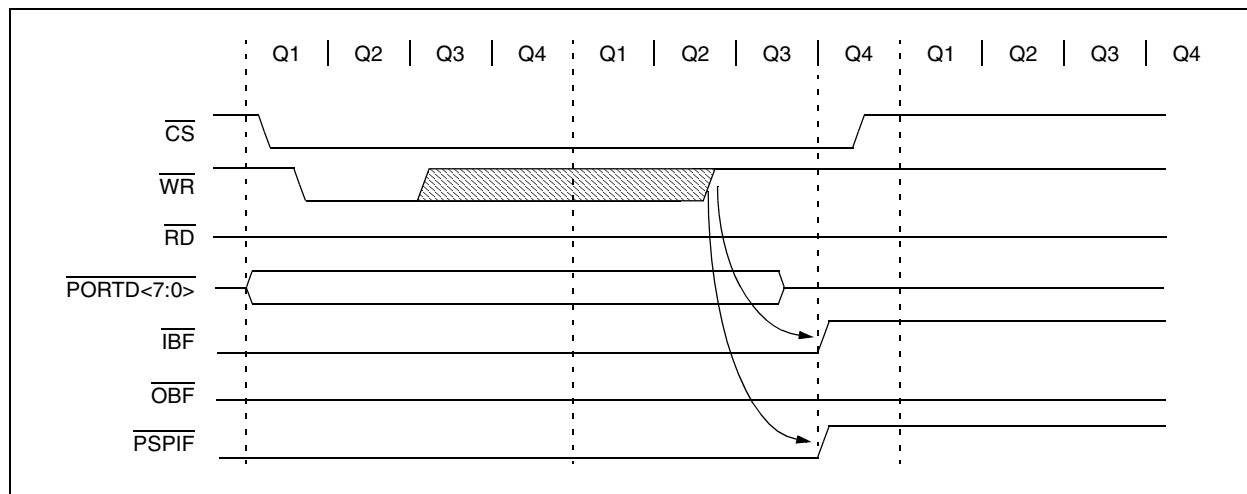
bit 4 **PSPMODE:** Parallel Slave Port Mode Select bit

1 = Parallel Slave Port mode

0 = General Purpose I/O mode

bit 3-0 **Unimplemented:** Read as '0'

**FIGURE 10-4: PARALLEL SLAVE PORT WRITE WAVEFORMS**



# PIC18F85J11 FAMILY

**FIGURE 10-5: PARALLEL SLAVE PORT READ WAVEFORMS**

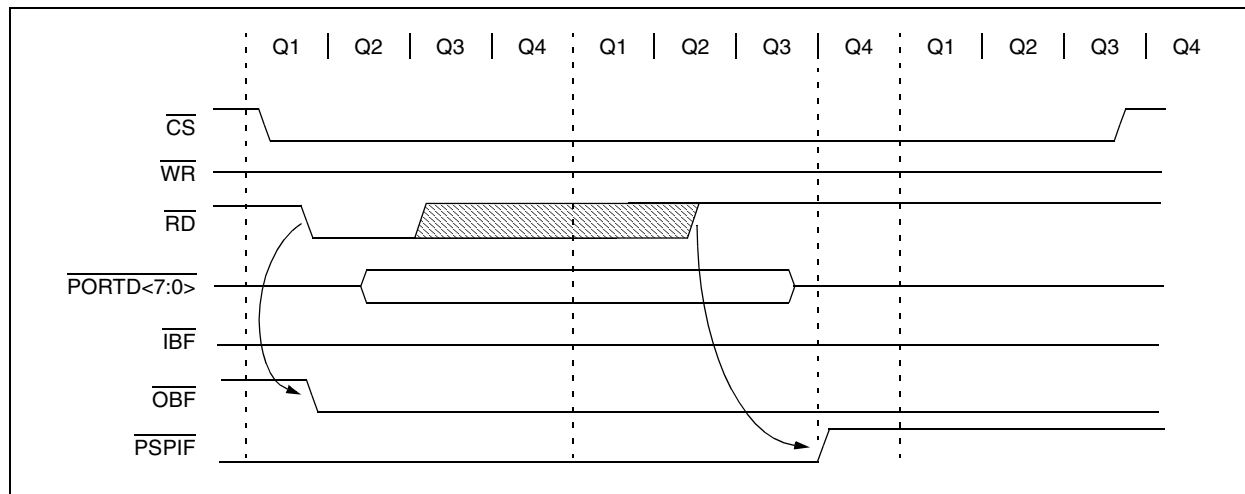


**TABLE 10-21: REGISTERS ASSOCIATED WITH PARALLEL SLAVE PORT**

| Name   | Bit 7    | Bit 6     | Bit 5  | Bit 4   | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|--------|----------|-----------|--------|---------|--------|--------|--------|--------|----------------------|
| PORTD  | RD7      | RD6       | RD5    | RD4     | RD3    | RD2    | RD1    | RD0    | 54                   |
| LATD   | LATD7    | LATD6     | LATD5  | LATD4   | LATD3  | LATD2  | LATD1  | LATD0  | 54                   |
| TRISD  | TRISD7   | TRISD6    | TRISD5 | TRISD4  | TRISD3 | TRISD2 | TRISD1 | TRISD0 | 54                   |
| PORTE  | RE7      | RE6       | RE5    | RE4     | RE3    | RE2    | RE1    | RE0    | 54                   |
| LATE   | LATE7    | LATE6     | LATE5  | LATE4   | LATE3  | LATE2  | LATE1  | LATE0  | 54                   |
| TRISE  | TRISE7   | TRISE6    | TRISE5 | TRISE4  | TRISE3 | TRISE2 | TRISE1 | TRISE0 | 54                   |
| PSPCON | IBF      | OBF       | IBOV   | PSPMODE | —      | —      | —      | —      | 53                   |
| INTCON | GIE/GIEH | PEIE/GIEL | TMR0IE | INT0IE  | RBIE   | TMR0IF | INT0IF | RBIF   | 51                   |
| PIR1   | PSPIF    | ADIF      | RC1IF  | TX1IF   | SSP1IF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1   | PSPIE    | ADIE      | RC1IE  | TX1IE   | SSP1IE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1   | PSPIP    | ADIP      | RC1IP  | TX1IP   | SSP1IP | —      | TMR2IP | TMR1IP | 53                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Parallel Slave Port.

## 11.0 TIMER0 MODULE

The Timer0 module incorporates the following features:

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes
- Readable and writable registers
- Dedicated, 8-bit software programmable prescaler
- Selectable clock source (internal or external)
- Edge select for external clock
- Interrupt on overflow

The T0CON register (Register 11-1) controls all aspects of the module's operation, including the prescale selection; it is both readable and writable.

A simplified block diagram of the Timer0 module in 8-bit mode is shown in Figure 11-1. Figure 11-2 shows a simplified block diagram of the Timer0 module in 16-bit mode.

### REGISTER 11-1: T0CON: TIMER0 CONTROL REGISTER

| R/W-1  | R/W-1  | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|--------|--------|-------|-------|-------|-------|-------|-------|
| TMR0ON | T08BIT | T0CS  | T0SE  | PSA   | T0PS2 | T0PS1 | T0PS0 |
| 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 **TMR0ON:** Timer0 On/Off Control bit

1 = Enables Timer0

0 = Stops Timer0

bit 6 **T08BIT:** Timer0 8-Bit/16-Bit Control bit

1 = Timer0 is configured as an 8-bit timer/counter

0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS:** Timer0 Clock Source Select bit

1 = Transition on T0CKI pin

0 = Internal instruction cycle clock (CLKO)

bit 4 **T0SE:** Timer0 Source Edge Select bit

1 = Increment on high-to-low transition on T0CKI pin

0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Timer0 Prescaler Assignment bit

1 = Timer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.

0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits

111 = 1:256 Prescale value

110 = 1:128 Prescale value

101 = 1:64 Prescale value

100 = 1:32 Prescale value

011 = 1:16 Prescale value

010 = 1:8 Prescale value

001 = 1:4 Prescale value

000 = 1:2 Prescale value

# PIC18F85J11 FAMILY

## 11.1 Timer0 Operation

Timer0 can operate as either a timer or a counter. The mode is selected with the T0CS bit (T0CON<5>). In Timer mode (T0CS = 0), the module increments on every clock by default unless a different prescaler value is selected (see **Section 11.3 “Prescaler”**). If the TMR0 register is written to, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0 register.

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

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

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

## 11.2 Timer0 Reads and Writes in 16-Bit Mode

TMR0H is not the actual high byte of Timer0 in 16-bit mode. It is actually a buffered version of the real high byte of Timer0 which is not directly readable nor writable (refer to Figure 11-2). TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte.

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

**FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)**

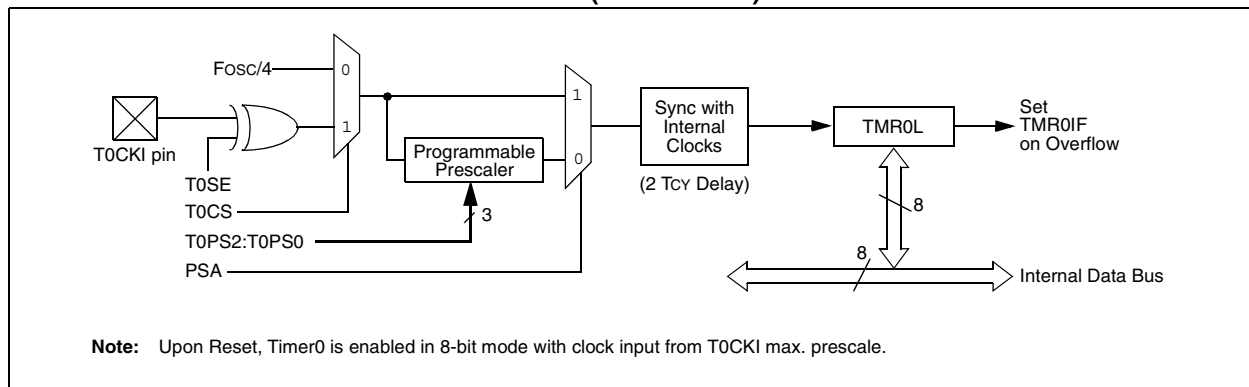
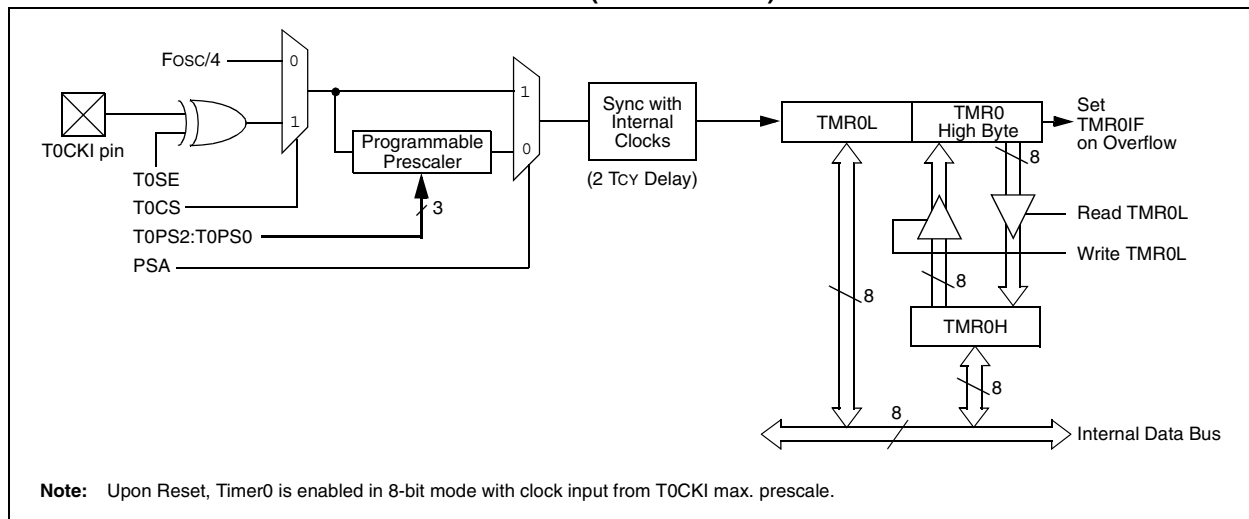


**FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)**



## 11.3 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not directly readable or writable. Its value is set by the PSA and T0PS2:T0PS0 bits (T0CON<3:0>) which determine the prescaler assignment and prescale ratio.

Clearing the PSA bit assigns the prescaler to the Timer0 module. When it is assigned, prescale values from 1:2 through 1:256, in power-of-2 increments, are selectable.

When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF TMR0, MOVWF TMR0, BSF TMR0, etc.) clear the prescaler count.

**Note:** Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

### 11.3.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control and can be changed “on-the-fly” during program execution.

## 11.4 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode, or from FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF flag bit. The interrupt can be masked by clearing the TMR0IE bit (INTCON<5>). Before re-enabling the interrupt, the TMR0IF bit must be cleared in software by the Interrupt Service Routine.

Since Timer0 is shut down in Sleep mode, the TMR0 interrupt cannot awaken the processor from Sleep.

**TABLE 11-1: REGISTERS ASSOCIATED WITH TIMER0**

| Name   | Bit 7                     | Bit 6                 | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0  | Reset Values on page |
|--------|---------------------------|-----------------------|--------|--------|--------|--------|--------|--------|----------------------|
| TMR0L  | Timer0 Register Low Byte  |                       |        |        |        |        |        |        | 52                   |
| TMR0H  | Timer0 Register High Byte |                       |        |        |        |        |        |        | 52                   |
| INTCON | GIE/GIEH                  | PEIE/GIEL             | TMR0IE | INT0IE | RBIE   | TMR0IF | INT0IF | RBIF   | 51                   |
| T0CON  | TMR0ON                    | T08BIT                | T0CS   | T0SE   | PSA    | T0PS2  | T0PS1  | T0PS0  | 52                   |
| TRISA  | TRISA7 <sup>(1)</sup>     | TRISA6 <sup>(1)</sup> | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1 | TRISA0 | 54                   |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used by Timer0.

**Note 1:** RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as ‘0’.

# PIC18F85J11 FAMILY

---

NOTES:



## 12.0 TIMER1 MODULE

The Timer1 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR1H and TMR1L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Reset on CCPx Special Event Trigger
- Device clock status flag (T1RUN)

A simplified block diagram of the Timer1 module is shown in Figure 12-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 12-2.

The module incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation.

Timer1 can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead.

Timer1 is controlled through the T1CON Control register (Register 12-1). It also contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled or disabled by setting or clearing control bit, TMR1ON (T1CON<0>).

### REGISTER 12-1: T1CON: TIMER1 CONTROL REGISTER

| R/W-0 | R-0   | R/W-0   | R/W-0   | R/W-0   | R/W-0  | R/W-0  | R/W-0  |
|-------|-------|---------|---------|---------|--------|--------|--------|
| RD16  | T1RUN | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON |
| 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 **RD16:** 16-Bit Read/Write Mode Enable bit  
 1 = Enables register read/write of Timer1 in one 16-bit operation  
 0 = Enables register read/write of Timer1 in two 8-bit operations
- bit 6 **T1RUN:** Timer1 System Clock Status bit  
 1 = Device clock is derived from Timer1 oscillator  
 0 = Device clock is derived from another source
- bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits  
 11 = 1:8 Prescale value  
 10 = 1:4 Prescale value  
 01 = 1:2 Prescale value  
 00 = 1:1 Prescale value
- bit 3 **T1OSCEN:** Timer1 Oscillator Enable bit  
 1 = Timer1 oscillator is enabled  
 0 = Timer1 oscillator is shut off  
 The oscillator inverter and feedback resistor are turned off to eliminate power drain.
- bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Select bit  
When TMR1CS = 1:  
 1 = Do not synchronize external clock input  
 0 = Synchronize external clock input  
When TMR1CS = 0:  
 This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 **TMR1CS:** Timer1 Clock Source Select bit  
 1 = External clock from pin RC0/T1OSO/T13CKI (on the rising edge)  
 0 = Internal clock (FOSC/4)
- bit 0 **TMR1ON:** Timer1 On bit  
 1 = Enables Timer1  
 0 = Stops Timer1

# PIC18F85J11 FAMILY

## 12.1 Timer1 Operation

Timer1 can operate in one of these modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON<1>). When TMR1CS is cleared (= 0), Timer1 increments on every internal instruction

cycle ( $F_{OSC}/4$ ). When the bit is set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

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

**FIGURE 12-1: TIMER1 BLOCK DIAGRAM (8-BIT MODE)**

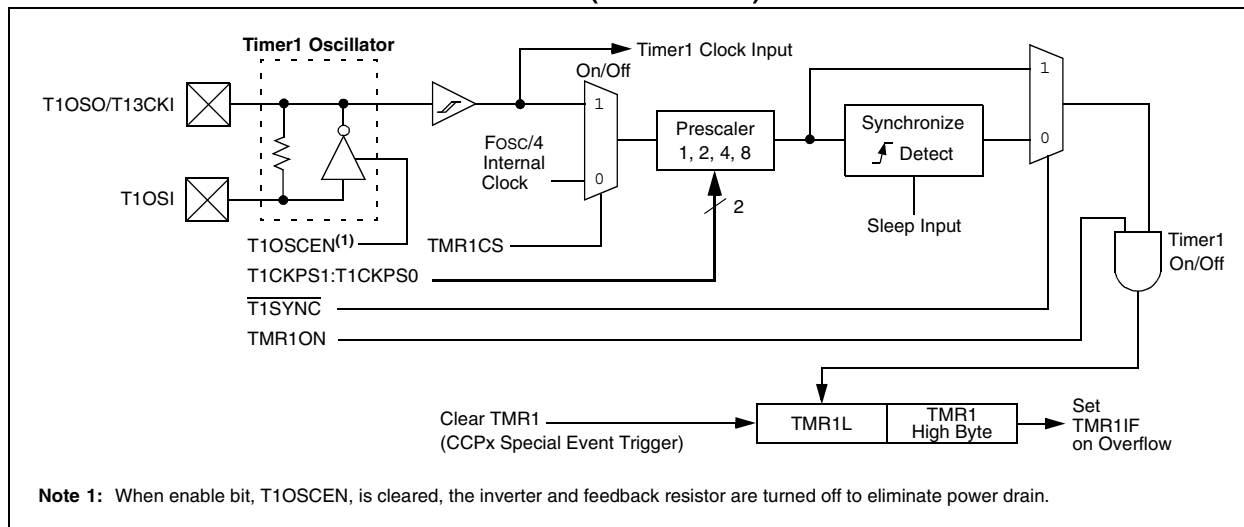
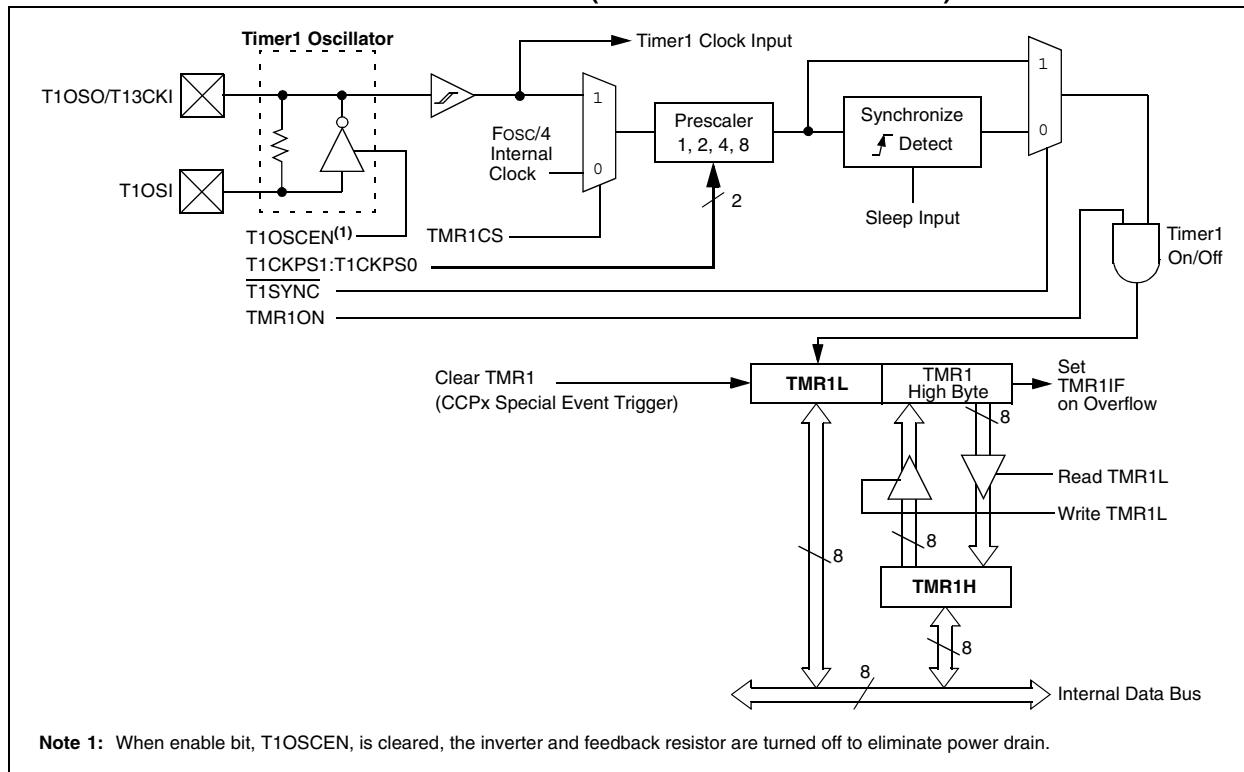


**FIGURE 12-2: TIMER1 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)**



## 12.2 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 12-2). When the RD16 control bit (T1CON<7>) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 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 Timer1 must also take place through the TMR1H Buffer register. The Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

## 12.3 Timer1 Oscillator

An on-chip crystal oscillator circuit is incorporated between pins T1OSI (input) and T1OSO (amplifier output). It is enabled by setting the Timer1 Oscillator Enable bit, T1OSCEN (T1CON<3>). The oscillator is a low-power circuit rated for 32 kHz crystals. It will continue to run during all power-managed modes. The circuit for a typical LP oscillator is shown in Figure 12-3. Table 12-1 shows the capacitor selection for the Timer1 oscillator.

The user must provide a software time delay to ensure proper start-up of the Timer1 oscillator.

**FIGURE 12-3: EXTERNAL COMPONENTS FOR THE TIMER1 LP OSCILLATOR**

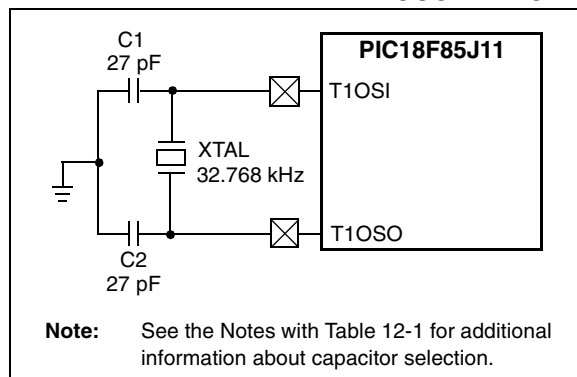


**TABLE 12-1: CAPACITOR SELECTION FOR THE TIMER1 OSCILLATOR<sup>(2,3,4)</sup>**

| Oscillator Type | Freq.      | C1                   | C2                   |
|-----------------|------------|----------------------|----------------------|
| LP              | 32.768 kHz | 27 pF <sup>(1)</sup> | 27 pF <sup>(1)</sup> |

**Note 1:** Microchip suggests these values as a starting point in validating the oscillator circuit.

**2:** Higher capacitance increases the stability of the oscillator but also increases the start-up time.

**3:** Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

**4:** Capacitor values are for design guidance only.

### 12.3.1 USING TIMER1 AS A CLOCK SOURCE

The Timer1 oscillator is also available as a clock source in power-managed modes. By setting the System Clock Select bits, SCS1:SCS0 (OSCCON<1:0>), to '01', the device switches to SEC\_RUN mode; both the CPU and peripherals are clocked from the Timer1 oscillator. If the IDLEN bit (OSCCON<7>) is cleared and a SLEEP instruction is executed, the device enters SEC\_IDLE mode. Additional details are available in **Section 3.0 "Power-Managed Modes"**.

Whenever the Timer1 oscillator is providing the clock source, the Timer1 system clock status flag, T1RUN (T1CON<6>), is set. This can be used to determine the controller's current clocking mode. It can also indicate the clock source being currently used by the Fail-Safe Clock Monitor. If the Fail-Safe Clock Monitor is enabled and the Timer1 oscillator fails while providing the clock, polling the T1RUN bit will indicate whether the clock is being provided by the Timer1 oscillator or another source.

# PIC18F85J11 FAMILY

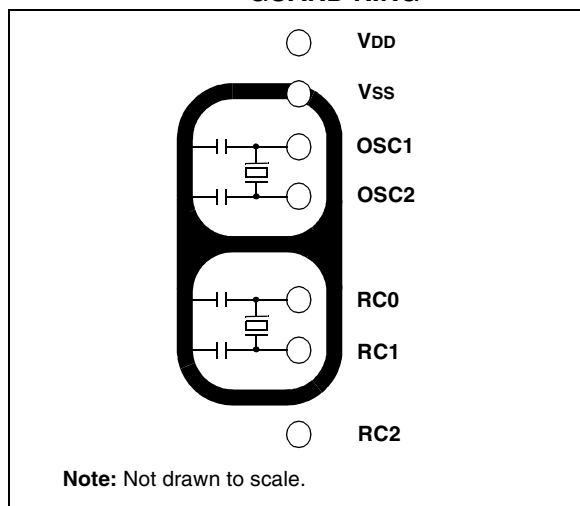
## 12.3.2 TIMER1 OSCILLATOR LAYOUT CONSIDERATIONS

The Timer1 oscillator circuit draws very little power during operation. Due to the low-power nature of the oscillator, it may also be sensitive to rapidly changing signals in close proximity.

The oscillator circuit, shown in Figure 12-3, should be located as close as possible to the microcontroller. There should be no circuits passing within the oscillator circuit boundaries other than VSS or VDD.

If a high-speed circuit must be located near the oscillator (such as the CCP1 pin in Output Compare or PWM mode, or the primary oscillator using the OSC2 pin), a grounded guard ring around the oscillator circuit, as shown in Figure 12-4, may be helpful when used on a single-sided PCB or in addition to a ground plane.

**FIGURE 12-4: OSCILLATOR CIRCUIT WITH GROUNDED GUARD RING**



## 12.4 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The Timer1 interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR1<0>). This interrupt can be enabled or disabled by setting or clearing the Timer1 Interrupt Enable bit, TMR1IE (PIE1<0>).

## 12.5 Resetting Timer1 Using the CCPx Special Event Trigger

If CCP1 or CCP2 is configured to use Timer1 and to generate a Special Event Trigger in Compare mode (CCPxM3:CCPxM0 = 1011), this signal will reset Timer3. The trigger from CCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 15.3.4 “Special Event Trigger”** for more information).

The module must be configured as either a timer or a synchronous counter to take advantage of this feature. When used this way, the CCPRxH:CCPRxL register pair effectively becomes a Period register for Timer1.

If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work.

In the event that a write to Timer1 coincides with a Special Event Trigger, the write operation will take precedence.

**Note:** The Special Event Triggers from the CCPx module will not set the TMR1IF interrupt flag bit (PIR1<0>).

## 12.6 Using Timer1 as a Real-Time Clock

Adding an external LP oscillator to Timer1 (such as the one described in **Section 12.3 “Timer1 Oscillator”** above) gives users the option to include RTC functionality to their applications. This is accomplished with an inexpensive watch crystal to provide an accurate time base and several lines of application code to calculate the time. When operating in Sleep mode and using a battery or supercapacitor as a power source, it can completely eliminate the need for a separate RTC device and battery backup.

The application code routine, *RTCisr*, shown in Example 12-1, demonstrates a simple method to increment a counter at one-second intervals using an Interrupt Service Routine. Incrementing the TMR1 register pair to overflow triggers the interrupt and calls the routine which increments the seconds counter by one. Additional counters for minutes and hours are incremented as the previous counter overflows.

Since the register pair is 16 bits wide, counting up to overflow the register directly from a 32.768 kHz clock would take 2 seconds. To force the overflow at the required one-second intervals, it is necessary to preload it. The simplest method is to set the MSb of TMR1H with a BSF instruction. Note that the TMR1L register is never preloaded or altered; doing so may introduce cumulative error over many cycles.

For this method to be accurate, Timer1 must operate in Asynchronous mode and the Timer1 overflow interrupt must be enabled (PIE1<0> = 1), as shown in the routine, *RTCinit*. The Timer1 oscillator must also be enabled and running at all times.

# PIC18F85J11 FAMILY

## EXAMPLE 12-1: IMPLEMENTING A REAL-TIME CLOCK USING A TIMER1 INTERRUPT SERVICE

```

RTCinit
    MOVLW    80h           ; Preload TMR1 register pair
    MOVWF    TMR1H         ; for 1 second overflow
    CLRF     TMR1L
    MOVLW    b'00001111'   ; Configure for external clock,
    MOVWF    T1CON         ; Asynchronous operation, external oscillator
    CLRF     secs          ; Initialize timekeeping registers
    CLRF     mins
    MOVLW    .12
    MOVWF    hours
    BSF      PIE1, TMR1IE   ; Enable Timer1 interrupt
    RETURN

RTCisr
    BSF      TMR1H, 7       ; Preload for 1 sec overflow
    BCF      PIR1, TMR1IF   ; Clear interrupt flag
    INCF     secs, F        ; Increment seconds
    MOVLW    .59            ; 60 seconds elapsed?
    CPFSGT   secs
    RETURN                    ; No, done
    CLRF     secs          ; Clear seconds
    INCF     mins, F        ; Increment minutes
    MOVLW    .59            ; 60 minutes elapsed?
    CPFSGT   mins
    RETURN                    ; No, done
    CLRF     mins          ; clear minutes
    INCF     hours, F       ; Increment hours
    MOVLW    .23            ; 24 hours elapsed?
    CPFSGT   hours
    RETURN                    ; No, done
    CLRF     hours          ; Reset hours
    RETURN                    ; Done
    
```

**TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 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   | 51                   |
| PIR1   | PSPIF                     | ADIF      | RC1IF   | TX1IF   | SSPIF   | —                    | TMR2IF | TMR1IF | 53                   |
| PIE1   | PSPIE                     | ADIE      | RC1IE   | TX1IE   | SSPIE   | —                    | TMR2IE | TMR1IE | 53                   |
| IPR1   | PSPIP                     | ADIP      | RC1IP   | TX1IP   | SSPIP   | —                    | TMR2IP | TMR1IP | 53                   |
| TMR1L  | Timer1 Register Low Byte  |           |         |         |         |                      |        |        | 52                   |
| TMR1H  | Timer1 Register High Byte |           |         |         |         |                      |        |        | 52                   |
| T1CON  | RD16                      | T1RUN     | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYN $\overline{C}$ | TMR1CS | TMR1ON | 52                   |

**Legend:** Shaded cells are not used by the Timer1 module.

# PIC18F85J11 FAMILY

---

NOTES:

## 13.0 TIMER2 MODULE

The Timer2 module incorporates the following features:

- 8-bit Timer and Period registers (TMR2 and PR2, respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match
- Optional use as the shift clock for the MSSP module

The module is controlled through the T2CON register (Register 13-1) which enables or disables the timer and configures the prescaler and postscaler. Timer2 can be shut off by clearing control bit, TMR2ON (T2CON<2>), to minimize power consumption.

A simplified block diagram of the module is shown in Figure 13-1.

## 13.1 Timer2 Operation

In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 4-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON<1:0>). The value of TMR2 is compared to that of the Period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler (see **Section 13.2 “Timer2 Interrupt”**).

The TMR2 and PR2 registers are both directly readable and writable. The TMR2 register is cleared on any device Reset while the PR2 register initializes at FFh. Both the prescaler and postscaler counters are cleared on the following events:

- a write to the TMR2 register
- a write to the T2CON register
- any device Reset (Power-on Reset,  $\overline{\text{MCLR}}$  Reset, Watchdog Timer Reset or Brown-out Reset)

TMR2 is not cleared when T2CON is written.

**REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER**

| U-0   | R/W-0    | R/W-0    | R/W-0    | R/W-0    | R/W-0  | R/W-0   | R/W-0   |
|-------|----------|----------|----------|----------|--------|---------|---------|
| —     | T2OUTPS3 | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 |
| 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 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

•  
•  
•

1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

# PIC18F85J11 FAMILY

## 13.2 Timer2 Interrupt

Timer2 can also generate an optional device interrupt. The Timer2 output signal (TMR2 to PR2 match) provides the input for the 4-bit output counter/postscaler. This counter generates the TMR2 match interrupt flag which is latched in TMR2IF (PIR1<1>). The interrupt is enabled by setting the TMR2 Match Interrupt Enable bit, TMR2IE (PIE1<1>).

A range of 16 postscale options (from 1:1 through 1:16 inclusive) can be selected with the postscaler control bits, T2OUTPS3:T2OUTPS0 (T2CON<6:3>).

## 13.3 Timer2 Output

The unscaled output of TMR2 is available primarily to the CCP modules where it is used as a time base for operations in PWM mode.

Timer2 can be optionally used as the shift clock source for the MSSP module operating in SPI mode. Additional information is provided in **Section 16.0 “Master Synchronous Serial Port (MSSP) Module”**.

**FIGURE 13-1: TIMER2 BLOCK DIAGRAM**

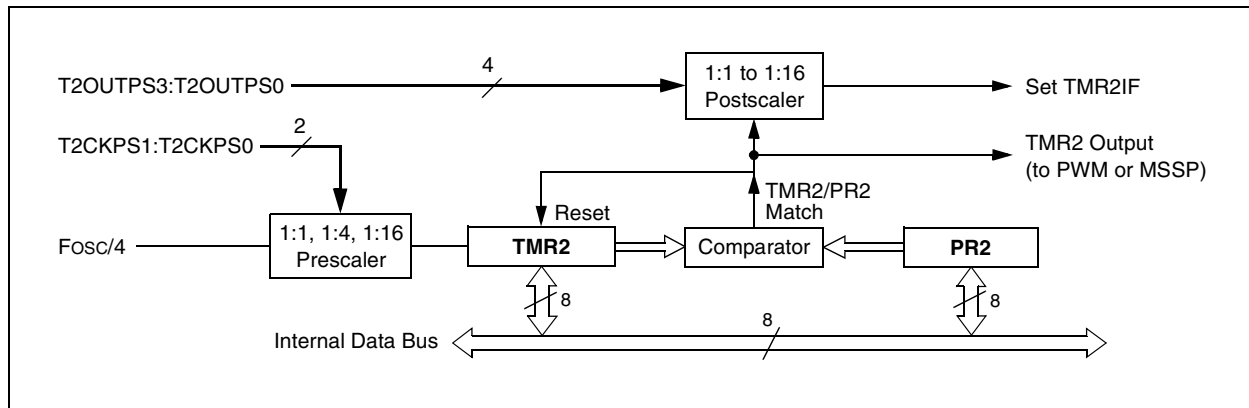


**TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER**

| 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    | 51                   |
| PIR1   | PSPIF                  | ADIF      | RC1IF    | TX1IF    | SSPIF    | —      | TMR2IF  | TMR1IF  | 53                   |
| PIE1   | PSPIE                  | ADIE      | RC1IE    | TX1IE    | SSPIE    | —      | TMR2IE  | TMR1IE  | 53                   |
| IPR1   | PSPIP                  | ADIP      | RC1IP    | TX1IP    | SSPIP    | —      | TMR2IP  | TMR1IP  | 53                   |
| TMR2   | Timer2 Register        |           |          |          |          |        |         |         | 52                   |
| T2CON  | —                      | T2OUTPS3  | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 52                   |
| PR2    | Timer2 Period Register |           |          |          |          |        |         |         | 52                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.



## 14.0 TIMER3 MODULE

The Timer3 timer/counter module incorporates these features:

- Software selectable operation as a 16-bit timer or counter
- Readable and writable 8-bit registers (TMR3H and TMR3L)
- Selectable clock source (internal or external) with device clock or Timer1 oscillator internal options
- Interrupt on overflow
- Module Reset on CCPx Special Event Trigger

A simplified block diagram of the Timer3 module is shown in Figure 14-1. A block diagram of the module's operation in Read/Write mode is shown in Figure 14-2.

The Timer3 module is controlled through the T3CON register (Register 14-1). It also selects the clock source options for the CCP modules. See **Section 15.2.2 “Timer1/Timer3 Mode Selection”** for more information.

**REGISTER 14-1: T3CON: TIMER3 CONTROL REGISTER**

| R/W-0 | R/W-0  | R/W-0   | R/W-0   | R/W-0  | R/W-0               | R/W-0  | R/W-0  |
|-------|--------|---------|---------|--------|---------------------|--------|--------|
| RD16  | T3CCP2 | T3CKPS1 | T3CKPS0 | T3CCP1 | $\overline{T3SYNC}$ | TMR3CS | TMR3ON |
| 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      **RD16:** 16-Bit Read/Write Mode Enable bit  
             1 = Enables register read/write of Timer3 in one 16-bit operation  
             0 = Enables register read/write of Timer3 in two 8-bit operations
- bit 6,3    **T3CCP2:T3CCP1:** Timer3 and Timer1 to CCPx Enable bits  
             1x = Timer3 is the capture/compare clock source for the CCP modules  
             01 = Timer3 is the capture/compare clock source for CCP2;  
                     Timer1 is the capture/compare clock source for CCP1  
             00 = Timer1 is the capture/compare clock source for the CCP modules
- bit 5-4    **T3CKPS1:T3CKPS0:** Timer3 Input Clock Prescale Select bits  
             11 = 1:8 Prescale value  
             10 = 1:4 Prescale value  
             01 = 1:2 Prescale value  
             00 = 1:1 Prescale value
- bit 2      **T3SYNC:** Timer3 External Clock Input Synchronization Control bit  
             (Not usable if the device clock comes from Timer1/Timer3.)  
             When TMR3CS = 1:  
             1 = Do not synchronize external clock input  
             0 = Synchronize external clock input  
             When TMR3CS = 0:  
             This bit is ignored. Timer3 uses the internal clock when TMR3CS = 0.
- bit 1      **TMR3CS:** Timer3 Clock Source Select bit  
             1 = External clock input from Timer1 oscillator or T13CKI (on the rising edge after the first falling edge)  
             0 = Internal clock (FOSC/4)
- bit 0      **TMR3ON:** Timer3 On bit  
             1 = Enables Timer3  
             0 = Stops Timer3

# PIC18F85J11 FAMILY

## 14.1 Timer3 Operation

Timer3 can operate in one of three modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

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

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

**FIGURE 14-1: TIMER3 BLOCK DIAGRAM (8-BIT MODE)**

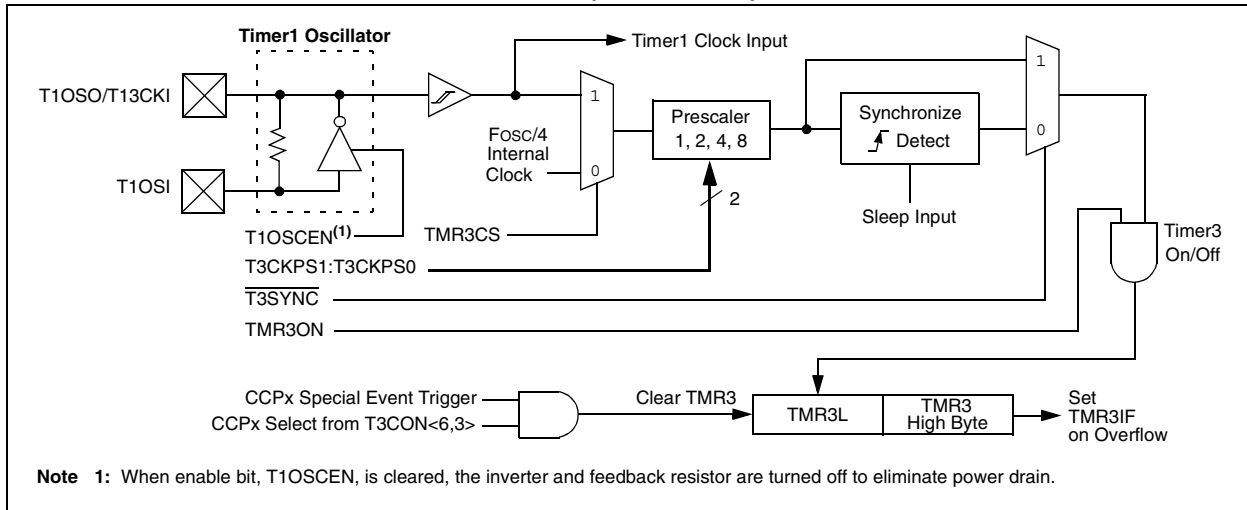
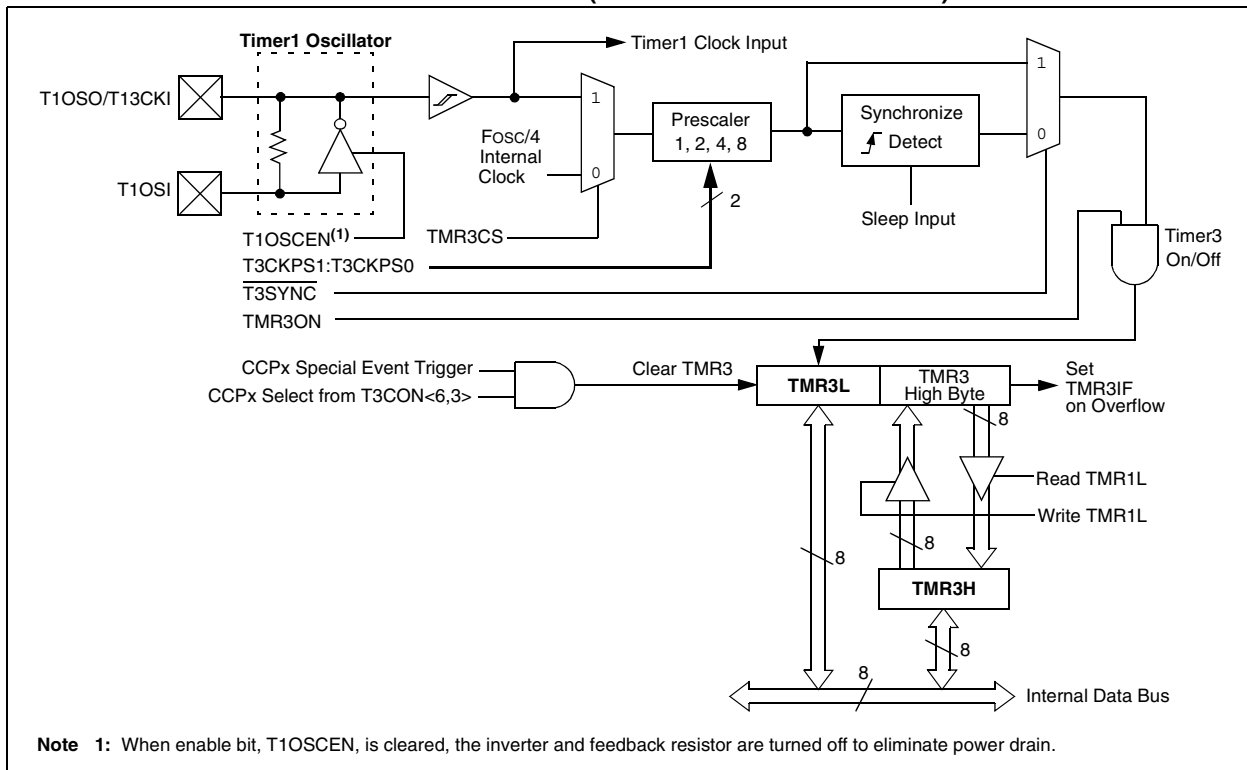


**FIGURE 14-2: TIMER3 BLOCK DIAGRAM (16-BIT READ/WRITE MODE)**



## 14.2 Timer3 16-Bit Read/Write Mode

Timer3 can be configured for 16-bit reads and writes (see Figure 14-2). When the RD16 control bit (T3CON<7>) 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.

## 14.3 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 12.0 “Timer1 Module”**.

## 14.4 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>).

## 14.5 Resetting Timer3 Using the CCPx Special Event Trigger

If CCP1 or CCP2 is configured to use Timer3 and to generate a Special Event Trigger in Compare mode (CCPxM3:CCPxM0 = 1011), this signal will reset Timer3. The trigger from CCP2 will also start an A/D conversion if the A/D module is enabled (see **Section 15.3.4 “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 a CCPx module, the write will take precedence.

**Note:** The Special Event Triggers from the CCPx module will not set the TMR3IF interrupt flag bit (PIR2<1>).

**TABLE 14-1: 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   | 51                   |
| PIR2   | OSCFIF                    | CMIF      | —       | —       | BCLIF   | LVDIF                | TMR3IF | —      | 53                   |
| PIE2   | OSCFIE                    | CMIE      | —       | —       | BCLIE   | LVDIE                | TMR3IE | —      | 53                   |
| IPR2   | OSCFIP                    | CMIP      | —       | —       | BCLIP   | LVDIP                | TMR3IP | —      | 53                   |
| TMR3L  | Timer3 Register Low Byte  |           |         |         |         |                      |        |        | 53                   |
| TMR3H  | Timer3 Register High Byte |           |         |         |         |                      |        |        | 53                   |
| T1CON  | RD16                      | T1RUN     | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYN $\overline{C}$ | TMR1CS | TMR1ON | 52                   |
| T3CON  | RD16                      | T3CCP2    | T3CKPS1 | T3CKPS0 | T3CCP1  | T3SYN $\overline{C}$ | TMR3CS | TMR3ON | 53                   |

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

# PIC18F85J11 FAMILY

---

NOTES:

## 15.0 CAPTURE/COMPARE/PWM (CCP) MODULES

PIC18F85J11 family devices have two CCP (Capture/Compare/PWM) modules, designated CCP1 and CCP2. Both modules implement standard Capture, Compare and Pulse-Width Modulation (PWM) modes.

Each CCP module contains a 16-bit register which can operate as a 16-bit Capture register, a 16-bit Compare register or a PWM Master/Slave Duty Cycle register. For the sake of clarity, all CCP module operation in the following sections is described with respect to CCP2, but is equally applicable to CCP1.

### REGISTER 15-1: CCPxCON: CCPx CONTROL REGISTER (CCP1, CCP2 MODULES)

| U-0   | U-0 | R/W-0 | R/W-0 | R/W-0  | R/W-0  | R/W-0  | R/W-0  |
|-------|-----|-------|-------|--------|--------|--------|--------|
| —     | —   | DCxB1 | DCxB0 | CCPxM3 | CCPxM2 | CCPxM1 | CCPxM0 |
| 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 **DCxB1:DCxB0:** PWM Duty Cycle bit 1 and bit 0 for CCPx Module

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two Least Significant bits (bit 1 and bit 0) of the 10-bit PWM duty cycle. The eight Most Significant bits (DCx9:DCx2) of the duty cycle are found in CCPxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Module Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0001 = Reserved

0010 = Compare mode, toggle output on match (CCPxIF bit is set)

0011 = Reserved

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode: initialize CCPx pin low; on compare match, force CCPx pin high (CCPxIF bit is set)

1001 = Compare mode: initialize CCPx pin high; on compare match, force CCPx pin low (CCPxIF bit is set)

1010 = Compare mode: generate software interrupt on compare match (CCPxIF bit is set, CCPx pin reflects I/O state)

1011 = Compare mode: Special Event Trigger; reset timer; start A/D conversion on CCPx match (CCPxIF bit is set)<sup>(1)</sup>

11xx = PWM mode

**Note 1:** CCPxM3:CCPxM0 = 1011 will only reset timer and not start A/D conversion on CCPx match.

# PIC18F85J11 FAMILY

## 15.1 CCP Module Configuration

Each Capture/Compare/PWM module is associated with a control register (generically, CCPxCON) and a data register (CCPRx). The data register, in turn, is comprised of two 8-bit registers: CCPRxL (low byte) and CCPRxH (high byte). All registers are both readable and writable.

### 15.1.1 CCP MODULES AND TIMER RESOURCES

The CCP modules utilize Timers 1, 2 or 3, depending on the mode selected. Timer1 and Timer3 are available to modules in Capture or Compare mode, while Timer2 is available for modules in PWM mode.

**TABLE 15-1: CCPx MODE – TIMER RESOURCE**

| CCPx Mode | Timer Resource   |
|-----------|------------------|
| Capture   | Timer1 or Timer3 |
| Compare   | Timer1 or Timer3 |
| PWM       | Timer2           |

The assignment of a particular timer to a module is determined by the timer to CCPx enable bits in the T3CON register (Register 14-1). Both modules may be active at any given time and may share the same timer resource if they are configured to operate in the same mode (Capture/Compare or PWM) at the same time. The interactions between the two modules are summarized in Table 15-2.

Depending on the configuration selected, up to four timers may be active at once, with modules in the same configuration (Capture/Compare or PWM) sharing timer resources. The possible configurations are shown in Figure 15-1.

### 15.1.2 OPEN-DRAIN OUTPUT OPTION

When operating in Output mode (i.e., in Compare or PWM modes), the drivers for the CCPx pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor and allows the output to communicate with external circuits without the need for additional level shifters.

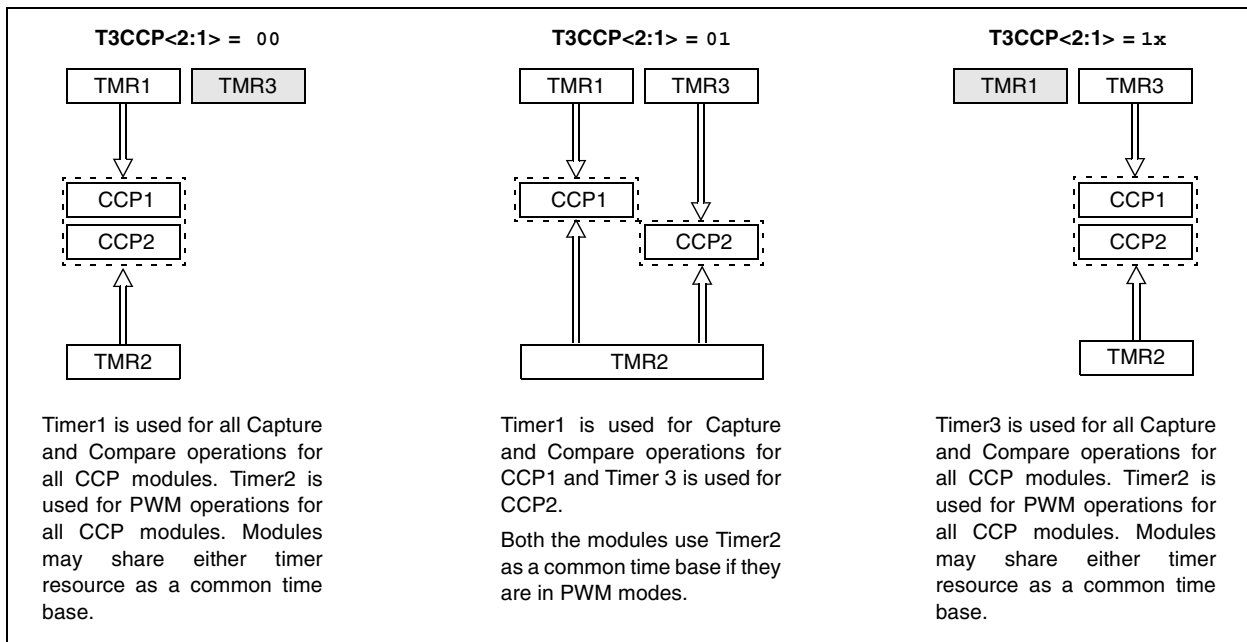
The open-drain output option is controlled by the CCP2OD and CCP1OD bits (TRISG<6:5>). Setting the appropriate bit configures the pin for the corresponding module for open-drain operation.

### 15.1.3 CCP2 PIN ASSIGNMENT

The pin assignment for CCP2 (Capture input, Compare and PWM output) can change, based on device configuration. The CCP2MX Configuration bit determines which pin CCP2 is multiplexed to. By default, it is assigned to RC1 (CCP2MX = 1). If the Configuration bit is cleared, CCP2 is multiplexed with RE7.

Changing the pin assignment of CCP2 does not automatically change any requirements for configuring the port pin. Users must always verify that the appropriate TRIS register is configured correctly for CCP2 operation, regardless of where it is located.

**FIGURE 15-1: CCPx AND TIMER INTERCONNECT CONFIGURATIONS**



# PIC18F85J11 FAMILY

**TABLE 15-2: INTERACTIONS BETWEEN CCP1 AND CCP2 FOR TIMER RESOURCES**

| CCP1 Mode | CCP2 Mode | Interaction                                                                                                                                                                                                                                                         |
|-----------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Capture   | Capture   | Each module can use TMR1 or TMR3 as the time base. The time base can be different for each CCP module.                                                                                                                                                              |
| Capture   | Compare   | CCP2 can be configured for the Special Event Trigger to reset TMR1 or TMR3 (depending upon which time base is used). Automatic A/D conversions on trigger event can also be done. Operation of CCP1 could be affected if it is using the same timer as a time base. |
| Compare   | Capture   | CCP1 can be configured for the Special Event Trigger to reset TMR1 or TMR3 (depending upon which time base is used). Operation of CCP2 could be affected if it is using the same timer as a time base.                                                              |
| Compare   | Compare   | Either module can be configured for the Special Event Trigger to reset the time base. Automatic A/D conversions on CCP2 trigger event can be done. Conflicts may occur if both modules are using the same time base.                                                |
| Capture   | PWM       | None                                                                                                                                                                                                                                                                |
| Compare   | PWM       | None                                                                                                                                                                                                                                                                |
| PWM       | Capture   | None                                                                                                                                                                                                                                                                |
| PWM       | Compare   | None                                                                                                                                                                                                                                                                |
| PWM       | PWM       | Both PWMs will have the same frequency and update rate (TMR2 interrupt).                                                                                                                                                                                            |

# PIC18F85J11 FAMILY

## 15.2 Capture Mode

In Capture mode, the CCPR2H:CCPR2L register pair captures the 16-bit value of the TMR1 or TMR3 register when an event occurs on the CCP2 pin (RB3, RC1 or RE7, depending on device configuration). An event is defined as one of the following:

- every falling edge
- every rising edge
- every 4th rising edge
- every 16th rising edge

The event is selected by the mode select bits, CCP2M3:CCP2M0 (CCP2CON<3:0>). When a capture is made, the interrupt request flag bit, CCP2IF (PIR3<2>), is set; it must be cleared in software. If another capture occurs before the value in register CCPR2 is read, the old captured value is overwritten by the new captured value.

### 15.2.1 CCPx PIN CONFIGURATION

In Capture mode, the appropriate CCPx pin should be configured as an input by setting the corresponding TRIS direction bit.

**Note:** If RB3/INT3/CCP2, RC1/T1OSI/CCP2 or RE7/CCP2 is configured as an output, a write to the port can cause a capture condition.

### 15.2.2 TIMER1/TIMER3 MODE SELECTION

The timers that are to be used with the capture feature (Timer1 and/or Timer3) must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work. The timer to be used with each CCP module is selected in the T3CON register (see **Section 15.1.1 “CCP Modules and Timer Resources”**).

## 15.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep the CCP2IE bit (PIE3<2>) clear to avoid false interrupts and should clear the flag bit, CCP2IF, following any such change in operating mode.

### 15.2.4 CCP PRESCALER

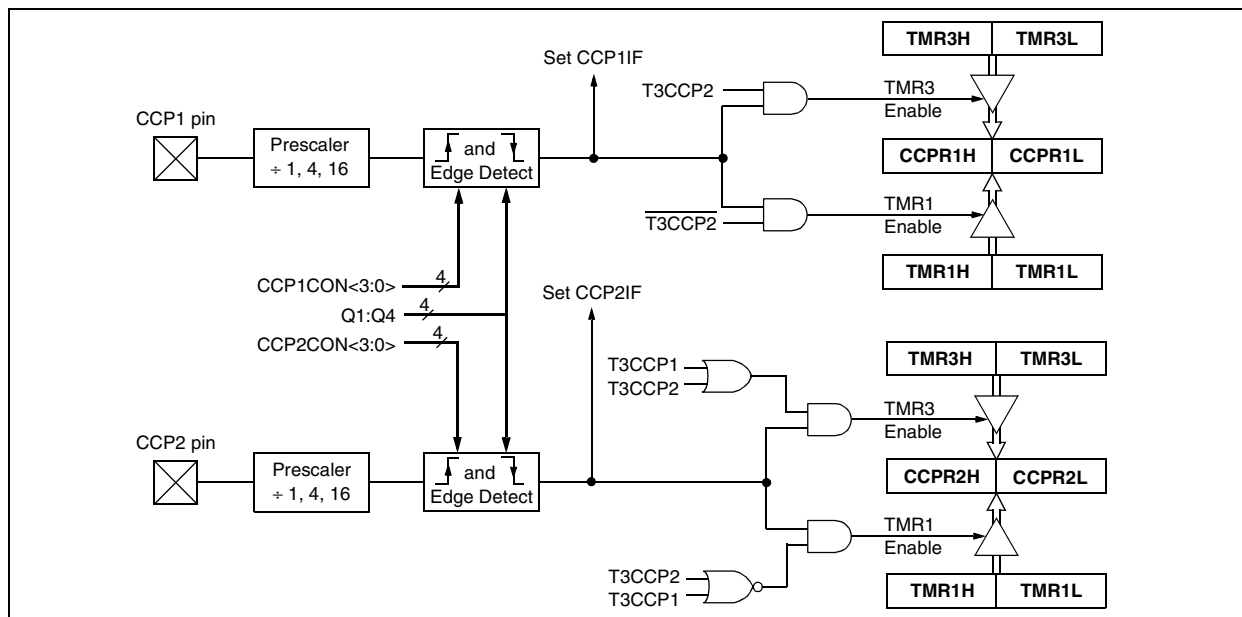
There are four prescaler settings in Capture mode. They are specified as part of the operating mode selected by the mode select bits (CCP2M3:CCP2M0). Whenever the CCP2 module is turned off, or the CCP2 module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 15-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the “false” interrupt.

### EXAMPLE 15-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF CCP2CON      ; Turn CCP module off
MOVLW NEW_CAPT_PS ; Load WREG with the
                  ; new prescaler mode
                  ; value and CCP ON
MOVWF CCP2CON      ; Load CCP2CON with
                  ; this value
```

**FIGURE 15-2: CAPTURE MODE OPERATION BLOCK DIAGRAM**





## 15.3 Compare Mode

In Compare mode, the 16-bit CCPR2 register value is constantly compared against either the TMR1 or TMR3 register pair value. When a match occurs, the CCP2 pin can be:

- driven high
- driven low
- toggled (high-to-low or low-to-high)
- remains 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 (CCP2M3:CCP2M0). At the same time, the interrupt flag bit, CCP2IF, is set.

### 15.3.1 CCPx PIN CONFIGURATION

The user must configure the CCPx pin as an output by clearing the appropriate TRIS bit.

**Note:** Clearing the CCP2CON register will force the RB3, RC1 or RE7 compare output latch (depending on device configuration) to the default low level. This is not the PORTB, PORTC or PORTE I/O data latch.

### 15.3.2 TIMER1/TIMER3 MODE SELECTION

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

### 15.3.3 SOFTWARE INTERRUPT MODE

When the Generate Software Interrupt mode is chosen (CCP2M3:CCP2M0 = 1010), the CCP2 pin is not affected. Only a CCP2 interrupt is generated, if enabled, and the CCP2IE bit is set.

### 15.3.4 SPECIAL EVENT TRIGGER

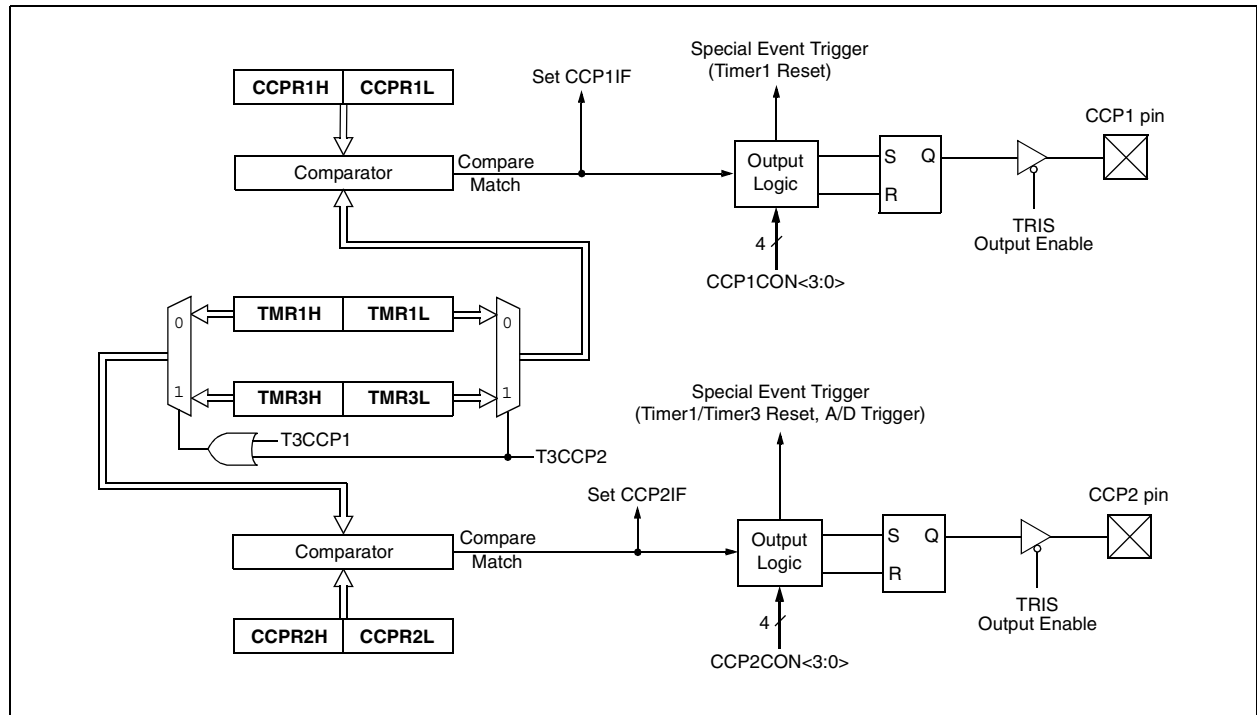
Both CCP modules are equipped with a Special Event Trigger. This is an internal hardware signal generated in Compare mode to trigger actions by other modules. The Special Event Trigger is enabled by selecting the Compare Special Event Trigger mode (CCP2M3:CCP2M0 = 1011).

For either CCP module, 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 for CCP2 can also start an A/D conversion. In order to do this, the A/D converter must already be enabled.

**Note:** The Special Event Trigger of CCP1 only resets Timer1/Timer3 and cannot start an A/D conversion even when the A/D converter is enabled.

**FIGURE 15-3: COMPARE MODE OPERATION BLOCK DIAGRAM**



# PIC18F85J11 FAMILY

**TABLE 15-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1 AND TIMER3**

| 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   | 51                   |
| RCON    | IPEN                                     | —         | CM      | RI      | TO      | PD                   | POR    | BOR    | 52                   |
| PIR3    | —                                        | —         | RC2IF   | TX2IF   | —       | CCP2IF               | CCP1IF | —      | 53                   |
| PIE3    | —                                        | —         | RC2IE   | TX2IE   | —       | CCP2IE               | CCP1IE | —      | 53                   |
| IPR3    | —                                        | —         | RC2IP   | TX2IP   | —       | CCP2IP               | CCP1IP | —      | 53                   |
| PIR2    | OSCFIF                                   | CMIF      | —       | —       | BCLIF   | LVDIF                | TMR3IF | —      | 53                   |
| PIE2    | OSCFIE                                   | CMIE      | —       | —       | BCLIE   | LVDIE                | TMR3IE | —      | 53                   |
| IPR2    | OSCFIP                                   | CMIP      | —       | —       | BCLIP   | LVDIP                | TMR3IP | —      | 53                   |
| TRISC   | TRISC7                                   | TRISC6    | TRISC5  | TRISC4  | TRISC3  | TRISC2               | TRISC1 | TRISC0 | 54                   |
| TRISE   | TRISE7                                   | TRISE6    | TRISE5  | TRISE4  | TRISE3  | —                    | TRISE1 | TRISE0 | 54                   |
| TRISG   | SPIOD                                    | CCP2OD    | CCP1OD  | TRISG4  | TRISG3  | TRISG2               | TRISG1 | TRISG0 | 54                   |
| TMR1L   | Timer1 Register Low Byte                 |           |         |         |         |                      |        |        | 52                   |
| TMR1H   | Timer1 Register High Byte                |           |         |         |         |                      |        |        | 52                   |
| T1CON   | RD16                                     | T1RUN     | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYN $\overline{C}$ | TMR1CS | TMR1ON | 52                   |
| TMR3H   | Timer3 Register High Byte                |           |         |         |         |                      |        |        | 53                   |
| TMR3L   | Timer3 Register Low Byte                 |           |         |         |         |                      |        |        | 53                   |
| T3CON   | RD16                                     | T3CCP2    | T3CKPS1 | T3CKPS0 | T3CCP1  | T3SYN $\overline{C}$ | TMR3CS | TMR3ON | 53                   |
| CCPR1L  | Capture/Compare/PWM Register 1 Low Byte  |           |         |         |         |                      |        |        | 54                   |
| CCPR1H  | Capture/Compare/PWM Register 1 High Byte |           |         |         |         |                      |        |        | 54                   |
| CCP1CON | —                                        | —         | DC1B1   | DC1B0   | CCP1M3  | CCP1M2               | CCP1M1 | CCP1M0 | 54                   |
| CCPR2L  | Capture/Compare/PWM Register 2 Low Byte  |           |         |         |         |                      |        |        | 55                   |
| CCPR2H  | Capture/Compare/PWM Register 2 High Byte |           |         |         |         |                      |        |        | 55                   |
| CCP2CON | —                                        | —         | DC2B1   | DC2B0   | CCP2M3  | CCP2M2               | CCP2M1 | CCP2M0 | 55                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by Capture/Compare, Timer1 or Timer3.

## 15.4 PWM Mode

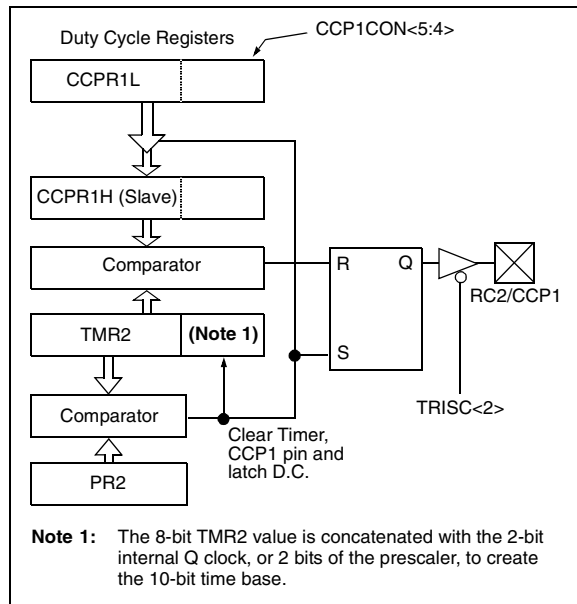
In Pulse-Width Modulation (PWM) mode, the CCP2 pin produces up to a 10-bit resolution PWM output. Since the CCP2 pin is multiplexed with a PORTB, PORTC or PORTE data latch, the appropriate TRIS bit must be cleared to make the CCP2 pin an output.

**Note:** Clearing the CCP2CON register will force the RB3, RC1 or RE7 output latch (depending on device configuration) to the default low level. This is not the PORTB, PORTC or PORTE I/O data latch.

Figure 15-4 shows a simplified block diagram of the CCP1 module in PWM mode.

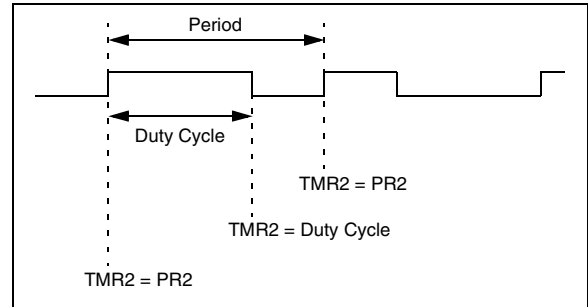
For a step-by-step procedure on how to set up the CCP module for PWM operation, see **Section 15.4.3 “Setup for PWM Operation”**.

**FIGURE 15-4: SIMPLIFIED PWM BLOCK DIAGRAM**



A PWM output (Figure 15-5) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

**FIGURE 15-5: PWM OUTPUT**



### 15.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula:

**EQUATION 15-1:**

$$\text{PWM Period} = (\text{PR2} + 1) \cdot 4 \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

PWM frequency is defined as  $1/[\text{PWM period}]$ .

When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

- TMR2 is cleared
- The CCP2 pin is set (exception: if PWM duty cycle = 0%, the CCP2 pin will not be set)
- The PWM duty cycle is latched from CCPR2L into CCPR2H

**Note:** The Timer2 postscalers (see **Section 13.0 “Timer2 Module”**) are not used in the determination of the PWM frequency. The postscaler could be used to have a servo update rate at a different frequency than the PWM output.

# PIC18F85J11 FAMILY

## 15.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR2L register and to the CCP2CON<5:4> bits. Up to 10-bit resolution is available. The CCPR2L contains the eight MSbs and the CCP2CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR2L:CCP2CON<5:4>. The following equation is used to calculate the PWM duty cycle in time:

### EQUATION 15-2:

$$\text{PWM Duty Cycle} = (\text{CCPR2L:CCP2CON<5:4>}) \cdot \text{TOSC} \cdot (\text{TMR2 Prescale Value})$$

CCPR2L and CCP2CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR2H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR2H is a read-only register.

The CCPR2H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.

When the CCPR2H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP2 pin is cleared.

The maximum PWM resolution (bits) for a given PWM frequency is given by the equation:

### EQUATION 15-3:

$$\text{PWM Resolution (max)} = \frac{\log\left(\frac{F_{\text{OSC}}}{F_{\text{PWM}}}\right)}{\log(2)} \text{ bits}$$

**Note:** If the PWM duty cycle value is longer than the PWM period, the CCP2 pin will not be cleared.

**TABLE 15-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz**

| PWM Frequency              | 2.44 kHz | 9.77 kHz | 39.06 kHz | 156.25 kHz | 312.50 kHz | 416.67 kHz |
|----------------------------|----------|----------|-----------|------------|------------|------------|
| Timer Prescaler (1, 4, 16) | 16       | 4        | 1         | 1          | 1          | 1          |
| PR2 Value                  | FFh      | FFh      | FFh       | 3Fh        | 1Fh        | 17h        |
| Maximum Resolution (bits)  | 14       | 12       | 10        | 8          | 7          | 6.58       |

# PIC18F85J11 FAMILY

## 15.4.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:

1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCP2L register and CCP2CON<5:4> bits.
3. Make the CCP2 pin an output by clearing the appropriate TRIS bit.
4. Set the TMR2 prescale value, then enable Timer2 by writing to T2CON.
5. Configure the CCP2 module for PWM operation.

**TABLE 15-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2**

| 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    | 51                   |
| RCON    | IPEN                                     | —         | CM       | RI       | TO       | PD     | POR     | BOR     | 52                   |
| PIR1    | PSPIF                                    | ADIF      | RC1IF    | TX1IF    | SSPIF    | —      | TMR2IF  | TMR1IF  | 53                   |
| PIE1    | PSPIE                                    | ADIE      | RC1IE    | TX1IE    | SSPIE    | —      | TMR2IE  | TMR1IE  | 53                   |
| IPR1    | PSPIP                                    | ADIP      | RC1IP    | TX1IP    | SSPIP    | —      | TMR2IP  | TMR1IP  | 53                   |
| TRISC   | TRISC7                                   | TRISC6    | TRISC5   | TRISC4   | TRISC3   | TRISC2 | TRISC1  | TRISC0  | 54                   |
| TRISE   | TRISE7                                   | TRISE6    | TRISE5   | TRISE4   | TRISE3   | —      | TRISE1  | TRISE0  | 54                   |
| TRISG   | SPIOD                                    | CCP2OD    | CCP1OD   | TRISG4   | TRISG3   | TRISG2 | TRISG1  | TRISG0  | 54                   |
| TMR2    | Timer2 Register                          |           |          |          |          |        |         |         | 52                   |
| PR2     | Timer2 Period Register                   |           |          |          |          |        |         |         | 52                   |
| T2CON   | —                                        | T2OUTPS3  | T2OUTPS2 | T2OUTPS1 | T2OUTPS0 | TMR2ON | T2CKPS1 | T2CKPS0 | 52                   |
| CCPR1L  | Capture/Compare/PWM Register 1 Low Byte  |           |          |          |          |        |         |         | 54                   |
| CCPR1H  | Capture/Compare/PWM Register 1 High Byte |           |          |          |          |        |         |         | 54                   |
| CCP1CON | —                                        | —         | DC1B1    | DC1B0    | CCP1M3   | CCP1M2 | CCP1M1  | CCP1M0  | 54                   |
| CCPR2L  | Capture/Compare/PWM Register 2 Low Byte  |           |          |          |          |        |         |         | 55                   |
| CCPR2H  | Capture/Compare/PWM Register 2 High Byte |           |          |          |          |        |         |         | 55                   |
| CCP2CON | —                                        | —         | DC2B1    | DC2B0    | CCP2M3   | CCP2M2 | CCP2M1  | CCP2M0  | 55                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by PWM or Timer2.

# PIC18F85J11 FAMILY

---

NOTES:

## 16.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

## 16.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface, useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I<sup>2</sup>C™)
  - Full Master mode
  - Slave mode (with general address call)

The I<sup>2</sup>C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

## 16.2 Control Registers

The MSSP module has three associated control registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2). The use of these registers and their individual bits differ significantly depending on whether the MSSP module is operated in SPI or I<sup>2</sup>C mode.

Additional details are provided under the individual sections.

## 16.3 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

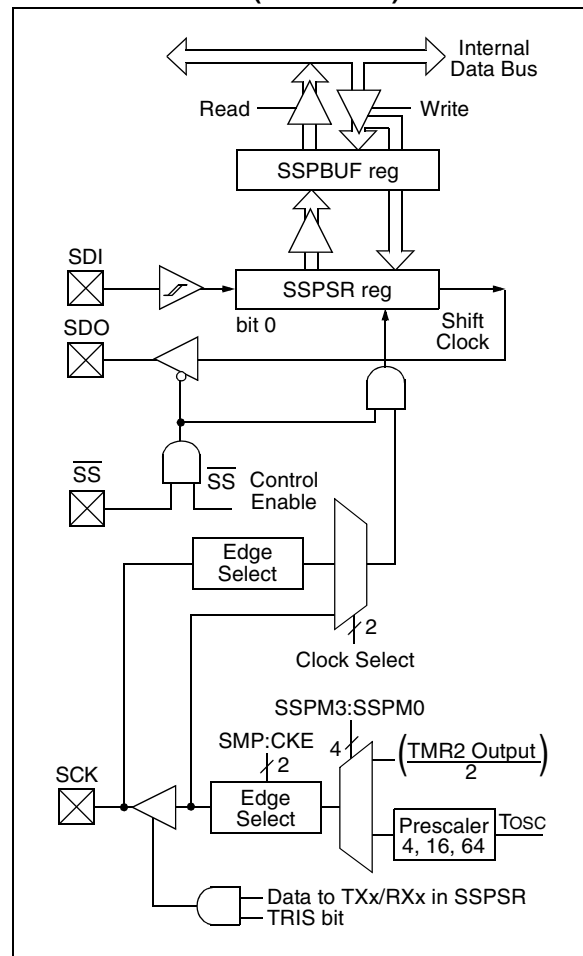
- Serial Data Out (SDO) – RC5/SDO
- Serial Data In (SDI) – RC4/SDI/SDA
- Serial Clock (SCK) – RC3/SCK/SCL

Additionally, a fourth pin may be used when in a Slave mode of operation:

- Slave Select ( $\overline{SS}$ ) – RF7/AN5/ $\overline{SS}$

Figure 16-1 shows the block diagram of the MSSP module when operating in SPI mode.

**FIGURE 16-1: MSSP BLOCK DIAGRAM (SPI MODE)**



# PIC18F85J11 FAMILY

## 16.3.1 REGISTERS

The MSSP module has four registers for SPI mode operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible

SSPCON1 and SSPSTAT are the control and status registers in SPI mode operation. The SSPCON1 register is readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

### REGISTER 16-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

| R/W-0 | R/W-0              | R-0 | R-0 | R-0 | R-0 | R0 | R-0   |
|-------|--------------------|-----|-----|-----|-----|----|-------|
| SMP   | CKE <sup>(1)</sup> | D/A | P   | S   | R/W | UA | BF    |
| 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 **SMP:** Sample bit

SPI Master mode:

1 = Input data sampled at end of data output time

0 = Input data sampled at middle of data output time

SPI Slave mode:

SMP must be cleared when SPI is used in Slave mode.

bit 6 **CKE:** SPI Clock Select bit<sup>(1)</sup>

1 = Transmit occurs on transition from active to Idle clock state

0 = Transmit occurs on transition from Idle to active clock state

bit 5 **D/A:** Data/Address bit

Used in I<sup>2</sup>C™ mode only.

bit 4 **P:** Stop bit

Used in I<sup>2</sup>C mode only. This bit is cleared when the MSSP module is disabled, SSPEN is cleared.

bit 3 **S:** Start bit

Used in I<sup>2</sup>C mode only.

bit 2 **R/W:** Read/Write Information bit

Used in I<sup>2</sup>C mode only.

bit 1 **UA:** Update Address bit

Used in I<sup>2</sup>C mode only.

bit 0 **BF:** Buffer Full Status bit (Receive mode only)

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

**Note 1:** Polarity of clock state is set by the CKP bit (SSPCON1<4>).



# PIC18F85J11 FAMILY

**REGISTER 16-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)**

| R/W-0 | R/W-0                | R/W-0                | R/W-0 | R/W-0                | R/W-0                | R/W-0                | R/W-0                |
|-------|----------------------|----------------------|-------|----------------------|----------------------|----------------------|----------------------|
| WCOL  | SSPOV <sup>(1)</sup> | SSPEN <sup>(2)</sup> | CKP   | SSPM3 <sup>(3)</sup> | SSPM2 <sup>(3)</sup> | SSPM1 <sup>(3)</sup> | SSPM0 <sup>(3)</sup> |
| 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      **WCOL:** Write Collision Detect bit (Transmit mode only)  
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)  
0 = No collision
- bit 6      **SSPOV:** Receive Overflow Indicator bit<sup>(1)</sup>  
SPI Slave mode:  
1 = A new byte is received while the SSPBUF register is still holding the previous data. In case of overflow, the data in SSPSR is lost. Overflow can only occur in Slave mode. The user must read the SSPBUF, even if only transmitting data, to avoid setting overflow (must be cleared in software).  
0 = No overflow
- bit 5      **SSPEN:** Master Synchronous Serial Port Enable bit<sup>(2)</sup>  
1 = Enables serial port and configures SCK, SDO, SDI and  $\overline{SS}$  as serial port pins  
0 = Disables serial port and configures these pins as I/O port pins
- bit 4      **CKP:** Clock Polarity Select bit  
1 = Idle state for clock is a high level  
0 = Idle state for clock is a low level
- bit 3-0    **SSPM3:SSPM0:** Master Synchronous Serial Port Mode Select bits<sup>(3)</sup>  
0101 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control disabled,  $\overline{SS}$  can be used as I/O pin  
0100 = SPI Slave mode, clock = SCK pin,  $\overline{SS}$  pin control enabled  
0011 = SPI Master mode, clock = TMR2 output/2  
0010 = SPI Master mode, clock = Fosc/64  
0001 = SPI Master mode, clock = Fosc/16  
0000 = SPI Master mode, clock = Fosc/4

**Note 1:** In Master mode, the overflow bit is not set, since each new reception (and transmission) is initiated by writing to the SSPBUF register.

**2:** When enabled, this pin must be properly configured as an input or output.

**3:** Bit combinations not specifically listed here are either reserved or implemented in I<sup>2</sup>C™ mode only.

# PIC18F85J11 FAMILY

## 16.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0> and SSPSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then, the Buffer Full detect bit, BF (SSPSTAT<0>), and the MSSP Interrupt Flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception

before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the Write Collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. The Buffer Full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 16-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the SSPSTAT register indicates the various status conditions.

### EXAMPLE 16-1: LOADING THE SSPBUF (SSPSR) REGISTER

|      |       |             |                                              |
|------|-------|-------------|----------------------------------------------|
| LOOP | BTFSS | SSPSTAT, BF | ;Has data been received (transmit complete)? |
|      | BRA   | LOOP        | ;No                                          |
|      | MOVF  | SSPBUF, W   | ;WREG reg = contents of SSPBUF               |
|      | MOVWF | RXDATA      | ;Save in user RAM, if data is meaningful     |
|      | MOVF  | TXDATA, W   | ;W reg = contents of TXDATA                  |
|      | MOVWF | SSPBUF      | ;New data to xmit                            |

## 16.3.3 ENABLING SPI I/O

To enable the serial port, MSSP Enable bit, SSPEN (SSPCON1<5>), must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, reinitialize the SSPCON registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and  $\overline{SS}$  pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- SDI is automatically controlled by the SPI module
- SDO must have TRISC<5> bit cleared
- SCK (Master mode) must have TRISC<3> bit cleared
- SCK (Slave mode) must have TRISC<3> bit set
- $\overline{SS}$  must have TRISF<7> bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

## 16.3.4 OPEN-DRAIN OUTPUT OPTION

The drivers for the SDO output and SCK clock pins can be optionally configured as open-drain outputs. This feature allows the voltage level on the pin to be pulled

to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters.

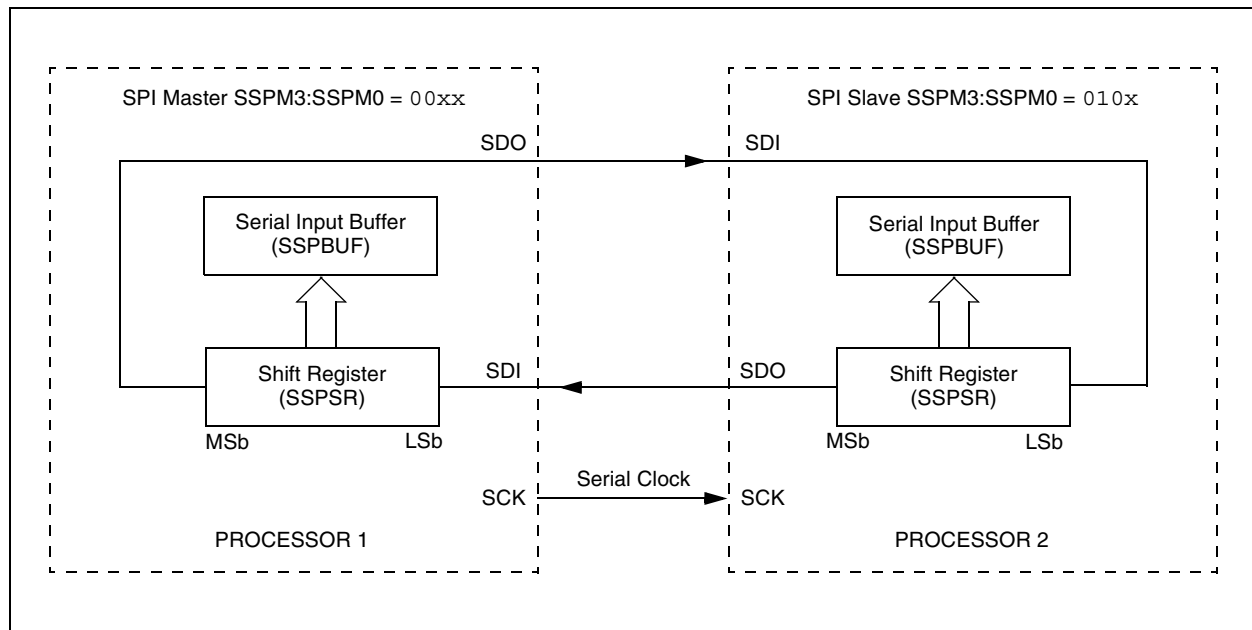
The open-drain output option is controlled by the SPIOD bit (TRISG<7>). Setting the bit configures both pins for open-drain operation.

## 16.3.5 TYPICAL CONNECTION

Figure 16-2 shows a typical connection between two microcontrollers. The master controller (Processor 1) initiates the data transfer by sending the SCK signal. Data is shifted out of both shift registers on their programmed clock edge and latched on the opposite edge of the clock. Both processors should be programmed to the same Clock Polarity (CKP), then both controllers would send and receive data at the same time. Whether the data is meaningful (or dummy data) depends on the application software. This leads to three scenarios for data transmission:

- Master sends data – Slave sends dummy data
- Master sends data – Slave sends data
- Master sends dummy data – Slave sends data

**FIGURE 16-2: SPI MASTER/SLAVE CONNECTION**



# PIC18F85J11 FAMILY

## 16.3.6 MASTER MODE

The master can initiate the data transfer at any time because it controls the SCK. The master determines when the slave (Processor 2, Figure 16-2) will broadcast data by the software protocol.

In Master mode, the data is transmitted/received as soon as the SSPBUF register is written to. If the SPI is only going to receive, the SDO output could be disabled (programmed as an input). The SSPSR register will continue to shift in the signal present on the SDI pin at the programmed clock rate. As each byte is received, it will be loaded into the SSPBUF register as if a normal received byte (interrupts and status bits appropriately set). This could be useful in receiver applications as a "Line Activity Monitor" mode.

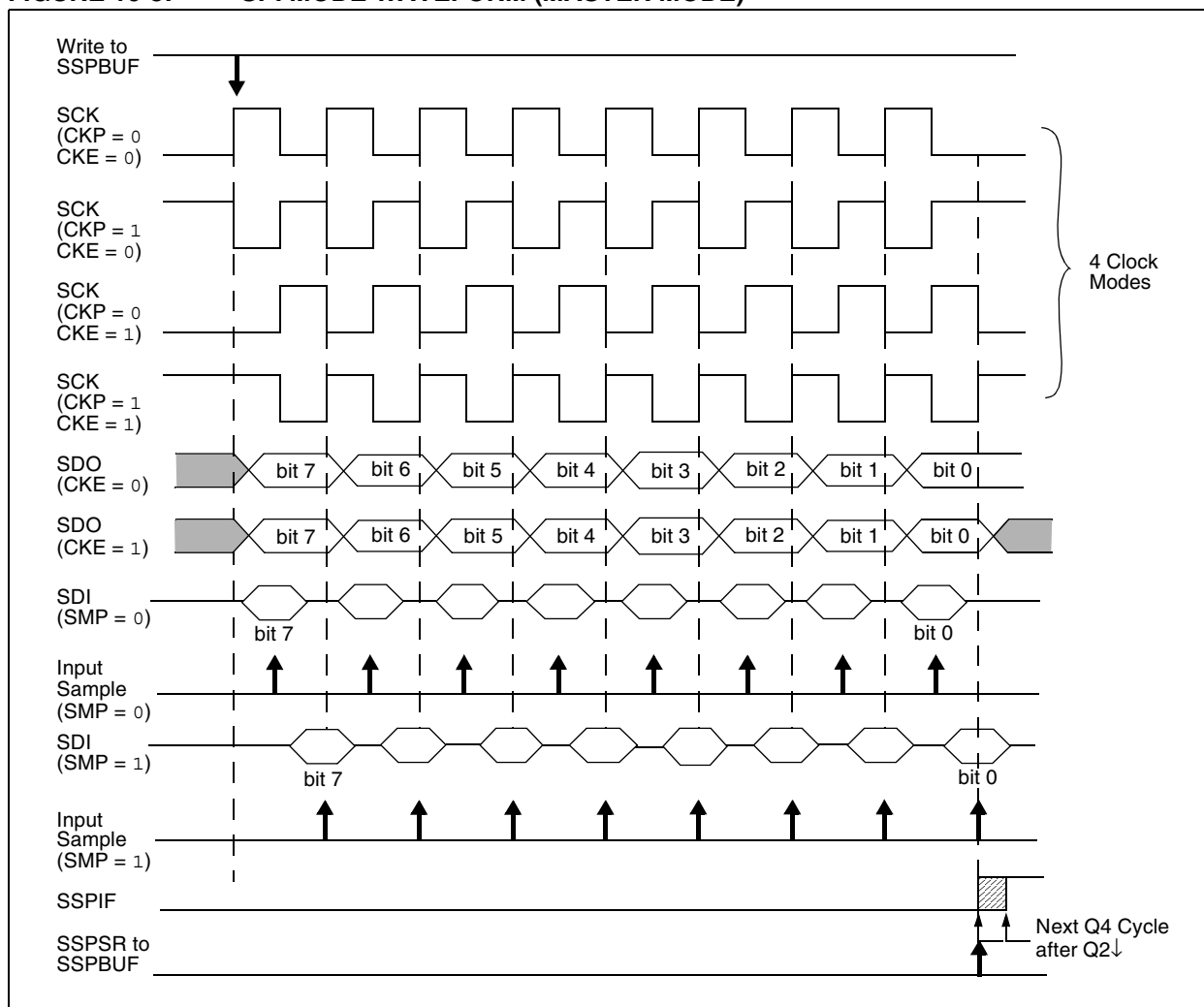
The clock polarity is selected by appropriately programming the CKP bit (SSPCON1<4>). This, then, would give waveforms for SPI communication as shown in Figure 16-3, Figure 16-5 and Figure 16-6, where the MSB is transmitted first. In Master mode, the SPI clock rate (bit rate) is user-programmable to be one of the following:

- $F_{OSC}/4$  (or  $T_{CY}$ )
- $F_{OSC}/16$  (or  $4 \cdot T_{CY}$ )
- $F_{OSC}/64$  (or  $16 \cdot T_{CY}$ )
- $\text{Timer2 output}/2$

This allows a maximum data rate (at 40 MHz) of 10.00 Mbps.

Figure 16-3 shows the waveforms for Master mode. When the CKE bit is set, the SDO data is valid before there is a clock edge on SCK. The change of the input sample is shown based on the state of the SMP bit. The time when the SSPBUF is loaded with the received data is shown.

**FIGURE 16-3: SPI MODE WAVEFORM (MASTER MODE)**



## 16.3.7 SLAVE MODE

In Slave mode, the data is transmitted and received as the external clock pulses appear on SCK. When the last bit is latched, the SSPIF interrupt flag bit is set.

Before enabling the module in SPI Slave mode, the clock line must match the proper Idle state. The clock line can be observed by reading the SCK pin. The Idle state is determined by the CKP bit (SSPCON1<4>).

While in Slave mode, the external clock is supplied by the external clock source on the SCK pin. This external clock must meet the minimum high and low times as specified in the electrical specifications.

While in Sleep mode, the slave can transmit/receive data. When a byte is received, the device will wake-up from Sleep.

## 16.3.8 SLAVE SELECT SYNCHRONIZATION

The  $\overline{SS}$  pin allows a Synchronous Slave mode. The SPI must be in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON1<3:0> = 04h). When the  $\overline{SS}$  pin is low, transmission and reception are enabled and the SDO pin is

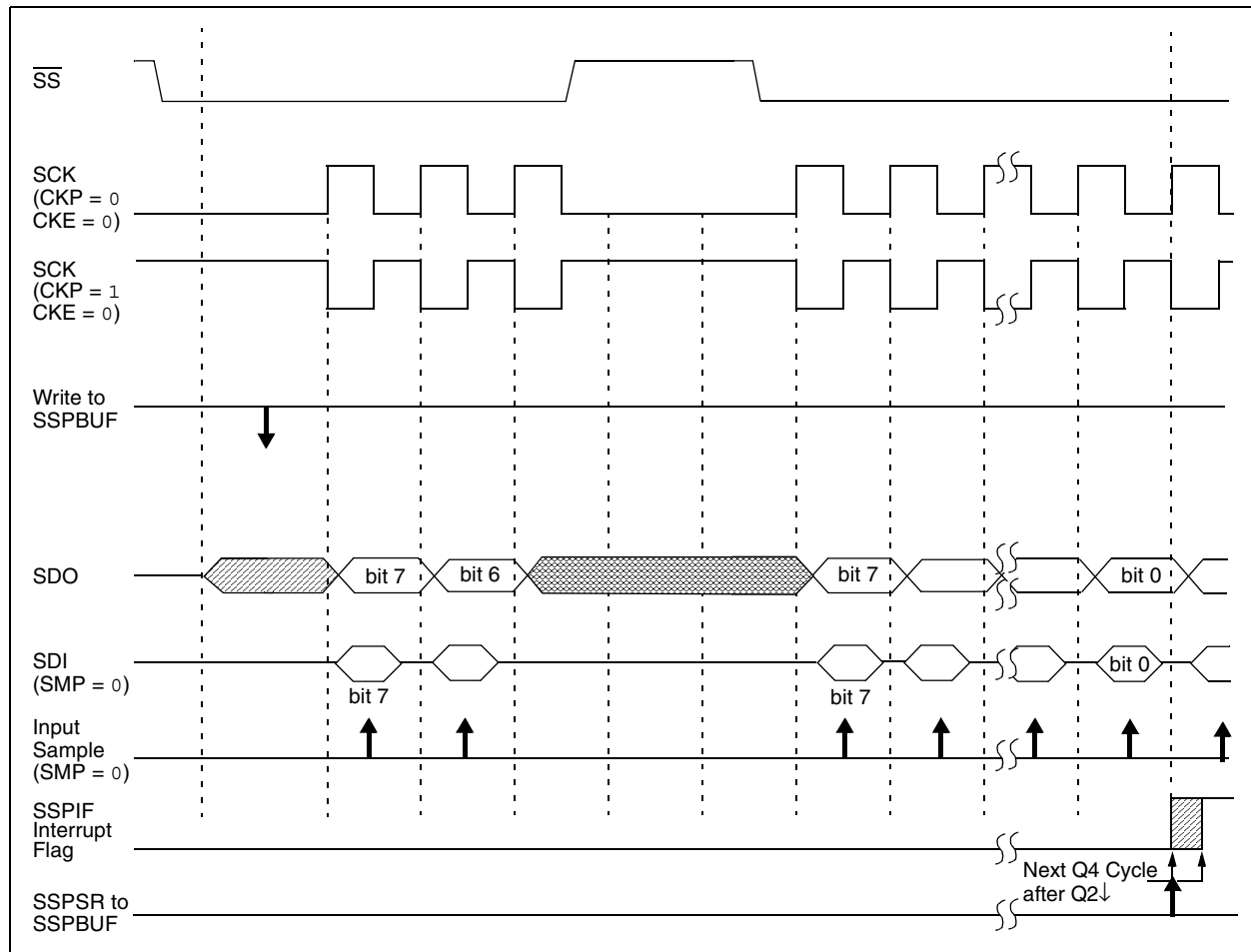
driven. When the  $\overline{SS}$  pin goes high, the SDO pin is no longer driven, even if in the middle of a transmitted byte and becomes a floating output. External pull-up/pull-down resistors may be desirable depending on the application.

- Note 1:** When the SPI is in Slave mode with  $\overline{SS}$  pin control enabled (SSPCON1<3:0> = 0100), the SPI module will reset if the  $\overline{SS}$  pin is set to VDD.
- 2:** If the SPI is used in Slave mode with CKE set, then the  $\overline{SS}$  pin control must be enabled.

When the SPI module resets, the bit counter is forced to '0'. This can be done by either forcing the  $\overline{SS}$  pin to a high level or clearing the SSPEN bit.

To emulate two-wire communication, the SDO pin can be connected to the SDI pin. When the SPI needs to operate as a receiver, the SDO pin can be configured as an input. This disables transmissions from the SDO. The SDI can always be left as an input (SDI function) since it cannot create a bus conflict.

**FIGURE 16-4: SLAVE SYNCHRONIZATION WAVEFORM**



# PIC18F85J11 FAMILY

FIGURE 16-5: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 0)

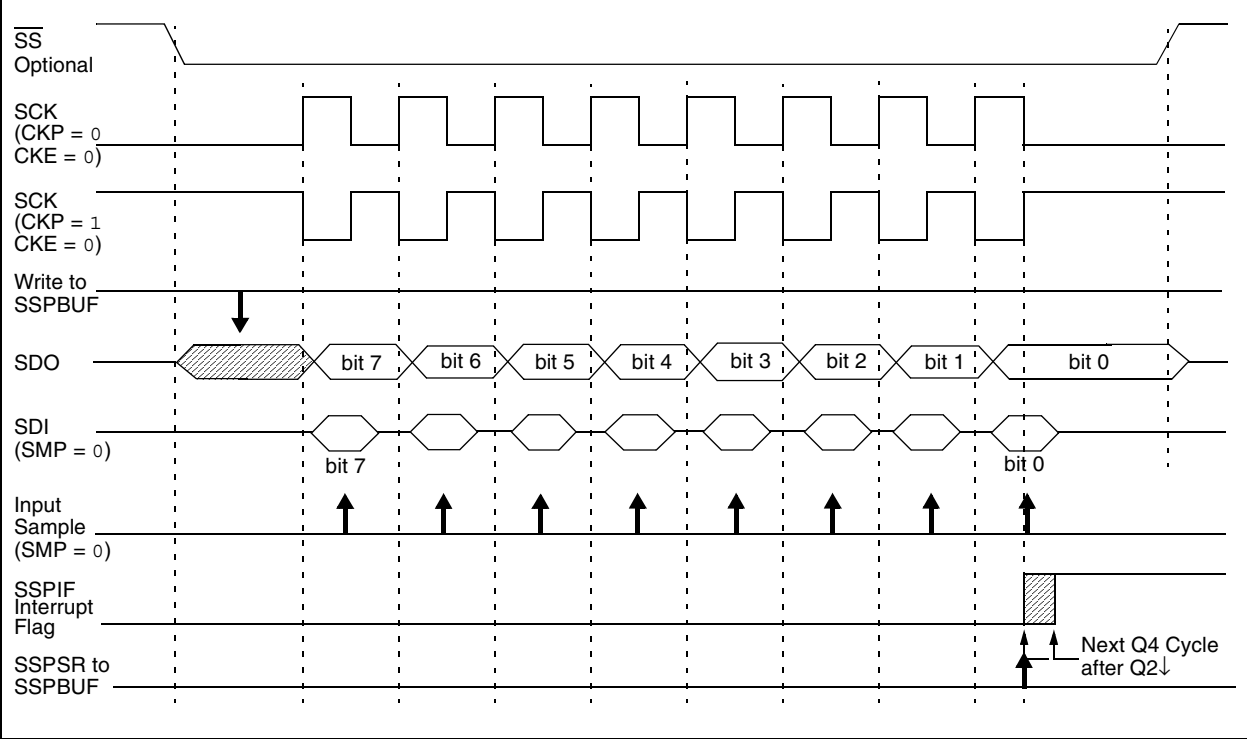
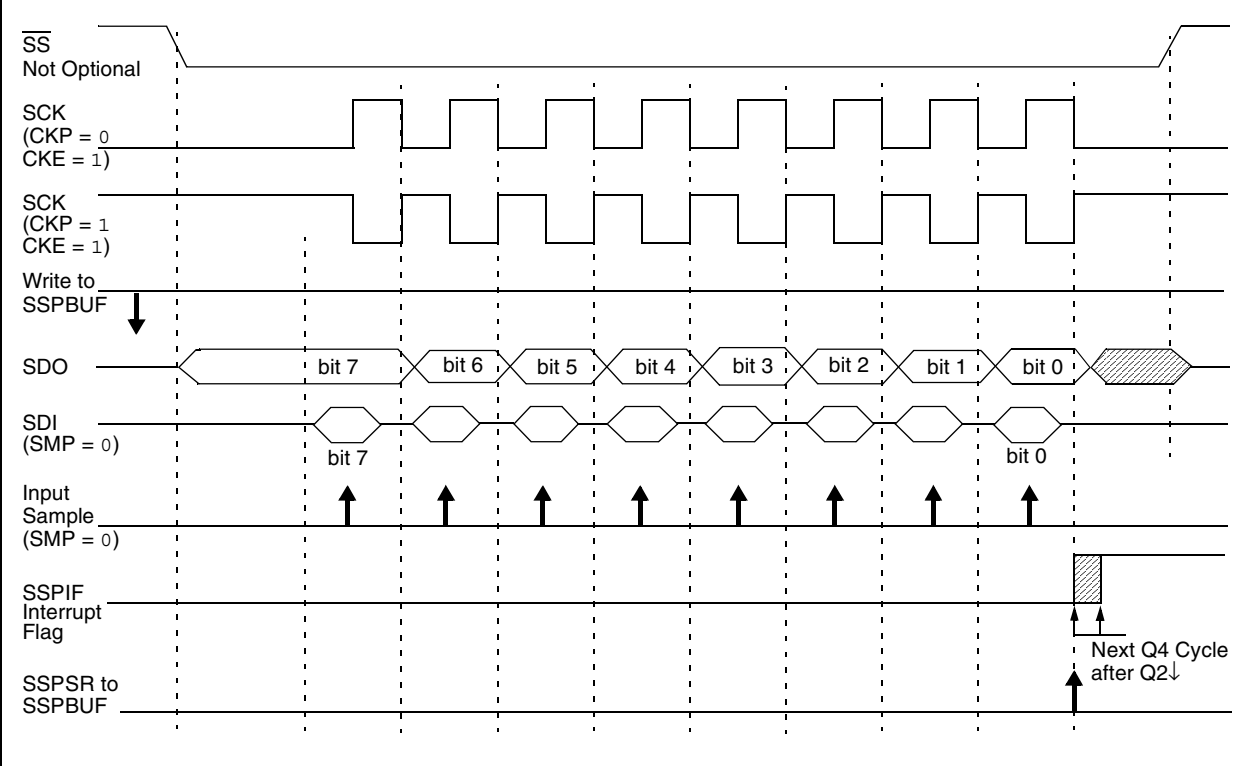


FIGURE 16-6: SPI MODE WAVEFORM (SLAVE MODE WITH CKE = 1)



## 16.3.9 OPERATION IN POWER-MANAGED MODES

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

In Idle modes, a clock is provided to the peripherals. That clock should be from the primary clock source, the secondary clock (Timer1 oscillator at 32.768 kHz) or the INTRC source. See **Section 2.3 “Clock Sources and Oscillator Switching”** for additional information.

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

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

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

In SPI Slave mode, the SPI Transmit/Receive Shift register operates asynchronously to the device. This allows the device to be placed in any power-managed

mode and data to be shifted into the SPI Transmit/Receive Shift register. When all 8 bits have been received, the MSSP interrupt flag bit will be set, and if enabled, will wake the device.

## 16.3.10 EFFECTS OF A RESET

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

## 16.3.11 BUS MODE COMPATIBILITY

Table 16-1 shows the compatibility between the standard SPI modes and the states of the CKP and CKE control bits.

**TABLE 16-1: SPI BUS MODES**

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

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

**TABLE 16-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

| 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   | 51                   |
| PIR1    | PSPIF                                 | ADIF      | RC1IF  | TX1IF  | SSPIF  | —      | TMR2IF | TMR1IF | 53                   |
| PIE1    | PSPIE                                 | ADIE      | RC1IE  | TX1IE  | SSPIE  | —      | TMR2IE | TMR1IE | 53                   |
| IPR1    | PSPIP                                 | ADIP      | RC1IP  | TX1IP  | SSPIP  | —      | TMR2IP | TMR1IP | 53                   |
| TRISC   | TRISC7                                | TRISC6    | TRISC5 | TRISC4 | TRISC3 | TRISC2 | TRISC1 | TRISC0 | 54                   |
| TRISF   | TRISF7                                | TRISF6    | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1 | —      | 54                   |
| TRISG   | SPIOD                                 | CCP2OD    | CCP1OD | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 | 54                   |
| SSPBUF  | MSSP Receive Buffer/Transmit Register |           |        |        |        |        |        |        | 52                   |
| SSPCON1 | WCOL                                  | SSPOV     | SSPEN  | CKP    | SSPM3  | SSPM2  | SSPM1  | SSPM0  | 52                   |
| SSPSTAT | SMP                                   | CKE       | D/Ā    | P      | S      | R/Ā    | UA     | BF     | 52                   |

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

# PIC18F85J11 FAMILY

## 16.4 I<sup>2</sup>C Mode

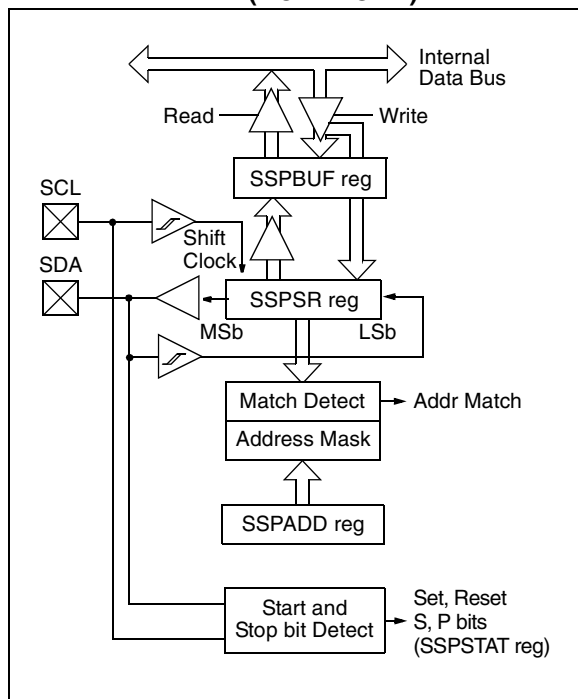
The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial Clock (SCL) – RC3/SCK/SCL
- Serial Data (SDA) – RC4/SDI/SDA

The user must configure these pins as inputs by setting the TRISC<4:3> bits.

**FIGURE 16-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MODE)**



### 16.4.1 REGISTERS

The MSSP module has six registers for I<sup>2</sup>C operation. These are:

- MSSP Control Register 1 (SSPCON1)
- MSSP Control Register 2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

Many of the bits in SSPCON2 assume different functions, depending on whether the module is operating in Master or Slave mode; bits <5:2> also assume different names in Slave mode. The different aspects of SSPCON2 are shown in Register 16-5 (for Master mode) and Register 16-6 (Slave mode).

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the MSSP is configured in I<sup>2</sup>C Slave mode. When the MSSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.



# PIC18F85J11 FAMILY

**REGISTER 16-3: SSPSTAT: MSSP STATUS REGISTER (I<sup>2</sup>C™ MODE)**

|       |       |     |                  |                  |     |    |       |
|-------|-------|-----|------------------|------------------|-----|----|-------|
| R/W-0 | R/W-0 | R-0 | R-0              | R-0              | R-0 | R0 | R-0   |
| SMP   | CKE   | D/A | P <sup>(1)</sup> | S <sup>(1)</sup> | R/W | UA | BF    |
| 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      **SMP:** Slew Rate Control bit  
In Master or Slave mode:  
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)  
0 = Slew rate control enabled for High-Speed mode (400 kHz)
- bit 6      **CKE:** SMBus Select bit  
In Master or Slave mode:  
1 = Enable SMBus specific inputs  
0 = Disable SMBus specific inputs
- bit 5      **D/A:** Data/Address bit  
In Master mode:  
Reserved.  
In Slave mode:  
1 = Indicates that the last byte received or transmitted was data  
0 = Indicates that the last byte received or transmitted was address
- bit 4      **P:** Stop bit<sup>(1)</sup>  
1 = Indicates that a Stop bit has been detected last  
0 = Stop bit was not detected last
- bit 3      **S:** Start bit<sup>(1)</sup>  
1 = Indicates that a Start bit has been detected last  
0 = Start bit was not detected last
- bit 2      **R/W:** Read/Write Information bit (I<sup>2</sup>C mode only)  
In Slave mode:<sup>(2)</sup>  
1 = Read  
0 = Write  
In Master mode:<sup>(3)</sup>  
1 = Transmit is in progress  
0 = Transmit is not in progress
- bit 1      **UA:** Update Address bit (10-Bit Slave mode only)  
1 = Indicates that the user needs to update the address in the SSPADD register  
0 = Address does not need to be updated
- bit 0      **BF:** Buffer Full Status bit  
In Transmit mode:  
1 = SSPBUF is full  
0 = SSPBUF is empty  
In Receive mode:  
1 = SSPBUF is full (does not include the  $\overline{\text{ACK}}$  and Stop bits)  
0 = SSPBUF is empty (does not include the  $\overline{\text{ACK}}$  and Stop bits)

**Note 1:** This bit is cleared on Reset and when SSPEN is cleared.

**2:** This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not  $\overline{\text{ACK}}$  bit.

**3:** ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Active mode.

# PIC18F85J11 FAMILY

## REGISTER 16-4: SSPCON1: MSSP CONTROL REGISTER 1 (I<sup>2</sup>C™ MODE)

| R/W-0 | R/W-0 | R/W-0                | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|-------|----------------------|-------|-------|-------|-------|-------|
| WCOL  | SSPOV | SSPEN <sup>(1)</sup> | CKP   | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| 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      **WCOL:** Write Collision Detect bit  
In Master Transmit mode:  
1 = A write to the SSPBUF register was attempted while the I<sup>2</sup>C conditions were not valid for a transmission to be started (must be cleared in software)  
0 = No collision  
In Slave Transmit mode:  
1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)  
0 = No collision  
In Receive mode (Master or Slave modes):  
This is a “don't care” bit.
- bit 6      **SSPOV:** Receive Overflow Indicator bit  
In Receive mode:  
1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software)  
0 = No overflow  
In Transmit mode:  
This is a “don't care” bit in Transmit mode.
- bit 5      **SSPEN:** Master Synchronous Serial Port Enable bit<sup>(1)</sup>  
1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins  
0 = Disables serial port and configures these pins as I/O port pins
- bit 4      **CKP:** SCK Release Control bit  
In Slave mode:  
1 = Release clock  
0 = Holds clock low (clock stretch), used to ensure data setup time  
In Master mode:  
Unused in this mode.
- bit 3-0      **SSPM3:SSPM0:** Synchronous Serial Port Mode Select bits  
1111 = I<sup>2</sup>C Slave mode, 10-bit address with Start and Stop bit interrupts enabled  
1110 = I<sup>2</sup>C Slave mode, 7-bit address with Start and Stop bit interrupts enabled  
1011 = I<sup>2</sup>C Firmware Controlled Master mode (slave Idle)  
1000 = I<sup>2</sup>C Master mode, clock = Fosc/(4 \* (SSPAD + 1))  
0111 = I<sup>2</sup>C Slave mode, 10-bit address  
0110 = I<sup>2</sup>C Slave mode, 7-bit address  
Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

**Note 1:** When enabled, the SDA and SCL pins must be configured as inputs.

# PIC18F85J11 FAMILY

**REGISTER 16-5: SSPCON2: MSSP CONTROL REGISTER 2 (I<sup>2</sup>C™ MASTER MODE)**

| R/W-0 | R/W-0   | R/W-0                | R/W-0                | R/W-0               | R/W-0              | R/W-0               | R/W-0              |
|-------|---------|----------------------|----------------------|---------------------|--------------------|---------------------|--------------------|
| GCEN  | ACKSTAT | ACKDT <sup>(1)</sup> | ACKEN <sup>(2)</sup> | RCEN <sup>(2)</sup> | PEN <sup>(2)</sup> | RSEN <sup>(2)</sup> | SEN <sup>(2)</sup> |
| 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      **GCEN:** General Call Enable bit  
Unused in Master mode.
- bit 6      **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)  
1 = Acknowledge was not received from slave  
0 = Acknowledge was received from slave
- bit 5      **ACKDT:** Acknowledge Data bit (Master Receive mode only)<sup>(1)</sup>  
1 = Not Acknowledge  
0 = Acknowledge
- bit 4      **ACKEN:** Acknowledge Sequence Enable bit<sup>(2)</sup>  
1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.  
0 = Acknowledge sequence Idle
- bit 3      **RCEN:** Receive Enable bit (Master Receive mode only)<sup>(2)</sup>  
1 = Enables Receive mode for I<sup>2</sup>C  
0 = Receive Idle
- bit 2      **PEN:** Stop Condition Enable bit<sup>(2)</sup>  
1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Stop condition Idle
- bit 1      **RSEN:** Repeated Start Condition Enable bit<sup>(2)</sup>  
1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Repeated Start condition Idle
- bit 0      **SEN:** Start Condition Enable bit<sup>(2)</sup>  
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.  
0 = Start condition Idle

**Note 1:** Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.

**2:** If the I<sup>2</sup>C module is active, these bits may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

# PIC18F85J11 FAMILY

## REGISTER 16-6: SSPCON2: MSSP CONTROL REGISTER 2 (I<sup>2</sup>C™ SLAVE MODE)

| R/W-0 | R/W-0   | R/W-0  | R/W-0  | R/W-0  | R/W-0  | R/W-0  | R/W-0              |
|-------|---------|--------|--------|--------|--------|--------|--------------------|
| GCEN  | ACKSTAT | ADMSK5 | ADMSK4 | ADMSK3 | ADMSK2 | ADMSK1 | SEN <sup>(1)</sup> |
| 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      **GCEN:** General Call Enable bit  
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR  
0 = General call address disabled
- bit 6      **ACKSTAT:** Acknowledge Status bit  
Unused in Slave mode.
- bit 5-2    **ADMSK5:ADMSK2:** Slave Address Mask Select bits  
1 = Masking of corresponding bits of SSPADD enabled  
0 = Masking of corresponding bits of SSPADD disabled
- bit 1      **ADMSK1:** Slave Address Least Significant bit(s) Mask Select bit  
In 7-Bit Address mode:  
1 = Masking of SSPADD<1> only enabled  
0 = Masking of SSPADD<1> only disabled  
In 10-Bit Address mode:  
1 = Masking of SSPADD<1:0> enabled  
0 = Masking of SSPADD<1:0> disabled
- bit 0      **SEN:** Stretch Enable bit<sup>(1)</sup>  
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)  
0 = Clock stretching is disabled

**Note 1:** If the I<sup>2</sup>C module is active, this bit may not be set (no spooling) and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

## 16.4.2 OPERATION

The MSSP module functions are enabled by setting the MSSP Enable bit, SSPEN (SSPCON1<5>).

The SSPCON1 register allows control of the I<sup>2</sup>C operation. Four mode selection bits (SSPCON1<3:0>) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Master mode,  
clock = (Fosc/4) x (SSPADD + 1)
- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Firmware Controlled Master mode,  
slave is Idle

Selection of any I<sup>2</sup>C mode, with the SSPEN bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC or TRISD bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

## 16.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I<sup>2</sup>C Slave mode hardware will always generate an interrupt on an exact address match. In addition, address masking will also allow the hardware to generate an interrupt for more than one address (up to 31 in 7-bit addressing and up to 63 in 10-bit addressing). Through the mode select bits, the user can also choose to interrupt on Start and Stop bits.

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (**ACK**) pulse and load the SSPBUF register with the received value currently in the SSPSR register.

Any combination of the following conditions will cause the MSSP module not to give this **ACK** pulse:

- The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.
- The MSSP Overflow bit, SSPOV (SSPCON1<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but the SSPIF bit is set. The BF bit is cleared by reading the SSPBUF register, while the SSPOV bit is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirement of the MSSP module, are shown in timing parameter 100 and parameter 101.

### 16.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit, BF, is set.
3. An **ACK** pulse is generated.
4. The MSSP Interrupt Flag bit, SSPIF, is set (and the interrupt is generated, if enabled) on the falling edge of the ninth SCL pulse.

In 10-Bit Addressing mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110 A9 A8 0', where 'A9' and 'A8' are the two MSBs of the address. The sequence of events for 10-bit addressing is as follows, with steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
4. Receive second (low) byte of address (SSPIF, BF and UA bits are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear UA bit.
6. Read the SSPBUF register (clears bit BF) and clear flag bit, SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (SSPIF and BF bits are set).
9. Read the SSPBUF register (clears BF bit) and clear flag bit, SSPIF.

# PIC18F85J11 FAMILY

## 16.4.3.2 Address Masking

Masking an address bit causes that bit to become a “don’t care”. When one address bit is masked, two addresses will be Acknowledged and cause an interrupt. It is possible to mask more than one address bit at a time, which makes it possible to Acknowledge up to 31 addresses in 7-Bit Addressing mode and up to 63 addresses in 10-Bit Addressing mode (see Example 16-2).

The I<sup>2</sup>C Slave behaves the same way, whether address masking is used or not. However, when address masking is used, the I<sup>2</sup>C slave can Acknowledge multiple addresses and cause interrupts. When this occurs, it is necessary to determine which address caused the interrupt by checking SSPBUF.

In 7-Bit Addressing mode, Address Mask bits, ADMSK<5:1> (SSPCON2<5:1>), mask the corresponding address bits in the SSPADD register. For any ADMSK bits that are set (ADMSK<n> = 1), the corresponding address bit is ignored (SSPADD<n> = x). For the module

to issue an address Acknowledge, it is sufficient to match only on addresses that do not have an active address mask.

In 10-Bit Addressing mode, ADMSK<5:2> bits mask the corresponding address bits in the SSPADD register. In addition, ADMSK1 simultaneously masks the two LSBs of the address (SSPADD<1:0>). For any ADMSK bits that are active (ADMSK<n> = 1), the corresponding address bit is ignored (SSPADD<n> = x). Also note that although in 10-Bit Addressing mode, the upper address bits reuse part of the SSPADD register bits, the address mask bits do not interact with those bits. They only affect the lower address bits.

**Note 1:** ADMSK1 masks the two Least Significant bits of the address.

**2:** The two Most Significant bits of the address are not affected by address masking.

## EXAMPLE 16-2: ADDRESS MASKING EXAMPLES

### 7-Bit Addressing:

SSPADD<7:1> = A0h (1010000) (SSPADD<0> is assumed to be '0')

ADMSK<5:1> = 00111

Addresses Acknowledged: A0h, A2h, A4h, A6h, A8h, AAh, ACh, AEh

### 10-Bit Addressing:

SSPADD<7:0> = A0h (10100000) (the two MSBs of the address are ignored in this example, since they are not affected by masking)

ADMSK<5:1> = 00111

Addresses Acknowledged: A0h, A1h, A2h, A3h, A4h, A5h, A6h, A7h, A8h, A9h, AAh, ABh, ACh, ADh, AEh, AFh

## 16.4.3.3 Reception

When the  $\overline{R/\overline{W}}$  bit of the address byte is clear and an address match occurs, the  $\overline{R/\overline{W}}$  bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low ( $\overline{ACK}$ ).

When the address byte overflow condition exists, then the no Acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is defined as either bit, BF (SSPSTAT<0>), is set or bit, SSPOV (SSPCON1<6>), is set.

An MSSP interrupt is generated for each data transfer byte. The interrupt flag bit, SSPIF, must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPCON2<0> = 1), SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit, CKP (SSPCON1<4>). See **Section 16.4.4 “Clock Stretching”** for more details.

## 16.4.3.4 Transmission

When the  $\overline{R/\overline{W}}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/\overline{W}}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit and pin RC3 is held low, regardless of SEN (see **Section 16.4.4 “Clock Stretching”** for more details). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register which also loads the SSPSR register. Then, pin RC3 should be enabled by setting bit, CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 16-10).

The  $\overline{ACK}$  pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not  $\overline{ACK}$ ), then the data transfer is complete. In this case, when the  $\overline{ACK}$  is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low ( $\overline{ACK}$ ), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3 must be enabled by setting bit, CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

# PIC18F85J11 FAMILY

FIGURE 16-8: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESSING)

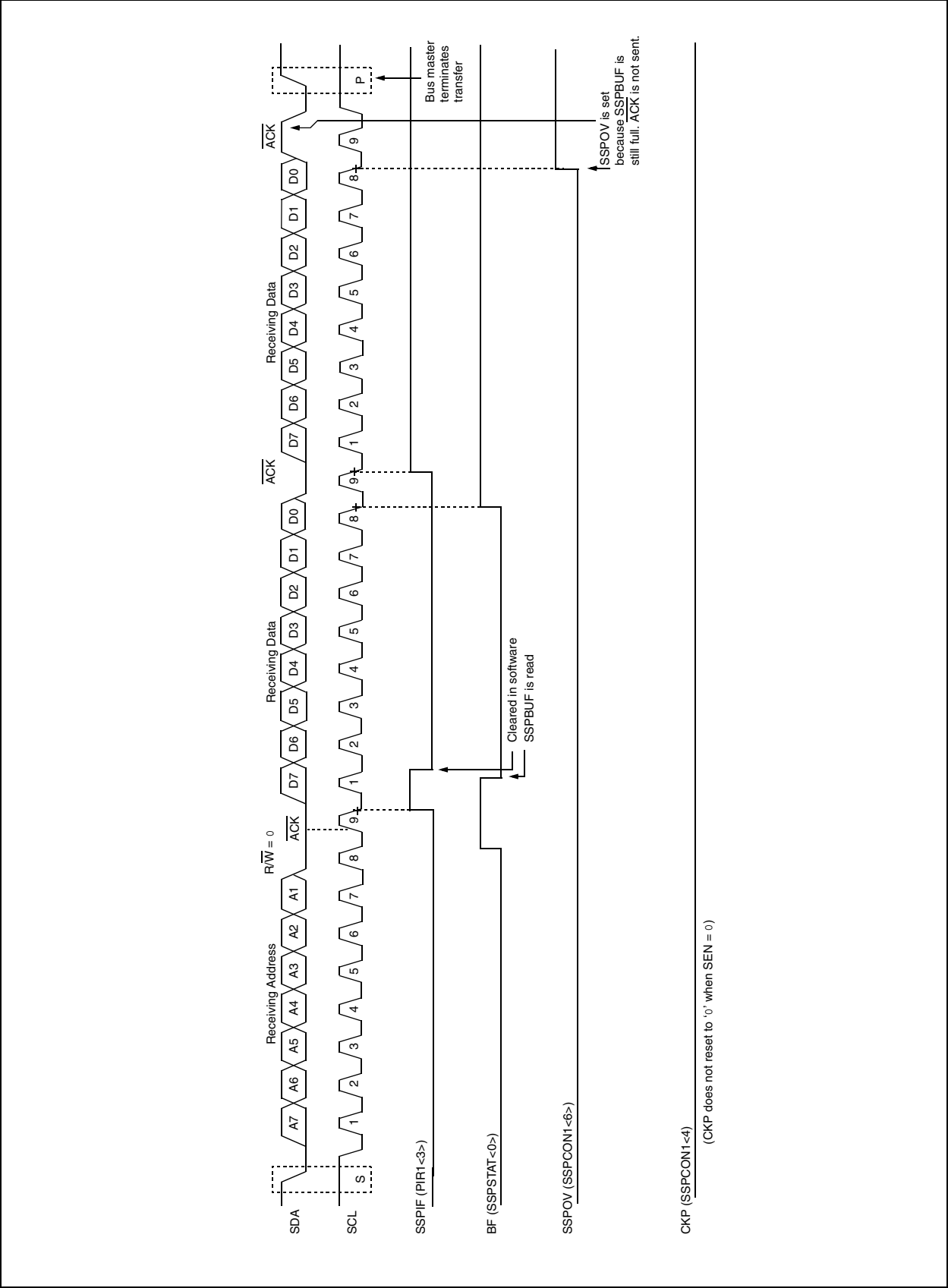
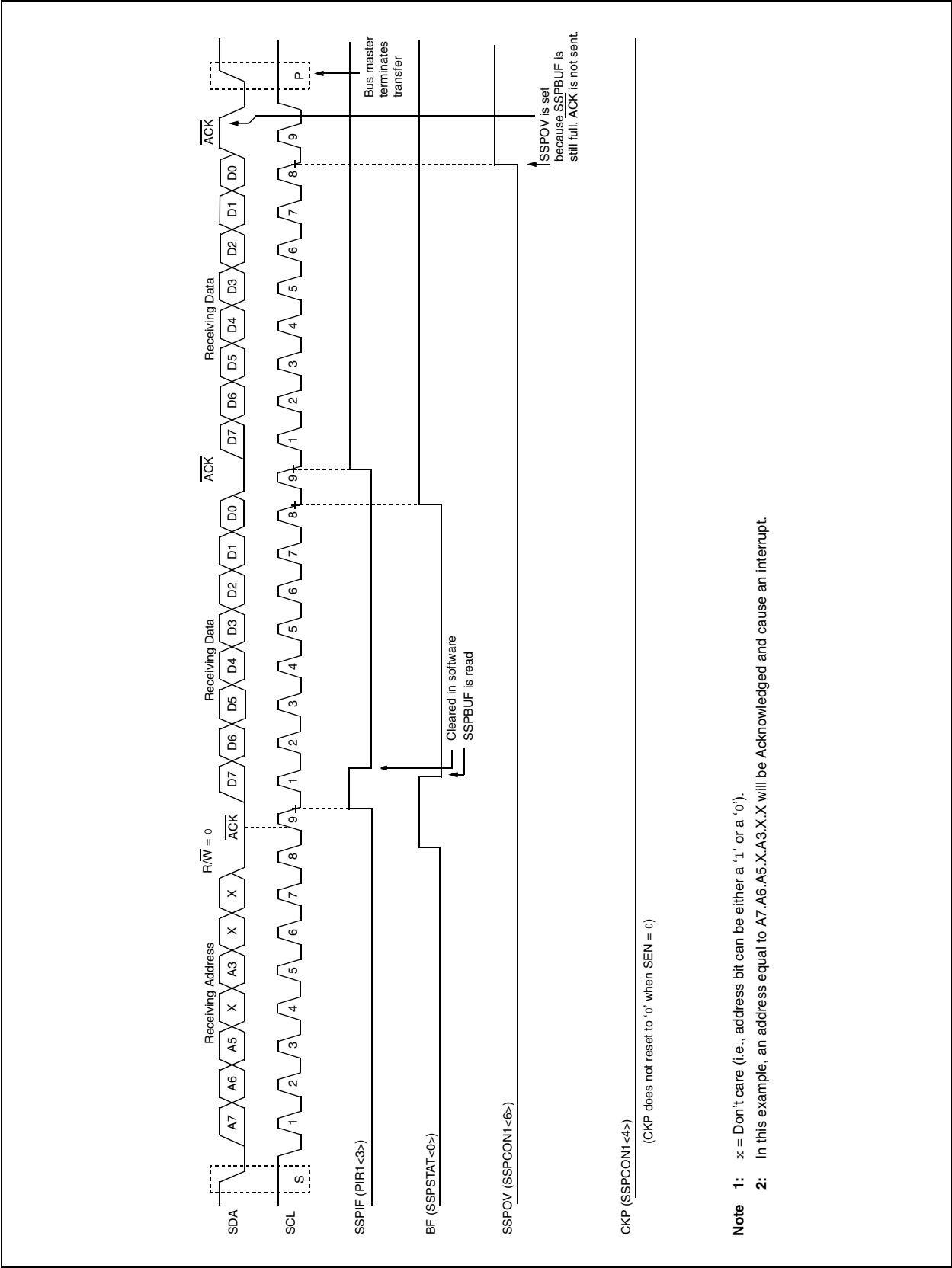




FIGURE 16-9: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01011 (RECEPTION, 7-BIT ADDRESSING)



# PIC18F85J11 FAMILY

FIGURE 16-10: I<sup>2</sup>C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESSING)

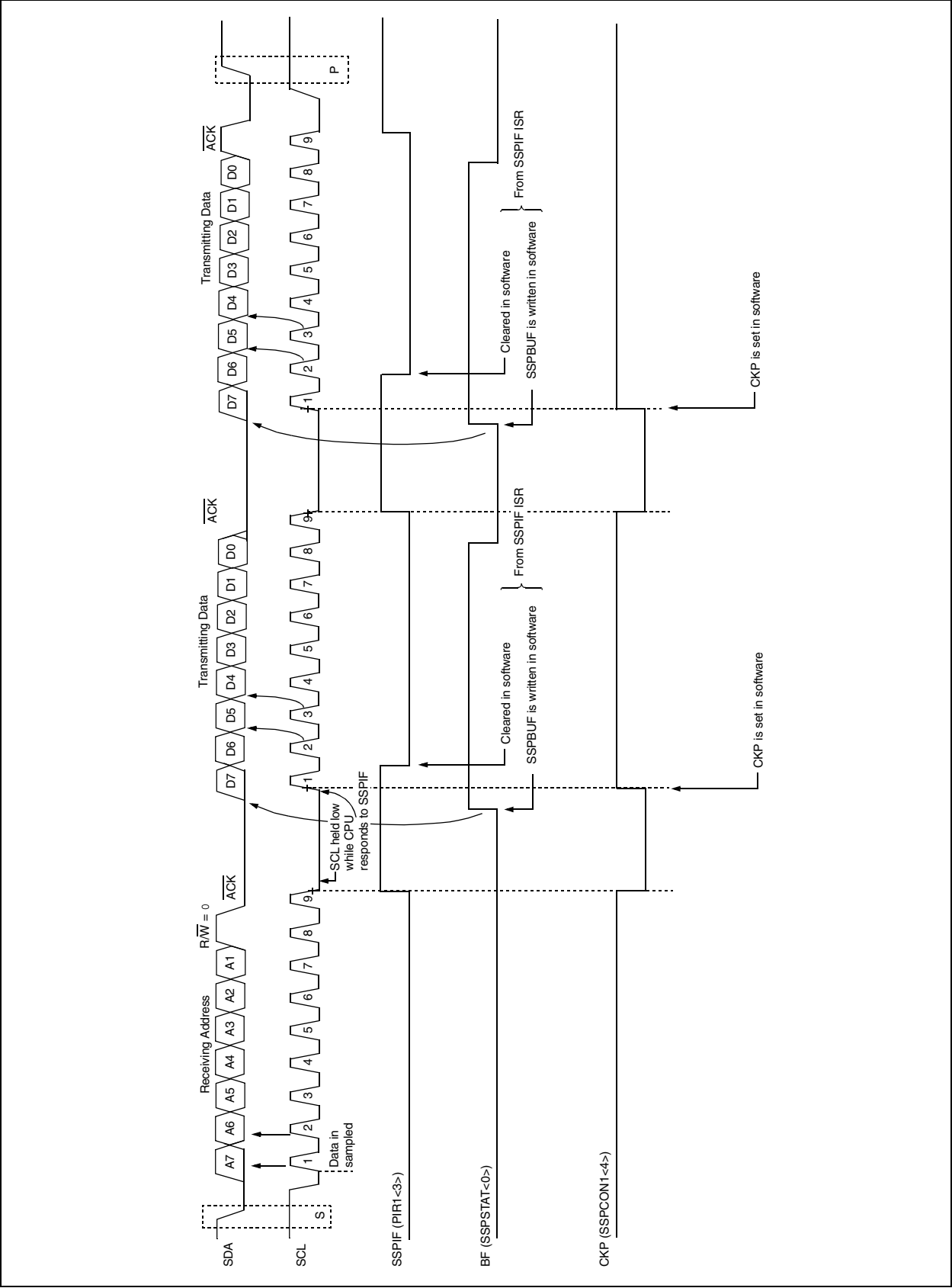
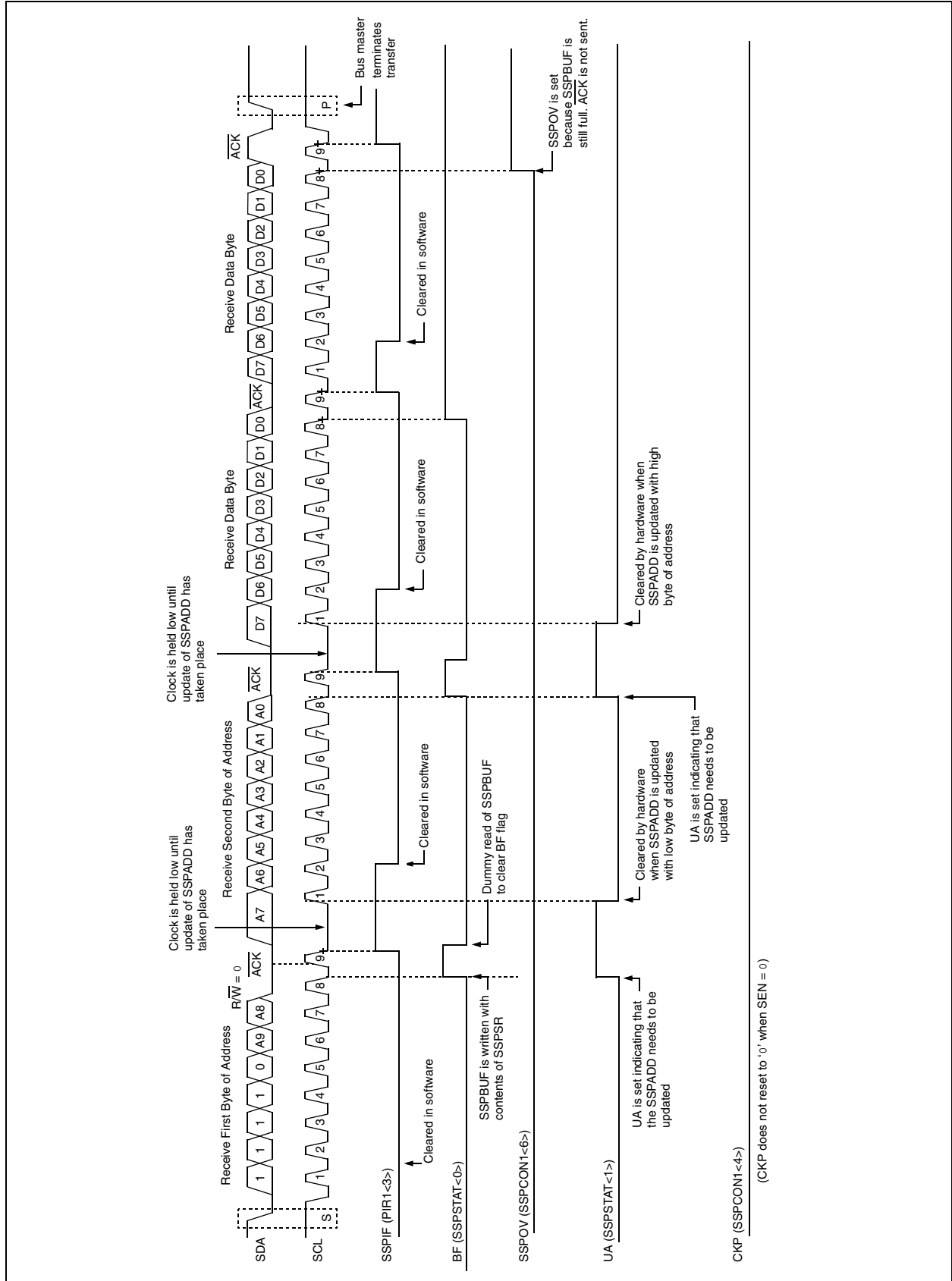


**FIGURE 16-11: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESSING)**



# PIC18F85J11 FAMILY

**FIGURE 16-12: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 0 AND ADMSK<5:1> = 01001 (RECEPTION, 10-BIT ADDRESSING)**

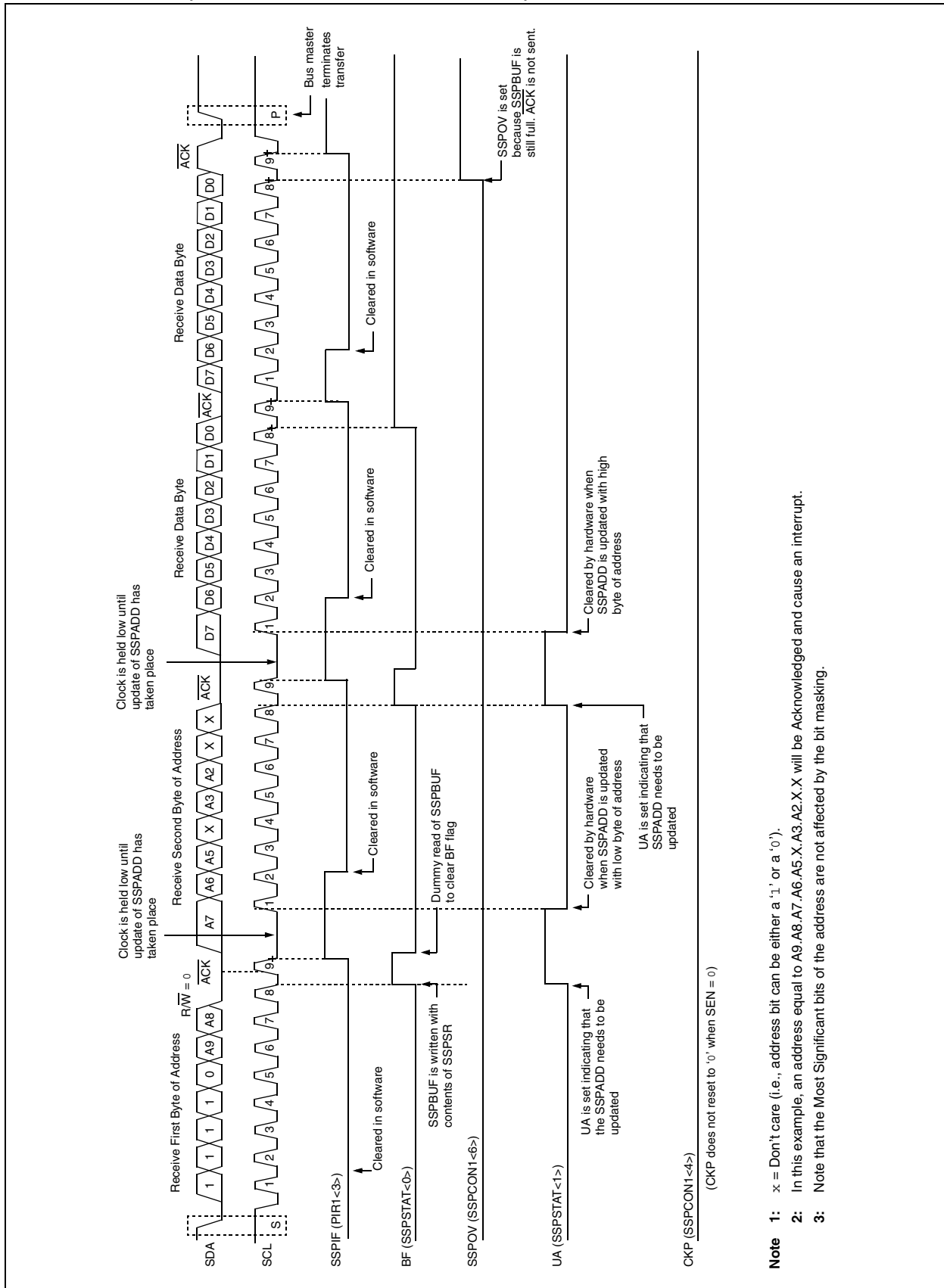
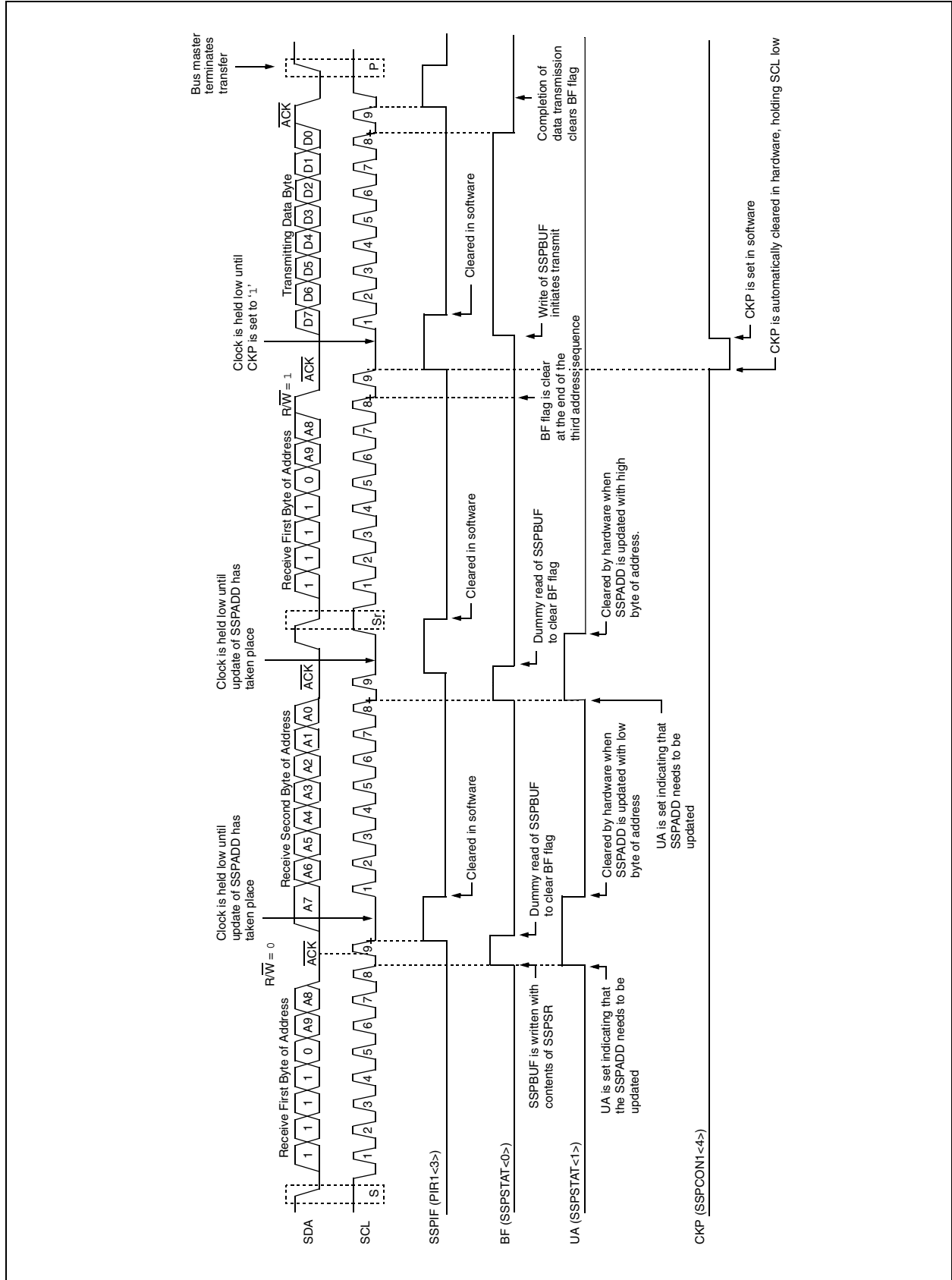


FIGURE 16-13: I<sup>2</sup>C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESSING)



# PIC18F85J11 FAMILY

## 16.4.4 CLOCK STRETCHING

Both 7-Bit and 10-Bit Slave modes implement automatic clock stretching during a transmit sequence.

The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

### 16.4.4.1 Clock Stretching for 7-Bit Slave Receive Mode (SEN = 1)

In 7-Bit Slave Receive mode when the BF bit is set, on the falling edge of the ninth clock at the end of the ACK sequence, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP bit being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring (see Figure 16-15).

**Note 1:** If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

### 16.4.4.2 Clock Stretching for 10-Bit Slave Receive Mode (SEN = 1)

In 10-Bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but the CKP bit is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/W bit cleared to '0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

**Note:** If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

### 16.4.4.3 Clock Stretching for 7-Bit Slave Transmit Mode

The 7-Bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock if the BF bit is clear. This occurs regardless of the state of the SEN bit.

The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 16-10).

**Note 1:** If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.

**2:** The CKP bit can be set in software regardless of the state of the BF bit.

### 16.4.4.4 Clock Stretching for 10-Bit Slave Transmit Mode

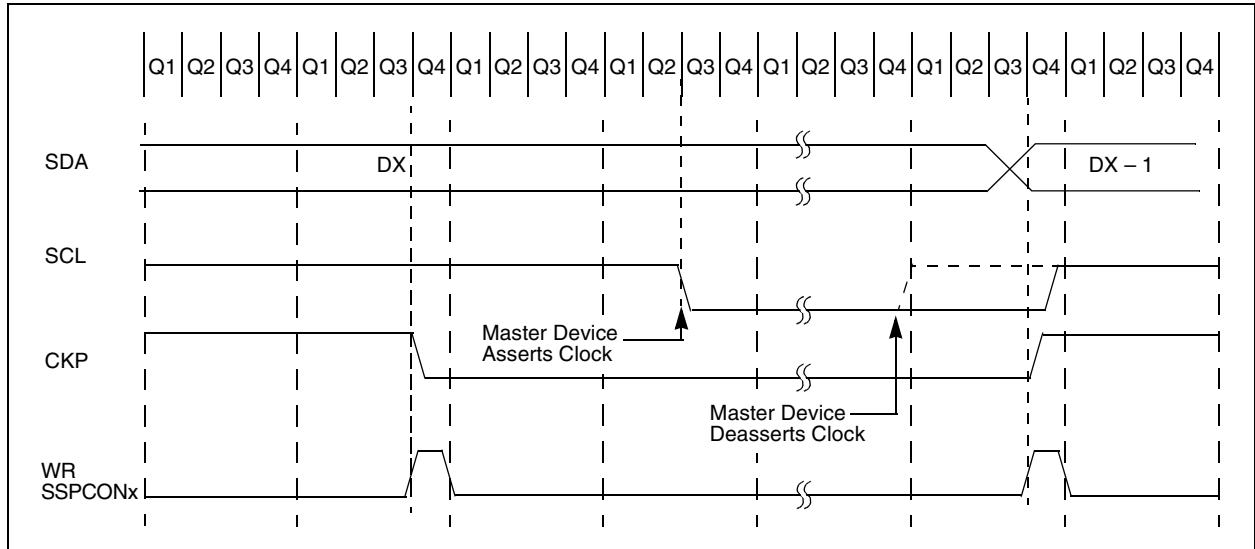
In 10-Bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-Bit Slave Receive mode. The first two addresses are followed by a third address sequence which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-Bit Slave Transmit mode (see Figure 16-13).

## 16.4.4.5 Clock Synchronization and the CKP bit

When the CKP bit is cleared, the SCL output is forced to '0'. However, clearing the CKP bit will not assert the SCL output low until the SCL output is already sampled low. Therefore, the CKP bit will not assert the SCL line until an external I<sup>2</sup>C master device has

already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I<sup>2</sup>C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 16-14).

**FIGURE 16-14: CLOCK SYNCHRONIZATION TIMING**



# PIC18F85J11 FAMILY

FIGURE 16-15: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESSING)

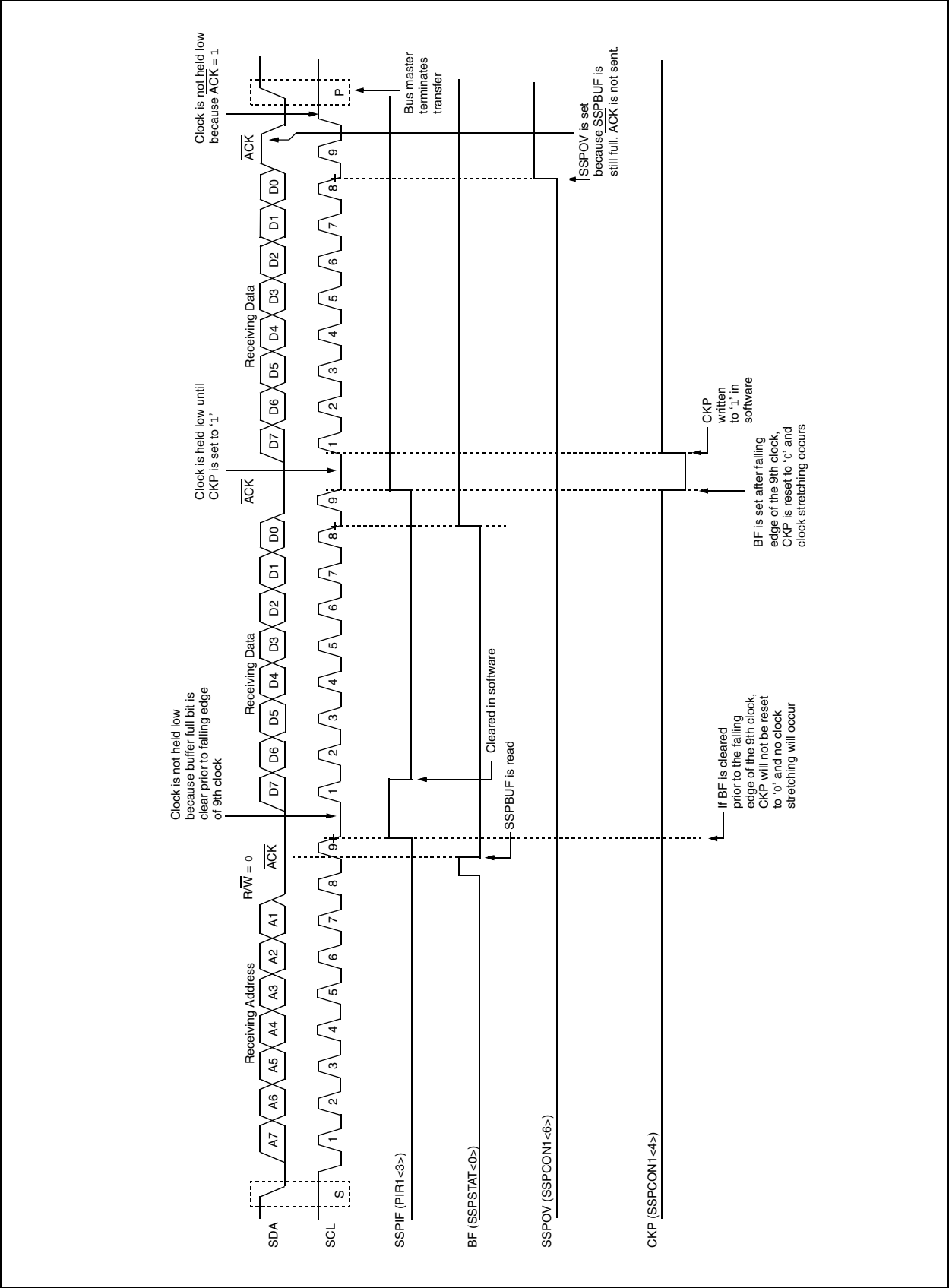
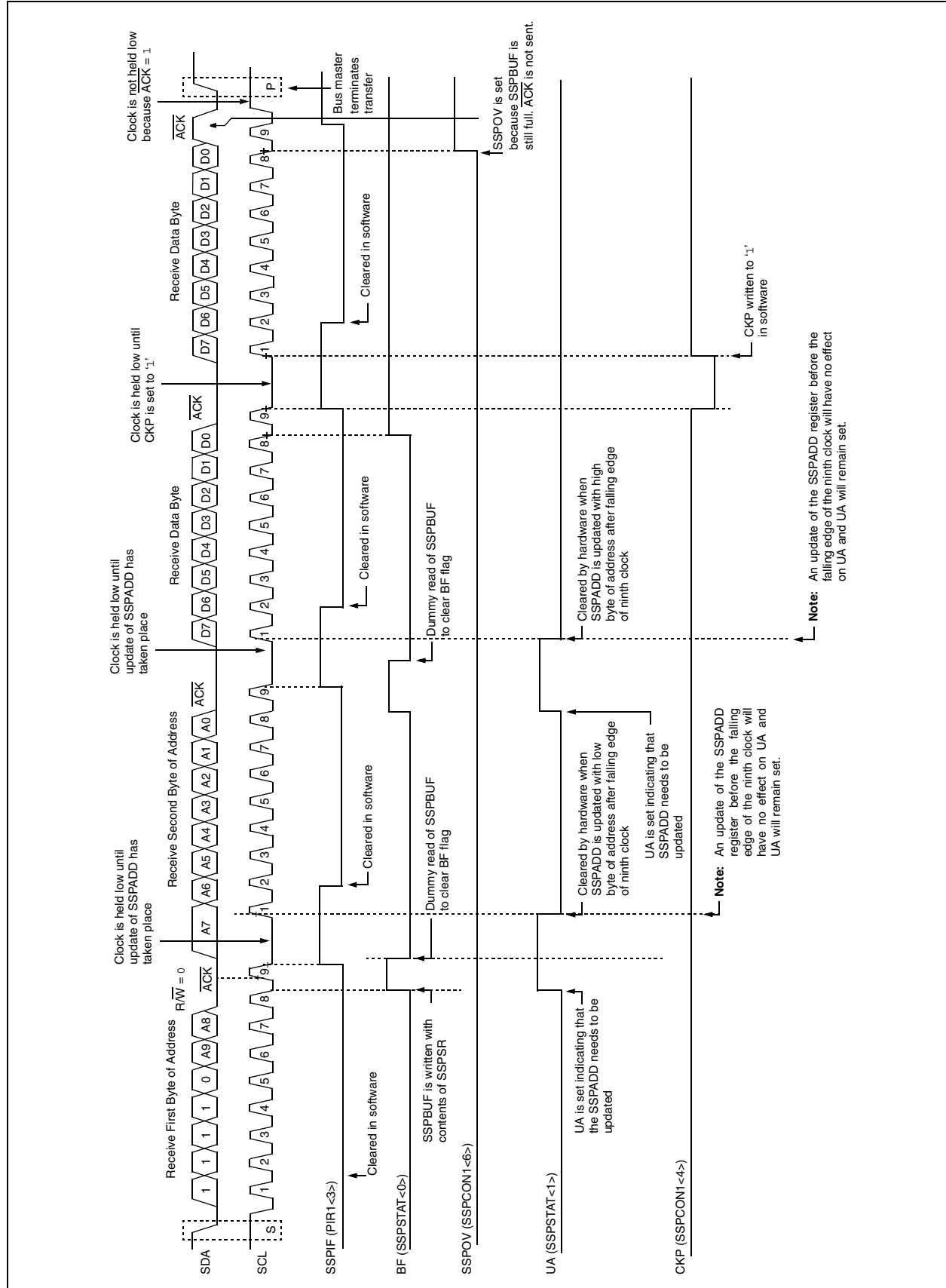




**FIGURE 16-16: I<sup>2</sup>C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESSING)**



# PIC18F85J11 FAMILY

## 16.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I<sup>2</sup>C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

The general call address is one of eight addresses reserved for specific purposes by the I<sup>2</sup>C protocol. It consists of all '0's with  $R/\bar{W} = 0$ .

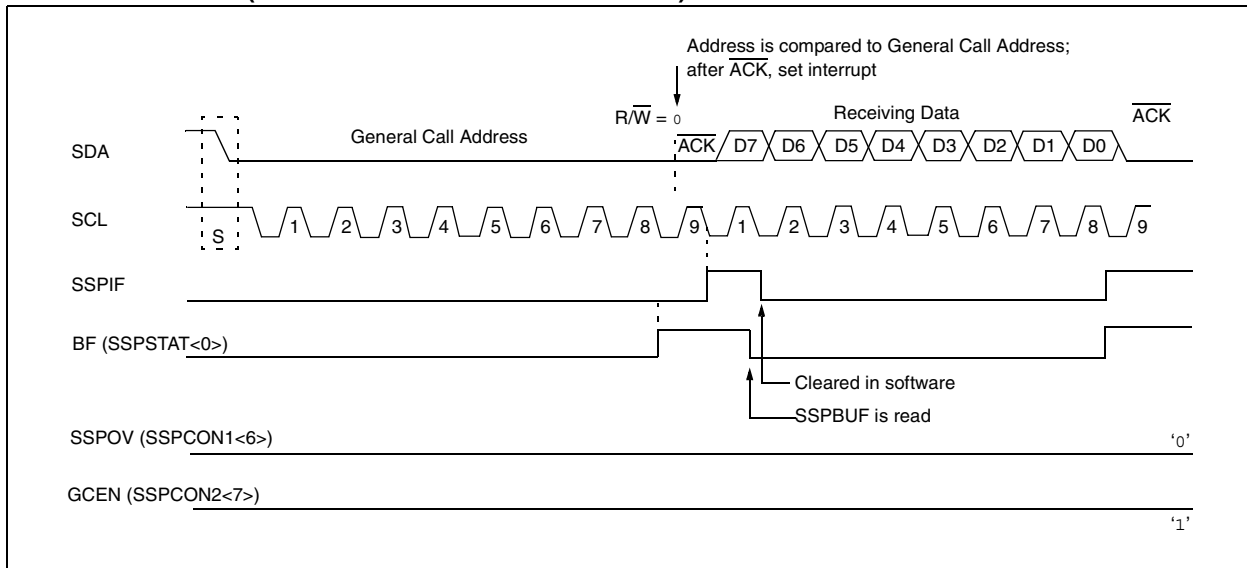
The general call address is recognized when the General Call Enable bit, GCEN, is enabled (SSPCON2<7> set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit ( $\bar{ACK}$  bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address.

In 10-Bit Addressing mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-Bit Addressing mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 16-17).

**FIGURE 16-17: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESSING MODE)**



## 16.4.6 MASTER MODE

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

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

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

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

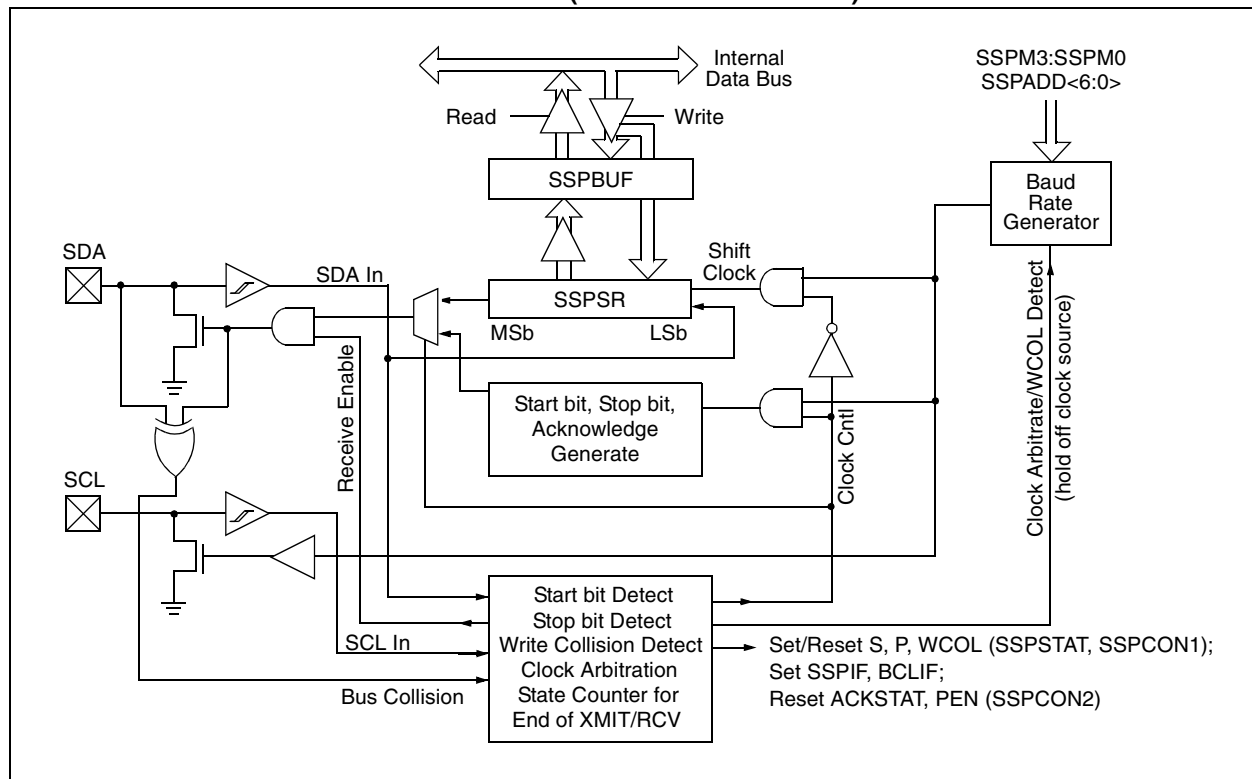
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I<sup>2</sup>C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6.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

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

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

- Start Condition
- Stop Condition
- Data Transfer Byte Transmitted/Received
- Acknowledge Transmit
- Repeated Start

**FIGURE 16-18: MSSP BLOCK DIAGRAM (I<sup>2</sup>C™ MASTER MODE)**



# PIC18F85J11 FAMILY

---

## 16.4.6.1 I<sup>2</sup>C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I<sup>2</sup>C bus will not be released.

In Master Transmitter mode, serial data is output through SDA, while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate the receive bit. Serial data is received via SDA, while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I<sup>2</sup>C operation. See **Section 16.4.7 "Baud Rate"** for more detail.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation takes place.
3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit, PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

## 16.4.7 BAUD RATE

In I<sup>2</sup>C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPADD register (Figure 16-19). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I<sup>2</sup>C Master mode, the BRG is reloaded automatically.

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

Table 16-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

### 16.4.7.1 Baud Rate Generation in Power-Managed Modes

When the device is operating in one of the power-managed modes, the clock source to the BRG may change frequency, or even stop, depending on the mode and clock source selected. Switching to a Run or Idle mode from either the secondary clock or internal oscillator is likely to change the clock rate to the BRG. In Sleep mode, the BRG will not be clocked at all.

**FIGURE 16-19: BAUD RATE GENERATOR BLOCK DIAGRAM**

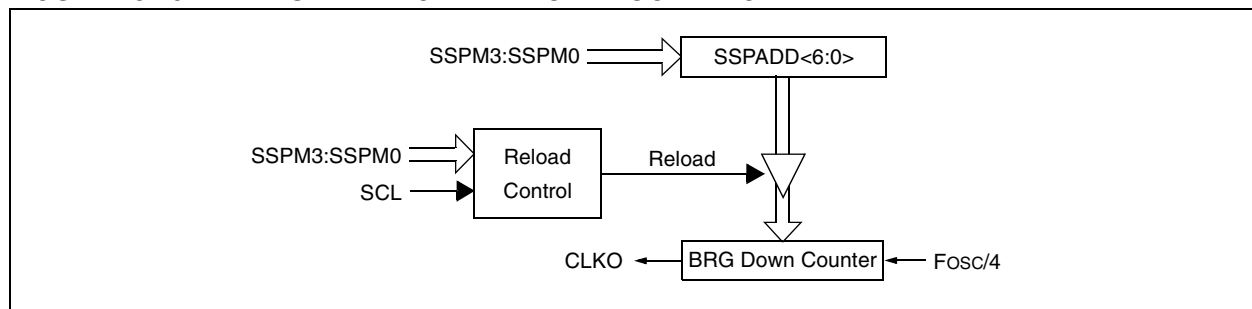


**TABLE 16-3: I<sup>2</sup>C™ CLOCK RATE w/BRG**

| Fcy    | Fcy * 2 | BRG Value | Fscl<br>(2 Rollovers of BRG) |
|--------|---------|-----------|------------------------------|
| 10 MHz | 20 MHz  | 18h       | 400 kHz <sup>(1)</sup>       |
| 10 MHz | 20 MHz  | 1Fh       | 312.5 kHz                    |
| 10 MHz | 20 MHz  | 63h       | 100 kHz                      |
| 4 MHz  | 8 MHz   | 09h       | 400 kHz <sup>(1)</sup>       |
| 4 MHz  | 8 MHz   | 0Ch       | 308 kHz                      |
| 4 MHz  | 8 MHz   | 27h       | 100 kHz                      |
| 1 MHz  | 2 MHz   | 02h       | 333 kHz <sup>(1)</sup>       |
| 1 MHz  | 2 MHz   | 09h       | 100 kHz                      |
| 1 MHz  | 2 MHz   | 00h       | 1 MHz <sup>(1)</sup>         |

**Note 1:** The I<sup>2</sup>C™ interface does not conform to the 400 kHz I<sup>2</sup>C 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.

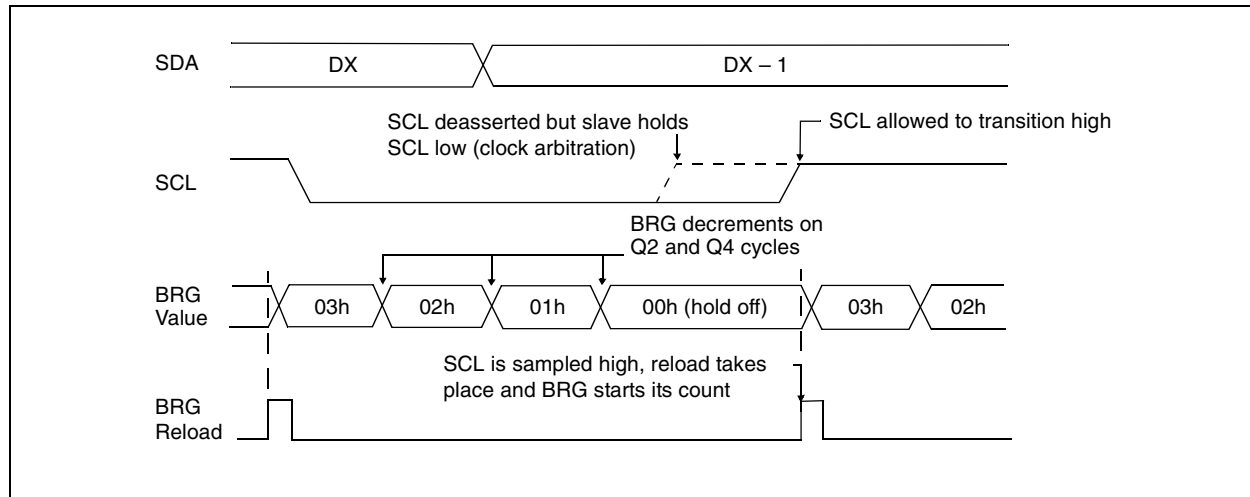
# PIC18F85J11 FAMILY

## 16.4.7.2 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the

SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 16-20).

**FIGURE 16-20: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



## 16.4.8 I<sup>2</sup>C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start Enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low while SCL is high is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware. The Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

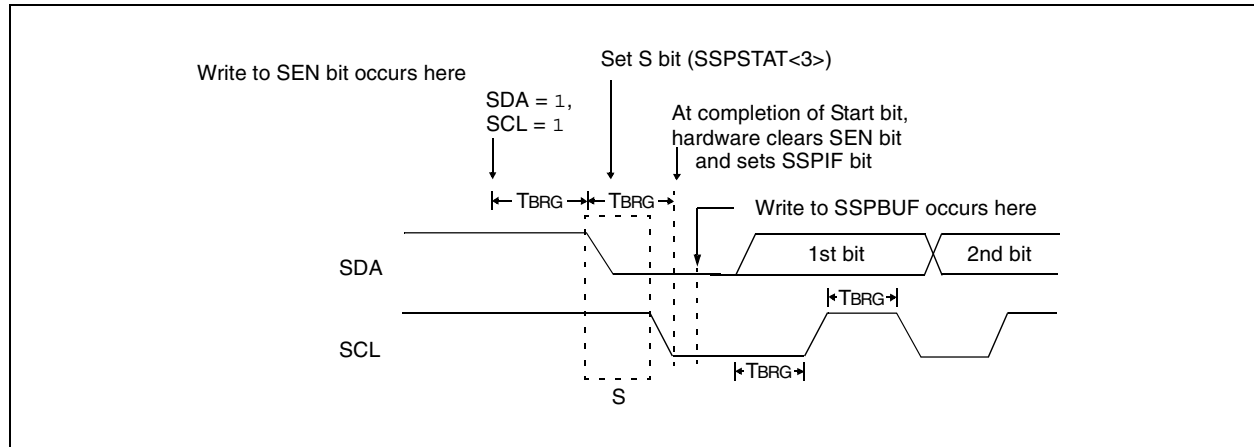
**Note:** If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs. The Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I<sup>2</sup>C module is reset into its Idle state.

### 16.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

**FIGURE 16-21: FIRST START BIT TIMING**



# PIC18F85J11 FAMILY

## 16.4.9 I<sup>2</sup>C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I<sup>2</sup>C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<6:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

**Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.

**2:** A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

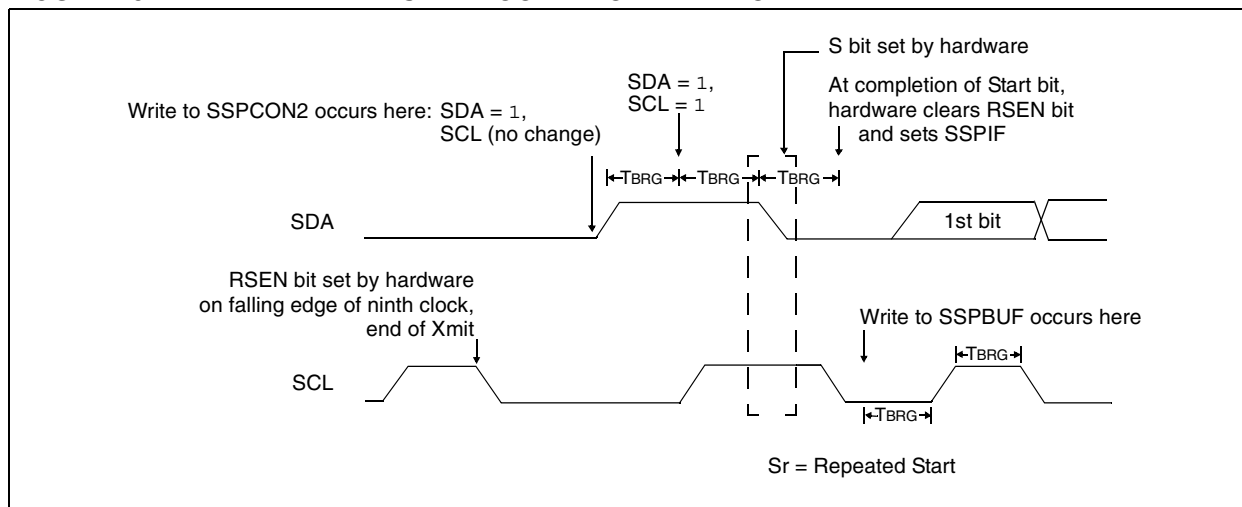
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-Bit Addressing mode or the default first address in 10-Bit Addressing mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-Bit Addressing mode) or eight bits of data (7-Bit Addressing mode).

### 16.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

**Note:** Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

**FIGURE 16-22: REPEATED START CONDITION WAVEFORM**





## 16.4.10 I<sup>2</sup>C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter 106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter 107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time if an address match occurred, or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared; if not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 16-23).

After the write to the SSPBUF, each bit of the address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF bit is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

### 16.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

### 16.4.10.2 WCOL Status Flag

If the user writes to the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur) after 2 Tcy after the SSPBUF write. If SSPBUF is rewritten within 2 Tcy, the WCOL bit is set and SSPBUF is updated. This may result in a corrupted transfer.

The user should verify that the WCOL is clear after each write to SSPBUF to ensure the transfer is correct. In all cases, WCOL must be cleared in software.

### 16.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge (ACK = 0) and is set when the slave does not Acknowledge (ACK = 1). A slave sends an Acknowledge when it has recognized its address (including a general call), or when the slave has properly received its data.

## 16.4.11 I<sup>2</sup>C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

**Note:** The MSSP module must be in an Idle state before the RCEN bit is set or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting, and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

### 16.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

### 16.4.11.2 SSPOV Status Flag

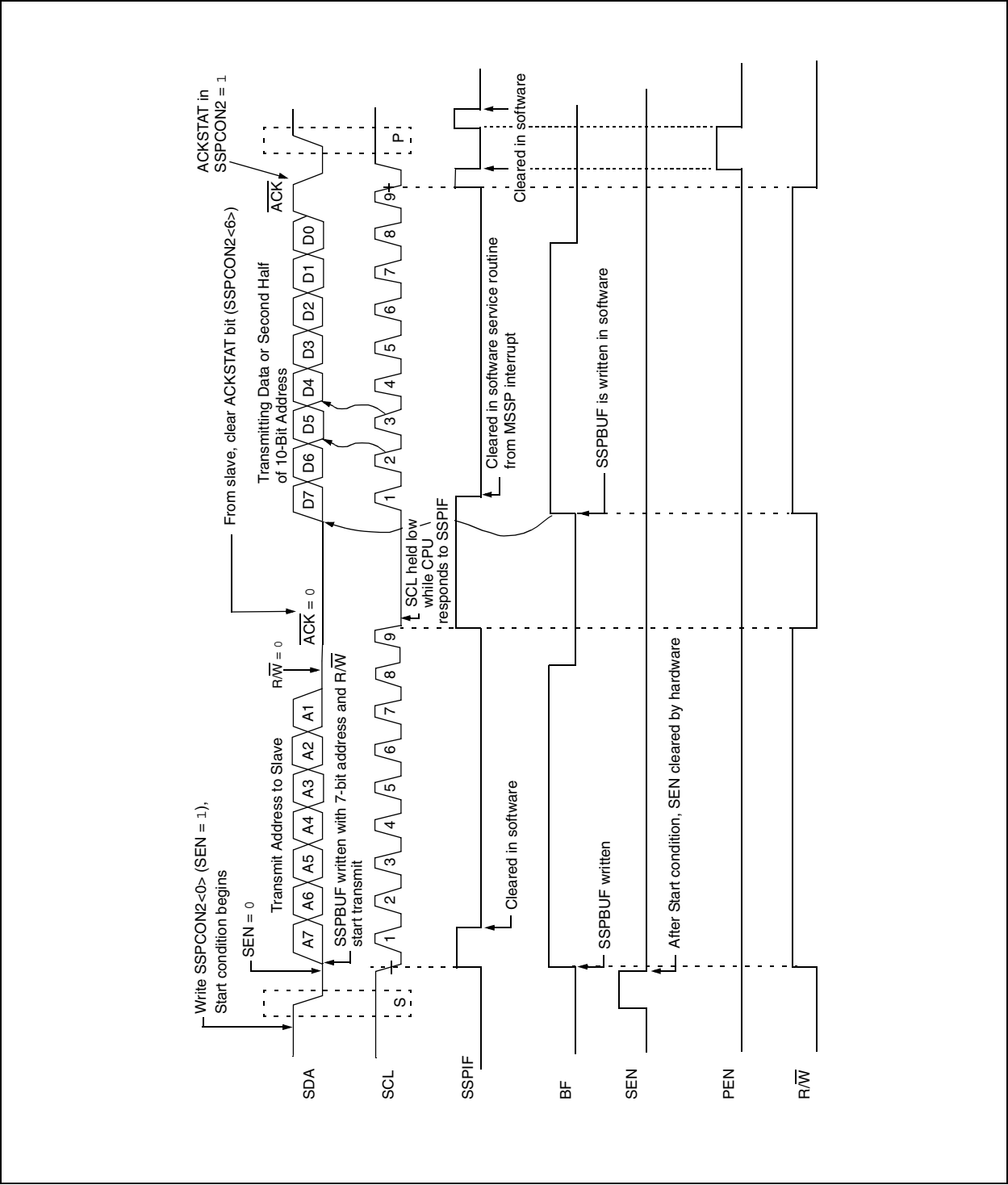
In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag bit is already set from a previous reception.

### 16.4.11.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

# PIC18F85J11 FAMILY

FIGURE 16-23: I<sup>2</sup>C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESSING)





# PIC18F85J11 FAMILY

## 16.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 16-25).

### 16.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

## 16.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to '0'. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 16-26).

### 16.4.13.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

**FIGURE 16-25: ACKNOWLEDGE SEQUENCE WAVEFORM**

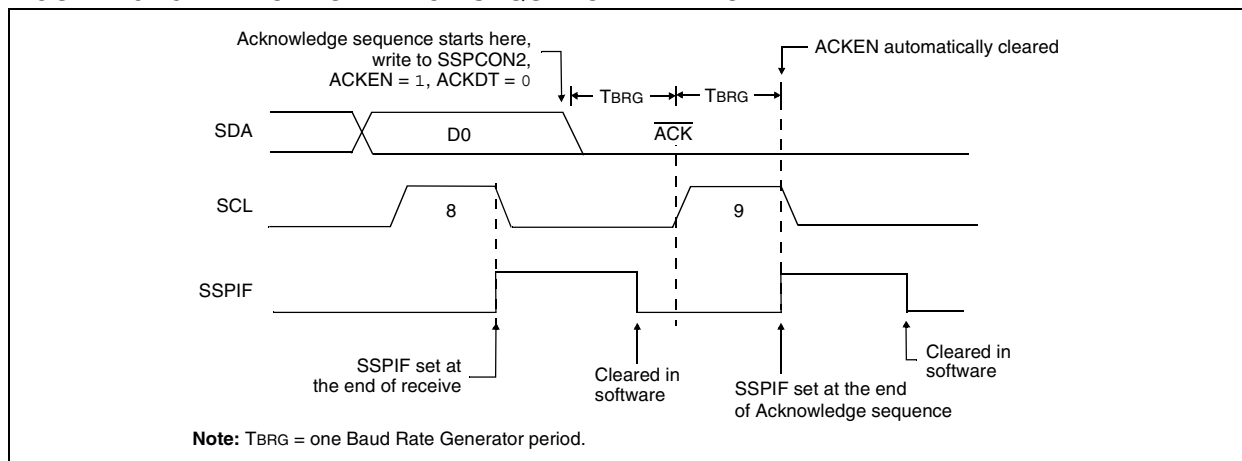
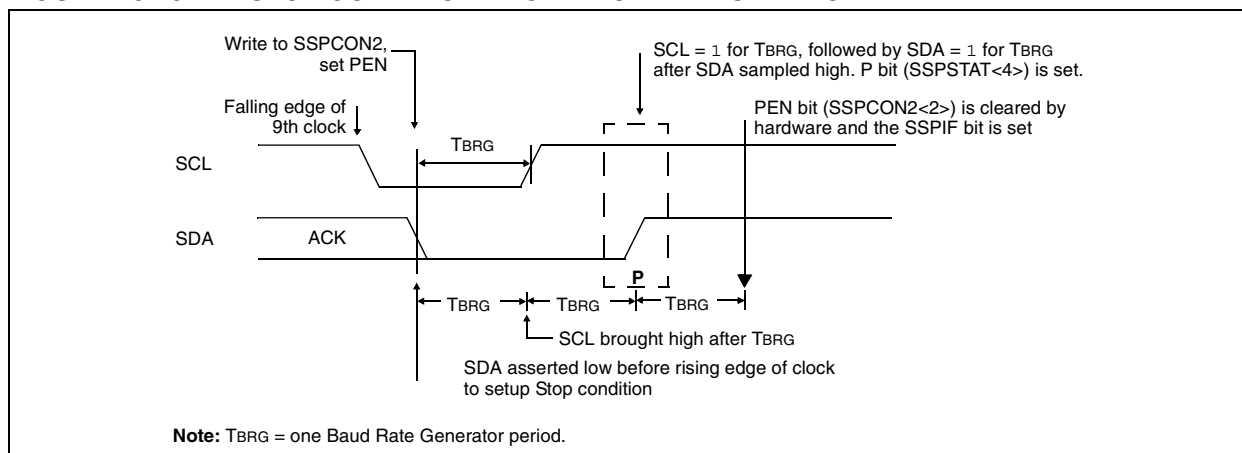


**FIGURE 16-26: STOP CONDITION RECEIVE OR TRANSMIT MODE**



## 16.4.14 SLEEP OPERATION

While in Sleep mode, the I<sup>2</sup>C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

## 16.4.15 EFFECTS OF A RESET

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

## 16.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I<sup>2</sup>C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the MSSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

- Address Transfer
- Data Transfer
- A Start Condition
- A Repeated Start Condition
- An Acknowledge Condition

## 16.4.17 MULTI-MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high, and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag, BCLIF and reset the I<sup>2</sup>C port to its Idle state (Figure 16-27).

If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

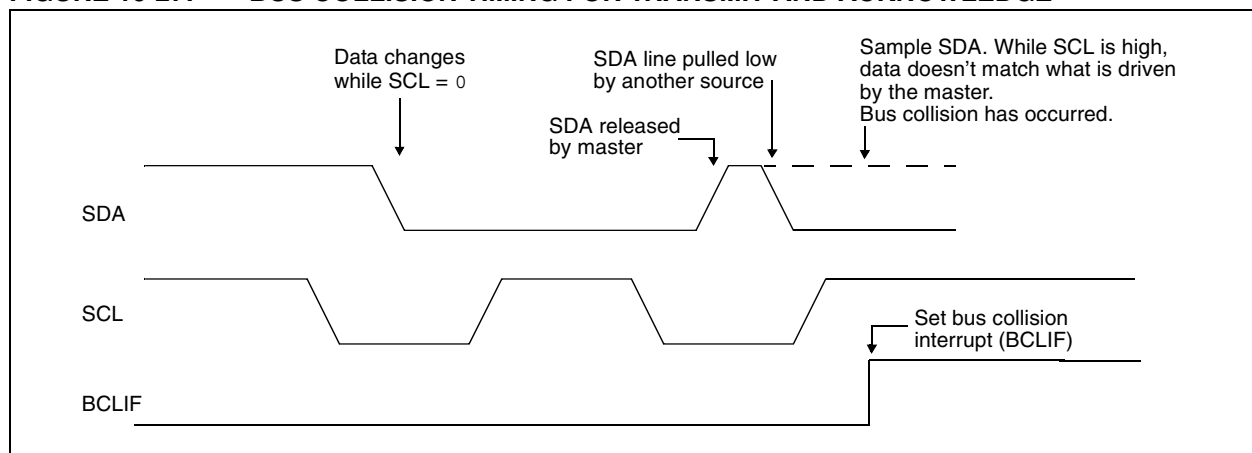
If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I<sup>2</sup>C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set.

A write to the SSPBUF will start the transmission of data at the first data bit regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I<sup>2</sup>C bus can be taken when the P bit is set in the SSPSTAT register, or the bus is Idle and the S and P bits are cleared.

**FIGURE 16-27: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE**



# PIC18F85J11 FAMILY

## 16.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- SDA or SCL are sampled low at the beginning of the Start condition (Figure 16-28).
- SCL is sampled low before SDA is asserted low (Figure 16-29).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- the Start condition is aborted;
- the BCLIF flag is set; and
- the MSSP module is reset to its Idle state (Figure 16-28).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0> and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs, because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 16-30). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0. If the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

**Note:** The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

**FIGURE 16-28: BUS COLLISION DURING START CONDITION (SDA ONLY)**

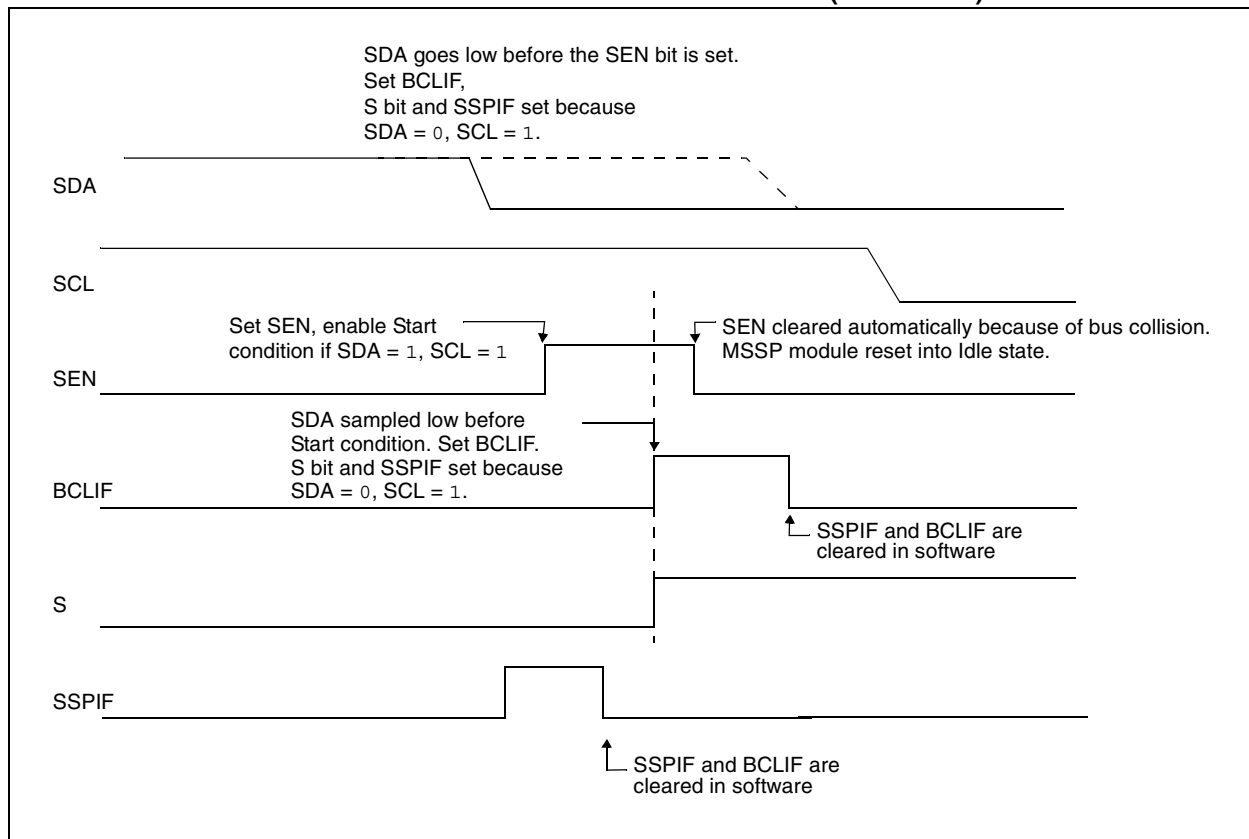


**FIGURE 16-29: BUS COLLISION DURING START CONDITION (SCL = 0)**

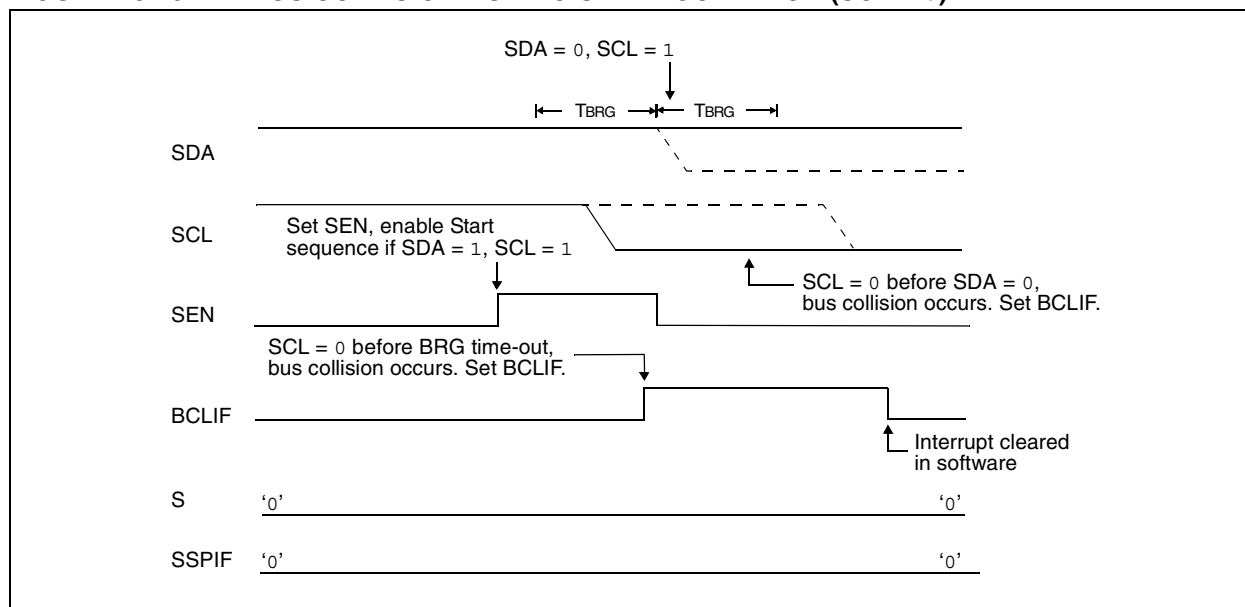
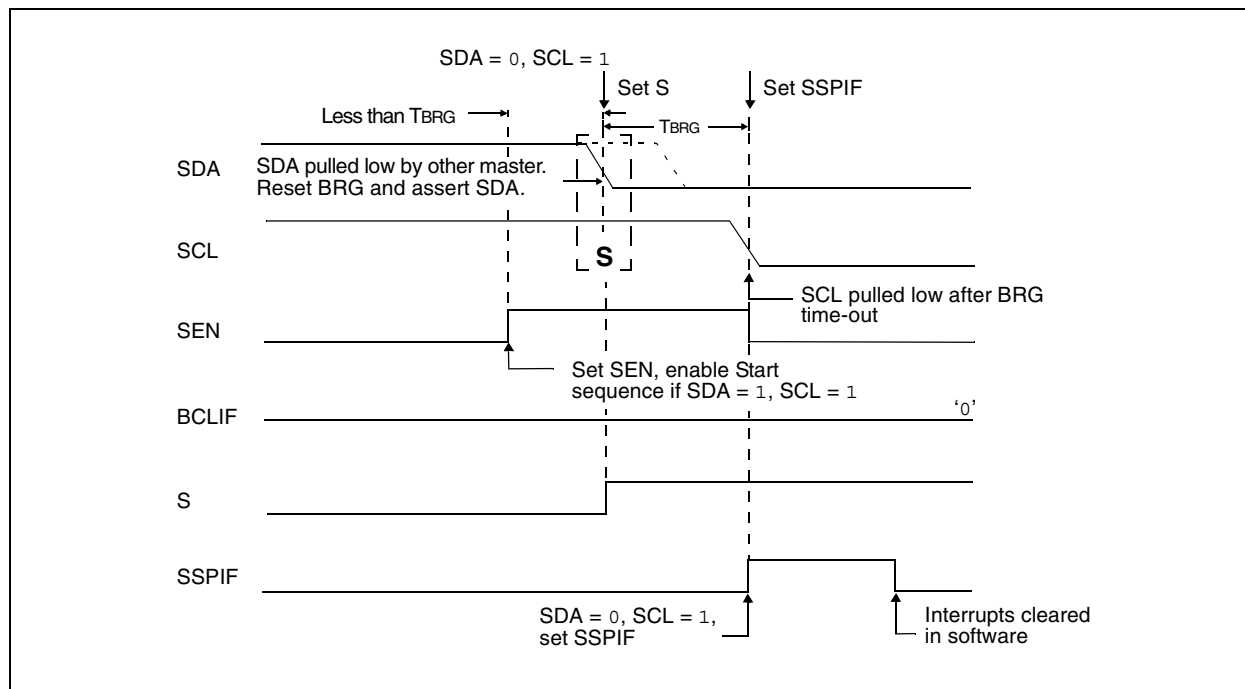


**FIGURE 16-30: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION**



# PIC18F85J11 FAMILY

## 16.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

- A low level is sampled on SDA when SCL goes from low level to high level.
- SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', see Figure 16-31). If SDA is sampled high, the BRG is reloaded and begins counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.

If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (see Figure 16-32).

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

**FIGURE 16-31: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)**

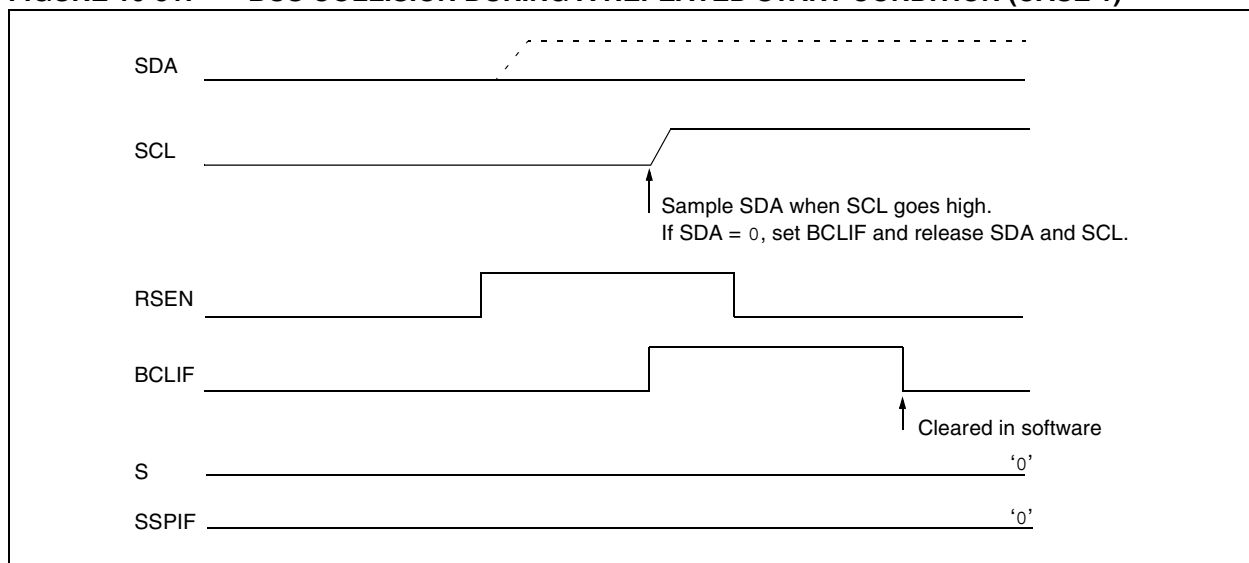
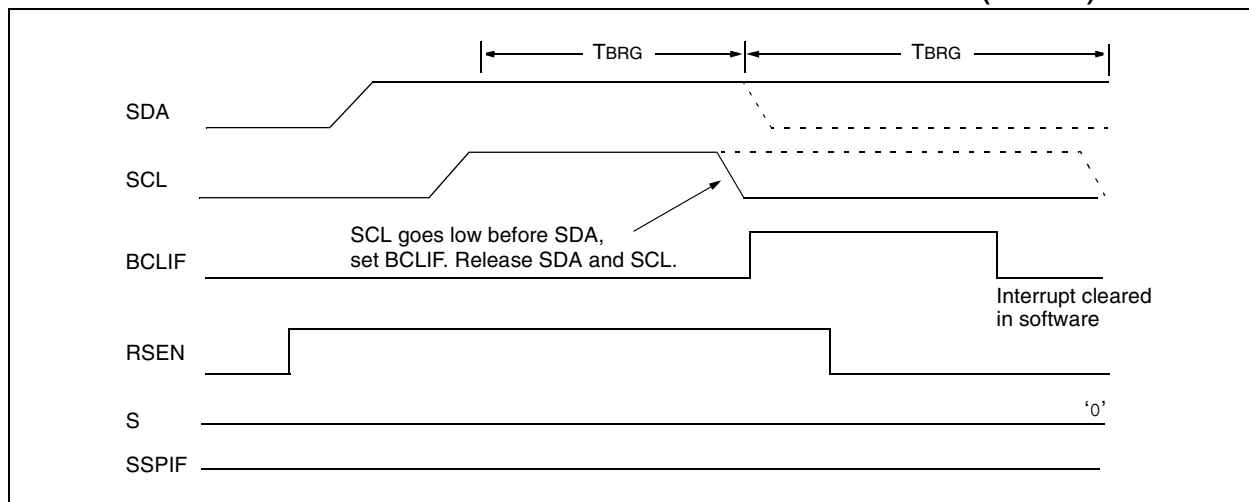


**FIGURE 16-32: BUS COLLISION DURING REPEATED START CONDITION (CASE 2)**





## 16.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

- After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.
- After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD<6:0> and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 16-33). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 16-34).

**FIGURE 16-33: BUS COLLISION DURING A STOP CONDITION (CASE 1)**

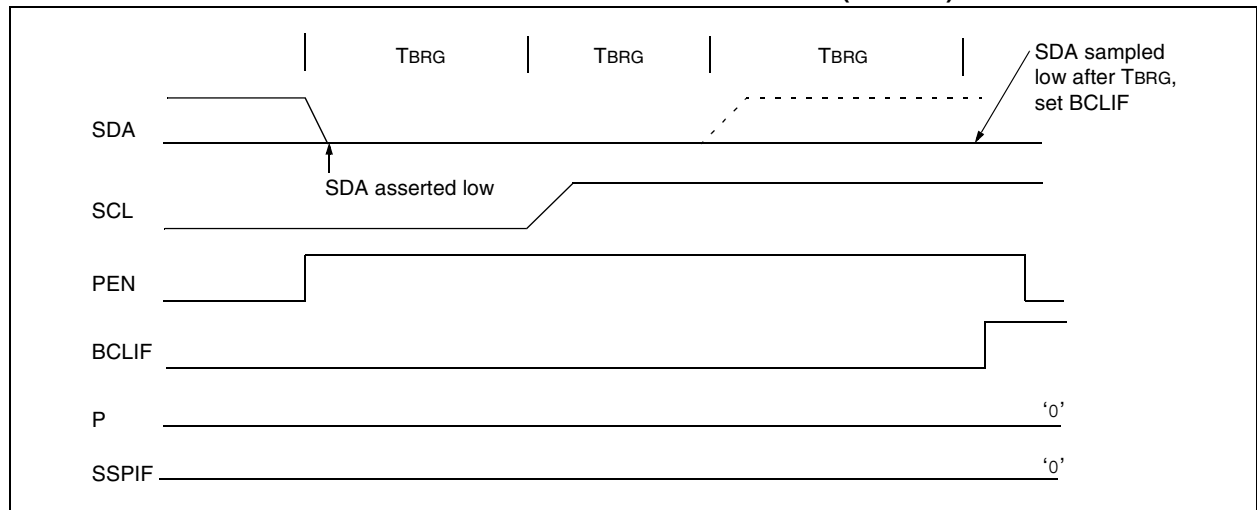
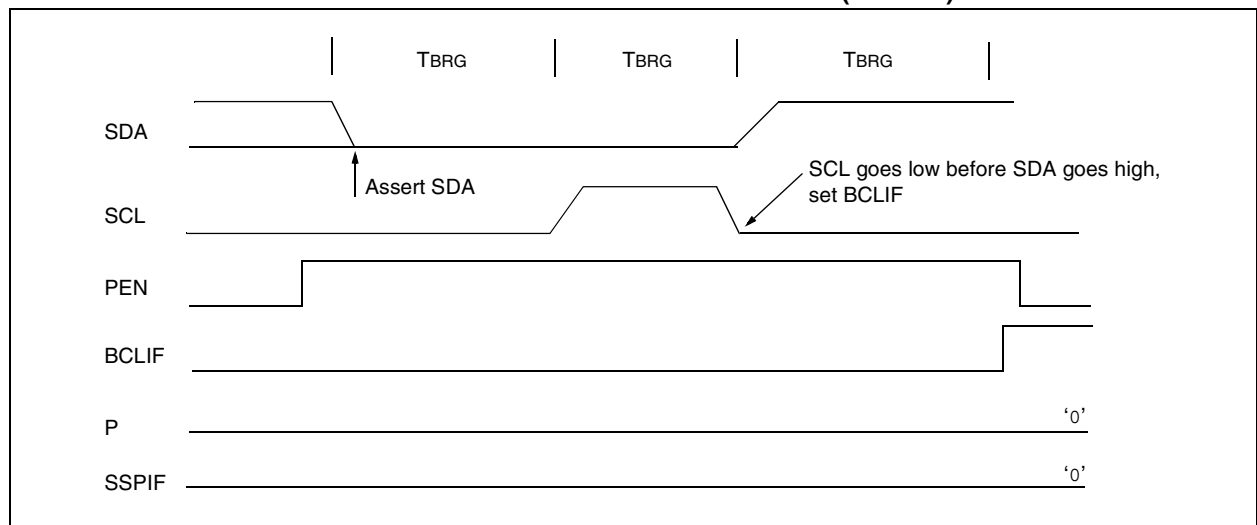


**FIGURE 16-34: BUS COLLISION DURING A STOP CONDITION (CASE 2)**



# PIC18F85J11 FAMILY

**TABLE 16-4: REGISTERS ASSOCIATED WITH I<sup>2</sup>C™ OPERATION**

| 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   | 51                   |
| PIR1    | PSPIF                                                                                                               | ADIF      | RC1IF                 | TX1IF                 | SSPIF                 | —                     | TMR2IF                | TMR1IF | 53                   |
| PIE1    | PSPIE                                                                                                               | ADIE      | RC1IE                 | TX1IE                 | SSPIE                 | —                     | TMR2IE                | TMR1IE | 53                   |
| IPR1    | PSPIP                                                                                                               | ADIP      | RC1IP                 | TX1IP                 | SSPIP                 | —                     | TMR2IP                | TMR1IP | 53                   |
| PIR2    | OSCFIF                                                                                                              | CMIF      | —                     | —                     | BCLIF                 | LVDIF                 | TMR3IF                | —      | 53                   |
| PIE2    | OSCFIE                                                                                                              | CMIE      | —                     | —                     | BCLIE                 | LVDIE                 | TMR3IE                | —      | 53                   |
| IPR2    | OSCFIP                                                                                                              | CMIP      | —                     | —                     | BCLIP                 | LVDIP                 | TMR3IP                | —      | 53                   |
| TRISC   | TRISC7                                                                                                              | TRISC6    | TRISC5                | TRISC4                | TRISC3                | TRISC2                | TRISC1                | TRISC0 | 54                   |
| SSPBUF  | MSSP Receive Buffer/Transmit Register                                                                               |           |                       |                       |                       |                       |                       |        | 52                   |
| SSPADD  | MSSP Address Register (I <sup>2</sup> C™ Slave mode), MSSP Baud Rate Reload Register (I <sup>2</sup> C Master mode) |           |                       |                       |                       |                       |                       |        | 52                   |
| SSPCON1 | WCOL                                                                                                                | SSPOV     | SSPEN                 | CKP                   | SSPM3                 | SSPM2                 | SSPM1                 | SSPM0  | 52                   |
| SSPCON2 | GCEN                                                                                                                | ACKSTAT   | ACKDT                 | ACKEN                 | RCEN                  | PEN                   | RSEN                  | SEN    | 52                   |
|         | GCEN                                                                                                                | ACKSTAT   | ADMSK5 <sup>(1)</sup> | ADMSK4 <sup>(1)</sup> | ADMSK3 <sup>(1)</sup> | ADMSK2 <sup>(1)</sup> | ADMSK1 <sup>(1)</sup> | SEN    |                      |
| SSPSTAT | SMP                                                                                                                 | CKE       | D/ $\bar{A}$          | P                     | S                     | R/ $\bar{W}$          | UA                    | BF     | 52                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used by the MSSP module in I<sup>2</sup>C™ mode.

**Note 1:** Alternate bit definitions for use in I<sup>2</sup>C Slave mode operations only.

## 17.0 ENHANCED UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (EUSART)

PIC18F85J11 family devices have three serial I/O modules: the MSSP module, discussed in the previous chapter and two Universal Synchronous Asynchronous Receiver Transmitter (USART) modules. (Generically, the USART is also known as a Serial Communications Interface or SCI.) The EUSART can be configured as a full-duplex, asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers. It can also be configured as a half-duplex, synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, serial EEPROMs, etc.

There are two distinct implementations of the USART module in these devices: the Enhanced USART (EUSART) discussed here and the Addressable USART discussed in the next chapter. For this device family, USART1 always refers to the EUSART, while USART2 is always the AUSART.

The EUSART and AUSART modules implement the same core features for serial communications; their basic operation is essentially the same. The EUSART module provides additional features, including Automatic Baud Rate Detection and calibration, automatic wake-up on Sync Break reception and 12-bit Break character transmit. These features make it ideally suited for use in Local Interconnect Network bus (LIN bus) systems.

The EUSART can be configured in the following modes:

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

The pins of the EUSART are multiplexed with the functions of PORTC (RC6/TX1/CK1 and RC7/RX1/DT1). In order to configure these pins as a EUSART:

- bit SPEN (RCSTA1<7>) must be set (= 1)
- bit TRISC<7> must be set (= 1)
- bit TRISC<6> must be set (= 1)

|                                                                                                        |
|--------------------------------------------------------------------------------------------------------|
| <b>Note:</b> The EUSART control will automatically reconfigure the pin from input to output as needed. |
|--------------------------------------------------------------------------------------------------------|

The driver for the TX1 output pin can also be optionally configured as an open-drain output. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters.

The open-drain output option is controlled by the U1OD bit (LATG<6>). Setting the bit configures the pin for open-drain operation.

### 17.1 Control Registers

The operation of the Enhanced USART module is controlled through three registers:

- Transmit Status and Control Register 1 (TXSTA1)
- Receive Status and Control Register 1 (RCSTA1)
- Baud Rate Control Register 1 (BAUDCON1)

The registers are described in Register 17-1, Register 17-2 and Register 17-3.

# PIC18F85J11 FAMILY

## REGISTER 17-1: TXSTA1: EUSART TRANSMIT STATUS AND CONTROL REGISTER

| R/W-0 | R/W-0 | R/W-0               | R/W-0 | R/W-0 | R/W-0 | R-1  | R/W-0 |
|-------|-------|---------------------|-------|-------|-------|------|-------|
| CSRC  | TX9   | TXEN <sup>(1)</sup> | SYNC  | SENDB | BRGH  | TRMT | TX9D  |
| 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      **CSRC:** Clock Source Select bit  
Asynchronous mode:  
Don't care.  
Synchronous mode:  
1 = Master mode (clock generated internally from BRG)  
0 = Slave mode (clock from external source)
- bit 6      **TX9:** 9-Bit Transmit Enable bit  
1 = Selects 9-bit transmission  
0 = Selects 8-bit transmission
- bit 5      **TXEN:** Transmit Enable bit<sup>(1)</sup>  
1 = Transmit enabled  
0 = Transmit disabled
- bit 4      **SYNC:** AUSART Mode Select bit  
1 = Synchronous mode  
0 = Asynchronous mode
- bit 3      **SENDB:** Send Break Character bit  
Asynchronous mode:  
1 = Send Sync Break on next transmission (cleared by hardware upon completion)  
0 = Sync Break transmission completed  
Synchronous mode:  
Don't care.
- bit 2      **BRGH:** High Baud Rate Select bit  
Asynchronous mode:  
1 = High speed  
0 = Low speed  
Synchronous mode:  
Unused in this mode.
- bit 1      **TRMT:** Transmit Shift Register Status bit  
1 = TSR empty  
0 = TSR full
- bit 0      **TX9D:** 9th Bit of Transmit Data  
Can be address/data bit or a parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.

# PIC18F85J11 FAMILY

**REGISTER 17-2: RCSTA1: EUSART RECEIVE STATUS AND CONTROL REGISTER**

|       |       |       |       |       |      |      |       |
|-------|-------|-------|-------|-------|------|------|-------|
| 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 (configures RX1/DT1 and TX1/CK1 pins as serial port pins)  
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 9-bit (RX9 = 0):  
Don't care.
- bit 2      **FERR:** Framing Error bit  
1 = Framing error (can be updated by reading RCREG1 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)  
0 = No overrun error
- bit 0      **RX9D:** 9th Bit of Received Data  
This can be address/data bit or a parity bit and must be calculated by user firmware.

# PIC18F85J11 FAMILY

## REGISTER 17-3: BAUDCON1: BAUD RATE CONTROL REGISTER 1

| R/W-0  | R-1  | U-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|--------|------|-----|-------|-------|-----|-------|-------|
| ABDOVF | RCMT | —   | SCKP  | BRG16 | —   | WUE   | ABDEN |
| 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      **ABDOVF:** Auto-Baud Acquisition Rollover Status bit  
1 = A BRG rollover has occurred during Auto-Baud Rate Detect mode (must be cleared in software)  
0 = No BRG rollover has occurred
- bit 6      **RCMT:** Receive Operation Idle Status bit  
1 = Receive operation is Idle  
0 = Receive operation is active
- bit 5      **Unimplemented:** Read as '0'
- bit 4      **SCKP:** Synchronous Clock Polarity Select bit  
Asynchronous mode:  
Unused in this mode.  
Synchronous mode:  
1 = Idle state for clock (CK1) is a high level  
0 = Idle state for clock (CK1) is a low level
- bit 3      **BRG16:** 16-Bit Baud Rate Register Enable bit  
1 = 16-bit Baud Rate Generator – SPBRGH1 and SPBRG1  
0 = 8-bit Baud Rate Generator – SPBRG1 only (Compatible mode), SPBRGH1 value ignored
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **WUE:** Wake-up Enable bit  
Asynchronous mode:  
1 = EUSART will continue to sample the RX1 pin – interrupt generated on falling edge; bit cleared in hardware on following rising edge  
0 = RX1 pin not monitored or rising edge detected  
Synchronous mode:  
Unused in this mode.
- bit 0      **ABDEN:** Auto-Baud Detect Enable bit  
Asynchronous mode:  
1 = Enable baud rate measurement on the next character. Requires reception of a Sync field (55h); cleared in hardware upon completion.  
0 = Baud rate measurement disabled or completed  
Synchronous mode:  
Unused in this mode.

## 17.2 EUSART Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit or 16-bit generator that supports both the Asynchronous and Synchronous modes of the EUSART. By default, the BRG operates in 8-bit mode; setting the BRG16 bit (BAUDCON1<3>) selects 16-bit mode.

The SPBRGH1:SPBRG1 register pair controls the period of a free-running timer. In Asynchronous mode, BRGH (TXSTA1<2>) and BRG16 (BAUDCON1<3>) bits also control the baud rate. In Synchronous mode, BRGH is ignored. Table 17-1 shows the formula for computation of the baud rate for different EUSART modes that only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRGH1:SPBRG1 registers can be calculated using the formulas in Table 17-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 17-1. Typical baud rates and error values for the various

Asynchronous modes are shown in Table 17-2. It may be advantageous to use the high baud rate (BRGH = 1) or the 16-bit BRG to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRGH1:SPBRG1 registers causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 17.2.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRG1 register pair.

### 17.2.2 SAMPLING

The data on the RX1 pin is sampled three times by a majority detect circuit to determine if a high or low level is present at the RX1 pin.

**TABLE 17-1: BAUD RATE FORMULAS**

| Configuration Bits |       |      | BRG/EUSART Mode     | Baud Rate Formula      |
|--------------------|-------|------|---------------------|------------------------|
| SYNC               | BRG16 | BRGH |                     |                        |
| 0                  | 0     | 0    | 8-Bit/Asynchronous  | $F_{osc}/[64 (n + 1)]$ |
| 0                  | 0     | 1    | 8-Bit/Asynchronous  | $F_{osc}/[16 (n + 1)]$ |
| 0                  | 1     | 0    | 16-Bit/Asynchronous |                        |
| 0                  | 1     | 1    | 16-Bit/Asynchronous | $F_{osc}/[4 (n + 1)]$  |
| 1                  | 0     | x    | 8-Bit/Synchronous   |                        |
| 1                  | 1     | x    | 16-Bit/Synchronous  |                        |

**Legend:** x = Don't care, n = Value of SPBRGH1:SPBRG1 register pair

### EXAMPLE 17-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, 8-bit BRG:

Desired Baud Rate =  $F_{osc}/(64 ([SPBRGH1:SPBRG1] + 1))$

Solving for SPBRGH1:SPBRG1:

$$\begin{aligned} X &= ((F_{osc}/\text{Desired Baud Rate})/64) - 1 \\ &= ((16000000/9600)/64) - 1 \\ &= [25.042] = 25 \end{aligned}$$

Calculated Baud Rate =  $16000000/(64 (25 + 1))$   
= 9615

Error =  $(\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate}$   
=  $(9615 - 9600)/9600 = 0.16\%$

**TABLE 17-2: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR**

| Name     | Bit 7                                         | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page |
|----------|-----------------------------------------------|-------|-------|-------|-------|-------|-------|-------|----------------------|
| TXSTA1   | CSRC                                          | TX9   | TXEN  | SYNC  | SENDB | BRGH  | TRMT  | TX9D  | 53                   |
| RCSTA1   | SPEN                                          | RX9   | SREN  | CREN  | ADDEN | FERR  | OERR  | RX9D  | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT  | —     | SCKP  | BRG16 | —     | WUE   | ABDEN | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |       |       |       |       |       |       |       | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |       |       |       |       |       |       |       | 53                   |

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

# PIC18F85J11 FAMILY

**TABLE 17-3: BAUD RATES FOR ASYNCHRONOUS MODES**

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz             |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                             | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 1.2           | —                             | —       | —                     | 1.221             | 1.73    | 255                   | 1.202             | 0.16    | 129                   | 1.201            | -0.16   | 103                   |
| 2.4           | 2.441                         | 1.73    | 255                   | 2.404             | 0.16    | 129                   | 2.404             | 0.16    | 64                    | 2.403            | -0.16   | 51                    |
| 9.6           | 9.615                         | 0.16    | 64                    | 9.766             | 1.73    | 31                    | 9.766             | 1.73    | 15                    | 9.615            | -0.16   | 12                    |
| 19.2          | 19.531                        | 1.73    | 31                    | 19.531            | 1.73    | 15                    | 19.531            | 1.73    | 7                     | —                | —       | —                     |
| 57.6          | 56.818                        | -1.36   | 10                    | 62.500            | 8.51    | 4                     | 52.083            | -9.58   | 2                     | —                | —       | —                     |
| 115.2         | 125.000                       | 8.51    | 4                     | 104.167           | -9.58   | 2                     | 78.125            | -32.18  | 1                     | —                | —       | —                     |

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 0 |         |                       |                  |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz              |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300                         | 0.16    | 207                   | 0.300            | -0.16   | 103                   | 0.300            | -0.16   | 51                    |
| 1.2           | 1.202                         | 0.16    | 51                    | 1.201            | -0.16   | 25                    | 1.201            | -0.16   | 12                    |
| 2.4           | 2.404                         | 0.16    | 25                    | 2.403            | -0.16   | 12                    | —                | —       | —                     |
| 9.6           | 8.929                         | -6.99   | 6                     | —                | —       | —                     | —                | —       | —                     |
| 19.2          | 20.833                        | 8.51    | 2                     | —                | —       | —                     | —                | —       | —                     |
| 57.6          | 62.500                        | 8.51    | 0                     | —                | —       | —                     | —                | —       | —                     |
| 115.2         | 62.500                        | -45.75  | 0                     | —                | —       | —                     | —                | —       | —                     |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz             |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                             | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 1.2           | —                             | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 2.4           | —                             | —       | —                     | —                 | —       | —                     | 2.441             | 1.73    | 255                   | 2.403            | -0.16   | 207                   |
| 9.6           | 9.766                         | 1.73    | 255                   | 9.615             | 0.16    | 129                   | 9.615             | 0.16    | 64                    | 9.615            | -0.16   | 51                    |
| 19.2          | 19.231                        | 0.16    | 129                   | 19.231            | 0.16    | 64                    | 19.531            | 1.73    | 31                    | 19.230           | -0.16   | 25                    |
| 57.6          | 58.140                        | 0.94    | 42                    | 56.818            | -1.36   | 21                    | 56.818            | -1.36   | 10                    | 55.555           | 3.55    | 8                     |
| 115.2         | 113.636                       | -1.36   | 21                    | 113.636           | -1.36   | 10                    | 125.000           | 8.51    | 4                     | —                | —       | —                     |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 0 |         |                       |                  |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz              |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                             | —       | —                     | —                | —       | —                     | 0.300            | -0.16   | 207                   |
| 1.2           | 1.202                         | 0.16    | 207                   | 1.201            | -0.16   | 103                   | 1.201            | -0.16   | 51                    |
| 2.4           | 2.404                         | 0.16    | 103                   | 2.403            | -0.16   | 51                    | 2.403            | -0.16   | 25                    |
| 9.6           | 9.615                         | 0.16    | 25                    | 9.615            | -0.16   | 12                    | —                | —       | —                     |
| 19.2          | 19.231                        | 0.16    | 12                    | —                | —       | —                     | —                | —       | —                     |
| 57.6          | 62.500                        | 8.51    | 3                     | —                | —       | —                     | —                | —       | —                     |
| 115.2         | 125.000                       | 8.51    | 1                     | —                | —       | —                     | —                | —       | —                     |



# PIC18F85J11 FAMILY

**TABLE 17-3: BAUD RATES FOR ASYNCHRONOUS MODES (CONTINUED)**

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 1 |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz             |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300                         | 0.00    | 8332                  | 0.300             | 0.02    | 4165                  | 0.300             | 0.02    | 2082                  | 0.300            | -0.04   | 1665                  |
| 1.2           | 1.200                         | 0.02    | 2082                  | 1.200             | -0.03   | 1041                  | 1.200             | -0.03   | 520                   | 1.201            | -0.16   | 415                   |
| 2.4           | 2.402                         | 0.06    | 1040                  | 2.399             | -0.03   | 520                   | 2.404             | 0.16    | 259                   | 2.403            | -0.16   | 207                   |
| 9.6           | 9.615                         | 0.16    | 259                   | 9.615             | 0.16    | 129                   | 9.615             | 0.16    | 64                    | 9.615            | -0.16   | 51                    |
| 19.2          | 19.231                        | 0.16    | 129                   | 19.231            | 0.16    | 64                    | 19.531            | 1.73    | 31                    | 19.230           | -0.16   | 25                    |
| 57.6          | 58.140                        | 0.94    | 42                    | 56.818            | -1.36   | 21                    | 56.818            | -1.36   | 10                    | 55.555           | 3.55    | 8                     |
| 115.2         | 113.636                       | -1.36   | 21                    | 113.636           | -1.36   | 10                    | 125.000           | 8.51    | 4                     | —                | —       | —                     |

| BAUD RATE (K) | SYNC = 0, BRGH = 0, BRG16 = 1 |         |                       |                  |         |                       |                  |         |                       |
|---------------|-------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz              |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)               | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300                         | 0.04    | 832                   | 0.300            | -0.16   | 415                   | 0.300            | -0.16   | 207                   |
| 1.2           | 1.202                         | 0.16    | 207                   | 1.201            | -0.16   | 103                   | 1.201            | -0.16   | 51                    |
| 2.4           | 2.404                         | 0.16    | 103                   | 2.403            | -0.16   | 51                    | 2.403            | -0.16   | 25                    |
| 9.6           | 9.615                         | 0.16    | 25                    | 9.615            | -0.16   | 12                    | —                | —       | —                     |
| 19.2          | 19.231                        | 0.16    | 12                    | —                | —       | —                     | —                | —       | —                     |
| 57.6          | 62.500                        | 8.51    | 3                     | —                | —       | —                     | —                | —       | —                     |
| 115.2         | 125.000                       | 8.51    | 1                     | —                | —       | —                     | —                | —       | —                     |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|------------------------------------------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz                                    |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)                                      | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300                                                | 0.00    | 33332                 | 0.300             | 0.00    | 16665                 | 0.300             | 0.00    | 8332                  | 0.300            | -0.01   | 6665                  |
| 1.2           | 1.200                                                | 0.00    | 8332                  | 1.200             | 0.02    | 4165                  | 1.200             | 0.02    | 2082                  | 1.200            | -0.04   | 1665                  |
| 2.4           | 2.400                                                | 0.02    | 4165                  | 2.400             | 0.02    | 2082                  | 2.402             | 0.06    | 1040                  | 2.400            | -0.04   | 832                   |
| 9.6           | 9.606                                                | 0.06    | 1040                  | 9.596             | -0.03   | 520                   | 9.615             | 0.16    | 259                   | 9.615            | -0.16   | 207                   |
| 19.2          | 19.193                                               | -0.03   | 520                   | 19.231            | 0.16    | 259                   | 19.231            | 0.16    | 129                   | 19.230           | -0.16   | 103                   |
| 57.6          | 57.803                                               | 0.35    | 172                   | 57.471            | -0.22   | 86                    | 58.140            | 0.94    | 42                    | 57.142           | 0.79    | 34                    |
| 115.2         | 114.943                                              | -0.22   | 86                    | 116.279           | 0.94    | 42                    | 113.636           | -1.36   | 21                    | 117.647          | -2.12   | 16                    |

| BAUD RATE (K) | SYNC = 0, BRGH = 1, BRG16 = 1 or SYNC = 1, BRG16 = 1 |         |                       |                  |         |                       |                  |         |                       |
|---------------|------------------------------------------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz                                     |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)                                      | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300                                                | 0.01    | 3332                  | 0.300            | -0.04   | 1665                  | 0.300            | -0.04   | 832                   |
| 1.2           | 1.200                                                | 0.04    | 832                   | 1.201            | -0.16   | 415                   | 1.201            | -0.16   | 207                   |
| 2.4           | 2.404                                                | 0.16    | 415                   | 2.403            | -0.16   | 207                   | 2.403            | -0.16   | 103                   |
| 9.6           | 9.615                                                | 0.16    | 103                   | 9.615            | -0.16   | 51                    | 9.615            | -0.16   | 25                    |
| 19.2          | 19.231                                               | 0.16    | 51                    | 19.230           | -0.16   | 25                    | 19.230           | -0.16   | 12                    |
| 57.6          | 58.824                                               | 2.12    | 16                    | 55.555           | 3.55    | 8                     | —                | —       | —                     |
| 115.2         | 111.111                                              | -3.55   | 8                     | —                | —       | —                     | —                | —       | —                     |

# PIC18F85J11 FAMILY

## 17.2.3 AUTO-BAUD RATE DETECT

The Enhanced USART module supports the automatic detection and calibration of baud rate. This feature is active only in Asynchronous mode and while the WUE bit is clear.

The automatic baud rate measurement sequence (Figure 17-1) begins whenever a Start bit is received and the ABDEN bit is set. The calculation is self-averaging.

In the Auto-Baud Rate Detect (ABD) mode, the clock to the BRG is reversed. Rather than the BRG clocking the incoming RX1 signal, the RX1 signal is timing the BRG. In ABD mode, the internal Baud Rate Generator is used as a counter to time the bit period of the incoming serial byte stream.

Once the ABDEN bit is set, the state machine will clear the BRG and look for a Start bit. The Auto-Baud Rate Detect must receive a byte with the value, 55h (ASCII “U”, which is also the LIN bus Sync character), in order to calculate the proper bit rate. The measurement is taken over both a low and high bit time in order to minimize any effects caused by asymmetry of the incoming signal. After a Start bit, the SPBRG1 begins counting up, using the preselected clock source on the first rising edge of RX1. After eight bits on the RX1 pin, or the fifth rising edge, an accumulated value totalling the proper BRG period is left in the SPBRGH1:SPBRG1 register pair. Once the 5th edge is seen (this should correspond to the Stop bit), the ABDEN bit is automatically cleared.

If a rollover of the BRG occurs (an overflow from FFFFh to 0000h), the event is trapped by the ABDOVF status bit (BAUDCON1<7>). It is set in hardware by BRG rollovers and can be set or cleared by the user in software. ABD mode remains active after rollover events and the ABDEN bit remains set (Figure 17-2).

While calibrating the baud rate period, the BRG registers are clocked at 1/8th the preconfigured clock rate. Note that the BRG clock will be configured by the BRG16 and BRGH bits. Independent of the BRG16 bit setting, both the SPBRG1 and SPBRGH1 will be used as a 16-bit counter. This allows the user to verify that no carry occurred for 8-bit modes by checking for 00h in the SPBRGH1 register. Refer to Table 17-4 for counter clock rates to the BRG.

While the ABD sequence takes place, the EUSART state machine is held in Idle. The RC1IF interrupt is set once the fifth rising edge on RX1 is detected. The value in the RCREG1 needs to be read to clear the RC1IF interrupt. The contents of RCREG1 should be discarded.

**Note 1:** If the WUE bit is set with the ABDEN bit, Auto-Baud Rate Detection will occur on the byte *following* the Break character.

**2:** It is up to the user to determine that the incoming character baud rate is within the range of the selected BRG clock source. Some combinations of oscillator frequency and EUSART baud rates are not possible due to bit error rates. Overall system timing and communication baud rates must be taken into consideration when using the Auto-Baud Rate Detection feature.

**TABLE 17-4: BRG COUNTER CLOCK RATES**

| BRG16 | BRGH | BRG Counter Clock |
|-------|------|-------------------|
| 0     | 0    | Fosc/512          |
| 0     | 1    | Fosc/128          |
| 1     | 0    | Fosc/128          |
| 1     | 1    | Fosc/32           |

**Note:** During the ABD sequence, SPBRG1 and SPBRGH1 are both used as a 16-bit counter, independent of the BRG16 setting.

### 17.2.3.1 ABD and EUSART Transmission

Since the BRG clock is reversed during ABD acquisition, the EUSART transmitter cannot be used during ABD. This means that whenever the ABDEN bit is set, TXREG1 cannot be written to. Users should also ensure that ABDEN does not become set during a transmit sequence. Failing to do this may result in unpredictable EUSART operation.

FIGURE 17-1: AUTOMATIC BAUD RATE CALCULATION

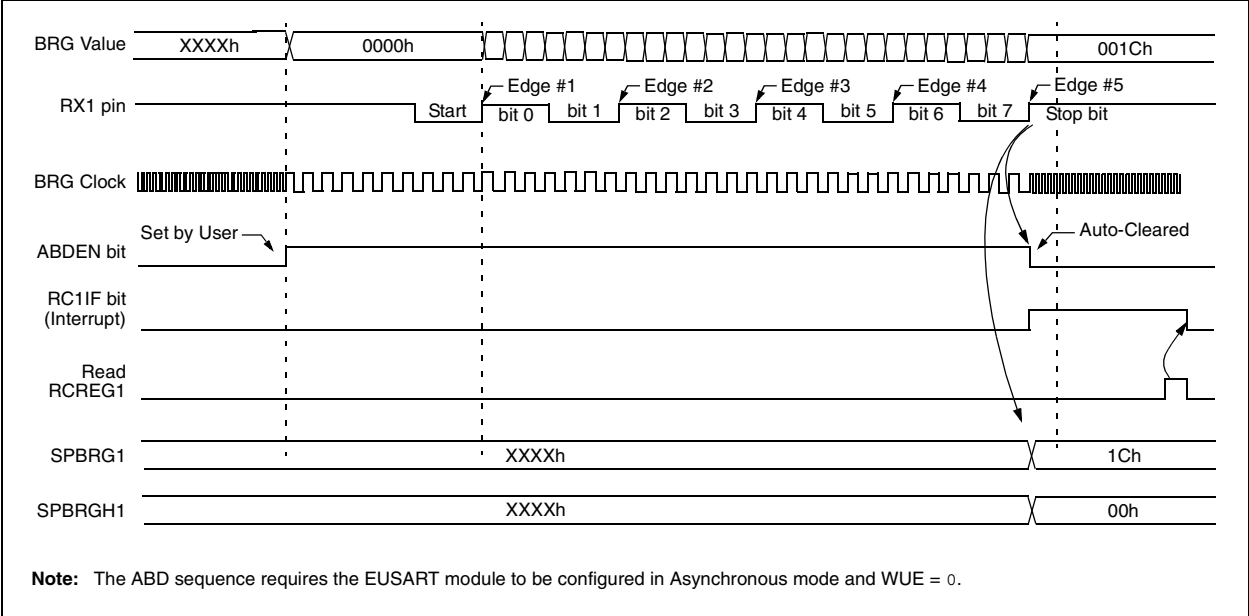
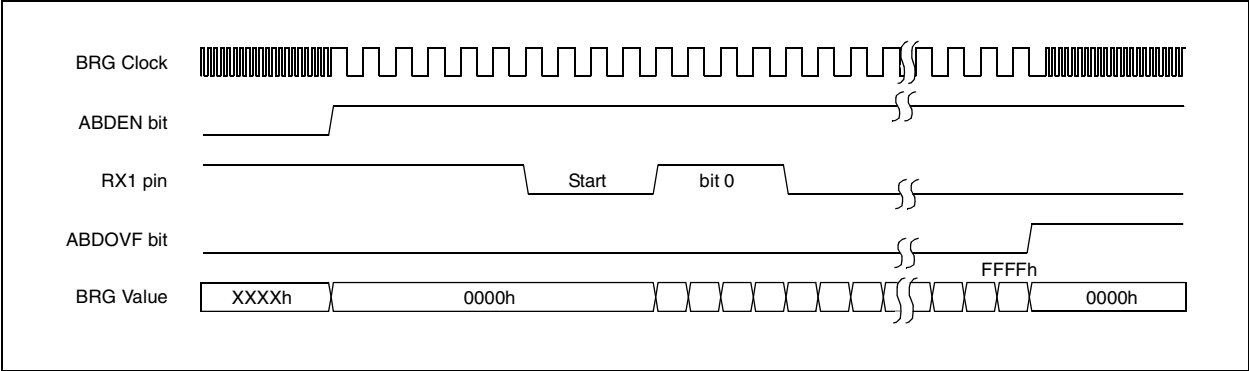


FIGURE 17-2: BRG OVERFLOW SEQUENCE



# PIC18F85J11 FAMILY

## 17.3 EUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA1<4>). In this mode, the EUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated 8-bit/16-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The EUSART transmits and receives the LSb first. The EUSART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH and BRG16 bits (TXSTA1<2> and BAUDCON1<3>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

When operating in Asynchronous mode, the EUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver
- Auto-Wake-up on Sync Break Character
- 12-Bit Break Character Transmit
- Auto-Baud Rate Detection

### 17.3.1 EUSART ASYNCHRONOUS TRANSMITTER

The EUSART transmitter block diagram is shown in Figure 17-3. The heart of the transmitter is the Transmit (Serial) Shift register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG1. The TXREG1 register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG1 register (if available).

Once the TXREG1 register transfers the data to the TSR register (occurs in one Tcy), the TXREG1 register is empty and the TX1IF flag bit (PIR1<4>) is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TX1IE (PIE1<4>). TX1IF will be set regardless of the state of TX1IE; it cannot be cleared in software. TX1IF is also not cleared immediately upon loading TXREG1, but becomes valid in the second instruction cycle following the load instruction. Polling TX1IF immediately following a load of TXREG1 will return invalid results.

While TX1IF indicates the status of the TXREG1 register, another bit, TRMT (TXSTA1<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.

**2:** Flag bit, TX1IF, is set when enable bit, TXEN, is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRGH1:SPBRG1 registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, TX1IE.
4. If 9-bit transmission is desired, set transmit bit, TX9; can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TX1IF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Load data to the TXREG1 register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 17-3: EUSART TRANSMIT BLOCK DIAGRAM**

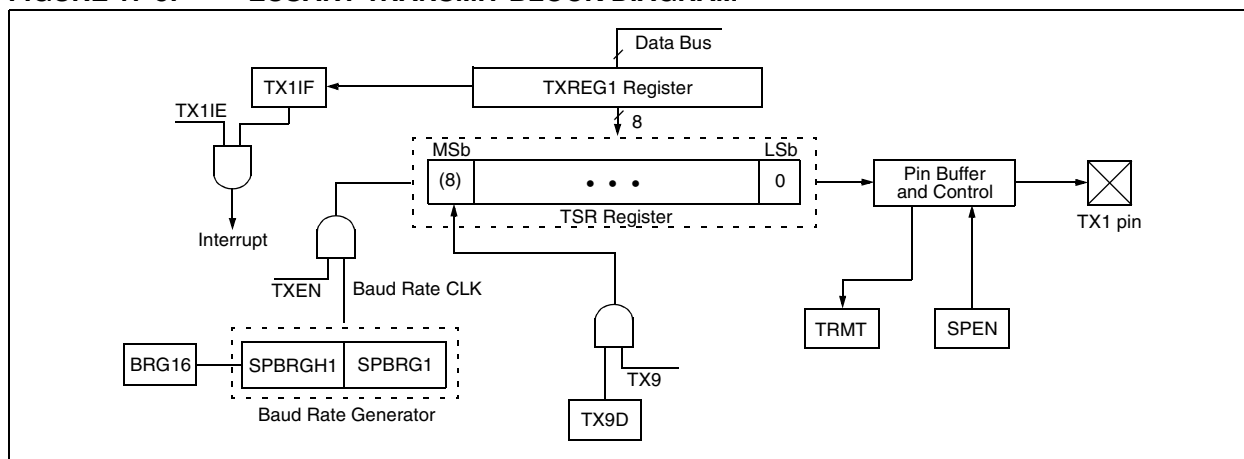


**FIGURE 17-4: ASYNCHRONOUS TRANSMISSION**

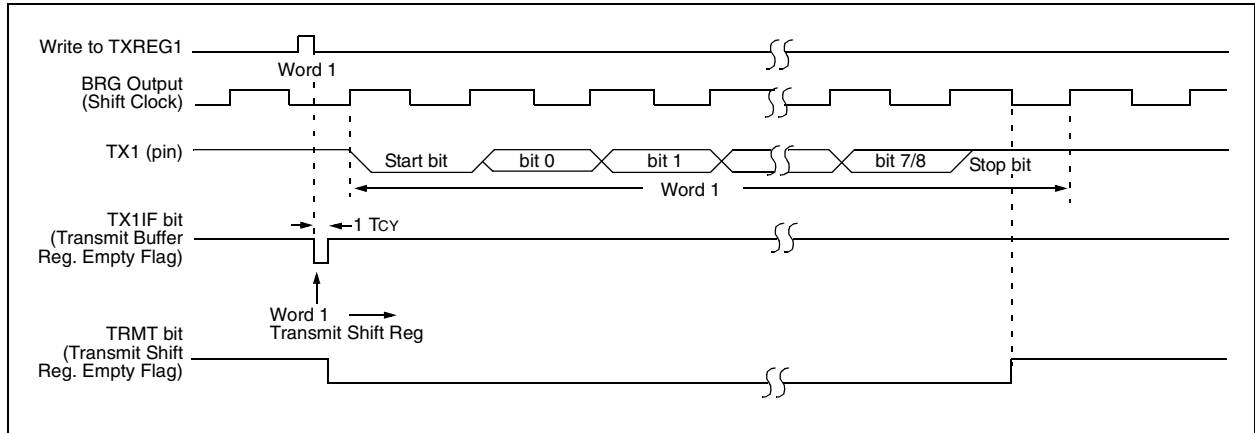


**FIGURE 17-5: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**

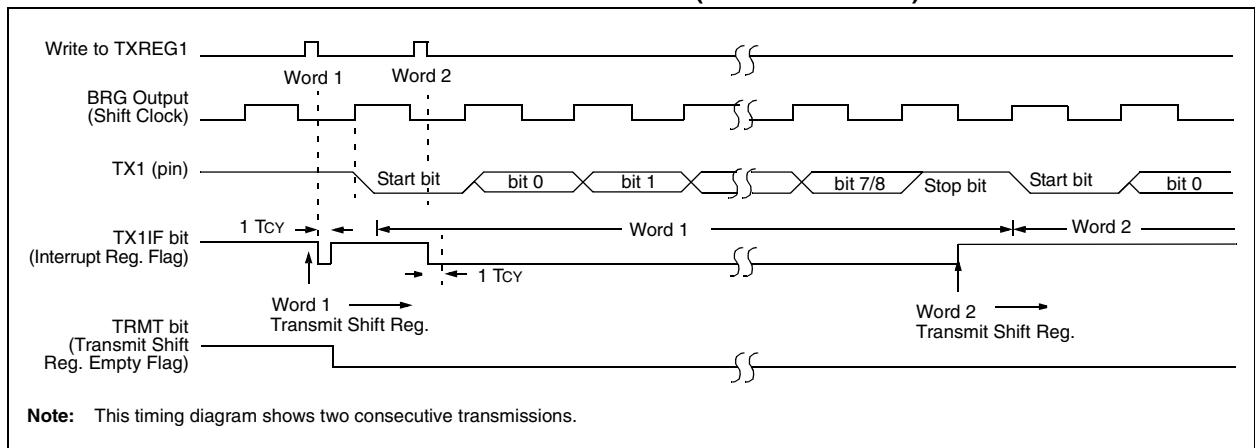


**TABLE 17-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D   | 53                   |
| TXREG1   | EUSART Transmit Register                      |           |        |        |       |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SENDB | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16 | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |       |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |       |        |        |        | 53                   |
| LATG     | U2OD                                          | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0  | 54                   |

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

# PIC18F85J11 FAMILY

## 17.3.2 EUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 17-6. The data is received on the RX1 pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

1. Initialize the SPBRGH1:SPBRG1 registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RC1IE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RC1IF, will be set when reception is complete and an interrupt will be generated if enable bit, RC1IE, was set.
7. Read the RCSTA1 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG1 register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

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

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

1. Initialize the SPBRGH1:SPBRG1 registers for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RC1IP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RC1IF bit will be set when reception is complete. The interrupt will be Acknowledged if the RC1IE and GIE bits are set.
8. Read the RCSTA1 register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG1 to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

**FIGURE 17-6: EUSART RECEIVE BLOCK DIAGRAM**

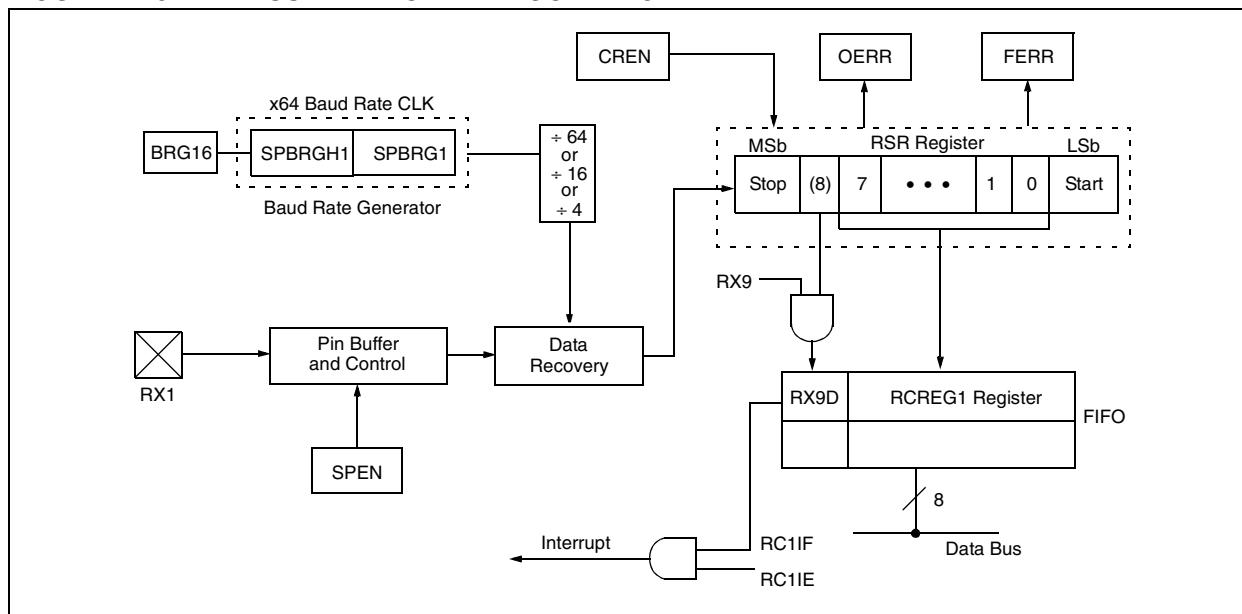


**FIGURE 17-7: ASYNCHRONOUS RECEPTION**

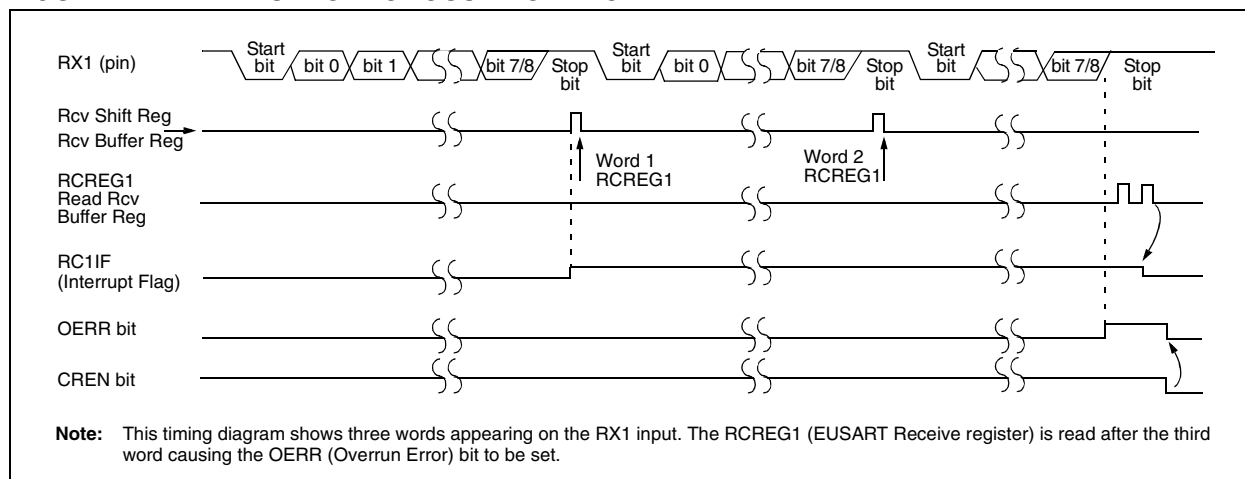


**TABLE 17-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D   | 53                   |
| RCREG1   | EUSART Receive Register                       |           |        |        |       |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SEnDB | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16 | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |       |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |       |        |        |        | 53                   |

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

# PIC18F85J11 FAMILY

## 17.3.4 AUTO-WAKE-UP ON SYNC BREAK CHARACTER

During Sleep mode, all clocks to the EUSART are suspended. Because of this, the Baud Rate Generator is inactive and a proper byte reception cannot be performed. The auto-wake-up feature allows the controller to wake-up due to activity on the RX1/DT1 line, while the EUSART is operating in Asynchronous mode.

The auto-wake-up feature is enabled by setting the WUE bit (BAUDCON<1>). Once set, the typical receive sequence on RX1/DT1 is disabled and the EUSART remains in an Idle state, monitoring for a wake-up event independent of the CPU mode. A wake-up event consists of a high-to-low transition on the RX1/DT1 line. (This coincides with the start of a Sync Break or a Wake-up Signal character for the LIN protocol.)

Following a wake-up event, the module generates an RC1IF interrupt. The interrupt is generated synchronously to the Q clocks in normal operating modes (Figure 17-8), and asynchronously, if the device is in Sleep mode (Figure 17-9). The interrupt condition is cleared by reading the RCREG1 register.

The WUE bit is automatically cleared once a low-to-high transition is observed on the RX1 line following the wake-up event. At this point, the EUSART module is in Idle mode and returns to normal operation. This signals to the user that the Sync Break event is over.

### 17.3.4.1 Special Considerations Using Auto-Wake-up

Since auto-wake-up functions by sensing rising edge transitions on RX1/DT1, information with any state changes before the Stop bit may signal a false

End-of-Character (EOC) and cause data or framing errors. Therefore, to work properly, the initial character in the transmission must be all '0's. This can be 00h (8 bytes) for standard RS-232 devices or 000h (12 bits) for LIN bus.

Oscillator start-up time must also be considered, especially in applications using oscillators with longer start-up intervals (i.e., XT or HS mode). The Sync Break (or Wake-up Signal) character must be of sufficient length and be followed by a sufficient interval to allow enough time for the selected oscillator to start and provide proper initialization of the EUSART.

### 17.3.4.2 Special Considerations Using the WUE Bit

The timing of WUE and RC1IF events may cause some confusion when it comes to determining the validity of received data. As noted, setting the WUE bit places the EUSART in an Idle mode. The wake-up event causes a receive interrupt by setting the RC1IF bit. The WUE bit is cleared after this when a rising edge is seen on RX1/DT1. The interrupt condition is then cleared by reading the RCREG1 register. Ordinarily, the data in RCREG1 will be dummy data and should be discarded.

The fact that the WUE bit has been cleared (or is still set) and the RC1IF flag is set should not be used as an indicator of the integrity of the data in RCREG1. Users should consider implementing a parallel method in firmware to verify received data integrity.

To assure that no actual data is lost, check the RCMT bit to verify that a receive operation is not in process. If a receive operation is not occurring, the WUE bit may then be set just prior to entering the Sleep mode.

**FIGURE 17-8: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING NORMAL OPERATION**

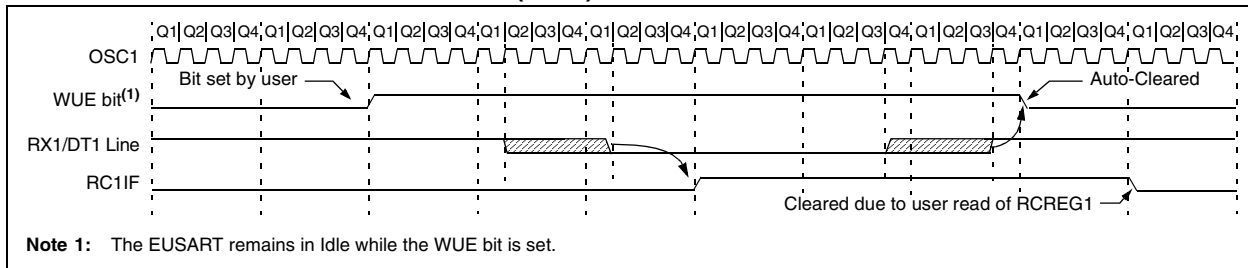
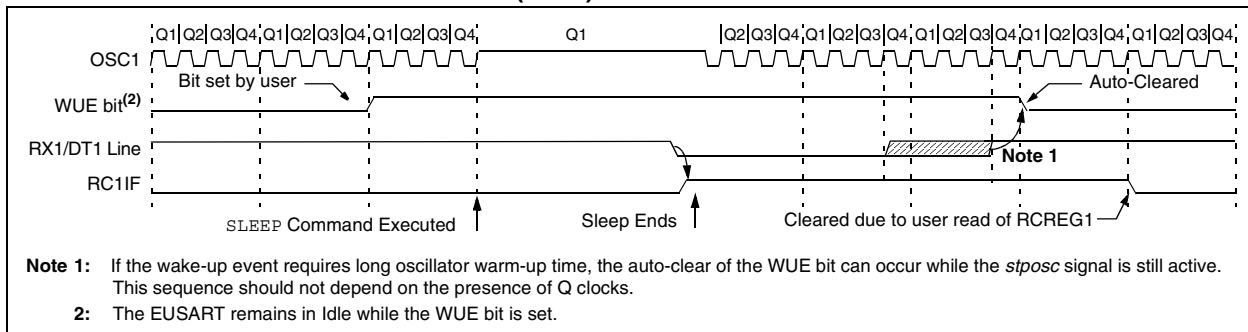


**FIGURE 17-9: AUTO-WAKE-UP BIT (WUE) TIMINGS DURING SLEEP**





## 17.3.5 BREAK CHARACTER SEQUENCE

The Enhanced USART module has the capability of sending the special Break character sequences that are required by the LIN bus standard. The Break character transmit consists of a Start bit, followed by twelve '0' bits and a Stop bit. The Frame Break character is sent whenever the SENDB and TXEN bits (TXSTA1<3> and TXSTA1<5>) are set while the Transmit Shift register is loaded with data. Note that the value of data written to TXREG1 will be ignored and all '0's will be transmitted.

The SENDB bit is automatically reset by hardware after the corresponding Stop bit is sent. This allows the user to preload the transmit FIFO with the next transmit byte following the Break character (typically, the Sync character in the LIN specification).

Note that the data value written to the TXREG1 for the Break character is ignored. The write simply serves the purpose of initiating the proper sequence.

The TRMT bit indicates when the transmit operation is active or Idle, just as it does during normal transmission. See Figure 17-10 for the timing of the Break character sequence.

### 17.3.5.1 Break and Sync Transmit Sequence

The following sequence will send a message frame header made up of a Break, followed by an Auto-Baud Sync byte. This sequence is typical of a LIN bus master.

1. Configure the EUSART for the desired mode.
2. Set the TXEN and SENDB bits to set up the Break character.
3. Load the TXREG1 with a dummy character to initiate transmission (the value is ignored).
4. Write '55h' to TXREG1 to load the Sync character into the transmit FIFO buffer.
5. After the Break has been sent, the SENDB bit is reset by hardware. The Sync character now transmits in the preconfigured mode.

When the TXREG1 becomes empty, as indicated by the TX1IF, the next data byte can be written to TXREG1.

## 17.3.6 RECEIVING A BREAK CHARACTER

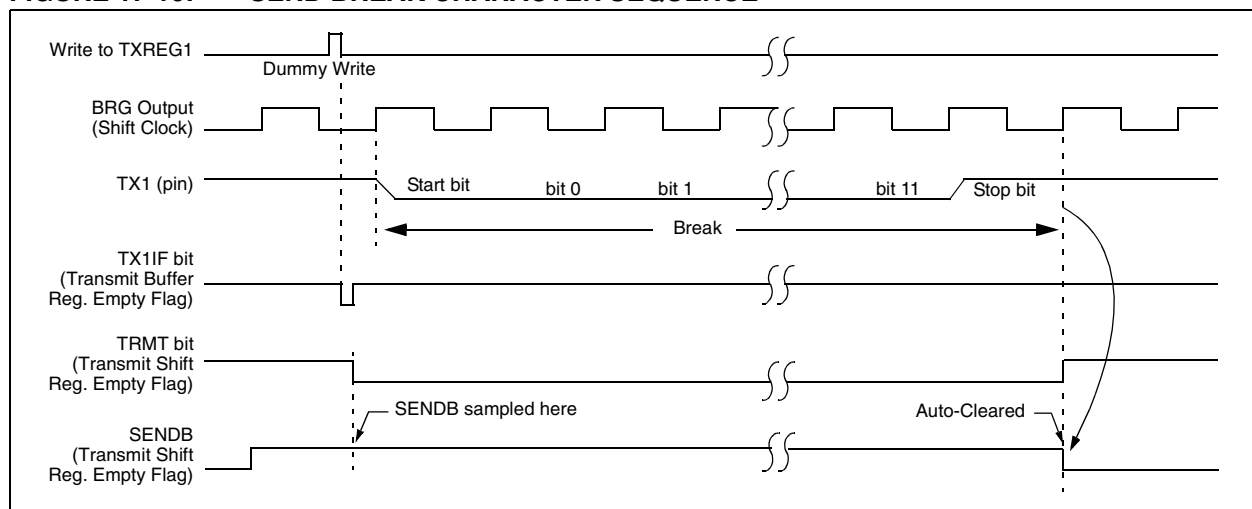
The Enhanced USART module can receive a Break character in two ways.

The first method forces configuration of the baud rate at a frequency of 9/13 the typical speed. This allows for the Stop bit transition to be at the correct sampling location (13 bits for Break versus Start bit and 8 data bits for typical data).

The second method uses the auto-wake-up feature described in **Section 17.3.4 "Auto-Wake-up on Sync Break Character"**. By enabling this feature, the EUSART will sample the next two transitions on RX1/DT1, cause an RC1IF interrupt and receive the next data byte followed by another interrupt.

Note that following a Break character, the user will typically want to enable the Auto-Baud Rate Detect feature. For both methods, the user can set the ABDEN bit once the TX1IF interrupt is observed.

**FIGURE 17-10: SEND BREAK CHARACTER SEQUENCE**



# PIC18F85J11 FAMILY

## 17.4 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA1<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA1<4>). In addition, enable bit, SPEN (RCSTA1<7>), is set in order to configure the TX1 and RX1 pins to CK1 (clock) and DT1 (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK1 line. Clock polarity is selected with the SCKP bit (BAUDCON1<4>). Setting SCKP sets the Idle state on CK1 as high, while clearing the bit sets the Idle state as low. This option is provided to support Microwire devices with this module.

### 17.4.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 17-3. The heart of the transmitter is the Transmit (Serial) Shift register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG1. The TXREG1 register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG1 (if available).

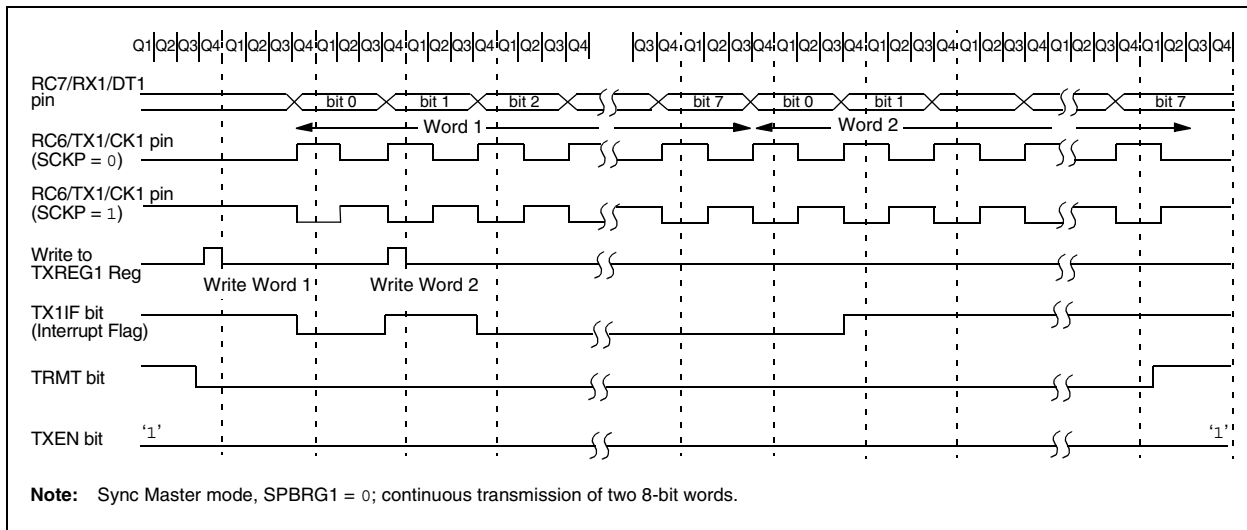
Once the TXREG1 register transfers the data to the TSR register (occurs in one T<sub>CYCLE</sub>), the TXREG1 is empty and the TX1IF flag bit (PIR1<4>) is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TX1IE (PIE1<4>). TX1IF is set regardless of the state of enable bit, TX1IE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREG1 register.

While flag bit TX1IF indicates the status of the TXREG1 register, another bit, TRMT (TXSTA1<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGH1:SPBRG1 registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit, TX1IE.
4. If 9-bit transmission is desired, set bit, TX9.
5. Enable the transmission by setting bit, TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG1 register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 17-11: SYNCHRONOUS TRANSMISSION**



# PIC18F85J11 FAMILY

**FIGURE 17-12: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

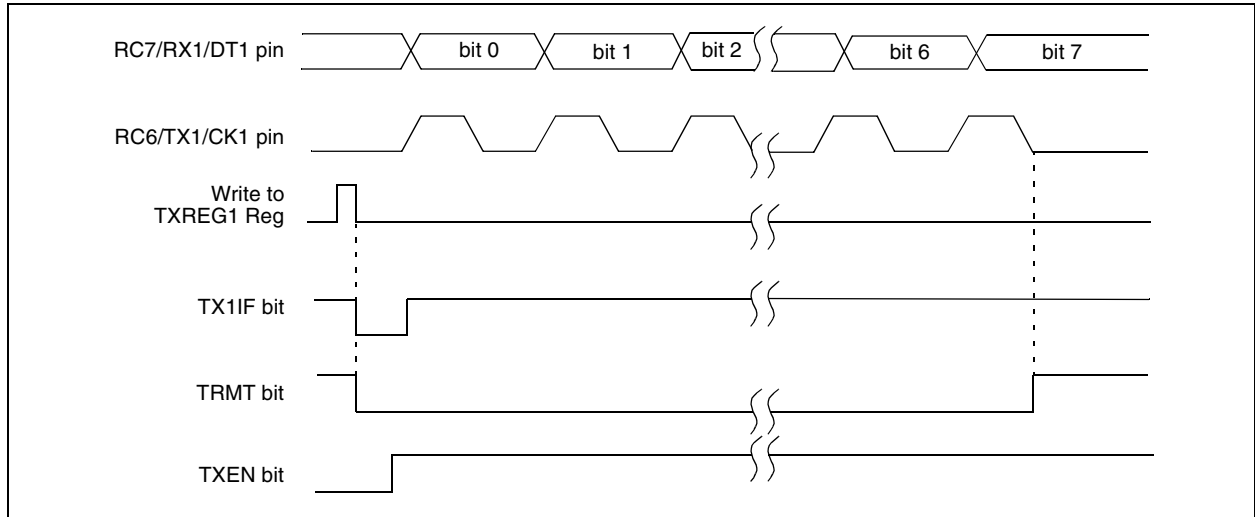


**TABLE 17-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D   | 53                   |
| TXREG1   | EUSART Transmit Register                      |           |        |        |       |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SEnDB | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16 | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |       |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |       |        |        |        | 53                   |
| LATG     | U2OD                                          | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0  | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

# PIC18F85J11 FAMILY

## 17.4.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA1<5>), or the Continuous Receive Enable bit, CREN (RCSTA1<4>). Data is sampled on the RX1 pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH1:SPBRG1 registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. Ensure bits, CREN and SREN, are clear.
4. If interrupts are desired, set enable bit, RC1IE.
5. If 9-bit reception is desired, set bit, RX9.
6. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
7. Interrupt flag bit, RC1IF, will be set when reception is complete and an interrupt will be generated if the enable bit, RC1IE, was set.
8. Read the RCSTA1 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG1 register.
10. If any error occurred, clear the error by clearing bit, CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 17-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**

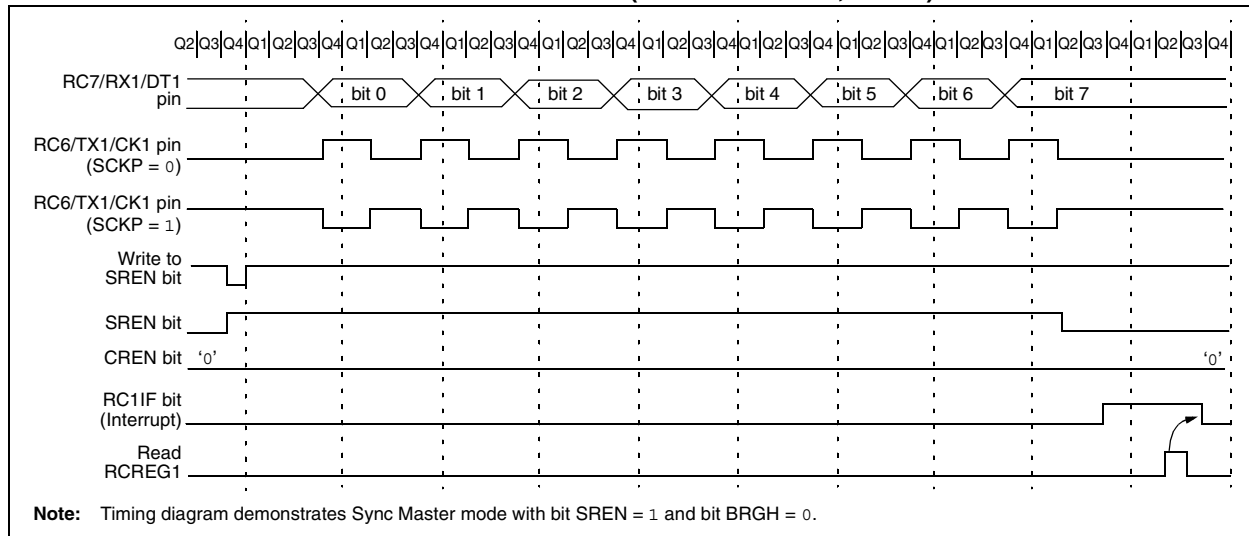


**TABLE 17-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D   | 53                   |
| RCREG1   | EUSART Receive Register                       |           |        |        |       |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SENDB | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16 | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |       |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |       |        |        |        | 53                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

## 17.5 EUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTA<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CK1 pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any low-power mode.

### 17.5.1 EUSART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the Sleep mode.

If two words are written to the TXREG1 and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in the TXREG1 register.
- Flag bit, TX1IF, will not be set.
- When the first word has been shifted out of TSR, the TXREG1 register will transfer the second word to the TSR and flag bit, TX1IF, will now be set.
- If enable bit, TX1IE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
- Clear bits, CREN and SREN.
- If interrupts are desired, set enable bit, TX1IE.
- If 9-bit transmission is desired, set bit, TX9.
- Enable the transmission by setting enable bit, TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG1 register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 17-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D   | 53                   |
| TXREG1   | EUSART Transmit Register                      |           |        |        |       |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SENDB | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16 | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |       |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |       |        |        |        | 53                   |
| LATG     | U2OD                                          | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0  | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

# PIC18F85J11 FAMILY

## 17.5.2 EUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical except in the case of Sleep or any Idle mode, and bit SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREG1 register. If the RC1IE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RC1IE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RC1IF, will be set when reception is complete. An interrupt will be generated if enable bit, RC1IE, was set.
6. Read the RCSTA1 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG1 register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 17-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

| 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   | 51                   |
| PIR1     | PSPIF                                         | ADIF      | RC1IF  | TX1IF  | SSPIF  | —      | TMR2IF | TMR1IF | 53                   |
| PIE1     | PSPIE                                         | ADIE      | RC1IE  | TX1IE  | SSPIE  | —      | TMR2IE | TMR1IE | 53                   |
| IPR1     | PSPIP                                         | ADIP      | RC1IP  | TX1IP  | SSPIP  | —      | TMR2IP | TMR1IP | 53                   |
| RCSTA1   | SPEN                                          | RX9       | SREN   | CREN   | ADDEN  | FERR   | OERR   | RX9D   | 53                   |
| RCREG1   | EUSART Receive Register                       |           |        |        |        |        |        |        | 53                   |
| TXSTA1   | CSRC                                          | TX9       | TXEN   | SYNC   | SENDER | BRGH   | TRMT   | TX9D   | 53                   |
| BAUDCON1 | ABDOVF                                        | RCMT      | —      | SCKP   | BRG16  | —      | WUE    | ABDEN  | 54                   |
| SPBRGH1  | EUSART Baud Rate Generator Register High Byte |           |        |        |        |        |        |        | 54                   |
| SPBRG1   | EUSART Baud Rate Generator Register Low Byte  |           |        |        |        |        |        |        | 53                   |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

## 18.0 ADDRESSABLE UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (AUSART)

The Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART) module is very similar in function to the Enhanced USART module discussed in the previous chapter. It is provided as an additional channel for serial communication with external devices for those situations that do not require Auto-Baud Detection or LIN bus support.

The AUSART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

The pins of the AUSART module are multiplexed with the functions of PORTG (RG1/TX2/CK2 and RG2/RX2/DT2, respectively). In order to configure these pins as an AUSART:

- bit SPEN (RCSTA2<7>) must be set (= 1)
- bit TRISG<2> must be set (= 1)
- bit TRISG<1> must be cleared (= 0) for Asynchronous and Synchronous Master modes
- bit TRISG<1> must be set (= 1) for Synchronous Slave mode

|                                                                                                        |
|--------------------------------------------------------------------------------------------------------|
| <b>Note:</b> The AUSART control will automatically reconfigure the pin from input to output as needed. |
|--------------------------------------------------------------------------------------------------------|

The driver for the TX2 output pin can also be optionally configured as an open-drain output. This feature allows the voltage level on the pin to be pulled to a higher level through an external pull-up resistor, and allows the output to communicate with external circuits without the need for additional level shifters.

The open-drain output option is controlled by the U2OD bit (LATG<7>). Setting this bit configures the pin for open-drain operation.

### 18.1 Control Registers

The operation of the Addressable USART module is controlled through two registers, TXSTA2 and RCSTA2. These are detailed in Register 18-1 and Register 18-2, respectively.

# PIC18F85J11 FAMILY

**REGISTER 18-1: TXSTA2: AUSART TRANSMIT STATUS AND CONTROL REGISTER**

| R/W-0 | R/W-0 | R/W-0               | R/W-0 | U-0 | R/W-0 | R-1  | R/W-0 |
|-------|-------|---------------------|-------|-----|-------|------|-------|
| CSRC  | TX9   | TXEN <sup>(1)</sup> | SYNC  | —   | BRGH  | TRMT | TX9D  |
| 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      **CSRC:** Clock Source Select bit  
Asynchronous mode:  
Don't care.  
Synchronous mode:  
1 = Master mode (clock generated internally from BRG)  
0 = Slave mode (clock from external source)
- bit 6      **TX9:** 9-Bit Transmit Enable bit  
1 = Selects 9-bit transmission  
0 = Selects 8-bit transmission
- bit 5      **TXEN:** Transmit Enable bit<sup>(1)</sup>  
1 = Transmit enabled  
0 = Transmit disabled
- bit 4      **SYNC:** AUSART Mode Select bit  
1 = Synchronous mode  
0 = Asynchronous mode
- bit 3      **Unimplemented:** Read as '0'
- bit 2      **BRGH:** High Baud Rate Select bit  
Asynchronous mode:  
1 = High speed  
0 = Low speed  
Synchronous mode:  
Unused in this mode.
- bit 1      **TRMT:** Transmit Shift Register Status bit  
1 = TSR empty  
0 = TSR full
- bit 0      **TX9D:** 9th Bit of Transmit Data  
Can be address/data bit or parity bit.

**Note 1:** SREN/CREN overrides TXEN in Sync mode.



# PIC18F85J11 FAMILY

**REGISTER 18-2: RCSTA2: AUSART RECEIVE STATUS AND CONTROL REGISTER**

| 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 | <b>SPEN:</b> Serial Port Enable bit<br>1 = Serial port enabled (configures RX2/DT2 and TX2/CK2 pins as serial port pins)<br>0 = Serial port disabled (held in Reset)                                                                                                                                                                                   |
| bit 6 | <b>RX9:</b> 9-Bit Receive Enable bit<br>1 = Selects 9-bit reception<br>0 = Selects 8-bit reception                                                                                                                                                                                                                                                     |
| bit 5 | <b>SREN:</b> Single Receive Enable bit<br><u>Asynchronous mode:</u><br>Don't care.<br><u>Synchronous mode – Master:</u><br>1 = Enables single receive<br>0 = Disables single receive<br>This bit is cleared after reception is complete.<br><u>Synchronous mode – Slave:</u><br>Don't care.                                                            |
| bit 4 | <b>CREN:</b> Continuous Receive Enable bit<br><u>Asynchronous mode:</u><br>1 = Enables receiver<br>0 = Disables receiver<br><u>Synchronous mode:</u><br>1 = Enables continuous receive until enable bit, CREN, is cleared (CREN overrides SREN)<br>0 = Disables continuous receive                                                                     |
| bit 3 | <b>ADDEN:</b> Address Detect Enable bit<br><u>Asynchronous mode 9-Bit (RX9 = 1):</u><br>1 = Enables address detection, enables interrupt and loads the receive buffer when RSR<8> is set<br>0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit<br><u>Asynchronous mode 9-Bit (RX9 = 0):</u><br>Don't care. |
| bit 2 | <b>FERR:</b> Framing Error bit<br>1 = Framing error (can be updated by reading RCREG2 register and receiving next valid byte)<br>0 = No framing error                                                                                                                                                                                                  |
| bit 1 | <b>OERR:</b> Overrun Error bit<br>1 = Overrun error (can be cleared by clearing the CREN bit)<br>0 = No overrun error                                                                                                                                                                                                                                  |
| bit 0 | <b>RX9D:</b> 9th Bit of Received Data<br>This can be address/data bit or parity bit and must be calculated by user firmware.                                                                                                                                                                                                                           |

# PIC18F85J11 FAMILY

## 18.2 AUSART Baud Rate Generator (BRG)

The BRG is a dedicated, 8-bit generator that supports both the Asynchronous and Synchronous modes of the AUSART.

The SPBRG2 register controls the period of a free-running timer. In Asynchronous mode, bit BRGH (TXSTA<2>) also controls the baud rate. In Synchronous mode, BRGH is ignored. Table 18-1 shows the formula for computation of the baud rate for different AUSART modes, which only apply in Master mode (internally generated clock).

Given the desired baud rate and FOSC, the nearest integer value for the SPBRG2 register can be calculated using the formulas in Table 18-1. From this, the error in baud rate can be determined. An example calculation is shown in Example 18-1. Typical baud rates and error values for the various Asynchronous modes are shown in Table 18-2. It may be advanta-

geous to use the high baud rate (BRGH = 1) to reduce the baud rate error, or achieve a slow baud rate for a fast oscillator frequency.

Writing a new value to the SPBRG2 register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

### 18.2.1 OPERATION IN POWER-MANAGED MODES

The device clock is used to generate the desired baud rate. When one of the power-managed modes is entered, the new clock source may be operating at a different frequency. This may require an adjustment to the value in the SPBRG2 register.

### 18.2.2 SAMPLING

The data on the RX2 pin is sampled three times by a majority detect circuit to determine if a high or low level is present at the RX2 pin.

**TABLE 18-1: BAUD RATE FORMULAS**

| Configuration Bits |      | BRG/AUSART Mode | Baud Rate Formula      |
|--------------------|------|-----------------|------------------------|
| SYNC               | BRGH |                 |                        |
| 0                  | 0    | Asynchronous    | $F_{osc}/[64 (n + 1)]$ |
| 0                  | 1    | Asynchronous    | $F_{osc}/[16 (n + 1)]$ |
| 1                  | x    | Synchronous     | $F_{osc}/[4 (n + 1)]$  |

**Legend:** x = Don't care, n = Value of SPBRG2 register

### EXAMPLE 18-1: CALCULATING BAUD RATE ERROR

For a device with FOSC of 16 MHz, desired baud rate of 9600, Asynchronous mode, BRGH = 0:

Desired Baud Rate =  $F_{osc}/(64 ([SPBRG2] + 1))$

Solving for SPBRG2:

$$\begin{aligned} X &= ((F_{osc}/\text{Desired Baud Rate})/64) - 1 \\ &= ((16000000/9600)/64) - 1 \\ &= [25.042] = 25 \end{aligned}$$

Calculated Baud Rate =  $16000000/(64 (25 + 1))$   
= 9615

Error =  $(\text{Calculated Baud Rate} - \text{Desired Baud Rate})/\text{Desired Baud Rate}$   
=  $(9615 - 9600)/9600 = 0.16\%$

**TABLE 18-2: REGISTERS ASSOCIATED WITH THE BAUD RATE GENERATOR**

| Name   | Bit 7                               | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Reset Values on Page |
|--------|-------------------------------------|-------|-------|-------|-------|-------|-------|-------|----------------------|
| TXSTA2 | CSRC                                | TX9   | TXEN  | SYNC  | —     | BRGH  | TRMT  | TX9D  | 55                   |
| RCSTA2 | SPEN                                | RX9   | SREN  | CREN  | ADDEN | FERR  | OERR  | RX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |       |       |       |       |       |       |       | 55                   |

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used by the BRG.

# PIC18F85J11 FAMILY

**TABLE 18-3: BAUD RATES FOR ASYNCHRONOUS MODES**

| BAUD RATE (K) | BRGH = 0          |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                 | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 1.2           | —                 | —       | —                     | 1.221             | 1.73    | 255                   | 1.202             | 0.16    | 129                   | 1.201            | -0.16   | 103                   |
| 2.4           | 2.441             | 1.73    | 255                   | 2.404             | 0.16    | 129                   | 2.404             | 0.16    | 64                    | 2.403            | -0.16   | 51                    |
| 9.6           | 9.615             | 0.16    | 64                    | 9.766             | 1.73    | 31                    | 9.766             | 1.73    | 15                    | 9.615            | -0.16   | 12                    |
| 19.2          | 19.531            | 1.73    | 31                    | 19.531            | 1.73    | 15                    | 19.531            | 1.73    | 7                     | —                | —       | —                     |
| 57.6          | 56.818            | -1.36   | 10                    | 62.500            | 8.51    | 4                     | 52.083            | -9.58   | 2                     | —                | —       | —                     |
| 115.2         | 125.000           | 8.51    | 4                     | 104.167           | -9.58   | 2                     | 78.125            | -32.18  | 1                     | —                | —       | —                     |

| BAUD RATE (K) | BRGH = 0         |         |                       |                  |         |                       |                  |         |                       |
|---------------|------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | 0.300            | 0.16    | 207                   | 0.300            | -0.16   | 103                   | 0.300            | -0.16   | 51                    |
| 1.2           | 1.202            | 0.16    | 51                    | 1.201            | -0.16   | 25                    | 1.201            | -0.16   | 12                    |
| 2.4           | 2.404            | 0.16    | 25                    | 2.403            | -0.16   | 12                    | —                | —       | —                     |
| 9.6           | 8.929            | -6.99   | 6                     | —                | —       | —                     | —                | —       | —                     |
| 19.2          | 20.833           | 8.51    | 2                     | —                | —       | —                     | —                | —       | —                     |
| 57.6          | 62.500           | 8.51    | 0                     | —                | —       | —                     | —                | —       | —                     |
| 115.2         | 62.500           | -45.75  | 0                     | —                | —       | —                     | —                | —       | —                     |

| BAUD RATE (K) | BRGH = 1          |         |                       |                   |         |                       |                   |         |                       |                  |         |                       |
|---------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|-------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 40.000 MHz |         |                       | Fosc = 20.000 MHz |         |                       | Fosc = 10.000 MHz |         |                       | Fosc = 8.000 MHz |         |                       |
|               | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)   | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                 | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 1.2           | —                 | —       | —                     | —                 | —       | —                     | —                 | —       | —                     | —                | —       | —                     |
| 2.4           | —                 | —       | —                     | —                 | —       | —                     | 2.441             | 1.73    | 255                   | 2.403            | -0.16   | 207                   |
| 9.6           | 9.766             | 1.73    | 255                   | 9.615             | 0.16    | 129                   | 9.615             | 0.16    | 64                    | 9.615            | -0.16   | 51                    |
| 19.2          | 19.231            | 0.16    | 129                   | 19.231            | 0.16    | 64                    | 19.531            | 1.73    | 31                    | 19.230           | -0.16   | 25                    |
| 57.6          | 58.140            | 0.94    | 42                    | 56.818            | -1.36   | 21                    | 56.818            | -1.36   | 10                    | 55.555           | 3.55    | 8                     |
| 115.2         | 113.636           | -1.36   | 21                    | 113.636           | -1.36   | 10                    | 125.000           | 8.51    | 4                     | —                | —       | —                     |

| BAUD RATE (K) | BRGH = 1         |         |                       |                  |         |                       |                  |         |                       |
|---------------|------------------|---------|-----------------------|------------------|---------|-----------------------|------------------|---------|-----------------------|
|               | Fosc = 4.000 MHz |         |                       | Fosc = 2.000 MHz |         |                       | Fosc = 1.000 MHz |         |                       |
|               | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) | Actual Rate (K)  | % Error | SPBRG Value (decimal) |
| 0.3           | —                | —       | —                     | —                | —       | —                     | 0.300            | -0.16   | 207                   |
| 1.2           | 1.202            | 0.16    | 207                   | 1.201            | -0.16   | 103                   | 1.201            | -0.16   | 51                    |
| 2.4           | 2.404            | 0.16    | 103                   | 2.403            | -0.16   | 51                    | 2.403            | -0.16   | 25                    |
| 9.6           | 9.615            | 0.16    | 25                    | 9.615            | -0.16   | 12                    | —                | —       | —                     |
| 19.2          | 19.231           | 0.16    | 12                    | —                | —       | —                     | —                | —       | —                     |
| 57.6          | 62.500           | 8.51    | 3                     | —                | —       | —                     | —                | —       | —                     |
| 115.2         | 125.000          | 8.51    | 1                     | —                | —       | —                     | —                | —       | —                     |

# PIC18F85J11 FAMILY

## 18.3 AUSART Asynchronous Mode

The Asynchronous mode of operation is selected by clearing the SYNC bit (TXSTA2<4>). In this mode, the AUSART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip, dedicated, 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator.

The AUSART transmits and receives the LSb first. The AUSART's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH bit (TXSTA2<2>). Parity is not supported by the hardware but can be implemented in software and stored as the 9th data bit.

When operating in Asynchronous mode, the AUSART module consists of the following important elements:

- Baud Rate Generator
- Sampling Circuit
- Asynchronous Transmitter
- Asynchronous Receiver

### 18.3.1 AUSART ASYNCHRONOUS TRANSMITTER

The AUSART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the Transmit (Serial) Shift register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG2. The TXREG2 register is loaded with data in software. The TSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG2 register (if available).

Once the TXREG2 register transfers the data to the TSR register (occurs in one Tcy), the TXREG2 register is empty and the TX2IF flag bit (PIR3<4>) is set. This interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TX2IE (PIE3<4>). TX2IF will be set regardless of the state of TX2IE; it cannot be cleared in software. TX2IF is also not cleared immediately upon loading TXREG2, but becomes valid in the second instruction cycle following the load instruction. Polling TX2IF immediately following a load of TXREG2 will return invalid results.

While TX2IF indicates the status of the TXREG2 register, another bit, TRMT (TXSTA2<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty.

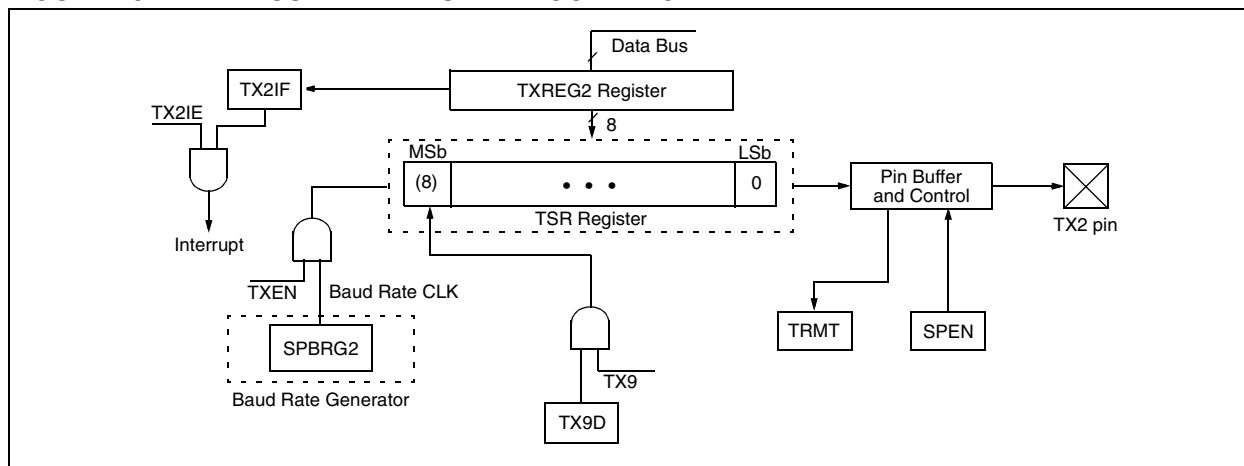
**Note 1:** The TSR register is not mapped in data memory so it is not available to the user.

**2:** Flag bit, TX2IF, is set when enable bit, TXEN, is set.

To set up an Asynchronous Transmission:

1. Initialize the SPBRG2 register for the appropriate baud rate. Set or clear the BRGH bit, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are desired, set enable bit, TX2IE.
4. If 9-bit transmission is desired, set transmit bit, TX9; can be used as address/data bit.
5. Enable the transmission by setting bit, TXEN, which will also set bit, TX2IF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit, TX9D.
7. Load data to the TXREG2 register (starts transmission).
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 18-1: AUSART TRANSMIT BLOCK DIAGRAM**



# PIC18F85J11 FAMILY

**FIGURE 18-2: ASYNCHRONOUS TRANSMISSION**

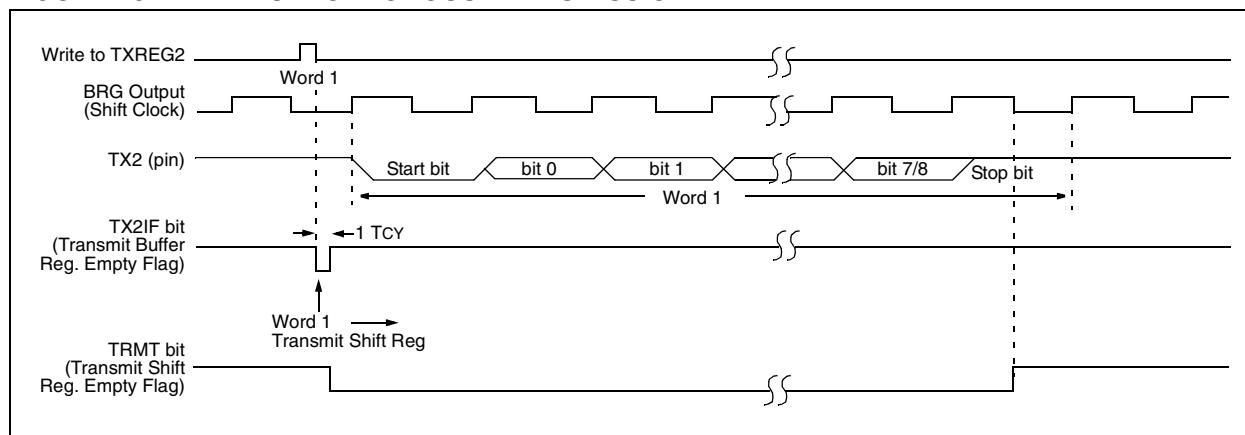


**FIGURE 18-3: ASYNCHRONOUS TRANSMISSION (BACK-TO-BACK)**

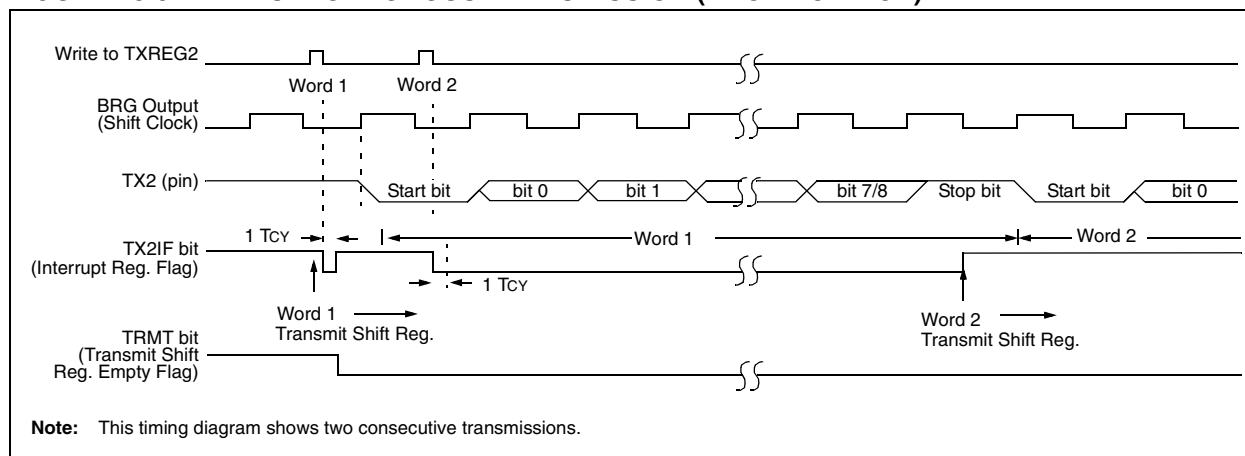


**TABLE 18-4: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| TXREG2 | AUSART Transmit Register            |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |
| LATG   | U2OD                                | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0 | 54                   |

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

# PIC18F85J11 FAMILY

## 18.3.2 AUSART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 18-4. The data is received on the RX2 pin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

To set up an Asynchronous Reception:

1. Initialize the SPBRG2 register for the appropriate baud rate. Set or clear the BRGH bit, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing bit, SYNC, and setting bit, SPEN.
3. If interrupts are desired, set enable bit, RC2IE.
4. If 9-bit reception is desired, set bit, RX9.
5. Enable the reception by setting bit, CREN.
6. Flag bit, RC2IF, will be set when reception is complete and an interrupt will be generated if enable bit, RC2IE, was set.
7. Read the RCSTA2 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG2 register.
9. If any error occurred, clear the error by clearing enable bit, CREN.
10. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

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

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

1. Initialize the SPBRG2 register for the appropriate baud rate. Set or clear the BRGH and BRG16 bits, as required, to achieve the desired baud rate.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RC2IP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RC2IF bit will be set when reception is complete. The interrupt will be Acknowledged if the RC2IE and GIE bits are set.
8. Read the RCSTA2 register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG2 to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

**FIGURE 18-4: AUSART RECEIVE BLOCK DIAGRAM**

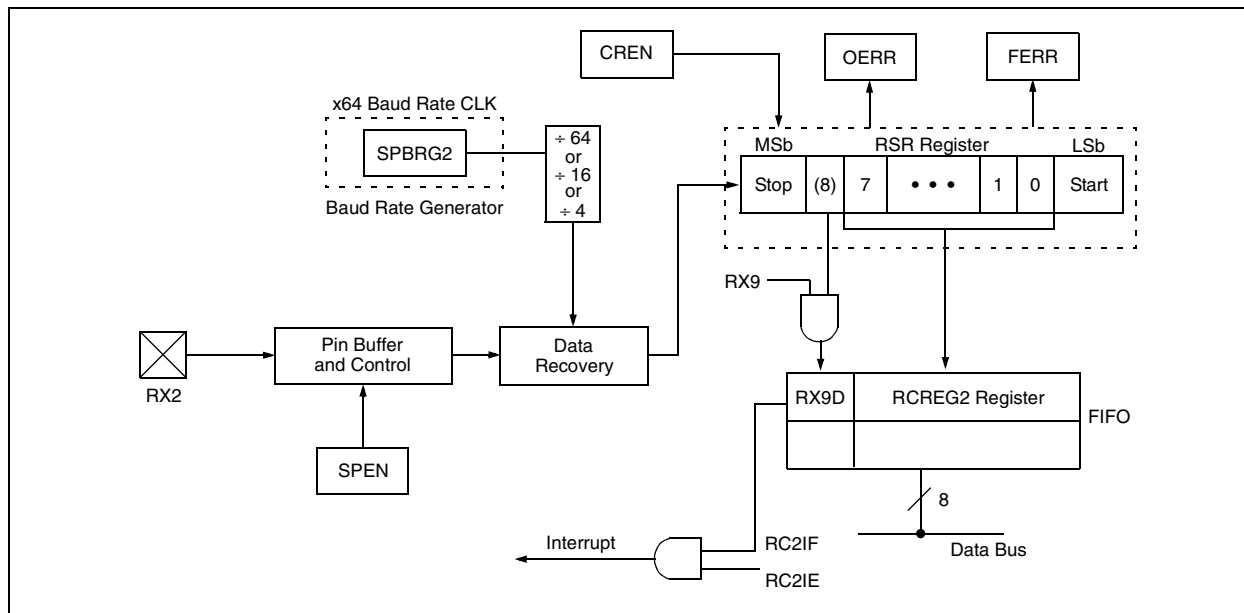


**FIGURE 18-5: ASYNCHRONOUS RECEPTION**

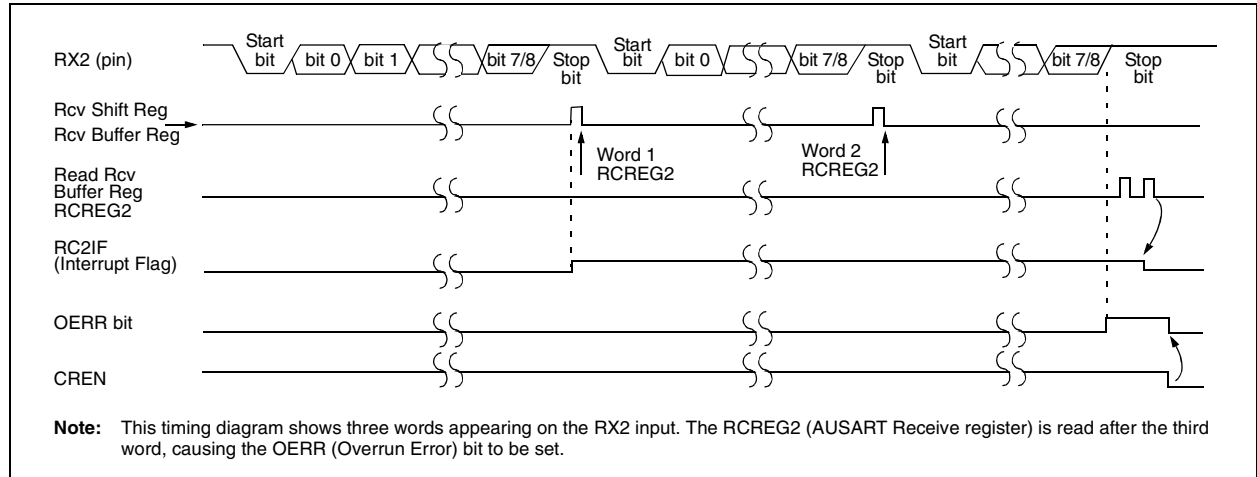


**TABLE 18-5: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| RCREG2 | AUSART Receive Register             |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |

**Legend:** — = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

# PIC18F85J11 FAMILY

## 18.4 AUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA2<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit, SYNC (TXSTA2<4>). In addition, enable bit, SPEN (RCSTA2<7>), is set in order to configure the TX2 and RX2 pins to CK2 (clock) and DT2 (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK2 line.

### 18.4.1 AUSART SYNCHRONOUS MASTER TRANSMISSION

The AUSART transmitter block diagram is shown in Figure 18-1. The heart of the transmitter is the Transmit (Serial) Shift register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG2. The TXREG2 register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG2 (if available).

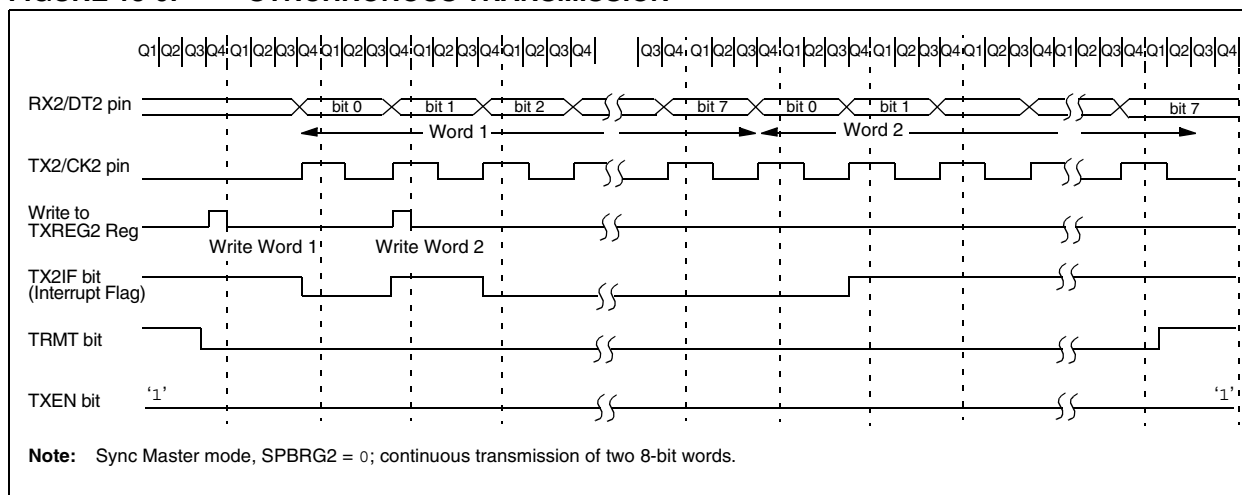
Once the TXREG2 register transfers the data to the TSR register (occurs in one T<sub>CYCLE</sub>), the TXREG2 is empty and the TX2IF flag bit (PIR3<4>) is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TX2IE (PIE3<4>). TX2IF is set regardless of the state of enable bit, TX2IE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREG2 register.

While flag bit, TX2IF, indicates the status of the TXREG2 register, another bit, TRMT (TXSTA2<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRG2 register for the appropriate baud rate.
2. Enable the synchronous master serial port by setting bits, SYNC, SPEN and CSRC.
3. If interrupts are desired, set enable bit, TX2IE.
4. If 9-bit transmission is desired, set bit, TX9.
5. Enable the transmission by setting bit, TXEN.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Start transmission by loading data to the TXREG2 register.
8. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 18-6: SYNCHRONOUS TRANSMISSION**





# PIC18F85J11 FAMILY

**FIGURE 18-7: SYNCHRONOUS TRANSMISSION (THROUGH TXEN)**

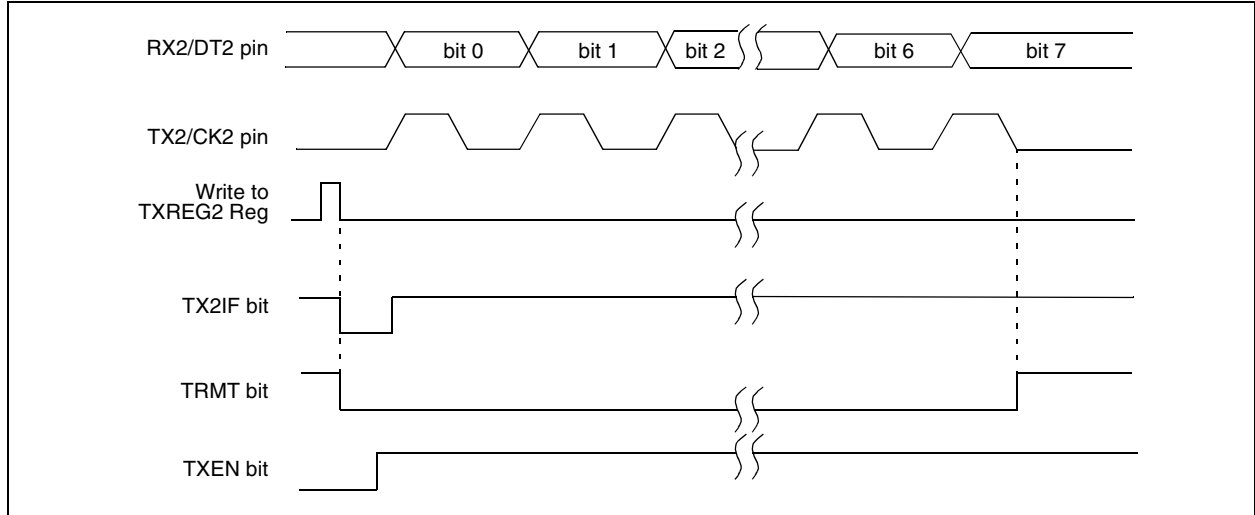


**TABLE 18-6: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| TXREG2 | AUSART Transmit Register            |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |
| LATG   | U2OD                                | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0 | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master transmission.

# PIC18F85J11 FAMILY

## 18.4.2 AUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA2<5>), or the Continuous Receive Enable bit, CREN (RCSTA2<4>). Data is sampled on the RX2 pin on the falling edge of the clock.

If enable bit, SREN, is set, only a single word is received. If enable bit, CREN, is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRG2 register for the appropriate baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If interrupts are desired, set enable bit, RC2IE.
5. If 9-bit reception is desired, set bit, RX9.
6. If a single reception is required, set bit, SREN. For continuous reception, set bit, CREN.
7. Interrupt flag bit, RC2IF, will be set when reception is complete and an interrupt will be generated if the enable bit, RC2IE, was set.
8. Read the RCSTA2 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
9. Read the 8-bit received data by reading the RCREG2 register.
10. If any error occurred, clear the error by clearing bit, CREN.
11. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**FIGURE 18-8: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)**

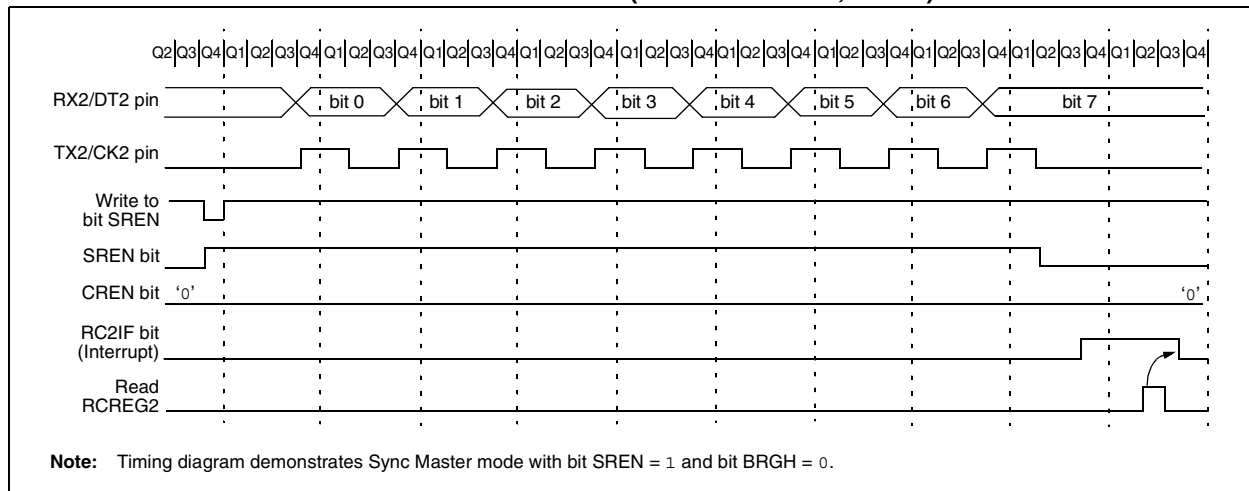


**TABLE 18-7: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| RCREG2 | AUSART Receive Register             |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

## 18.5 AUSART Synchronous Slave Mode

Synchronous Slave mode is entered by clearing bit, CSRC (TXSTA2<7>). This mode differs from the Synchronous Master mode in that the shift clock is supplied externally at the CK2 pin (instead of being supplied internally in Master mode). This allows the device to transfer or receive data while in any low-power mode.

### 18.5.1 AUSART SYNCHRONOUS SLAVE TRANSMIT

The operation of the Synchronous Master and Slave modes are identical except in the case of the Sleep mode.

If two words are written to the TXREG2 and then the SLEEP instruction is executed, the following will occur:

- The first word will immediately transfer to the TSR register and transmit.
- The second word will remain in the TXREG2 register.
- Flag bit, TX2IF, will not be set.
- When the first word has been shifted out of TSR, the TXREG2 register will transfer the second word to the TSR and flag bit, TX2IF, will now be set.
- If enable bit, TX2IE, is set, the interrupt will wake the chip from Sleep. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Transmission:

- Enable the synchronous slave serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
- Clear bits, CREN and SREN.
- If interrupts are desired, set enable bit, TX2IE.
- If 9-bit transmission is desired, set bit, TX9.
- Enable the transmission by setting enable bit, TXEN.
- If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
- Start transmission by loading data to the TXREG2 register.
- If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 18-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| TXREG2 | AUSART Transmit Register            |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |
| LATG   | U2OD                                | U1OD      | —      | LATG4  | LATG3 | LATG2  | LATG1  | LATG0 | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used for synchronous slave transmission.

# PIC18F85J11 FAMILY

## 18.5.2 AUSART SYNCHRONOUS SLAVE RECEPTION

The operation of the Synchronous Master and Slave modes is identical except in the case of Sleep or any Idle mode, and bit, SREN, which is a “don’t care” in Slave mode.

If receive is enabled by setting the CREN bit prior to entering Sleep or any Idle mode, then a word may be received while in this low-power mode. Once the word is received, the RSR register will transfer the data to the RCREG2 register. If the RC2IE enable bit is set, the interrupt generated will wake the chip from the low-power mode. If the global interrupt is enabled, the program will branch to the interrupt vector.

To set up a Synchronous Slave Reception:

1. Enable the synchronous master serial port by setting bits, SYNC and SPEN, and clearing bit, CSRC.
2. If interrupts are desired, set enable bit, RC2IE.
3. If 9-bit reception is desired, set bit, RX9.
4. To enable reception, set enable bit, CREN.
5. Flag bit, RC2IF, will be set when reception is complete. An interrupt will be generated if enable bit, RC2IE, was set.
6. Read the RCSTA2 register to get the 9th bit (if enabled) and determine if any error occurred during reception.
7. Read the 8-bit received data by reading the RCREG2 register.
8. If any error occurred, clear the error by clearing bit, CREN.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

**TABLE 18-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION**

| 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  | 51                   |
| PIR3   | —                                   | —         | RC2IF  | TX2IF  | —     | CCP2IF | CCP1IF | —     | 53                   |
| PIE3   | —                                   | —         | RC2IE  | TX2IE  | —     | CCP2IE | CCP1IE | —     | 53                   |
| IPR3   | —                                   | —         | RC2IP  | TX2IP  | —     | CCP2IP | CCP1IP | —     | 53                   |
| RCSTA2 | SPEN                                | RX9       | SREN   | CREN   | ADDEN | FERR   | OERR   | RX9D  | 55                   |
| RCREG2 | AUSART Receive Register             |           |        |        |       |        |        |       | 55                   |
| TXSTA2 | CSRC                                | TX9       | TXEN   | SYNC   | —     | BRGH   | TRMT   | TX9D  | 55                   |
| SPBRG2 | AUSART Baud Rate Generator Register |           |        |        |       |        |        |       | 55                   |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used for synchronous slave reception.

## 19.0 10-BIT ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) converter module has 12 inputs for all PIC18F85J11 family devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital number.

The module has five registers:

- A/D Result Register High Byte (ADRESH)
- A/D Result Register Low Byte (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register, shown in Register 19-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 19-2, configures the functions of the port pins. The ADCON2 register, shown in Register 19-3, configures the A/D clock source, programmed acquisition time and justification.

### REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0   | R/W-0 |
|-------|-----|-------|-------|-------|-------|---------|-------|
| ADCAL | —   | CHS3  | CHS2  | CHS1  | CHS0  | GO/DONE | ADON  |
| 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      **ADCAL:** A/D Calibration bit  
             1 = Calibration is performed on next A/D conversion  
             0 = Normal A/D converter operation (no calibration is performed)
- bit 6      **Unimplemented:** Read as '0'
- bit 5-2      **CHS3:CHS0:** Analog Channel Select bits  
             0000 = Channel 00 (AN0)  
             0001 = Channel 01 (AN1)  
             0010 = Channel 02 (AN2)  
             0011 = Channel 03 (AN3)  
             0100 = Channel 04 (AN4)  
             0101 = Channel 05 (AN5)  
             0110 = Channel 06 (AN6)  
             0111 = Channel 07 (AN7)  
             1000 = Channel 08 (AN8)  
             1001 = Channel 09 (AN9)  
             1010 = Channel 10 (AN10)  
             1011 = Channel 11 (AN11)  
             11xx = Unused
- bit 1      **GO/DONE:** A/D Conversion Status bit  
             When ADON = 1:  
             1 = A/D conversion in progress  
             0 = A/D Idle
- bit 0      **ADON:** A/D On bit  
             1 = A/D converter module is enabled  
             0 = A/D converter module is disabled

# PIC18F85J11 FAMILY

## REGISTER 19-2: ADCON1: A/D CONTROL REGISTER 1

|       |     |       |       |       |       |       |       |
|-------|-----|-------|-------|-------|-------|-------|-------|
| U-0   | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| —     | —   | VCFG1 | VCFG0 | 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 7-6 **Unimplemented:** Read as '0'

bit 5 **VCFG1:** Voltage Reference Configuration bit (VREF- source)

1 = VREF- (AN2)

0 = AVSS

bit 4 **VCFG0:** Voltage Reference Configuration bit (VREF+ source)

1 = VREF+ (AN3)

0 = AVDD

bit 3-0 **PCFG3:PCFG0:** A/D Port Configuration Control bits:

| PCFG<3:0> | AN11 | AN10 | AN9 | AN8 | AN7 | AN6 | AN5 | AN4 | AN3 | AN2 | AN1 | AN0 |
|-----------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0000      | A    | A    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0001      | A    | A    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0010      | A    | A    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0011      | A    | A    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0100      | D    | A    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0101      | D    | D    | A   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0110      | D    | D    | D   | A   | A   | A   | A   | A   | A   | A   | A   | A   |
| 0111      | D    | D    | D   | D   | A   | A   | A   | A   | A   | A   | A   | A   |
| 1000      | D    | D    | D   | D   | D   | A   | A   | A   | A   | A   | A   | A   |
| 1001      | D    | D    | D   | D   | D   | D   | A   | A   | A   | A   | A   | A   |
| 1010      | D    | D    | D   | D   | D   | D   | D   | A   | A   | A   | A   | A   |
| 1011      | D    | D    | D   | D   | D   | D   | D   | D   | A   | A   | A   | A   |
| 1100      | D    | D    | D   | D   | D   | D   | D   | D   | D   | A   | A   | A   |
| 1101      | D    | D    | D   | D   | D   | D   | D   | D   | D   | D   | A   | A   |
| 1110      | D    | D    | D   | D   | D   | D   | D   | D   | D   | D   | D   | A   |
| 1111      | D    | D    | D   | D   | D   | D   | D   | D   | D   | D   | D   | D   |

A = Analog input

D = Digital I/O

# PIC18F85J11 FAMILY

**REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2**

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|-----|-------|-------|-------|-------|-------|-------|
| ADFM  | —   | ACQT2 | ACQT1 | ACQT0 | ADCS2 | ADCS1 | ADCS0 |
| 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 **ADFM:** A/D Result Format Select bit

1 = Right justified

0 = Left justified

bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT2:ACQT0:** A/D Acquisition Time Select bits

111 = 20 TAD

110 = 16 TAD

101 = 12 TAD

100 = 8 TAD

011 = 6 TAD

010 = 4 TAD

001 = 2 TAD

000 = 0 TAD<sup>(1)</sup>

bit 2-0 **ADCS2:ADCS0:** A/D Conversion Clock Select bits

111 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>

110 = FOSC/64

101 = FOSC/16

100 = FOSC/4

011 = FRC (clock derived from A/D RC oscillator)<sup>(1)</sup>

010 = FOSC/32

001 = FOSC/8

000 = FOSC/2

**Note 1:** If the A/D FRC clock source is selected, a delay of one T<sub>CY</sub> (instruction cycle) is added before the A/D clock starts. This allows the **SLEEP** instruction to be executed before starting a conversion.

# PIC18F85J11 FAMILY

The analog reference voltage is software selectable to either the device's positive and negative supply voltage (AVDD and AVSS), or the voltage level on the RA3/AN3/VREF+ and RA2/AN2/VREF- pins.

The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D conversion clock must be derived from the A/D's internal RC oscillator.

The output of the sample and hold is the input into the converter, which generates the result via successive approximation.

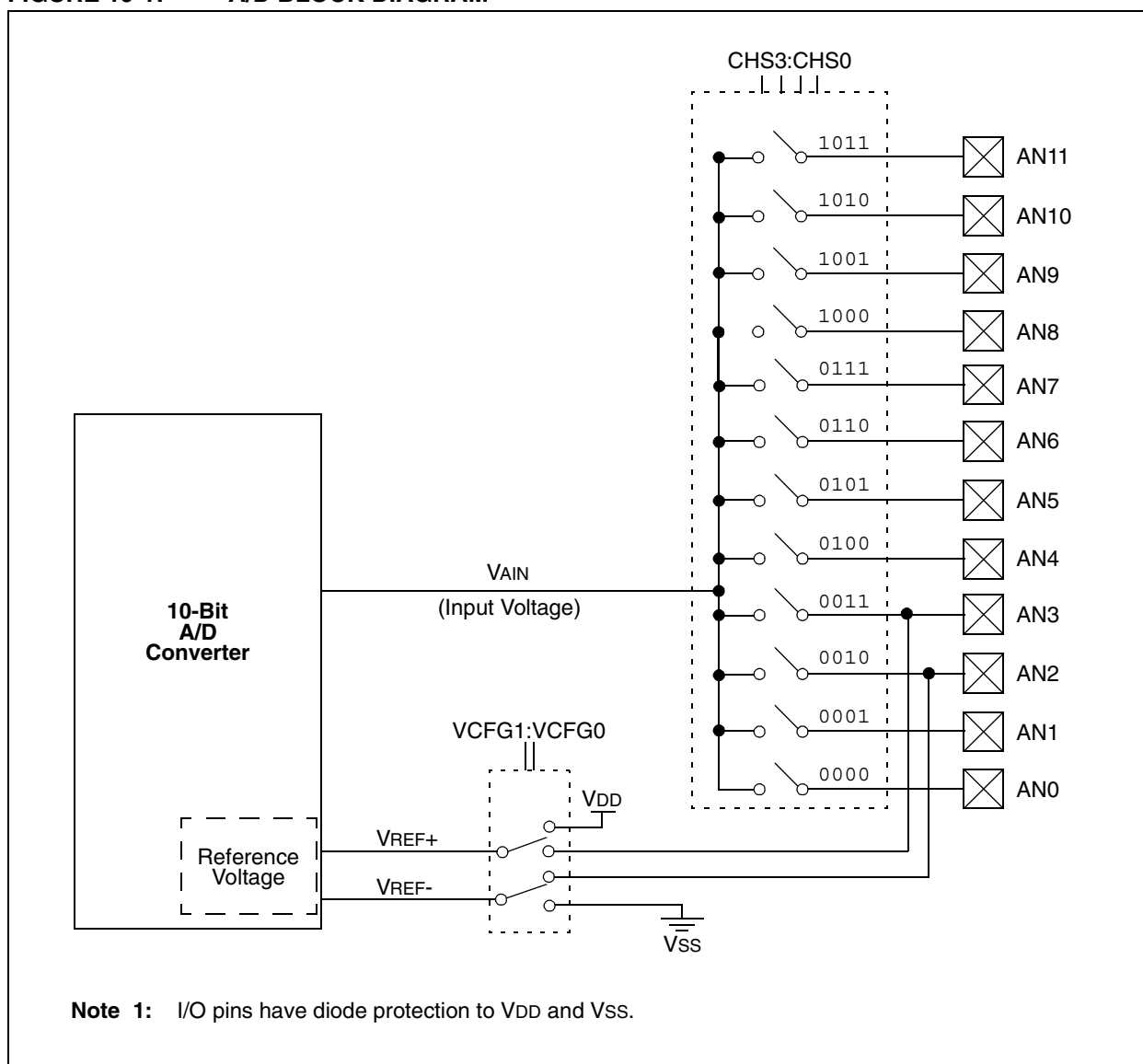
Each port pin associated with the A/D converter can be configured as an analog input or as a digital I/O. The ADRESH and ADRESL registers contain the result of

the A/D conversion. When the A/D conversion is complete, the result is loaded into the ADRESH:ADRESL register pair, the GO/DONE bit (ADCON0<1>) is cleared and the A/D Interrupt Flag bit, ADIF, is set.

A device Reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion in progress is aborted. The value in the ADRESH:ADRESL register pair is not modified for a Power-on Reset. These registers will contain unknown data after a Power-on Reset.

The block diagram of the A/D module is shown in Figure 19-1.

**FIGURE 19-1: A/D BLOCK DIAGRAM<sup>(1)</sup>**



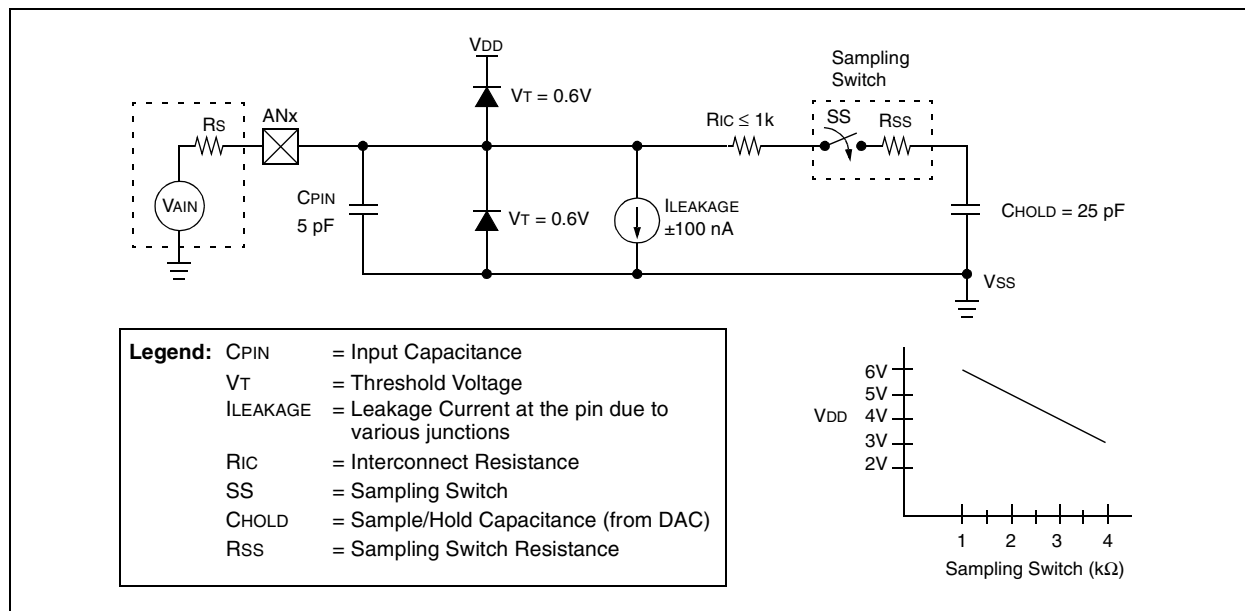


After the A/D module has been configured as desired, the selected channel must be acquired before the conversion is started. The analog input channels must have their corresponding TRIS bits selected as inputs. To determine acquisition time, see **Section 19.1 “A/D Acquisition Requirements”**. After this acquisition time has elapsed, the A/D conversion can be started. An acquisition time can be programmed to occur between setting the GO/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 analog pins, voltage reference and digital I/O (ADCON1)
  - Select A/D input channel (ADCON0)
  - Select A/D acquisition time (ADCON2)
  - Select A/D conversion clock (ADCON2)
  - 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/DONE bit (ADCON0<1>)
5. Wait for A/D conversion to complete, by either:
  - Polling for the GO/DONE bit to be cleared
  - OR
  - Waiting for the A/D interrupt
6. Read A/D Result registers (ADRESH:ADRESL); clear ADIF bit, 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 T<sub>AD</sub>. A minimum wait of 2 T<sub>AD</sub> is required before next acquisition starts.

**FIGURE 19-2: ANALOG INPUT MODEL**



# PIC18F85J11 FAMILY

## 19.1 A/D Acquisition Requirements

For the A/D converter to meet its specified accuracy, the charge holding capacitor (CHOLD) must be allowed to fully charge to the input channel voltage level. The analog input model is shown in Figure 19-2. The source impedance (Rs) and the internal sampling switch (Rss) impedance directly affect the time required to charge the capacitor CHOLD. The sampling switch (Rss) impedance varies over the device voltage (VDD). The source impedance affects the offset voltage at the analog input (due to pin leakage current). **The maximum recommended impedance for analog sources is 2.5 kΩ.** After the analog input channel is selected (changed), the channel must be sampled for at least the minimum acquisition time before starting a conversion.

**Note:** When the conversion is started, the holding capacitor is disconnected from the input pin.

To calculate the minimum acquisition time, Equation 19-1 may be used. This equation assumes that 1/2 LSB error is used (1024 steps for the A/D). The 1/2 LSB error is the maximum error allowed for the A/D to meet its specified resolution.

Equation 19-3 shows the calculation of the minimum required acquisition time, TACQ. This calculation is based on the following application system assumptions:

|                  |   |                    |
|------------------|---|--------------------|
| CHOLD            | = | 25 pF              |
| Rs               | = | 2.5 kΩ             |
| Conversion Error | ≤ | 1/2 LSB            |
| VDD              | = | 3V → Rss = 2 kΩ    |
| Temperature      | = | 85°C (system max.) |

### EQUATION 19-1: A/D ACQUISITION TIME

|      |   |                                                                                     |
|------|---|-------------------------------------------------------------------------------------|
| TACQ | = | Amplifier Settling Time + Holding Capacitor Charging Time + Temperature Coefficient |
|      | = | TAMP + TC + TCOFF                                                                   |

### EQUATION 19-2: A/D MINIMUM CHARGING TIME

|       |   |                                                                                  |
|-------|---|----------------------------------------------------------------------------------|
| VHOLD | = | $(V_{REF} - (V_{REF}/2048)) \cdot (1 - e^{-(T_c/CHOLD)(R_{IC} + R_{SS} + R_s)})$ |
| or    |   |                                                                                  |
| Tc    | = | $-(CHOLD)(R_{IC} + R_{SS} + R_s) \ln(1/2048)$                                    |

### EQUATION 19-3: CALCULATING THE MINIMUM REQUIRED ACQUISITION TIME

|                                                                                             |   |                                                                                                                                        |
|---------------------------------------------------------------------------------------------|---|----------------------------------------------------------------------------------------------------------------------------------------|
| TACQ                                                                                        | = | TAMP + TC + TCOFF                                                                                                                      |
| TAMP                                                                                        | = | 0.2 μs                                                                                                                                 |
| TCOFF                                                                                       | = | $(Temp - 25^{\circ}C)(0.02 \mu s/^{\circ}C)$<br>$(85^{\circ}C - 25^{\circ}C)(0.02 \mu s/^{\circ}C)$<br>1.2 μs                          |
| Temperature coefficient is only required for temperatures > 25°C. Below 25°C, TCOFF = 0 ms. |   |                                                                                                                                        |
| Tc                                                                                          | = | $-(CHOLD)(R_{IC} + R_{SS} + R_s) \ln(1/2048) \mu s$<br>$-(25 pF)(1 k\Omega + 2 k\Omega + 2.5 k\Omega) \ln(0.0004883) \mu s$<br>1.05 μs |
| TACQ                                                                                        | = | 0.2 μs + 1 μs + 1.2 μs<br>2.4 μs                                                                                                       |

## 19.2 Selecting and Configuring Automatic Acquisition Time

The ADCON2 register allows the user to select an acquisition time that occurs each time the GO/DONE bit is set.

When the GO/DONE bit is set, sampling is stopped and a conversion begins. The user is responsible for ensuring the required acquisition time has passed between selecting the desired input channel and setting the GO/DONE bit. This occurs when the ACQT2:ACQT0 bits (ADCON2<5:3>) remain in their Reset state ('000') and is compatible with devices that do not offer programmable acquisition times.

If desired, the ACQT bits can be set to select a programmable acquisition time for the A/D module. When the GO/DONE bit is set, the A/D module continues to sample the input for the selected acquisition time, then automatically begins a conversion. Since the acquisition time is programmed, there may be no need to wait for an acquisition time between selecting a channel and setting the GO/DONE bit.

In either case, when the conversion is completed, the GO/DONE bit is cleared, the ADIF flag is set and the A/D begins sampling the currently selected channel again. If an acquisition time is programmed, there is nothing to indicate if the acquisition time has ended or if the conversion has begun.

## 19.3 Selecting the A/D Conversion Clock

The A/D conversion time per bit is defined as TAD. The A/D conversion requires 11 TAD per 10-bit conversion. The source of the A/D conversion clock is software selectable.

There are seven possible options for TAD:

- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal RC Oscillator

For correct A/D conversions, the A/D conversion clock (TAD) must be as short as possible but greater than the minimum TAD (see parameter 130 in Table 25-24 for more information).

Table 19-1 shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

**TABLE 19-1: TAD vs. DEVICE OPERATING FREQUENCIES**

| AD Clock Source (TAD) |             | Maximum Device Frequency |
|-----------------------|-------------|--------------------------|
| Operation             | ADCS2:ADCS0 |                          |
| 2 TOSC                | 000         | 2.86 MHz                 |
| 4 TOSC                | 100         | 5.71 MHz                 |
| 8 TOSC                | 001         | 11.43 MHz                |
| 16 TOSC               | 101         | 22.86 MHz                |
| 32 TOSC               | 010         | 40.0 MHz                 |
| 64 TOSC               | 110         | 40.0 MHz                 |
| RC <sup>(1)</sup>     | x11         | 1.00 MHz <sup>(2)</sup>  |

**Note 1:** The RC source has a typical TAD time of 4  $\mu$ s.

**2:** For device frequencies above 1 MHz, the device must be in Sleep mode for the entire conversion or the A/D accuracy may be out of specification.

## 19.4 Configuring Analog Port Pins

The ADCON1, TRISA, TRISF and TRISH registers control the operation of the A/D port pins. The port pins needed as analog inputs must have their corresponding TRIS bits set (input). If the TRIS bit is cleared (output), the digital output level (VOH or VOL) will be converted.

The A/D operation is independent of the state of the CHS3:CHS0 bits and the TRIS bits.

**Note 1:** When reading the PORT register, all pins configured as analog input channels will read as cleared (a low level). Pins configured as digital inputs will convert an analog input. Analog levels on a digitally configured input will be accurately converted.

**2:** Analog levels on any pin defined as a digital input may cause the digital input buffer to consume current out of the device's specification limits.

# PIC18F85J11 FAMILY

## 19.5 A/D Conversions

Figure 19-3 shows the operation of the A/D converter after the  $\overline{\text{GO/DONE}}$  bit has been set and the ACQT2:ACQT0 bits are cleared. A conversion is started after the following instruction to allow entry into Sleep mode before the conversion begins.

Figure 19-4 shows the operation of the A/D converter after the  $\overline{\text{GO/DONE}}$  bit has been set, the ACQT2:ACQT0 bits have been set to '010' and a 4 TAD acquisition time has been selected before the conversion starts.

Clearing the  $\overline{\text{GO/DONE}}$  bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. This means the ADRESH:ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH:ADRESL registers).

After the A/D conversion is completed or aborted, a 2 TAD wait is required before the next acquisition can be started. After this wait, acquisition on the selected channel is automatically started.

**Note:** The  $\overline{\text{GO/DONE}}$  bit should **NOT** be set in the same instruction that turns on the A/D.

## 19.6 Use of the CCP2 Trigger

An A/D conversion can be started by the "Special Event Trigger" of the CCP2 module. This requires that the CCP2M3:CCP2M0 bits (CCP2CON<3:0>) be programmed as '1011' and that the A/D module is enabled (ADON bit is set). When the trigger occurs, the  $\overline{\text{GO/DONE}}$  bit will be set, starting the A/D acquisition and conversion, and the Timer1 (or Timer3) counter will be reset to zero. Timer1 (or Timer3) is reset to automatically repeat the A/D acquisition period with minimal software overhead (moving ADRESH/ADRESL to the desired location). The appropriate analog input channel must be selected and the minimum acquisition period is either timed by the user, or an appropriate TACQ time is selected before the Special Event Trigger sets the  $\overline{\text{GO/DONE}}$  bit (starts a conversion).

If the A/D module is not enabled (ADON is cleared), the Special Event Trigger will be ignored by the A/D module but will still reset the Timer1 (or Timer3) counter.

**FIGURE 19-3: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 000, TACQ = 0)**

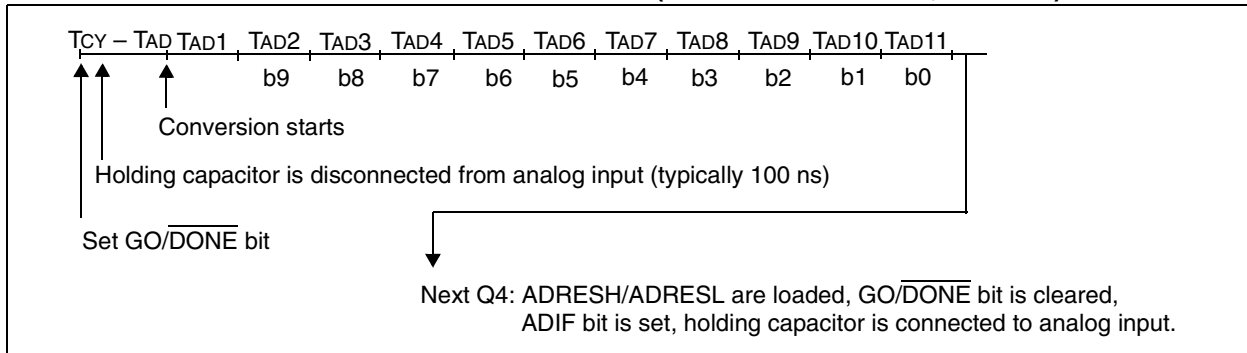
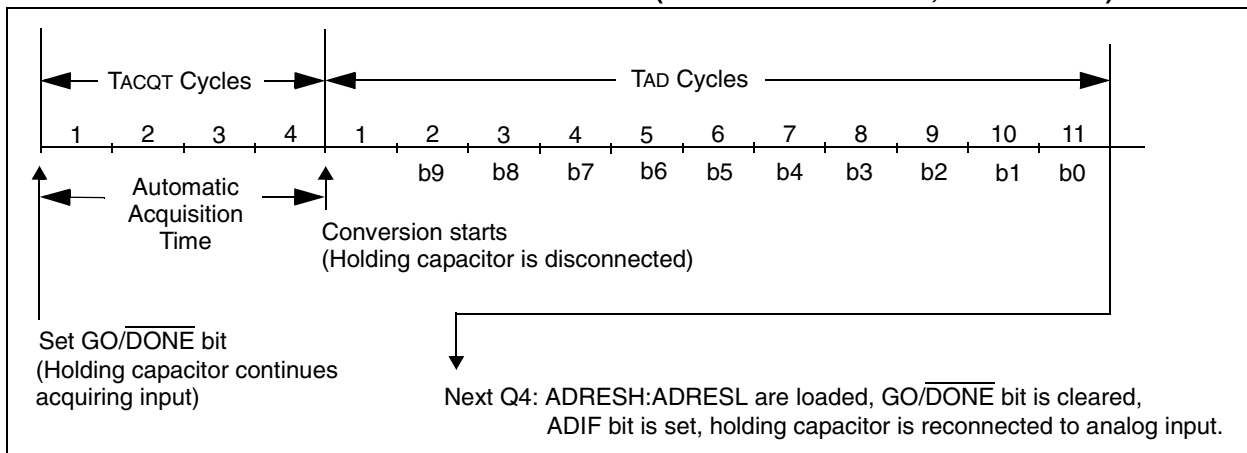


**FIGURE 19-4: A/D CONVERSION TAD CYCLES (ACQT2:ACQT0 = 010, TACQ = 4 TAD)**



## 19.7 A/D Converter Calibration

The A/D converter in the PIC18F85J11 family of devices includes a self-calibration feature which compensates for any offset generated within the module. The calibration process is automated and is initiated by setting the ADCAL bit (ADCON0<7>). The next time the GO/DONE bit is set, the module will perform a “dummy” conversion (that is, with reading none of the input channels) and store the resulting value internally to compensate for offset. Thus, subsequent offsets will be compensated.

The calibration process assumes that the device is in a relatively steady-state operating condition. If A/D calibration is used, it should be performed after each device Reset or if there are other major changes in operating conditions.

## 19.8 Operation in Power-Managed Modes

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

If the A/D is expected to operate while the device is in a power-managed mode, the ACQT2:ACQT0 and ADCS2:ADCS0 bits in ADCON2 should be updated in accordance with the power-managed mode clock that will be used. After the power-managed mode is entered (either of the power-managed Run modes), an A/D acquisition or conversion may be started. Once an acquisition or conversion is started, the device should continue to be clocked by the same power-managed mode clock source until the conversion has been completed. If desired, the device may be placed into the corresponding power-managed Idle mode during the conversion.

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

Operation in the Sleep mode requires the A/D RC clock to be selected. If bits, ACQT2:ACQT0, are set to ‘000’ and a conversion is started, the conversion will be delayed one instruction cycle to allow execution of the SLEEP instruction and entry to Sleep mode. The IDLEN and SCS<1:0> bits in the OSCCON register must have already been cleared prior to starting the conversion.

**TABLE 19-2: SUMMARY OF A/D REGISTERS**

| 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   | 51                   |
| PIR1    | PSPIF                         | ADIF                  | RC1IF  | TX1IF  | SSPIF  | —      | TMR2IF  | TMR1IF | 53                   |
| PIE1    | PSPIE                         | ADIE                  | RC1IE  | TX1IE  | SSPIE  | —      | TMR2IE  | TMR1IE | 53                   |
| IPR1    | PSPIP                         | ADIP                  | RC1IP  | TX1IP  | SSPIP  | —      | TMR2IP  | TMR1IP | 53                   |
| PIR3    | —                             | —                     | RC2IF  | TX2IF  | —      | CCP2IF | CCP1IF  | —      | 53                   |
| PIE3    | —                             | —                     | RC2IE  | TX2IE  | —      | CCP2IE | CCP1IE  | —      | 53                   |
| IPR3    | —                             | —                     | RC2IP  | TX2IP  | —      | CCP2IP | CCP1IP  | —      | 53                   |
| ADRESH  | A/D Result Register High Byte |                       |        |        |        |        |         |        | 53                   |
| ADRESL  | A/D Result Register Low Byte  |                       |        |        |        |        |         |        | 53                   |
| ADCON0  | ADCAL                         | —                     | CHS3   | CHS2   | CHS1   | CHS0   | GO/DONE | ADON   | 53                   |
| ADCON1  | —                             | —                     | VCFG1  | VCFG0  | PCFG3  | PCFG2  | PCFG1   | PCFG0  | 53                   |
| ADCON2  | ADFM                          | —                     | ACQT2  | ACQT1  | ACQT0  | ADCS2  | ADCS1   | ADCS0  | 53                   |
| CCP2CON | —                             | —                     | DC2B1  | DC2B0  | CCP2M3 | CCP2M2 | CCP2M1  | CCP2M0 | 54                   |
| PORTA   | RA7 <sup>(1)</sup>            | RA6 <sup>(1)</sup>    | RA5    | RA4    | RA3    | RA2    | RA1     | RA0    | 54                   |
| TRISA   | TRISA7 <sup>(1)</sup>         | TRISA6 <sup>(1)</sup> | TRISA5 | TRISA4 | TRISA3 | TRISA2 | TRISA1  | TRISA0 | 54                   |
| PORTF   | RF7                           | RF6                   | RF5    | RF4    | RF3    | RF2    | RF1     | —      | 54                   |
| TRISF   | TRISF5                        | TRISF4                | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1  | —      | 54                   |

**Legend:** — = unimplemented, read as ‘0’. Shaded cells are not used for A/D conversion.

**Note 1:** RA6/RA7 and their associated latch and direction bits are configured as port pins only when the internal oscillator is selected as the default clock source (FOSC2 Configuration bit = 0); otherwise, they are disabled and these bits read as ‘0’.

# PIC18F85J11 FAMILY

---

NOTES:

## 20.0 COMPARATOR MODULE

The analog comparator module contains two comparators that can be configured in a variety of ways. The inputs can be selected from the analog inputs multiplexed with pins RF1 through RF6, as well as the on-chip voltage reference (see **Section 21.0 “Comparator Voltage Reference Module”**). The digital outputs (normal or inverted) are available at the pin level and can also be read through the control register.

The CMCON register (Register 20-1) selects the comparator input and output configuration. Block diagrams of the various comparator configurations are shown in Figure 20-1.

### REGISTER 20-1: CMCON: COMPARATOR MODULE CONTROL REGISTER

| R-0   | R-0   | R/W-0 | R/W-0 | R/W-0 | R/W-1 | R/W-1 | R/W-1 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| C2OUT | C1OUT | C2INV | C1INV | CIS   | CM2   | CM1   | CM0   |
| 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 **C2OUT**: Comparator 2 Output bit

When C2INV = 0:

1 = C2 VIN+ > C2 VIN-

0 = C2 VIN+ < C2 VIN-

When C2INV = 1:

1 = C2 VIN+ < C2 VIN-

0 = C2 VIN+ > C2 VIN-

bit 6 **C1OUT**: Comparator 1 Output bit

When C1INV = 0:

1 = C1 VIN+ > C1 VIN-

0 = C1 VIN+ < C1 VIN-

When C1INV = 1:

1 = C1 VIN+ < C1 VIN-

0 = C1 VIN+ > C1 VIN-

bit 5 **C2INV**: Comparator 2 Output Inversion bit

1 = C2 output inverted

0 = C2 output not inverted

bit 4 **C1INV**: Comparator 1 Output Inversion bit

1 = C1 output inverted

0 = C1 output not inverted

bit 3 **CIS**: Comparator Input Switch bit

When CM2:CM0 = 110:

1 = C1 VIN- connects to RF5/AN10/CVREF

C2 VIN- connects to RF3/AN8

0 = C1 VIN- connects to RF6/AN11

C2 VIN- connects to RF4/AN9

bit 2-0 **CM2:CM0**: Comparator Mode bits

Figure 20-1 shows the Comparator modes and the CM2:CM0 bit settings.

# PIC18F85J11 FAMILY

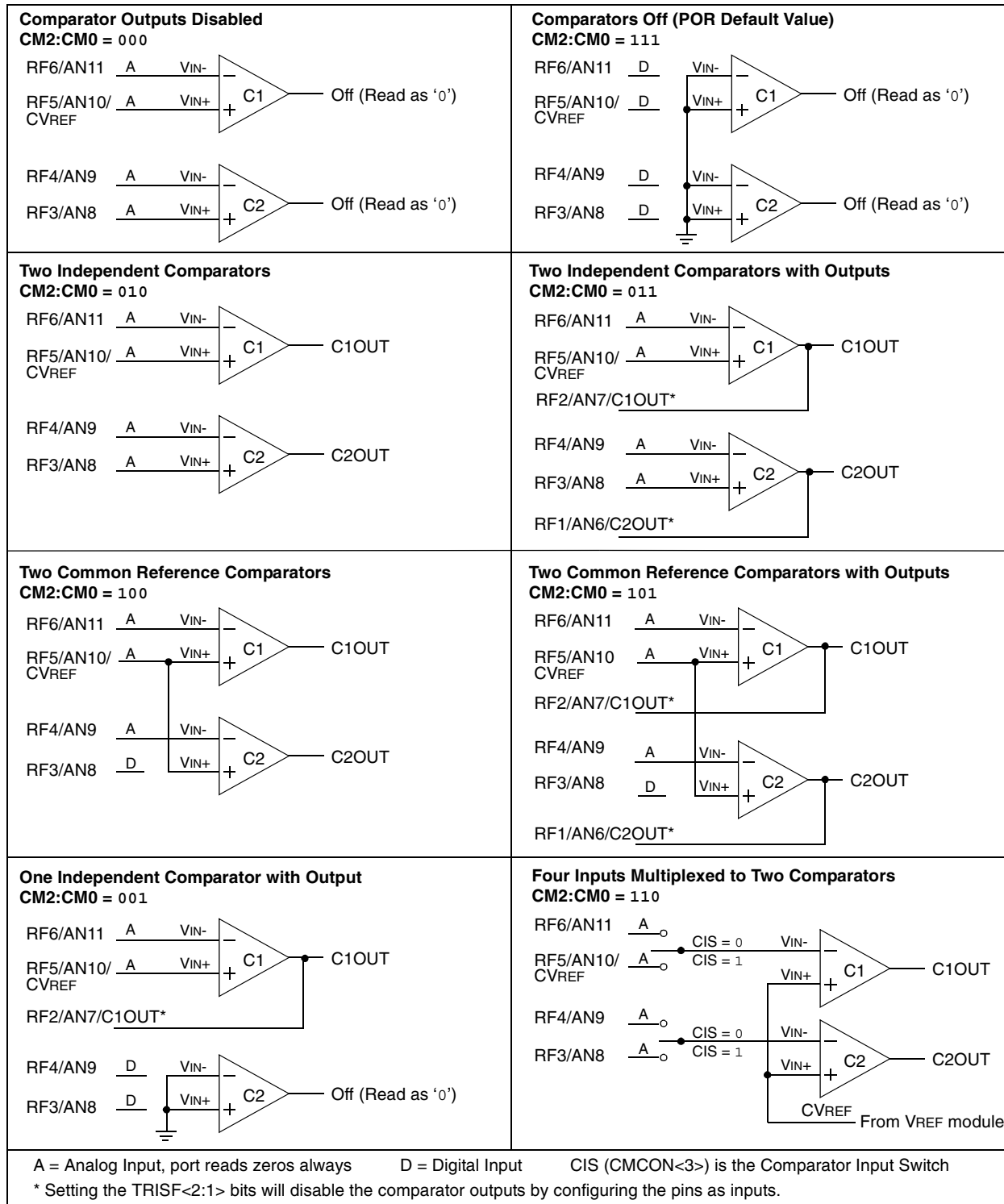
## 20.1 Comparator Configuration

There are eight modes of operation for the comparators, shown in Figure 20-1. Bits CM2:CM0 of the CMCON register are used to select these modes. The TRISF register controls the data direction of the comparator pins for each mode. If the Comparator

mode is changed, the comparator output level may not be valid for the specified mode change delay shown in Section 25.0 “Electrical Characteristics”.

**Note:** Comparator interrupts should be disabled during a Comparator mode change; otherwise, a false interrupt may occur.

**FIGURE 20-1: COMPARATOR I/O OPERATING MODES**





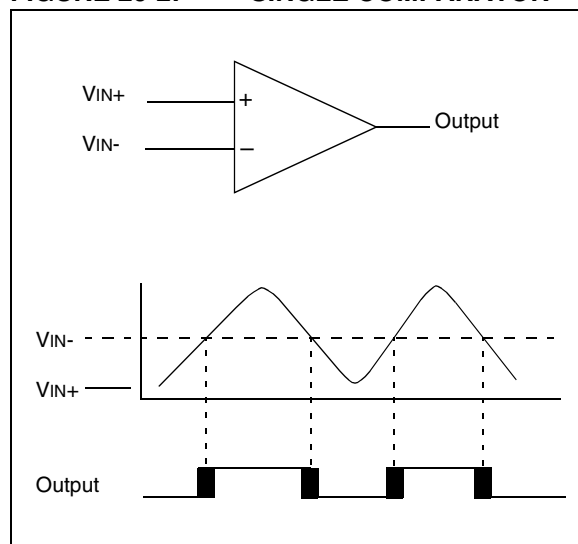
## 20.2 Comparator Operation

A single comparator is shown in Figure 20-2, along with the relationship between the analog input levels and the digital output. When the analog input at  $V_{IN+}$  is less than the analog input  $V_{IN-}$ , the output of the comparator is a digital low level. When the analog input at  $V_{IN+}$  is greater than the analog input  $V_{IN-}$ , the output of the comparator is a digital high level. The shaded areas of the output of the comparator in Figure 20-2 represent the uncertainty due to input offsets and response time.

## 20.3 Comparator Reference

Depending on the comparator operating mode, either an external or internal voltage reference may be used. The analog signal present at  $V_{IN-}$  is compared to the signal at  $V_{IN+}$  and the digital output of the comparator is adjusted accordingly (Figure 20-2).

**FIGURE 20-2: SINGLE COMPARATOR**



### 20.3.1 EXTERNAL REFERENCE SIGNAL

When external voltage references are used, the comparator module can be configured to have the comparators operate from the same or different reference sources. However, threshold detector applications may require the same reference. The reference signal must be between  $V_{SS}$  and  $V_{DD}$  and can be applied to either pin of the comparator(s).

### 20.3.2 INTERNAL REFERENCE SIGNAL

The comparator module also allows the selection of an internally generated voltage reference from the comparator voltage reference module. This module is described in more detail in **Section 21.0 "Comparator Voltage Reference Module"**.

The internal reference is only available in the mode where four inputs are multiplexed to two comparators ( $CM2:CM0 = 110$ ). In this mode, the internal voltage reference is applied to the  $V_{IN+}$  pin of both comparators.

## 20.4 Comparator Response Time

Response time is the minimum time, after selecting a new reference voltage or input source, before the comparator output has a valid level. If the internal reference is changed, the maximum delay of the internal voltage reference must be considered when using the comparator outputs. Otherwise, the maximum delay of the comparators should be used (see **Section 25.0 "Electrical Characteristics"**).

## 20.5 Comparator Outputs

The comparator outputs are read through the CMCON register. These bits are read-only. The comparator outputs may also be directly output to the RF1 and RF2 I/O pins. When enabled, multiplexors in the output path of the RF1 and RF2 pins will switch and the output of each pin will be the unsynchronized output of the comparator. The uncertainty of each of the comparators is related to the input offset voltage and the response time given in the specifications. Figure 20-3 shows the comparator output block diagram.

The TRISF bits will still function as output enables/disables for the RF1 and RF2 pins while in this mode.

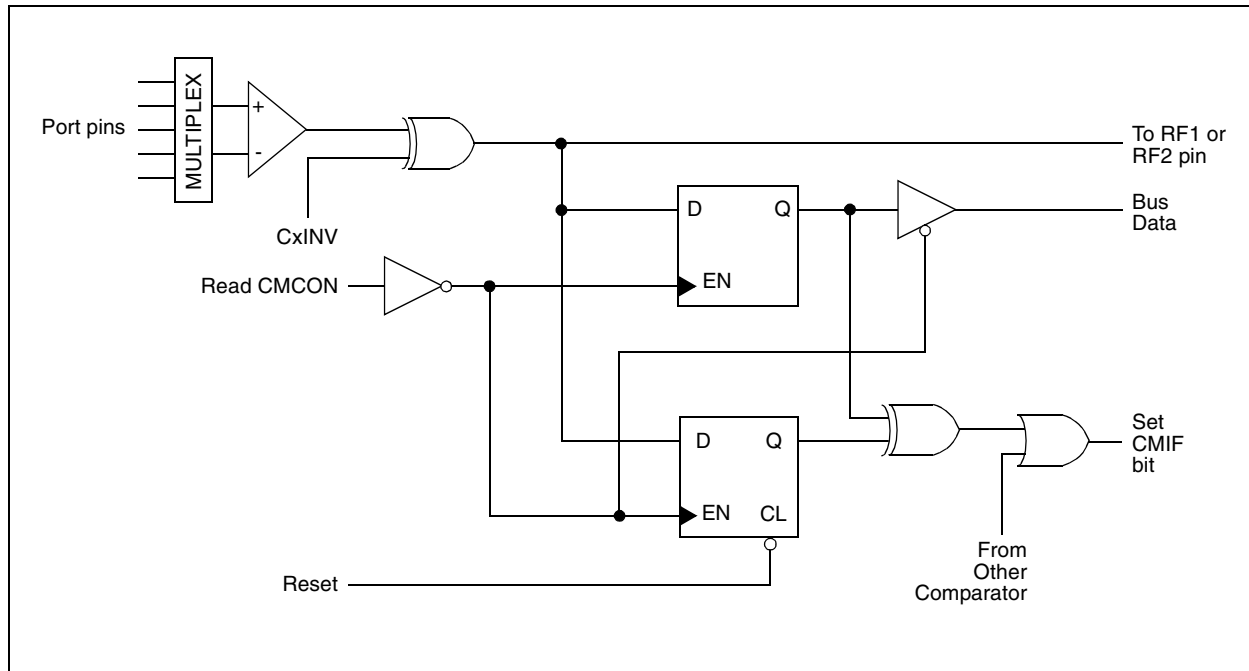
The polarity of the comparator outputs can be changed using the C2INV and C1INV bits ( $CMCON<5:4>$ ).

**Note 1:** When reading the PORT register, all pins configured as analog inputs will read as '0'. Pins configured as digital inputs will convert an analog input according to the Schmitt Trigger input specification.

**2:** Analog levels on any pin defined as a digital input may cause the input buffer to consume more current than is specified.

# PIC18F85J11 FAMILY

**FIGURE 20-3: COMPARATOR OUTPUT BLOCK DIAGRAM**



## 20.6 Comparator Interrupts

The comparator interrupt flag is set whenever there is a change in the output value of either comparator. Software will need to maintain information about the status of the output bits, as read from CMCON<7:6>, to determine the actual change that occurred. The CMIF bit (PIR2<6>) is the Comparator Interrupt Flag. The CMIF bit must be reset by clearing it. Since it is also possible to write a '1' to this register, a simulated interrupt may be initiated.

Both the CMIE bit (PIE2<6>) and the PEIE bit (INTCON<6>) must be set to enable the interrupt. In addition, the GIE bit (INTCON<7>) must also be set. If any of these bits are clear, the interrupt is not enabled, though the CMIF bit will still be set if an interrupt condition occurs.

**Note:** If a change in the CMCON register (C1OUT or C2OUT) should occur when a read operation is being executed (start of the Q2 cycle), then the CMIF (PIR2<6>) interrupt flag may not get set.

The user, in the Interrupt Service Routine, can clear the interrupt in the following manner:

- Any read or write of CMCON will end the mismatch condition.
- Clear flag bit, CMIF.

A mismatch condition will continue to set flag bit, CMIF. Reading CMCON will end the mismatch condition and allow flag bit, CMIF, to be cleared.

## 20.7 Comparator Operation During Sleep

When a comparator is active and the device is placed in Sleep mode, the comparator remains active and the interrupt is functional if enabled. This interrupt will wake-up the device from Sleep mode when enabled. Each operational comparator will consume additional current, as shown in the comparator specifications. To minimize power consumption while in Sleep mode, turn off the comparators (CM2:CM0 = 111) before entering Sleep. If the device wakes up from Sleep, the contents of the CMCON register are not affected.

## 20.8 Effects of a Reset

A device Reset forces the CMCON register to its Reset state, causing the comparator modules to be turned off (CM2:CM0 = 111). However, the input pins (RF3 through RF6) are configured as analog inputs by default on device Reset. The I/O configuration for these pins is determined by the setting of the PCFG3:PCFG0 bits (ADCON1<3:0>). Therefore, device current is minimized when analog inputs are present at Reset time.

## 20.9 Analog Input Connection Considerations

A simplified circuit for an analog input is shown in Figure 20-4. Since the analog pins are connected to a digital output, they have reverse biased diodes to VDD and VSS. The analog input, therefore, must be between VSS and VDD. If the input voltage deviates from this

range by more than 0.6V in either direction, one of the diodes is forward biased and a latch-up condition may occur. A maximum source impedance of 10 kΩ is recommended for the analog sources. Any external component connected to an analog input pin, such as a capacitor or a Zener diode, should have very little leakage current.

**FIGURE 20-4: COMPARATOR ANALOG INPUT MODEL**

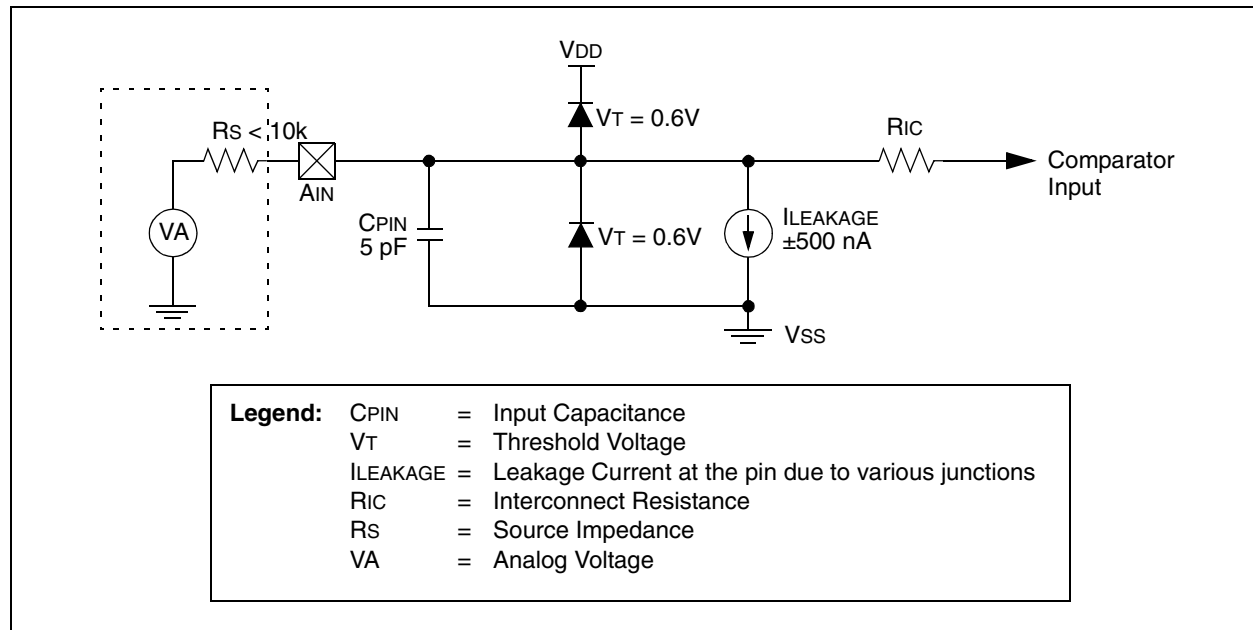


**TABLE 20-1: REGISTERS ASSOCIATED WITH COMPARATOR MODULE**

| 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  | 51                   |
| PIR2   | OSCFIF   | CMIF      | —      | —      | BCLIF  | LVDIF  | TMR3IF | —     | 53                   |
| PIE2   | OSCFIE   | CMIE      | —      | —      | BCLIE  | LVDIE  | TMR3IE | —     | 53                   |
| IPR2   | OSCFIP   | CMIP      | —      | —      | BCLIP  | LVDIP  | TMR3IP | —     | 53                   |
| CMCON  | C2OUT    | C1OUT     | C2INV  | C1INV  | CIS    | CM2    | CM1    | CM0   | 53                   |
| CVRCON | CVREN    | CVROE     | CVRR   | CVRSS  | CVR3   | CVR2   | CVR1   | CVR0  | 53                   |
| PORTF  | RF7      | RF6       | RF5    | RF4    | RF3    | RF2    | RF1    | —     | 54                   |
| LATF   | LATF7    | LATF6     | LATF5  | LATF4  | LATF3  | LATF2  | LATF1  | —     | 54                   |
| TRISF  | TRISF7   | TRISF6    | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1 | —     | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are unused by the comparator module.

# PIC18F85J11 FAMILY

---

NOTES:

## 21.0 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.

A block diagram of the module is shown in Figure 21-1. The resistor ladder is segmented to provide two ranges of CVREF 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 VDD/VSS or an external voltage reference.

### 21.1 Configuring the Comparator Voltage Reference

The comparator voltage reference module is controlled through the CVRCON register (Register 21-1). The comparator voltage reference provides two ranges of output voltage, each with 16 distinct levels.

The range to be used is selected by the CVRR bit (CVRCON<5>). The primary difference between the ranges is the size of the steps selected by the CVREF Selection bits (CVR3:CVR0), with one range offering finer resolution. The equations used to calculate the output of the comparator voltage reference are as follows:

If CVRR = 1:

$$CVREF = ((CVR3:CVR0)/24) \times (CVRSRC)$$

If CVRR = 0:

$$CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) \times (CVRSRC)$$

The comparator reference supply voltage can come from either VDD and VSS, or the external VREF+ and VREF- that are multiplexed with RA2 and RA3. The voltage source is selected by the CVRSS bit (CVRCON<4>).

The settling time of the comparator voltage reference must be considered when changing the CVREF output (see Table 25-3 in **Section 25.0 "Electrical Characteristics"**).

**REGISTER 21-1: CVRCON: COMPARATOR VOLTAGE REFERENCE CONTROL REGISTER**

| R/W-0 | R/W-0                | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|----------------------|-------|-------|-------|-------|-------|-------|
| CVREN | CVROE <sup>(1)</sup> | CVRR  | CVRSS | CVR3  | CVR2  | CVR1  | CVR0  |
| 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

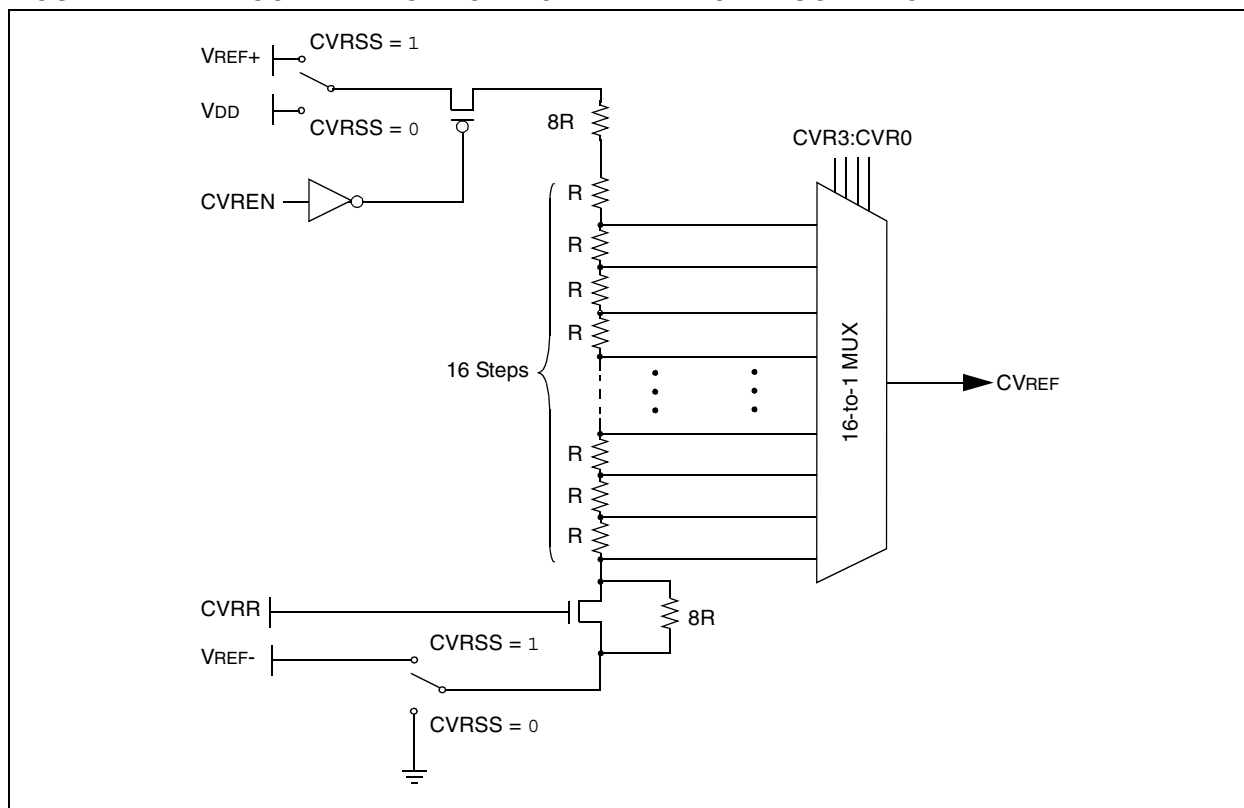
x = Bit is unknown

- bit 7 **CVREN:** Comparator Voltage Reference Enable bit  
 1 = CVREF circuit powered on  
 0 = CVREF circuit powered down
- bit 6 **CVROE:** Comparator VREF Output Enable bit<sup>(1)</sup>  
 1 = CVREF voltage level is also output on the RF5/AN10/CVREF pin  
 0 = CVREF voltage is disconnected from the RF5/AN10/CVREF pin
- bit 5 **CVRR:** Comparator VREF Range Selection bit  
 1 = 0 to 0.667 CVRSRC, with CVRSRC/24 step size (low range)  
 0 = 0.25 CVRSRC to 0.75 CVRSRC, with CVRSRC/32 step size (high range)
- bit 4 **CVRSS:** Comparator VREF Source Selection bit  
 1 = Comparator reference source, CVRSRC = (VREF+) – (VREF-)  
 0 = Comparator reference source, CVRSRC = VDD – VSS
- bit 3-0 **CVR3:CVR0:** Comparator VREF Value Selection bits (0 ≤ (CVR3:CVR0) ≤ 15)  
When CVRR = 1:  
 $CVREF = ((CVR3:CVR0)/24) \times (CVRSRC)$   
When CVRR = 0:  
 $CVREF = (CVRSRC/4) + ((CVR3:CVR0)/32) \times (CVRSRC)$

**Note 1:** CVROE overrides the TRISF<5> bit setting.

# PIC18F85J11 FAMILY

**FIGURE 21-1: COMPARATOR VOLTAGE REFERENCE BLOCK DIAGRAM**



## 21.2 Comparator Voltage Reference Accuracy/Error

The full range of comparator voltage reference cannot be realized due to the construction of the module. The transistors on the top and bottom of the resistor ladder network (Figure 21-1) keep CVREF from approaching the reference source rails. The voltage reference is derived from the reference source; therefore, the CVREF output changes with fluctuations in that source. The tested absolute accuracy of the voltage reference can be found in **Section 25.0 “Electrical Characteristics”**.

## 21.3 Operation During Sleep

When the device wakes up from Sleep through an interrupt or a Watchdog Timer time-out, the contents of the CVRCON register are not affected. To minimize current consumption in Sleep mode, the comparator voltage reference should be disabled.

## 21.4 Effects of a Reset

A device Reset disables the comparator voltage reference by clearing bit, CVREN (CVRCON<7>). This Reset also disconnects the reference from the RA2 pin by clearing bit, CVROE (CVRCON<6>) and selects the high-voltage range by clearing bit, CVRR (CVRCON<5>). The CVR<3:0> value select bits are also cleared.

## 21.5 Connection Considerations

The comparator voltage reference module operates independently of the comparator module. The output of the reference generator may be connected to the RF5 pin if the CVROE bit is set. Enabling the voltage reference output onto RA2 when it is configured as a digital input will increase current consumption. Connecting RF5 as a digital output with CVRSS enabled will also increase current consumption.

The RF5 pin can be used as a simple D/A output with limited drive capability. Due to the limited current drive capability, a buffer must be used on the comparator voltage reference output for external connections to VREF. Figure 21-2 shows an example buffering technique.

FIGURE 21-2: COMPARATOR VOLTAGE REFERENCE OUTPUT BUFFER EXAMPLE

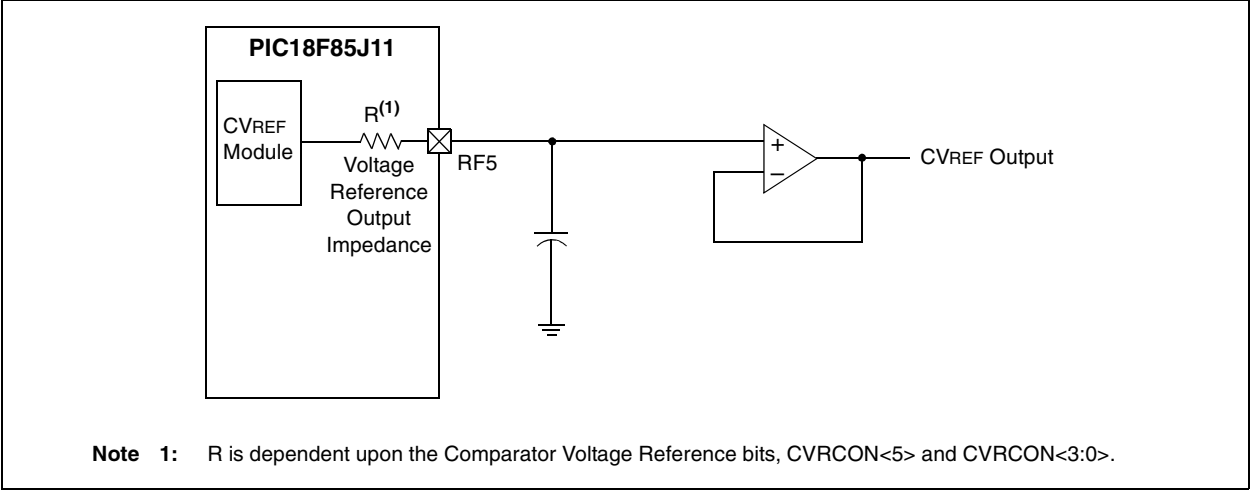


TABLE 21-1: REGISTERS ASSOCIATED WITH COMPARATOR VOLTAGE REFERENCE

| Name   | Bit 7  | Bit 6  | Bit 5  | Bit 4  | Bit 3  | Bit 2  | Bit 1  | Bit 0 | Reset Values on page |
|--------|--------|--------|--------|--------|--------|--------|--------|-------|----------------------|
| CVRCON | CVREN  | CVROE  | CVRR   | CVRSS  | CVR3   | CVR2   | CVR1   | CVR0  | 53                   |
| CMCON  | C2OUT  | C1OUT  | C2INV  | C1INV  | CIS    | CM2    | CM1    | CM0   | 53                   |
| TRISF  | TRISF7 | TRISF6 | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1 | —     | 54                   |

**Legend:** — = unimplemented, read as '0'. Shaded cells are not used with the comparator voltage reference.

# PIC18F85J11 FAMILY

---

NOTES:



## 22.0 SPECIAL FEATURES OF THE CPU

PIC18F85J11 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
- Two-Speed Start-up
- Code Protection
- In-Circuit Serial Programming

The oscillator can be configured for the application depending on frequency, power, accuracy and cost. All of the options are discussed in detail in **Section 2.0 “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 PIC18F85J11 family of devices have a configurable Watchdog Timer 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.

### 22.1 Configuration Bits

The Configuration bits can be programmed (read as ‘0’), or left unprogrammed (read as ‘1’), to select various device configurations. These bits are mapped starting at program memory location 300000h. A complete list is shown in Table 22-2. A detailed explanation of the various bit functions is provided in Register 22-1 through Register 22-6.

### 22.1.1 CONSIDERATIONS FOR CONFIGURING THE PIC18F85J11 FAMILY DEVICES

Devices of the PIC18F85J11 family do not use persistent memory registers to store configuration information. The configuration bytes are implemented as volatile memory which means that configuration data must be programmed each time the device is powered up.

Configuration data is stored in the four words at the top of the on-chip program memory space, known as the Flash Configuration Words. It is stored in program memory in the same order shown in Table 22-2, with CONFIG1L at the lowest address and CONFIG3H at the highest. The data is automatically loaded in the proper Configuration registers during device power-up.

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

The volatile memory cells used for the Configuration bits always reset to ‘1’ on Power-on Resets. For all other types of Reset events, the previously programmed values are maintained and used without reloading from program memory.

The four Most Significant bits of CONFIG1H, CONFIG2H and CONFIG3H in program memory should also be ‘1111’. This makes these Configuration Words appear to be NOP instructions in the remote event that their locations are ever executed by accident. Since Configuration bits are not implemented in the corresponding locations, writing ‘1’s to these locations has no effect on device operation.

To prevent inadvertent configuration changes during code execution, all programmable Configuration bits are write-once. After a bit is initially programmed during a power cycle, it cannot be written to again. Changing a device configuration requires that power to the device be cycled.

**TABLE 22-1: MAPPING OF THE FLASH CONFIGURATION WORDS TO THE CONFIGURATION REGISTERS**

| Configuration Byte | Code Space Address | Configuration Register Address |
|--------------------|--------------------|--------------------------------|
| CONFIG1L           | XXXF8h             | 300000h                        |
| CONFIG1H           | XXXF9h             | 300001h                        |
| CONFIG2L           | XXXFAh             | 300002h                        |
| CONFIG2H           | XXXFBh             | 300003h                        |
| CONFIG3L           | XXXFCh             | 300004h                        |
| CONFIG3H           | XXXFDh             | 300005h                        |

# PIC18F85J11 FAMILY

**TABLE 22-2: CONFIGURATION BITS AND DEVICE IDs**

| File Name                       | Bit 7            | Bit 6            | Bit 5            | Bit 4            | Bit 3            | Bit 2  | Bit 1  | Bit 0  | Default/<br>Unprogrammed<br>Value <sup>(1)</sup> |
|---------------------------------|------------------|------------------|------------------|------------------|------------------|--------|--------|--------|--------------------------------------------------|
| 300000h CONFIG1L                | DEBUG            | XINST            | STVREN           | —                | —                | —      | —      | WDTEN  | 111- ---1                                        |
| 300001h CONFIG1H                | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(3)</sup> | CP0    | —      | —      | ---- 01--                                        |
| 300002h CONFIG2L                | IESO             | FCMEN            | —                | —                | —                | FOSC2  | FOSC1  | FOSC0  | 11-- -111                                        |
| 300003h CONFIG2H                | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | WDTPS3           | WDTPS2 | WDTPS1 | WDTPS0 | ---- 1111                                        |
| 300004h CONFIG3L <sup>(4)</sup> | WAIT             | BW               | EMB1             | EMB0             | EASHFT           | —      | —      | —      | 1111 1---                                        |
| 300005h CONFIG3H                | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | — <sup>(2)</sup> | —                | —      | —      | CCP2MX | ---- ---1                                        |
| 3FFFFEh DEVID1                  | DEV2             | DEV1             | DEV0             | REV4             | REV3             | REV2   | REV1   | REV0   | xxxx xxxx <sup>(5)</sup>                         |
| 3FFFFFh DEVID2                  | DEV10            | DEV9             | DEV8             | DEV7             | DEV6             | DEV5   | DEV4   | DEV3   | 0000 10x1 <sup>(5)</sup>                         |

**Legend:** x = unknown, — = unimplemented. Shaded cells are unimplemented, read as '0'.

**Note 1:** Values reflect the unprogrammed state as received from the factory and following Power-on Resets. In all other Reset states, the configuration bytes maintain their previously programmed states.

**2:** The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

**3:** This bit should always be maintained as '0'.

**4:** CONFIG3L is implemented in 80-pin devices only.

**5:** See Register 22-7 and Register 22-8 for DEVID values. These registers are read-only and cannot be programmed by the user.

# PIC18F85J11 FAMILY

## REGISTER 22-1: CONFIG1L: CONFIGURATION REGISTER 1 LOW (BYTE ADDRESS 300000h)

| R/WO-1 | R/WO-1 | R/WO-1 | U-0 | U-0 | U-0 | U-0 | R/WO-1 |
|--------|--------|--------|-----|-----|-----|-----|--------|
| DEBUG  | XINST  | STVREN | —   | —   | —   | —   | WDTEN  |
| bit 7  |        |        |     |     |     |     | bit 0  |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

- bit 7 **DEBUG:** Background Debugger Enable bit  
 1 = Background debugger disabled; RB6 and RB7 configured as general purpose I/O pins  
 0 = Background debugger enabled; RB6 and RB7 are dedicated to in-circuit debug
- bit 6 **XINST:** Extended Instruction Set Enable bit  
 1 = Instruction set extension and Indexed Addressing mode enabled  
 0 = Instruction set extension and Indexed Addressing mode disabled (Legacy mode)
- bit 5 **STVREN:** Stack Overflow/Underflow Reset Enable bit  
 1 = Reset on stack overflow/underflow enabled  
 0 = Reset on stack overflow/underflow disabled
- bit 4-1 **Unimplemented:** Read as '0'
- bit 0 **WDTEN:** Watchdog Timer Enable bit  
 1 = WDT enabled  
 0 = WDT disabled (control is placed on SWDTEN bit)

## REGISTER 22-2: CONFIG1H: CONFIGURATION REGISTER 1 HIGH (BYTE ADDRESS 300001h)

| U-0   | U-0  | U-0  | U-0  | U-0  | R/WO-1 | U-0 | U-0   |
|-------|------|------|------|------|--------|-----|-------|
| —(1)  | —(1) | —(1) | —(1) | —(2) | CP0    | —   | —     |
| bit 7 |      |      |      |      |        |     | bit 0 |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

- bit 7-4 **Unimplemented:** Read as '1' <sup>(1)</sup>
- bit 3 **Unimplemented:** Read as '0' <sup>(2)</sup>
- bit 2 **CP0:** Code Protection bit  
 1 = Program memory is not code-protected  
 0 = Program memory is code-protected
- bit 1-0 **Unimplemented:** Read as '0'

**Note 1:** The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

**2:** This bit should always be maintained as '0'.

# PIC18F85J11 FAMILY

## REGISTER 22-3: CONFIG2L: CONFIGURATION REGISTER 2 LOW (BYTE ADDRESS 300002h)

| R/WO-1 | R/WO-1 | U-0 | U-0 | U-0 | R/WO-1 | R/WO-1 | R/WO-1 |
|--------|--------|-----|-----|-----|--------|--------|--------|
| IESO   | FCMEN  | —   | —   | —   | FOSC2  | FOSC1  | FOSC0  |
| bit 7  |        |     |     |     |        |        | bit 0  |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

bit 7 **IESO:** Two-Speed Start-up (Internal/External Oscillator Switchover) Control bit

1 = Two-Speed Start-up enabled

0 = Two-Speed Start-up disabled

bit 6 **FCMEN:** Fail-Safe Clock Monitor Enable bit

1 = Fail-Safe Clock Monitor enabled

0 = Fail-Safe Clock Monitor disabled

bit 5-3 **Unimplemented:** Read as '0'

bit 2-0 **FOSC2:FOSC0:** Oscillator Selection bits

111 = OSC1/OSC2 as primary; EC oscillator with CLKO function and software controlled PLL (ECPLL)

110 = OSC1/OSC2 as primary; EC oscillator with CLKO function (EC)

101 = OSC1/OSC2 as primary; HS oscillator with software controlled PLL (HSPLL)

100 = OSC1/OSC2 as primary; HS oscillator (HS)

011 = INTOSC with CLKO as primary; port function on RA7; EC oscillator with CLKO function and software controlled PLL (ECPLL)

010 = INTOSC with CLKO as primary; port function on RA7; EC oscillator with CLKO function

001 = INTOSC as primary with port function on RA6/RA7; HS oscillator with software controlled PLL (HSPLL)

000 = INTOSC as primary with port function on RA6/RA7; HS oscillator (HS)

# PIC18F85J11 FAMILY

## REGISTER 22-4: CONFIG2H: CONFIGURATION REGISTER 2 HIGH (BYTE ADDRESS 300003h)

| U-0   | U-0  | U-0  | U-0  | R/WO-1 | R/WO-1 | R/WO-1 | R/WO-1 |
|-------|------|------|------|--------|--------|--------|--------|
| —(1)  | —(1) | —(1) | —(1) | WDTPS3 | WDTPS2 | WDTPS1 | WDTPS0 |
| bit 7 |      |      |      | bit 0  |        |        |        |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

bit 7-4      **Unimplemented:** Read as '1' <sup>(1)</sup>

bit 3-0      **WDTPS3:WDTPS0:** Watchdog Timer Postscale Select bits

1111 = 1:32,768

1110 = 1:16,384

1101 = 1:8,192

1100 = 1:4,096

1011 = 1:2,048

1010 = 1:1,024

1001 = 1:512

1000 = 1:256

0111 = 1:128

0110 = 1:64

0101 = 1:32

0100 = 1:16

0011 = 1:8

0010 = 1:4

0001 = 1:2

0000 = 1:1

**Note 1:** The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

# PIC18F85J11 FAMILY

## REGISTER 22-5: CONFIG3L: CONFIGURATION REGISTER 3 LOW (BYTE ADDRESS 300004h)<sup>(1)</sup>

|        |        |        |        |        |     |     |       |
|--------|--------|--------|--------|--------|-----|-----|-------|
| R/WO-1 | R/WO-1 | R/WO-1 | R/WO-1 | R/WO-1 | U-0 | U-0 | U-0   |
| WAIT   | BW     | EMB1   | EMB0   | EASHFT | —   | —   | —     |
| bit 7  |        |        |        |        |     |     | bit 0 |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

bit 7 **WAIT:** External Bus Wait Enable bit

1 = Wait selections from MEMCON.WAIT<1:0> are unavailable and the device will not wait

0 = Wait programmed by MEMCON.WAIT<1:0>

bit 6 **BW:** Data Bus Width Select bit

1 = 16-Bit External Bus mode

0 = 8-Bit External Bus mode

bit 5-4 **EMB1:EMB0:** External Memory Bus Configuration bits

00 = Extended Microcontroller mode – 20-Bit Address mode

01 = Extended Microcontroller mode – 16-Bit Address mode

10 = Extended Microcontroller mode – 12-Bit Address mode

11 = Microcontroller mode – external bus disabled

bit 3 **EASHFT:** External Address Bus Shift Enable bit

1 = Address shifting enabled – external address bus is shifted to start at 000000h

0 = Address shifting disabled – external address bus reflects the PC value

bit 2-0 **Unimplemented:** Read as '0'

**Note 1:** CONFIG3L and its associated bits are implemented only in 80-pin devices.

## REGISTER 22-6: CONFIG3H: CONFIGURATION REGISTER 3 HIGH (BYTE ADDRESS 300005h)

|                  |                  |                  |                  |     |     |     |        |
|------------------|------------------|------------------|------------------|-----|-----|-----|--------|
| U-0              | U-0              | U-0              | U-0              | U-0 | U-0 | U-0 | R/WO-1 |
| — <sup>(1)</sup> | — <sup>(1)</sup> | — <sup>(1)</sup> | — <sup>(1)</sup> | —   | —   | —   | CCP2MX |
| bit 7            |                  |                  |                  |     |     |     | bit 0  |

### Legend:

R = Readable bit

WO = Write-Once bit

U = Unimplemented bit, read as '0'

-n = Value when device is unprogrammed

'1' = Bit is set

'0' = Bit is cleared

bit 7-1 **Unimplemented:** Read as '1'<sup>(1)</sup>

bit 0 **CCP2MX:** CCP2 MUX bit

1 = CCP2 is multiplexed with RC1

0 = CCP2 is multiplexed with RE7 in Microcontroller mode (all devices) or with RB3 in Extended Microcontroller mode (80-pin devices only)

**Note 1:** The value of these bits in program memory should always be '1'. This ensures that the location is executed as a NOP if it is accidentally executed.

# PIC18F85J11 FAMILY

## REGISTER 22-7: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F85J11 FAMILY DEVICES

|       |      |      |      |      |      |      |       |
|-------|------|------|------|------|------|------|-------|
| R     | R    | R    | R    | R    | R    | R    | R     |
| DEV2  | DEV1 | DEV0 | REV4 | REV3 | REV2 | REV1 | REV0  |
| bit 7 |      |      |      |      |      |      | bit 0 |

### Legend:

R = Read-only bit

bit 7-5 **DEV2:DEV0:** Device ID bits

111 = PIC18F85J11

101 = PIC18F84J11

100 = PIC18F83J11

011 = PIC18F65J11

001 = PIC18F64J11

000 = PIC18F63J11

bit 4-0 **REV4:REV0:** Revision ID bits

These bits are used to indicate the device revision.

## REGISTER 22-8: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F85J11 FAMILY DEVICES

|                      |                     |                     |                     |                     |                     |                     |                     |
|----------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| R                    | R                   | R                   | R                   | R                   | R                   | R                   | R                   |
| DEV10 <sup>(1)</sup> | DEV9 <sup>(1)</sup> | DEV8 <sup>(1)</sup> | DEV7 <sup>(1)</sup> | DEV6 <sup>(1)</sup> | DEV5 <sup>(1)</sup> | DEV4 <sup>(1)</sup> | DEV3 <sup>(1)</sup> |
| bit 7                |                     |                     |                     |                     |                     |                     | bit 0               |

### Legend:

R = Read-only bit

bit 7-0 **DEV10:DEV3:** Device ID bits<sup>(1)</sup>

These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.

0011 1001 = PIC18F6XJ11/8XJ11 devices

**Note 1:** The values for DEV10:DEV3 may be shared with other device families. The specific device is always identified by using the entire DEV10:DEV0 bit sequence.

# PIC18F85J11 FAMILY

## 22.2 Watchdog Timer (WDT)

For PIC18F85J11 family devices, the WDT is driven by the INTRC oscillator. When the WDT is enabled, the clock source is also enabled. The nominal WDT period is 4 ms and has the same stability as the INTRC oscillator.

The 4 ms period of the WDT is multiplied by a 16-bit postscaler. Any output of the WDT postscaler is selected by a multiplexor, controlled by the WDTPS bits in Configuration Register 2H. Available periods range from 4 ms to 131.072 seconds (2.18 minutes). The WDT and postscaler are cleared whenever a **SLEEP** or **CLRWDT** instruction is executed, or a clock failure (primary or Timer1 oscillator) has occurred.

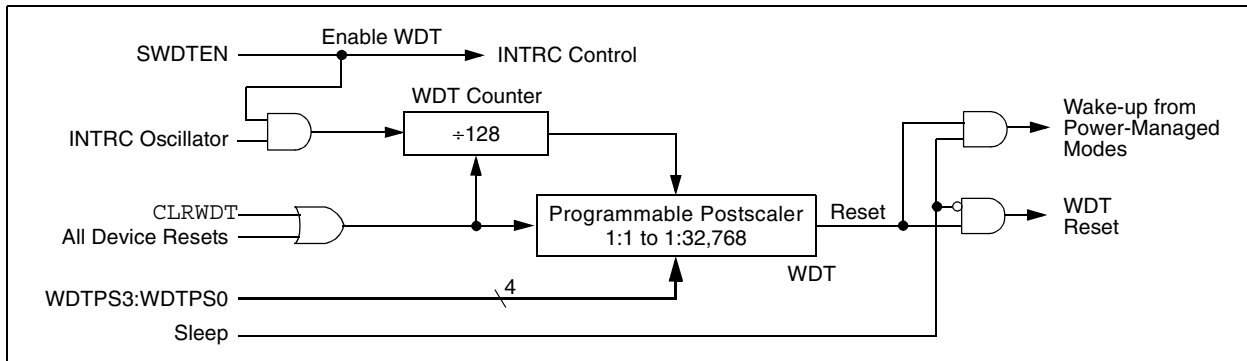
**Note 1:** The **CLRWDT** and **SLEEP** instructions clear the WDT and postscaler counts when executed.

**2:** When a **CLRWDT** instruction is executed, the postscaler count will be cleared.

### 22.2.1 CONTROL REGISTER

The WDTCON register (Register 22-9) is a readable and writable register. The SWDTEN bit enables or disables WDT operation. This allows software to override the WDTEN Configuration bit and enable the WDT only if it has been disabled by the Configuration bit.

**FIGURE 22-1: WDT BLOCK DIAGRAM**



**REGISTER 22-9: WDTCON: WATCHDOG TIMER CONTROL REGISTER**

| R/W-0                 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0                 |
|-----------------------|-----|-----|-----|-----|-----|-----|-----------------------|
| REGSLP <sup>(1)</sup> | —   | —   | —   | —   | —   | —   | SWDTEN <sup>(2)</sup> |
| 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 **REGSLP:** Voltage Regulator Low-Power Operation Enable bit

1 = On-chip regulator enters low-power operation when device enters Sleep mode

0 = On-chip regulator continues to operate normally in Sleep mode

bit 6-1 **Unimplemented:** Read as '0'

bit 0 **SWDTEN:** Software Controlled Watchdog Timer Enable bit<sup>(1)</sup>

1 = Watchdog Timer is on

0 = Watchdog Timer is off

**Note 1:** The REGSLP bit is automatically cleared when a Low-Voltage Detect condition occurs.

**2:** This bit has no effect if the Configuration bit, WDTEN, is enabled.

**TABLE 22-3: SUMMARY OF WATCHDOG TIMER REGISTERS**

| Name   | Bit 7  | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0  | Reset Values on Page |
|--------|--------|-------|-------|-------|-------|-------|-------|--------|----------------------|
| RCON   | IPEN   | —     | CM    | RI    | TO    | PD    | POR   | BOR    | 52                   |
| WDTCON | REGSLP | —     | —     | —     | —     | —     | —     | SWDTEN | 52                   |

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



## 22.3 On-Chip Voltage Regulator

All of the PIC18F85J11 family devices power their core digital logic at a nominal 2.5V. For designs that are required to operate at a higher typical voltage, such as 3.3V, all devices in the PIC18F85J11 family incorporate an on-chip regulator that allows the device to run its core logic from VDD.

The regulator is controlled by the ENVREG pin. Tying VDD to the pin enables the regulator, which in turn, provides power to the core from the other VDD pins. When the regulator is enabled, a low-ESR filter capacitor must be connected to the VDDCORE/VCAP pin (Figure 22-2). This helps to maintain the stability of the regulator. The recommended value for the filter capacitor is provided in **Section 25.3 “DC Characteristics: PIC18F85J11 Family (Industrial)”**.

If ENVREG is tied to VSS, the regulator is disabled. In this case, separate power for the core logic at a nominal 2.5V must be supplied to the device on the VDDCORE/VCAP pin to run the I/O pins at higher voltage levels, typically 3.3V. Alternatively, the VDDCORE/VCAP and VDD pins can be tied together to operate at a lower nominal voltage. Refer to Figure 22-2 for possible configurations.

### 22.3.1 VOLTAGE REGULATION AND LOW-VOLTAGE DETECTION

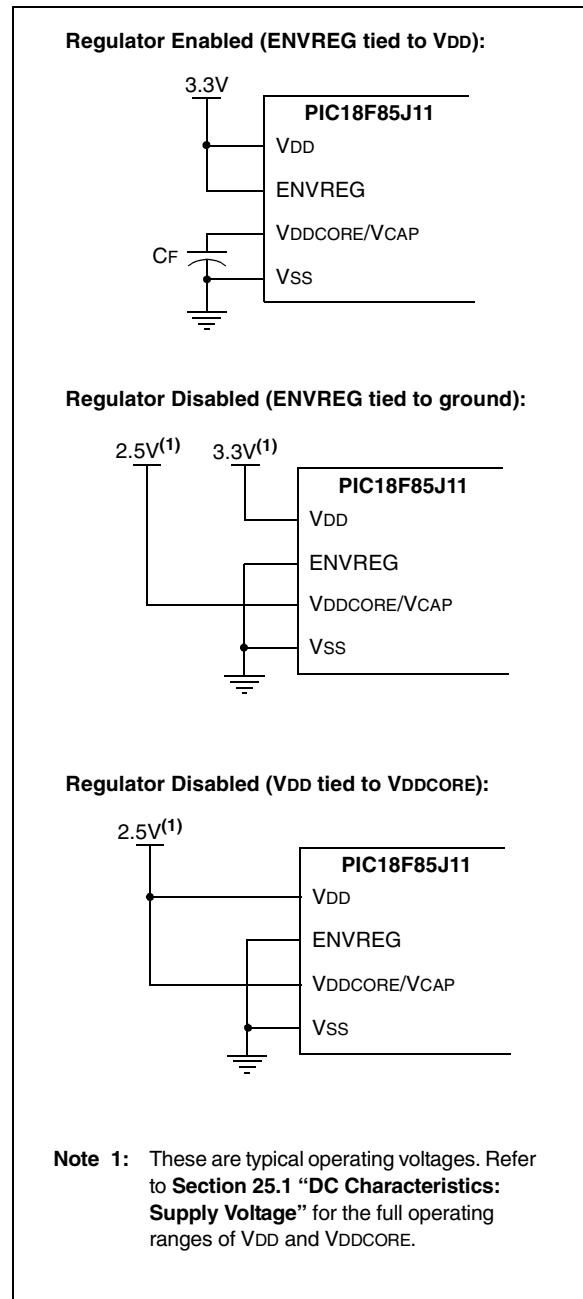
When it is enabled, the on-chip regulator provides a constant voltage of 2.5V nominal to the digital core logic. The regulator can provide this level from a VDD of about 2.5V, all the way up to the device's VDDMAX. It does not have the capability to boost VDD levels below 2.5V.

In order to prevent “brown-out” conditions, the regulator enters Tracking mode when the voltage drops too low for the regulator. In Tracking mode, the regulator output follows VDD, with a typical voltage drop of 100 mV.

The on-chip regulator includes a simple Low-Voltage Detect (LVD) circuit. If VDD drops too low to maintain approximately 2.45V on VDDCORE, the circuit sets the Low-Voltage Detect Interrupt Flag, LVDIF (PIR2<2>), and clears the REGSLP (WDTCON<7>) bit if it was set.

This can be used to generate an interrupt and put the application into a low-power operational mode or to trigger an orderly shutdown. Low-Voltage Detection is only available when the regulator is enabled.

**FIGURE 22-2: CONNECTIONS FOR THE ON-CHIP REGULATOR**



# PIC18F85J11 FAMILY

---

## 22.3.2 ON-CHIP REGULATOR AND BOR

When the on-chip regulator is enabled, PIC18F85J11 family devices also have a simple Brown-out Reset capability. If the voltage supplied to the regulator falls to a level that is inadequate to maintain a regulated output for full-speed operation, the regulator Reset circuitry will generate a Brown-out Reset. This event is captured by the BOR flag bit (RCON<0>).

The operation of the Brown-out Reset is described in more detail in **Section 4.4 “Brown-out Reset (BOR)”** and **Section 4.4.1 “Detecting BOR”**.

## 22.3.3 POWER-UP REQUIREMENTS

The on-chip regulator is designed to meet the power-up requirements for the device. If the application does not use the regulator, then strict power-up conditions must be adhered to. While powering up, VDDCORE must never exceed VDD by 0.3 volts.

## 22.3.4 OPERATION IN SLEEP MODE

When enabled, the on-chip regulator always consumes a small incremental amount of current over IDD. This includes when the device is in Sleep mode, even though the core digital logic does not require power. To provide additional savings in applications where power resources are critical, the regulator can be configured to automatically disable itself whenever the device goes into Sleep mode. This feature is controlled by the REGSLP bit (WDTCON<7>). Setting this bit disables the regulator in Sleep mode and reduces its current consumption to a minimum.

Substantial Sleep-mode power savings can be obtained by setting the REGSLP bit, but this will increase device wake-up time to ensure the regulator has enough time to stabilize.

The REGSLP bit is cleared automatically by hardware when a Low-Voltage Detect condition occurs. The REGSLP bit can be set again in software, which would keep the voltage regulator in Low-Power mode. This is not recommended, however, if any write operations to the Flash will be performed.

## 22.4 Two-Speed Start-up

The Two-Speed Start-up feature helps to minimize the latency period, from oscillator start-up to code execution, by allowing the microcontroller to use the INTRC oscillator as a clock source until the primary clock source is available. It is enabled by setting the IESO Configuration bit.

Two-Speed Start-up should be enabled only if the primary oscillator mode is HS or HSPLL (Crystal-Based) modes. Since the EC and ECPLL modes do not require an Oscillator Start-up Timer delay, Two-Speed Start-up should be disabled.

When enabled, Resets and wake-ups from Sleep mode cause the device to configure itself to run from the internal oscillator block as the clock source, following the time-out of the Power-up Timer, after a Power-on Reset is enabled. This allows almost immediate code execution while the primary oscillator starts and the OST is running. Once the OST times out, the device automatically switches to PRI\_RUN mode.

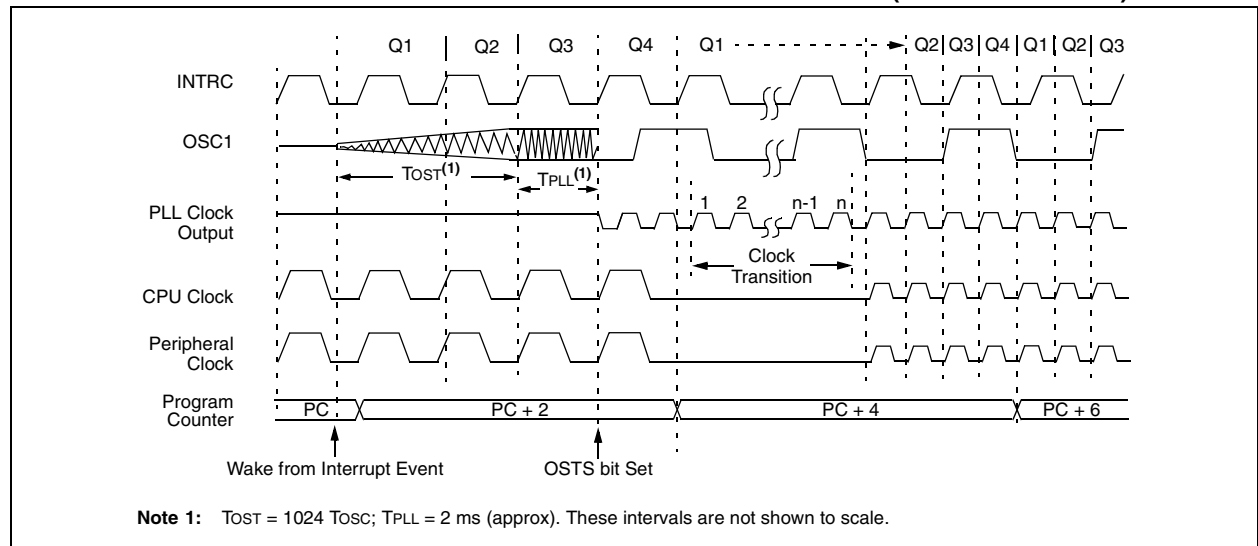
In all other power-managed modes, Two-Speed Start-up is not used. The device will be clocked by the currently selected clock source until the primary clock source becomes available. The setting of the IESO bit is ignored.

### 22.4.1 SPECIAL CONSIDERATIONS FOR USING TWO-SPEED START-UP

While using the INTRC oscillator in Two-Speed Start-up, the device still obeys the normal command sequences for entering power-managed modes, including serial `SLEEP` instructions (refer to **Section 3.1.4 “Multiple Sleep Commands”**). In practice, this means that user code can change the `SCS1:SCS0` bit settings or issue `SLEEP` instructions before the OST times out. This would allow an application to briefly wake-up, perform routine “housekeeping” tasks and return to Sleep before the device starts to operate from the primary oscillator.

User code can also check if the primary clock source is currently providing the device clocking by checking the status of the OSTS bit (`OSCCON<3>`). If the bit is set, the primary oscillator is providing the clock. Otherwise, the internal oscillator block is providing the clock during wake-up from Reset or Sleep mode.

**FIGURE 22-3: TIMING TRANSITION FOR TWO-SPEED START-UP (INTRC TO HSPLL)**



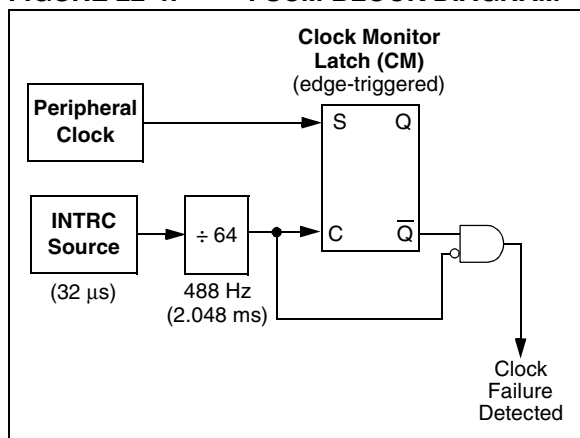
# PIC18F85J11 FAMILY

## 22.5 Fail-Safe Clock Monitor

The Fail-Safe Clock Monitor (FSCM) allows the microcontroller to continue operation in the event of an external oscillator failure by automatically switching the device clock to the internal oscillator block. The FSCM function is enabled by setting the FCMEN Configuration bit.

When FSCM is enabled, the INTRC oscillator runs at all times to monitor clocks to peripherals and provides a backup clock in the event of a clock failure. Clock monitoring (shown in Figure 22-4) is accomplished by creating a sample clock signal which is the INTRC output divided by 64. This allows ample time between FSCM sample clocks for a peripheral clock edge to occur. The peripheral device clock and the sample clock are presented as inputs to the Clock Monitor latch (CM). The CM is set on the falling edge of the device clock source but cleared on the rising edge of the sample clock.

**FIGURE 22-4: FSCM BLOCK DIAGRAM**



Clock failure is tested for on the falling edge of the sample clock. If a sample clock falling edge occurs while CM is still set, a clock failure has been detected (Figure 22-5). This causes the following:

- the FSCM generates an oscillator fail interrupt by setting bit, OSCFIF (PIR2<7>);
- the device clock source is switched to the internal oscillator block (OSCCON is not updated to show the current clock source – this is the fail-safe condition); and
- the WDT is reset.

During switchover, the postscaler frequency from the internal oscillator block may not be sufficiently stable for timing sensitive applications. In these cases, it may be desirable to select another clock configuration and enter an alternate power-managed mode. This can be done to attempt a partial recovery or execute a controlled shutdown. See **Section 3.1.4 “Multiple Sleep Commands”** and **Section 22.4.1 “Special Considerations for Using Two-Speed Start-up”** for more details.

The FSCM will detect failures of the primary or secondary clock sources only. If the internal oscillator block fails, no failure would be detected, nor would any action be possible.

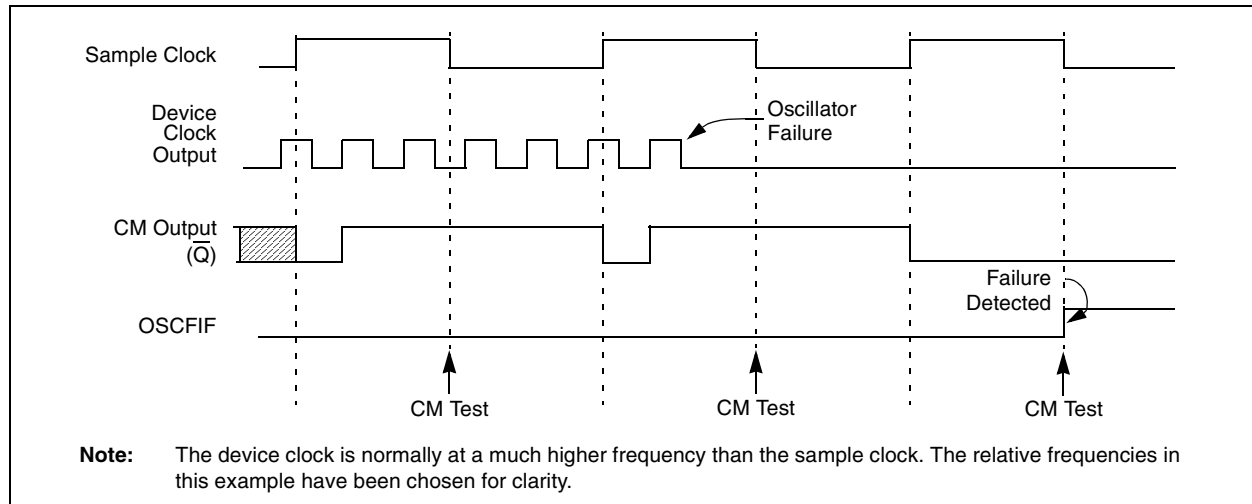
### 22.5.1 FSCM AND THE WATCHDOG TIMER

Both the FSCM and the WDT are clocked by the INTRC oscillator. Since the WDT operates with a separate divider and counter, disabling the WDT has no effect on the operation of the INTRC oscillator when the FSCM is enabled.

As already noted, the clock source is switched to the INTRC clock when a clock failure is detected. This may mean a substantial change in the speed of code execution. If the WDT is enabled with a small prescale value, a decrease in clock speed allows a WDT time-out to occur and a subsequent device Reset. For this reason, Fail-Safe Clock Monitor events also reset the WDT and postscaler, allowing it to start timing from when execution speed was changed and decreasing the likelihood of an erroneous time-out.

If the interrupt is disabled, subsequent interrupts while in Idle mode will cause the CPU to begin executing instructions while being clocked by the INTRC source.

**FIGURE 22-5: FSCM TIMING DIAGRAM**



## 22.5.2 EXITING FAIL-SAFE OPERATION

The fail-safe condition is terminated by either a device Reset or by entering a power-managed mode. On Reset, the controller starts the primary clock source specified in Configuration Register 2H (with any required start-up delays that are required for the oscillator mode, such as OST or PLL timer). The INTRC oscillator provides the device clock until the primary clock source becomes ready (similar to a Two-Speed Start-up). The clock source is then switched to the primary clock (indicated by the OSTS bit in the OSCCON register becoming set). The Fail-Safe Clock Monitor then resumes monitoring the peripheral clock.

The primary clock source may never become ready during start-up. In this case, operation is clocked by the INTOSC multiplexor. The OSCCON register will remain in its Reset state until a power-managed mode is entered.

## 22.5.3 FSCM INTERRUPTS IN POWER-MANAGED MODES

By entering a power-managed mode, the clock multiplexor selects the clock source selected by the OSCCON register. Fail-Safe Clock Monitoring of the power-managed clock source resumes in the power-managed mode.

If an oscillator failure occurs during power-managed operation, the subsequent events depend on whether or not the oscillator failure interrupt is enabled. If enabled (OSCFIF = 1), code execution will be clocked by the INTRC multiplexor. An automatic transition back to the failed clock source will not occur.

## 22.5.4 POR OR WAKE-UP FROM SLEEP

The FSCM is designed to detect oscillator failure at any point after the device has exited Power-on Reset (POR) or low-power Sleep mode. When the primary device clock is either EC or INTRC mode, monitoring can begin immediately following these events.

For HS or HSPLL modes, the situation is somewhat different. Since the oscillator may require a start-up time considerably longer than the FSCM sample clock time, a false clock failure may be detected. To prevent this, the internal oscillator block is automatically configured as the device clock and functions until the primary clock is stable (the OST and PLL timers have timed out). This is identical to Two-Speed Start-up mode. Once the primary clock is stable, the INTRC returns to its role as the FSCM source.

**Note:** The same logic that prevents false oscillator failure interrupts on POR, or wake from Sleep, will also prevent the detection of the oscillator's failure to start at all following these events. This can be avoided by monitoring the OSTS bit and using a timing routine to determine if the oscillator is taking too long to start. Even so, no oscillator failure interrupt will be flagged.

As noted in **Section 22.4.1 "Special Considerations for Using Two-Speed Start-up"**, it is also possible to select another clock configuration and enter an alternate power-managed mode while waiting for the primary clock to become stable. When the new power-managed mode is selected, the primary clock is disabled.

# PIC18F85J11 FAMILY

---

## 22.6 Program Verification and Code Protection

For all devices in the PIC18F85J11 family of devices, the on-chip program memory space is treated as a single block. Code protection for this block is controlled by one Configuration bit, CP0. This bit inhibits external reads and writes to the program memory space. It has no direct effect in normal execution mode.

### 22.6.1 CONFIGURATION REGISTER PROTECTION

The Configuration registers are protected against untoward changes or reads in two ways. The primary protection is the write-once feature of the Configuration bits which prevents reconfiguration once the bit has been programmed during a power cycle. To safeguard against unpredictable events, Configuration bit changes resulting from individual cell level disruptions (such as ESD events) will cause a parity error and trigger a device Reset.

The data for the Configuration registers is derived from the Flash Configuration Words in program memory. When the CP0 bit is set, the source data for device configuration is also protected as a consequence.

## 22.7 In-Circuit Serial Programming

PIC18F85J11 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.

## 22.8 In-Circuit Debugger

When the  $\overline{\text{DEBUG}}$  Configuration bit is programmed to '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 22-4 shows which resources are required by the background debugger.

**TABLE 22-4: DEBUGGER RESOURCES**

|                 |           |
|-----------------|-----------|
| I/O pins:       | RB6, RB7  |
| Stack:          | 2 levels  |
| Program Memory: | 512 bytes |
| Data Memory:    | 10 bytes  |

## 23.0 INSTRUCTION SET SUMMARY

The PIC18F85J11 family of devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of 8 new instructions for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

### 23.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC<sup>®</sup> MCU instruction sets, while maintaining an easy migration from these PIC MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type, and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- **Byte-Oriented** operations
- **Bit-Oriented** operations
- **Literal** operations
- **Control** operations

The PIC18 instruction set summary in Table 23-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 23-1 shows the opcode field descriptions.

Most **byte-oriented** instructions have three operands:

1. The File Select Register (specified by 'f').
2. The destination of the result (specified by 'd').
3. The accessed memory (specified by 'a').

The File Select Register designator, 'f', specifies which File Select Register is to be used by the instruction. The destination designator, 'd', specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the WREG register. If 'd' is one, the result is placed in the File Select Register specified in the instruction.

All **bit-oriented** instructions have three operands:

1. The File Select Register (specified by 'f').
2. The bit in the File Select Register (specified by 'b').
3. The accessed memory (specified by 'a').

The bit field designator, 'b', selects the number of the bit affected by the operation, while the File Select Register designator, 'f', represents the number of the file in which the bit is located.

The **literal** instructions may use some of the following operands:

- A literal value to be loaded into a File Select Register (specified by 'k').
- The desired FSR register to load the literal value into (specified by 'f').
- No operand required (specified by '—').

The **control** instructions may use some of the following operands:

- A program memory address (specified by 'n').
- The mode of the **CALL** or **RETURN** instructions (specified by 's').
- The mode of the table read and table write instructions (specified by 'm').
- No operand required (specified by '—').

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the 4 MSBs are '1's. If this second word is executed as an instruction (by itself), it will execute as a **NOOP**.

All single-word instructions are executed in a single instruction cycle unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a **NOOP**.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1  $\mu$ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2  $\mu$ s. Two-word branch instructions (if true) would take 3  $\mu$ s.

Figure 23-1 shows the general formats that the instructions can have. All examples use the convention 'nnh' to represent a hexadecimal number.

The instruction set summary, shown in Table 23-2, lists the standard instructions recognized by the Microchip MPASM<sup>™</sup> Assembler.

**Section 23.1.1 "Standard Instruction Set"** provides a description of each instruction.

# PIC18F85J11 FAMILY

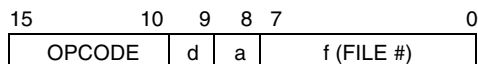
**TABLE 23-1: OPCODE FIELD DESCRIPTIONS**

| Field           | Description                                                                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| a               | RAM access bit:<br>a = 0: RAM location in Access RAM (BSR register is ignored)<br>a = 1: RAM bank is specified by BSR register                                 |
| bbb             | Bit address within an 8-bit File Select Register (0 to 7).                                                                                                     |
| BSR             | Bank Select Register. Used to select the current RAM bank.                                                                                                     |
| C, DC, Z, OV, N | ALU STATUS bits: <b>C</b> arry, <b>D</b> igit <b>C</b> arry, <b>Z</b> ero, <b>O</b> verflow, <b>N</b> egative.                                                 |
| d               | Destination select bit:<br>d = 0: store result in WREG<br>d = 1: store result in File Select Register f                                                        |
| dest            | Destination: either the WREG register or the specified register file location.                                                                                 |
| f               | 8-bit register file address (00h to FFh) or 2-bit FSR designator (0h to 3h).                                                                                   |
| f <sub>s</sub>  | 12-bit register file address (000h to FFFh). This is the source address.                                                                                       |
| f <sub>d</sub>  | 12-bit register file address (000h to FFFh). This is the destination address.                                                                                  |
| GIE             | Global Interrupt Enable bit.                                                                                                                                   |
| k               | Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).                                                                      |
| label           | Label name.                                                                                                                                                    |
| mm              | The mode of the TBLPTR register for the table read and table write instructions.<br>Only used with table read and table write instructions:                    |
| *               | No change to register (such as TBLPTR with table reads and writes)                                                                                             |
| *+              | Post-Increment register (such as TBLPTR with table reads and writes)                                                                                           |
| *-              | Post-Decrement register (such as TBLPTR with table reads and writes)                                                                                           |
| +*              | Pre-Increment register (such as TBLPTR with table reads and writes)                                                                                            |
| n               | The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions.                   |
| PC              | Program Counter.                                                                                                                                               |
| PCL             | Program Counter Low Byte.                                                                                                                                      |
| PCH             | Program Counter High Byte.                                                                                                                                     |
| PCLATH          | Program Counter High Byte Latch.                                                                                                                               |
| PCLATU          | Program Counter Upper Byte Latch.                                                                                                                              |
| PD              | Power-Down bit.                                                                                                                                                |
| PRODH           | Product of Multiply High Byte.                                                                                                                                 |
| PRODL           | Product of Multiply Low Byte.                                                                                                                                  |
| s               | Fast Call/Return mode select bit:<br>s = 0: do not update into/from shadow registers<br>s = 1: certain registers loaded into/from shadow registers (Fast mode) |
| TBLPTR          | 21-bit Table Pointer (points to a program memory location).                                                                                                    |
| TABLAT          | 8-bit Table Latch.                                                                                                                                             |
| TO              | Time-out bit.                                                                                                                                                  |
| TOS             | Top-of-Stack.                                                                                                                                                  |
| u               | Unused or unchanged.                                                                                                                                           |
| WDT             | Watchdog Timer.                                                                                                                                                |
| WREG            | Working register (accumulator).                                                                                                                                |
| x               | Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.   |
| z <sub>s</sub>  | 7-bit offset value for Indirect Addressing of register files (source).                                                                                         |
| z <sub>d</sub>  | 7-bit offset value for Indirect Addressing of register files (destination).                                                                                    |
| { }             | Optional argument.                                                                                                                                             |
| [text]          | Indicates an Indexed Address.                                                                                                                                  |
| (text)          | The contents of text.                                                                                                                                          |
| [expr] <n>      | Specifies bit n of the register indicated by the pointer, expr.                                                                                                |
| →               | Assigned to.                                                                                                                                                   |
| < >             | Register bit field.                                                                                                                                            |
| ∈               | In the set of.                                                                                                                                                 |
| italics         | User-defined term (font is Courier New).                                                                                                                       |



**FIGURE 23-1: GENERAL FORMAT FOR INSTRUCTIONS**

**Byte-Oriented File Select Register operations**

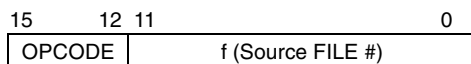


d = 0 for result destination to be WREG register  
d = 1 for result destination to be File Select Register (f)  
a = 0 to force Access Bank  
a = 1 for BSR to select bank  
f = 8-bit File Select Register address

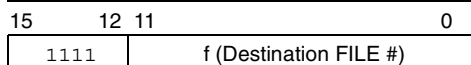
**Example Instruction**

ADDWF MYREG, W, B

**Byte to Byte move operations (2-word)**

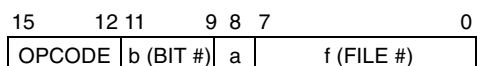


MOVFF MYREG1, MYREG2



f = 12-bit File Select Register address

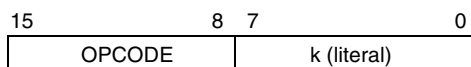
**Bit-Oriented File Select Register operations**



BSF MYREG, bit, B

b = 3-bit position of bit in File Select Register (f)  
a = 0 to force Access Bank  
a = 1 for BSR to select bank  
f = 8-bit File Select Register address

**Literal operations**

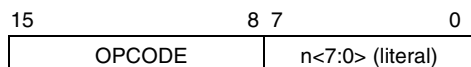


MOVLW 7Fh

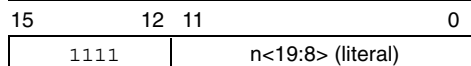
k = 8-bit immediate value

**Control operations**

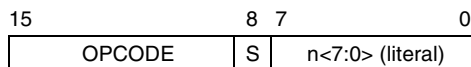
**CALL, GOTO and Branch operations**



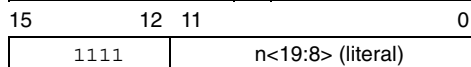
GOTO Label



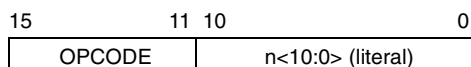
n = 20-bit immediate value



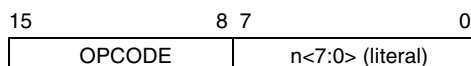
CALL MYFUNC



S = Fast bit



BRA MYFUNC



BC MYFUNC

# PIC18F85J11 FAMILY

**TABLE 23-2: PIC18F85J11 FAMILY INSTRUCTION SET**

| Mnemonic,<br>Operands                 | Description                                                  | Cycles     | 16-bit Instruction Word |      |      |      | Status<br>Affected | Notes      |
|---------------------------------------|--------------------------------------------------------------|------------|-------------------------|------|------|------|--------------------|------------|
|                                       |                                                              |            | MSb                     |      | LSb  |      |                    |            |
| BYTE-ORIENTED OPERATIONS              |                                                              |            |                         |      |      |      |                    |            |
| ADDWF f, d, a                         | Add WREG and f                                               | 1          | 0010                    | 01da | ffff | ffff | C, DC, Z, OV, N    | 1, 2       |
| ADDWFC f, d, a                        | Add WREG and Carry bit to f                                  | 1          | 0010                    | 00da | ffff | ffff | C, DC, Z, OV, N    | 1, 2       |
| ANDWF f, d, a                         | AND WREG with f                                              | 1          | 0001                    | 01da | ffff | ffff | Z, N               | 1,2        |
| CLRF f, a                             | Clear f                                                      | 1          | 0110                    | 101a | ffff | ffff | Z                  | 2          |
| COMF f, d, a                          | Complement f                                                 | 1          | 0001                    | 11da | ffff | ffff | Z, N               | 1, 2       |
| CPFSEQ f, a                           | Compare f with WREG, Skip =                                  | 1 (2 or 3) | 0110                    | 001a | ffff | ffff | None               | 4          |
| CPFSGT f, a                           | Compare f with WREG, Skip >                                  | 1 (2 or 3) | 0110                    | 010a | ffff | ffff | None               | 4          |
| CPFSLT f, a                           | Compare f with WREG, Skip <                                  | 1 (2 or 3) | 0110                    | 000a | ffff | ffff | None               | 1, 2       |
| DECF f, d, a                          | Decrement f                                                  | 1          | 0000                    | 01da | ffff | ffff | C, DC, Z, OV, N    | 1, 2, 3, 4 |
| DECFSZ f, d, a                        | Decrement f, Skip if 0                                       | 1 (2 or 3) | 0010                    | 11da | ffff | ffff | None               | 1, 2, 3, 4 |
| DCFSNZ f, d, a                        | Decrement f, Skip if Not 0                                   | 1 (2 or 3) | 0100                    | 11da | ffff | ffff | None               | 1, 2       |
| INCF f, d, a                          | Increment f                                                  | 1          | 0010                    | 10da | ffff | ffff | C, DC, Z, OV, N    | 1, 2, 3, 4 |
| INCFSZ f, d, a                        | Increment f, Skip if 0                                       | 1 (2 or 3) | 0011                    | 11da | ffff | ffff | None               | 4          |
| INFSNZ f, d, a                        | Increment f, Skip if Not 0                                   | 1 (2 or 3) | 0100                    | 10da | ffff | ffff | None               | 1, 2       |
| IORWF f, d, a                         | Inclusive OR WREG with f                                     | 1          | 0001                    | 00da | ffff | ffff | Z, N               | 1, 2       |
| MOVF f, d, a                          | Move f                                                       | 1          | 0101                    | 00da | ffff | ffff | Z, N               | 1          |
| MOVFF f <sub>s</sub> , f <sub>d</sub> | Move f <sub>s</sub> (source) to f <sub>d</sub> (destination) | 2          | 1100                    | ffff | ffff | ffff | None               |            |
|                                       |                                                              |            | 1111                    | ffff | ffff | ffff |                    |            |
| MOVWF f, a                            | Move WREG to f                                               | 1          | 0110                    | 111a | ffff | ffff | None               |            |
| MULWF f, a                            | Multiply WREG with f                                         | 1          | 0000                    | 001a | ffff | ffff | None               | 1, 2       |
| NEGF f, a                             | Negate f                                                     | 1          | 0110                    | 110a | ffff | ffff | C, DC, Z, OV, N    |            |
| RLCF f, d, a                          | Rotate Left f through Carry                                  | 1          | 0011                    | 01da | ffff | ffff | C, Z, N            | 1, 2       |
| RLNCF f, d, a                         | Rotate Left f (No Carry)                                     | 1          | 0100                    | 01da | ffff | ffff | Z, N               |            |
| RRCF f, d, a                          | Rotate Right f through Carry                                 | 1          | 0011                    | 00da | ffff | ffff | C, Z, N            |            |
| RRNCF f, d, a                         | Rotate Right f (No Carry)                                    | 1          | 0100                    | 00da | ffff | ffff | Z, N               |            |
| SETF f, a                             | Set f                                                        | 1          | 0110                    | 100a | ffff | ffff | None               | 1, 2       |
| SUBFWB f, d, a                        | Subtract f from WREG with Borrow                             | 1          | 0101                    | 01da | ffff | ffff | C, DC, Z, OV, N    |            |
| SUBWF f, d, a                         | Subtract WREG from f                                         | 1          | 0101                    | 11da | ffff | ffff | C, DC, Z, OV, N    | 1, 2       |
| SUBWFB f, d, a                        | Subtract WREG from f with Borrow                             | 1          | 0101                    | 10da | ffff | ffff | C, DC, Z, OV, N    |            |
| SWAPF f, d, a                         | Swap Nibbles in f                                            | 1          | 0011                    | 10da | ffff | ffff | None               | 4          |
| TSTFSZ f, a                           | Test f, Skip if 0                                            | 1 (2 or 3) | 0110                    | 011a | ffff | ffff | None               | 1, 2       |
| XORWF f, d, a                         | Exclusive OR WREG with f                                     | 1          | 0001                    | 10da | ffff | ffff | Z, N               |            |

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

# PIC18F85J11 FAMILY

**TABLE 23-2: PIC18F85J11 FAMILY INSTRUCTION SET (CONTINUED)**

| Mnemonic,<br>Operands   |         | Description                    | Cycles     | 16-bit Instruction Word |      |      |      | Status<br>Affected                              | Notes |
|-------------------------|---------|--------------------------------|------------|-------------------------|------|------|------|-------------------------------------------------|-------|
|                         |         |                                |            | MSb                     |      | LSb  |      |                                                 |       |
| BIT-ORIENTED OPERATIONS |         |                                |            |                         |      |      |      |                                                 |       |
| BCF                     | f, b, a | Bit Clear f                    | 1          | 1001                    | bbba | ffff | ffff | None                                            | 1, 2  |
| BSF                     | f, b, a | Bit Set f                      | 1          | 1000                    | bbba | ffff | ffff | None                                            | 1, 2  |
| BTFSC                   | f, b, a | Bit Test f, Skip if Clear      | 1 (2 or 3) | 1011                    | bbba | ffff | ffff | None                                            | 3, 4  |
| BTFSS                   | f, b, a | Bit Test f, Skip if Set        | 1 (2 or 3) | 1010                    | bbba | ffff | ffff | None                                            | 3, 4  |
| BTG                     | f, b, a | Bit Toggle f                   | 1          | 0111                    | bbba | ffff | ffff | None                                            | 1, 2  |
| CONTROL OPERATIONS      |         |                                |            |                         |      |      |      |                                                 |       |
| BC                      | n       | Branch if Carry                | 1 (2)      | 1110                    | 0010 | nnnn | nnnn | None                                            | 4     |
| BN                      | n       | Branch if Negative             | 1 (2)      | 1110                    | 0110 | nnnn | nnnn | None                                            |       |
| BNC                     | n       | Branch if Not Carry            | 1 (2)      | 1110                    | 0011 | nnnn | nnnn | None                                            |       |
| BNN                     | n       | Branch if Not Negative         | 1 (2)      | 1110                    | 0111 | nnnn | nnnn | None                                            |       |
| BNOV                    | n       | Branch if Not Overflow         | 1 (2)      | 1110                    | 0101 | nnnn | nnnn | None                                            |       |
| BNZ                     | n       | Branch if Not Zero             | 1 (2)      | 1110                    | 0001 | nnnn | nnnn | None                                            |       |
| BOV                     | n       | Branch if Overflow             | 1 (2)      | 1110                    | 0100 | nnnn | nnnn | None                                            |       |
| BRA                     | n       | Branch Unconditionally         | 2          | 1101                    | 0nnn | nnnn | nnnn | None                                            |       |
| BZ                      | n       | Branch if Zero                 | 1 (2)      | 1110                    | 0000 | nnnn | nnnn | None                                            |       |
| CALL                    | n, s    | Call Subroutine    1st word    | 2          | 1110                    | 110s | kkkk | kkkk | None                                            |       |
|                         |         | 2nd word                       |            | 1111                    | kkkk | kkkk | kkkk |                                                 |       |
| CLRWDT                  | —       | Clear Watchdog Timer           | 1          | 0000                    | 0000 | 0000 | 0100 | $\overline{\text{TO}}$ , $\overline{\text{PD}}$ |       |
| DAW                     | —       | Decimal Adjust WREG            | 1          | 0000                    | 0000 | 0000 | 0111 | C                                               |       |
| GOTO                    | n       | Go to Address    1st word      | 2          | 1110                    | 1111 | kkkk | kkkk | None                                            |       |
|                         |         | 2nd word                       |            | 1111                    | kkkk | kkkk | kkkk |                                                 |       |
| NOP                     | —       | No Operation                   | 1          | 0000                    | 0000 | 0000 | 0000 | None                                            |       |
| NOP                     | —       | No Operation                   | 1          | 1111                    | xxxx | xxxx | xxxx | None                                            |       |
| POP                     | —       | Pop Top of Return Stack (TOS)  | 1          | 0000                    | 0000 | 0000 | 0110 | None                                            |       |
| PUSH                    | —       | Push Top of Return Stack (TOS) | 1          | 0000                    | 0000 | 0000 | 0101 | None                                            |       |
| RCALL                   | n       | Relative Call                  | 2          | 1101                    | 1nnn | nnnn | nnnn | None                                            |       |
| RESET                   |         | Software Device Reset          | 1          | 0000                    | 0000 | 1111 | 1111 | All                                             |       |
| RETFIE                  | s       | Return from Interrupt Enable   | 2          | 0000                    | 0000 | 0001 | 000s | GIE/GIEH,<br>PEIE/GIEL                          |       |
| RETLW                   | k       | Return with Literal in WREG    | 2          | 0000                    | 1100 | kkkk | kkkk | None                                            |       |
| RETURN                  | s       | Return from Subroutine         | 2          | 0000                    | 0000 | 0001 | 001s | None                                            |       |
| SLEEP                   | —       | Go into Standby mode           | 1          | 0000                    | 0000 | 0000 | 0011 | $\overline{\text{TO}}$ , $\overline{\text{PD}}$ |       |

**Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

- 2:** If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

# PIC18F85J11 FAMILY

**TABLE 23-2: PIC18F85J11 FAMILY INSTRUCTION SET (CONTINUED)**

| Mnemonic,<br>Operands                   | Description                                       | Cycles | 16-bit Instruction Word |      |      |      | Status<br>Affected | Notes |
|-----------------------------------------|---------------------------------------------------|--------|-------------------------|------|------|------|--------------------|-------|
|                                         |                                                   |        | MSb                     |      | LSb  |      |                    |       |
| LITERAL OPERATIONS                      |                                                   |        |                         |      |      |      |                    |       |
| ADDLW k                                 | Add Literal and WREG                              | 1      | 0000                    | 1111 | kkkk | kkkk | C, DC, Z, OV, N    |       |
| ANDLW k                                 | AND Literal with WREG                             | 1      | 0000                    | 1011 | kkkk | kkkk | Z, N               |       |
| IORLW k                                 | Inclusive OR Literal with WREG                    | 1      | 0000                    | 1001 | kkkk | kkkk | Z, N               |       |
| LFSR f, k                               | Move Literal (12-bit) 2nd word to FSR(f) 1st word | 2      | 1110                    | 1110 | 00ff | kkkk | None               |       |
|                                         |                                                   |        | 1111                    | 0000 | kkkk | kkkk |                    |       |
| MOVLB k                                 | Move Literal to BSR<3:0>                          | 1      | 0000                    | 0001 | 0000 | kkkk | None               |       |
| MOVLW k                                 | Move Literal to WREG                              | 1      | 0000                    | 1110 | kkkk | kkkk | None               |       |
| MULLW k                                 | Multiply Literal with WREG                        | 1      | 0000                    | 1101 | kkkk | kkkk | None               |       |
| RETLW k                                 | Return with Literal in WREG                       | 2      | 0000                    | 1100 | kkkk | kkkk | None               |       |
| SUBLW k                                 | Subtract WREG from Literal                        | 1      | 0000                    | 1000 | kkkk | kkkk | C, DC, Z, OV, N    |       |
| XORLW k                                 | Exclusive OR Literal with WREG                    | 1      | 0000                    | 1010 | kkkk | kkkk | Z, N               |       |
| DATA MEMORY ↔ PROGRAM MEMORY OPERATIONS |                                                   |        |                         |      |      |      |                    |       |
| TBLRD*                                  | Table Read                                        | 2      | 0000                    | 0000 | 0000 | 1000 | None               |       |
| TBLRD*+                                 | Table Read with Post-Increment                    |        | 0000                    | 0000 | 0000 | 1001 | None               |       |
| TBLRD*-                                 | Table Read with Post-Decrement                    | 0000   | 0000                    | 0000 | 1010 | None |                    |       |
| TBLRD*+                                 | Table Read with Pre-Increment                     | 0000   | 0000                    | 0000 | 1011 | None |                    |       |
| TBLWT*                                  | Table Write                                       | 2      | 0000                    | 0000 | 0000 | 1100 | None               |       |
| TBLWT*+                                 | Table Write with Post-Increment                   |        | 0000                    | 0000 | 0000 | 1101 | None               |       |
| TBLWT*-                                 | Table Write with Post-Decrement                   |        | 0000                    | 0000 | 0000 | 1110 | None               |       |
| TBLWT*+                                 | Table Write with Pre-Increment                    |        | 0000                    | 0000 | 0000 | 1111 | None               |       |

- Note 1:** When a PORT register is modified as a function of itself (e.g., `MOVF PORTB, 1, 0`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2:** If this instruction is executed on the TMR0 register (and, where applicable,  $d = 1$ ), the prescaler will be cleared if assigned.
- 3:** If the Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.
- 4:** Some instructions are two-word instructions. The second word of these instructions will be executed as a NOP unless the first word of the instruction retrieves the information embedded in these 16 bits. This ensures that all program memory locations have a valid instruction.

# PIC18F85J11 FAMILY

## 23.1.1 STANDARD INSTRUCTION SET

### ADDLW ADD Literal to W

|                   |                                                                                         |                  |              |            |      |      |      |      |
|-------------------|-----------------------------------------------------------------------------------------|------------------|--------------|------------|------|------|------|------|
| Syntax:           | ADDLW k                                                                                 |                  |              |            |      |      |      |      |
| Operands:         | $0 \leq k \leq 255$                                                                     |                  |              |            |      |      |      |      |
| Operation:        | $(W) + k \rightarrow W$                                                                 |                  |              |            |      |      |      |      |
| Status Affected:  | N, OV, C, DC, Z                                                                         |                  |              |            |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0000</td><td>1111</td><td>kkkk</td><td>kkkk</td></tr></table> |                  |              |            | 0000 | 1111 | kkkk | kkkk |
| 0000              | 1111                                                                                    | kkkk             | kkkk         |            |      |      |      |      |
| Description:      | The contents of W are added to the 8-bit literal 'k' and the result is placed in W.     |                  |              |            |      |      |      |      |
| Words:            | 1                                                                                       |                  |              |            |      |      |      |      |
| Cycles:           | 1                                                                                       |                  |              |            |      |      |      |      |
| Q Cycle Activity: |                                                                                         |                  |              |            |      |      |      |      |
|                   | Q1                                                                                      | Q2               | Q3           | Q4         |      |      |      |      |
|                   | Decode                                                                                  | Read literal 'k' | Process Data | Write to W |      |      |      |      |

**Example:**           ADDLW   15h

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

### ADDWF ADD W to f

| Syntax:           | ADDWF f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |              |                      |  |      |      |      |      |        |                   |              |                      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|----------------------|--|------|------|------|------|--------|-------------------|--------------|----------------------|
| Operands:         | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |              |                      |  |      |      |      |      |        |                   |              |                      |
| Operation:        | $(W) + (f) \rightarrow \text{dest}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |              |                      |  |      |      |      |      |        |                   |              |                      |
| Status Affected:  | N, OV, C, DC, Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |              |                      |  |      |      |      |      |        |                   |              |                      |
| Encoding:         | <table border="1"><tr><td>0010</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |              |                      |  | 0010 | 01da | ffff | ffff |        |                   |              |                      |
| 0010              | 01da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | ffff         | ffff                 |  |      |      |      |      |        |                   |              |                      |
| Description:      | <p>Add W to register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |              |                      |  |      |      |      |      |        |                   |              |                      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |              |                      |  |      |      |      |      |        |                   |              |                      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |              |                      |  |      |      |      |      |        |                   |              |                      |
| Q Cycle Activity: | <table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                         |              |                      |  | Q1   | Q2   | Q3   | Q4   | Decode | Read register 'f' | Process Data | Write to destination |
| Q1                | Q2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Q3           | Q4                   |  |      |      |      |      |        |                   |              |                      |
| Decode            | Read register 'f'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Process Data | Write to destination |  |      |      |      |      |        |                   |              |                      |

**Example:**           ADDWF   REG, 0, 0

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

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

# PIC18F85J11 FAMILY

## ADDWFC

## ADD W and Carry bit to f

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

Operands:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

Operation:  $(W) + (f) + (C) \rightarrow \text{dest}$

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0010 | 00da | ffff | ffff |
|------|------|------|------|

Description: Add W, the Carry flag and data memory location 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in data memory location 'f'.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** ADDWFC REG, 0, 1

Before Instruction

Carry bit = 1  
REG = 02h  
W = 4Dh

After Instruction

Carry bit = 0  
REG = 02h  
W = 50h

## ANDLW

## AND Literal with W

Syntax: ANDLW k

Operands:  $0 \leq k \leq 255$

Operation:  $(W) .\text{AND. } k \rightarrow W$

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 1011 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are ANDed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4         |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

**Example:** ANDLW 05Fh

Before Instruction

W = A3h

After Instruction

W = 03h

# PIC18F85J11 FAMILY

## ANDWF

## AND W with f

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

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

Operation: (W) .AND. (f)  $\rightarrow$  dest

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0001 | 01da | ffff | ffff |
|------|------|------|------|

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

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** ANDWF REG, 0, 0

Before Instruction

W = 17h  
 REG = C2h

After Instruction

W = 02h  
 REG = C2h

## BC

## Branch if Carry

Syntax: BC n

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

Operation: if Carry bit is '1',  
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1110 | 0010 | nnnn | nnnn |
|------|------|------|------|

Description: If the Carry bit is '1', then the program will branch.

The 2's complement number '2n' is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

**Example:** HERE BC 5

Before Instruction

PC = address (HERE)

After Instruction

If Carry = 1;  
 PC = address (HERE + 12)  
 If Carry = 0;  
 PC = address (HERE + 2)

# PIC18F85J11 FAMILY

## BCF Bit Clear f

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                   |              |                    |      |      |      |      |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------|--------------------|------|------|------|------|
| Syntax:           | BCF f, b {,a}                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                   |              |                    |      |      |      |      |
| Operands:         | $0 \leq f \leq 255$<br>$0 \leq b \leq 7$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                          |                   |              |                    |      |      |      |      |
| Operation:        | $0 \rightarrow f < b >$                                                                                                                                                                                                                                                                                                                                                                                                                                             |                   |              |                    |      |      |      |      |
| Status Affected:  | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |                    |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>1001</td><td>bbba</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                             |                   |              |                    | 1001 | bbba | ffff | ffff |
| 1001              | bbba                                                                                                                                                                                                                                                                                                                                                                                                                                                                | ffff              | ffff         |                    |      |      |      |      |
| Description:      | <p>Bit 'b' in register 'f' is cleared.</p> <p>If 'a' is '0', the Access Bank is selected.<br/>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                   |              |                    |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                   |              |                    |      |      |      |      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                   |              |                    |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                   |              |                    |      |      |      |      |
|                   | Q1                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Q2                | Q3           | Q4                 |      |      |      |      |
|                   | Decode                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Read register 'f' | Process Data | Write register 'f' |      |      |      |      |

**Example:**      BCF      FLAG\_REG,    7, 0

Before Instruction  
     FLAG\_REG = C7h  
 After Instruction  
     FLAG\_REG = 47h

## BN Branch if Negative

|                   |                                                                                                                                                                                                                                                                                                    |                  |              |              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------|--------------|
| Syntax:           | BN n                                                                                                                                                                                                                                                                                               |                  |              |              |
| Operands:         | $-128 \leq n \leq 127$                                                                                                                                                                                                                                                                             |                  |              |              |
| Operation:        | if Negative bit is '1',<br>$(PC) + 2 + 2n \rightarrow PC$                                                                                                                                                                                                                                          |                  |              |              |
| Status Affected:  | None                                                                                                                                                                                                                                                                                               |                  |              |              |
| Encoding:         | 1110                                                                                                                                                                                                                                                                                               | 0110             | nnnn         | nnnn         |
| Description:      | <p>If the Negative bit is '1', then the program will branch.</p> <p>The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be <math>PC + 2 + 2n</math>. This instruction is then a two-cycle instruction.</p> |                  |              |              |
| Words:            | 1                                                                                                                                                                                                                                                                                                  |                  |              |              |
| Cycles:           | 1(2)                                                                                                                                                                                                                                                                                               |                  |              |              |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                    |                  |              |              |
| If Jump:          |                                                                                                                                                                                                                                                                                                    |                  |              |              |
|                   | Q1                                                                                                                                                                                                                                                                                                 | Q2               | Q3           | Q4           |
|                   | Decode                                                                                                                                                                                                                                                                                             | Read literal 'n' | Process Data | Write to PC  |
|                   | No operation                                                                                                                                                                                                                                                                                       | No operation     | No operation | No operation |
| If No Jump:       |                                                                                                                                                                                                                                                                                                    |                  |              |              |
|                   | Q1                                                                                                                                                                                                                                                                                                 | Q2               | Q3           | Q4           |
|                   | Decode                                                                                                                                                                                                                                                                                             | Read literal 'n' | Process Data | No operation |

**Example:**      HERE      BN    Jump

Before Instruction  
     PC            =    address (HERE)  
 After Instruction  
     If Negative    =    1;  
     PC            =    address (Jump)  
     If Negative    =    0;  
     PC            =    address (HERE + 2)



# PIC18F85J11 FAMILY

## BNC Branch if Not Carry

|                  |                                                                                                                                                                                                                                                                                    |      |      |      |      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:          | BNC n                                                                                                                                                                                                                                                                              |      |      |      |      |
| Operands:        | -128 ≤ n ≤ 127                                                                                                                                                                                                                                                                     |      |      |      |      |
| Operation:       | if Carry bit is '0',<br>(PC) + 2 + 2n → PC                                                                                                                                                                                                                                         |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                               |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>1110</td><td>0011</td><td>nnnn</td><td>nnnn</td></tr></table>                                                                                                                                                                                            | 1110 | 0011 | nnnn | nnnn |
| 1110             | 0011                                                                                                                                                                                                                                                                               | nnnn | nnnn |      |      |
| Description:     | <p>If the Carry bit is '0', then the program will branch.</p> <p>The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be PC + 2 + 2n. This instruction is then a two-cycle instruction.</p> |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                  |      |      |      |      |
| Cycles:          | 1(2)                                                                                                                                                                                                                                                                               |      |      |      |      |

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example:                      HERE                      BNC    Jump

Before Instruction

PC = address (HERE)

After Instruction

If Carry = 0;  
PC = address (Jump)  
If Carry = 1;  
PC = address (HERE + 2)

## BNN Branch if Not Negative

|                  |                                                                                                                                                                                                                                                                                                    |      |      |      |      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:          | BNN n                                                                                                                                                                                                                                                                                              |      |      |      |      |
| Operands:        | $-128 \leq n \leq 127$                                                                                                                                                                                                                                                                             |      |      |      |      |
| Operation:       | if Negative bit is '0',<br>$(PC) + 2 + 2n \rightarrow PC$                                                                                                                                                                                                                                          |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                                               |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>1110</td><td>0111</td><td>nnnn</td><td>nnnn</td></tr></table>                                                                                                                                                                                                            | 1110 | 0111 | nnnn | nnnn |
| 1110             | 0111                                                                                                                                                                                                                                                                                               | nnnn | nnnn |      |      |
| Description:     | <p>If the Negative bit is '0', then the program will branch.</p> <p>The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be <math>PC + 2 + 2n</math>. This instruction is then a two-cycle instruction.</p> |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                                  |      |      |      |      |
| Cycles:          | 1(2)                                                                                                                                                                                                                                                                                               |      |      |      |      |

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example:                      HERE                      BNN    Jump

Before Instruction

PC = address (HERE)

After Instruction

If Negative = 0;  
PC = address (Jump)  
If Negative = 1;  
PC = address (HERE + 2)

# PIC18F85J11 FAMILY

## BNOV Branch if Not Overflow

Syntax: BNOV n

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

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

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1110 | 0101 | nnnn | nnnn |
|------|------|------|------|

Description: If the Overflow bit is '0', then the program will branch.

The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example:                      HERE                      BNOV    Jump

Before Instruction  
PC = address (HERE)

After Instruction  
If Overflow = 0;  
PC = address (Jump)  
If Overflow = 1;  
PC = address (HERE + 2)

## BNZ Branch if Not Zero

Syntax: BNZ n

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

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

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1110 | 0001 | nnnn | nnnn |
|------|------|------|------|

Description: If the Zero bit is '0', then the program will branch.

The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example:                      HERE                      BNZ    Jump

Before Instruction  
PC = address (HERE)

After Instruction  
If Zero = 0;  
PC = address (Jump)  
If Zero = 1;  
PC = address (HERE + 2)

# PIC18F85J11 FAMILY

## BRA Unconditional Branch

Syntax: BRA n

Operands:  $-1024 \leq n \leq 1023$

Operation:  $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1101 | 0nnn | nnnn | nnnn |
|------|------|------|------|

Description: Add the 2's complement number, '2n', to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

Example:                HERE        BRA    Jump

Before Instruction  
PC = address (HERE)

After Instruction  
PC = address (Jump)

## BSF Bit Set f

Syntax: BSF f, b {,a}

Operands:  $0 \leq f \leq 255$   
 $0 \leq b \leq 7$   
 $a \in [0, 1]$

Operation:  $1 \rightarrow f \langle b \rangle$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1000 | bbba | ffff | ffff |
|------|------|------|------|

Description: Bit 'b' in register 'f' is set.

If 'a' is '0', the Access Bank is selected.  
If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:                BSF        FLAG\_REG, 7, 1

Before Instruction  
FLAG\_REG = 0Ah

After Instruction  
FLAG\_REG = 8Ah

# PIC18F85J11 FAMILY

## BTFSC Bit Test File, Skip if Clear

Syntax: BTFSC f, b {,a}

Operands:  $0 \leq f \leq 255$   
 $0 \leq b \leq 7$   
 $a \in [0, 1]$

Operation: skip if (f<b>) = 0

Status Affected: None

|           |      |      |      |      |
|-----------|------|------|------|------|
| Encoding: | 1011 | bbba | ffff | ffff |
|-----------|------|------|------|------|

Description: If bit 'b' in register 'f' is '0', then the next instruction is skipped. If bit 'b' is '0', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this 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  $f \leq 95$  (5Fh). See

**Section 23.2.3 “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 | No operation |

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  | BTFSC | FLAG, 1, 0 |
| FALSE | :     |            |
| TRUE  | :     |            |

Before Instruction

PC = address (HERE)

After Instruction

If FLAG<1> = 0;  
PC = address (TRUE)  
If FLAG<1> = 1;  
PC = address (FALSE)

## BTFSS Bit Test File, Skip if Set

Syntax: BTFSS f, b {,a}

Operands:  $0 \leq f \leq 255$   
 $0 \leq b < 7$   
 $a \in [0, 1]$

Operation: skip if (f<b>) = 1

Status Affected: None

|           |      |      |      |      |
|-----------|------|------|------|------|
| Encoding: | 1010 | bbba | ffff | ffff |
|-----------|------|------|------|------|

Description: If bit 'b' in register 'f' is '1', then the next instruction is skipped. If bit 'b' is '1', then the next instruction fetched during the current instruction execution is discarded and a NOP is executed instead, making this 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  $f \leq 95$  (5Fh). See

**Section 23.2.3 “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 | No operation |

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  | BTFSS | FLAG, 1, 0 |
| FALSE | :     |            |
| TRUE  | :     |            |

Before Instruction

PC = address (HERE)

After Instruction

If FLAG<1> = 0;  
PC = address (FALSE)  
If FLAG<1> = 1;  
PC = address (TRUE)

# PIC18F85J11 FAMILY

## BTG

### Bit Toggle f

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                   |              |                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------|--------------------|
| Syntax:           | BTG f, b {,a}                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                   |              |                    |
| Operands:         | $0 \leq f \leq 255$<br>$0 \leq b < 7$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                          |                   |              |                    |
| Operation:        | $(\overline{f < b}) \rightarrow f < b$                                                                                                                                                                                                                                                                                                                                                                                                                                           |                   |              |                    |
| Status Affected:  | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                   |              |                    |
| Encoding:         | 0111                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | bbba              | ffff         | ffff               |
| Description:      | <p>Bit 'b' in data memory location 'f' is inverted.</p> <p>If 'a' is '0', the Access Bank is selected.<br/>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                   |              |                    |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |                    |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |                    |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                   |              |                    |
|                   | Q1                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Q2                | Q3           | Q4                 |
|                   | Decode                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Read register 'f' | Process Data | Write register 'f' |

**Example:** BTG PORTC, 4, 0

Before Instruction:

PORTC = 0111 0101 [75h]

After Instruction:

PORTC = 0110 0101 [65h]

## BOV

### Branch if Overflow

|                   |                                                                                                                                                                                                                                                                                                    |      |      |      |      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:           | BOV n                                                                                                                                                                                                                                                                                              |      |      |      |      |
| Operands:         | $-128 \leq n \leq 127$                                                                                                                                                                                                                                                                             |      |      |      |      |
| Operation:        | if Overflow bit is '1',<br>$(PC) + 2 + 2n \rightarrow PC$                                                                                                                                                                                                                                          |      |      |      |      |
| Status Affected:  | None                                                                                                                                                                                                                                                                                               |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>1110</td><td>0100</td><td>nnnn</td><td>nnnn</td></tr></table>                                                                                                                                                                                                            | 1110 | 0100 | nnnn | nnnn |
| 1110              | 0100                                                                                                                                                                                                                                                                                               | nnnn | nnnn |      |      |
| Description:      | <p>If the Overflow bit is '1', then the program will branch.</p> <p>The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be <math>PC + 2 + 2n</math>. This instruction is then a two-cycle instruction.</p> |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                  |      |      |      |      |
| Cycles:           | 1(2)                                                                                                                                                                                                                                                                                               |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                    |      |      |      |      |
| If Jump:          |                                                                                                                                                                                                                                                                                                    |      |      |      |      |

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

**Example:** HERE BOV Jump

Before Instruction

PC = address (HERE)

After Instruction

If Overflow = 1;

PC = address (Jump)

If Overflow = 0;

PC = address (HERE + 2)

# PIC18F85J11 FAMILY

## BZ Branch if Zero

Syntax: BZ n

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

Operation: if Zero bit is '1',  
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding:

|      |      |      |      |
|------|------|------|------|
| 1110 | 0000 | nnnn | nnnn |
|------|------|------|------|

Description: If the Zero bit is '1', then the program will branch.

The 2's complement number, '2n', is added to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is then a two-cycle instruction.

Words: 1

Cycles: 1(2)

Q Cycle Activity:

If Jump:

| Q1           | Q2               | Q3           | Q4           |
|--------------|------------------|--------------|--------------|
| Decode       | Read literal 'n' | Process Data | Write to PC  |
| No operation | No operation     | No operation | No operation |

If No Jump:

| Q1     | Q2               | Q3           | Q4           |
|--------|------------------|--------------|--------------|
| Decode | Read literal 'n' | Process Data | No operation |

Example:                      HERE                      BZ                      Jump

Before Instruction  
PC = address (HERE)

After Instruction  
If Zero = 1;  
PC = address (Jump)  
If Zero = 0;  
PC = address (HERE + 2)

## CALL Subroutine Call

Syntax: CALL k {,s}

Operands:  $0 \leq k \leq 1048575$   
 $s \in [0, 1]$

Operation:  $(PC) + 4 \rightarrow TOS$ ,  
 $k \rightarrow PC<20:1>$ ;  
if  $s = 1$ ,  
 $(W) \rightarrow WS$ ,  
 $(STATUS) \rightarrow STATUSS$ ,  
 $(BSR) \rightarrow BSRS$

Status Affected: None

Encoding:

|      |                     |                    |                   |
|------|---------------------|--------------------|-------------------|
| 1110 | 110s                | k <sub>7</sub> kkk | kkkk <sub>0</sub> |
| 1111 | k <sub>19</sub> kkk | kkkk               | kkkk <sub>8</sub> |

1st word ( $k<7:0>$ )  
2nd word ( $k<19:8>$ )

Description: Subroutine call of entire 2-Mbyte memory range. First, return address ( $PC + 4$ ) is pushed onto the return stack. If 's' = 1, the W, STATUS and BSR registers are also pushed into their respective shadow registers, WS, STATUSS and BSRS. If 's' = 0, no update occurs (default). Then, the 20-bit value, 'k', is loaded into  $PC<20:1>$ . CALL is a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1           | Q2                                          | Q3               | Q4                                      |
|--------------|---------------------------------------------|------------------|-----------------------------------------|
| Decode       | Read literal 'k'<7:0>,&br/>Push PC to stack | Push PC to stack | Read literal 'k'<19:8>,&br/>Write to PC |
| No operation | No operation                                | No operation     | No operation                            |

Example:                      HERE                      CALL                      THERE, 1

Before Instruction  
PC = address (HERE)

After Instruction  
PC = address (THERE)  
TOS = address (HERE + 4)  
WS = W  
BSRS = BSR  
STATUSS = STATUS

# PIC18F85J11 FAMILY

| CLRF              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Clear f           |              |                    |  |      |      |      |      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------|--------------------|--|------|------|------|------|
| Syntax:           | CLRF f {,a}                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |                    |  |      |      |      |      |
| Operands:         | $0 \leq f \leq 255$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |              |                    |  |      |      |      |      |
| Operation:        | $000h \rightarrow f$ ,<br>$1 \rightarrow Z$                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |                    |  |      |      |      |      |
| Status Affected:  | Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                   |              |                    |  |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0110</td><td>101a</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                    |                   |              |                    |  | 0110 | 101a | ffff | ffff |
| 0110              | 101a                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | ffff              | ffff         |                    |  |      |      |      |      |
| Description:      | <p>Clears the contents of the specified register.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                   |              |                    |  |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                   |              |                    |  |      |      |      |      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                   |              |                    |  |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                   |              |                    |  |      |      |      |      |
|                   | Q1                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Q2                | Q3           | Q4                 |  |      |      |      |      |
|                   | Decode                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Read register 'f' | Process Data | Write register 'f' |  |      |      |      |      |

**Example:** CLRF FLAG\_REG, 1

Before Instruction  
FLAG\_REG = 5Ah  
After Instruction  
FLAG\_REG = 00h

| CLRWD             |                                                                                                                                                    | Clear Watchdog Timer |              |              |      |      |      |      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|--------------|--------------|------|------|------|------|
| Syntax:           | CLRWD                                                                                                                                              |                      |              |              |      |      |      |      |
| Operands:         | None                                                                                                                                               |                      |              |              |      |      |      |      |
| Operation:        | 000h → WDT,<br>000h → WDT postscaler,<br>1 → $\overline{TO}$ ,<br>1 → $\overline{PD}$                                                              |                      |              |              |      |      |      |      |
| Status Affected:  | $\overline{TO}$ , $\overline{PD}$                                                                                                                  |                      |              |              |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0000</td><td>0000</td><td>0000</td><td>0100</td></tr></table>                                                            |                      |              |              | 0000 | 0000 | 0000 | 0100 |
| 0000              | 0000                                                                                                                                               | 0000                 | 0100         |              |      |      |      |      |
| Description:      | CLRWD instruction resets the Watchdog Timer. It also resets the postscaler of the WDT. Status bits, $\overline{TO}$ and $\overline{PD}$ , are set. |                      |              |              |      |      |      |      |
| Words:            | 1                                                                                                                                                  |                      |              |              |      |      |      |      |
| Cycles:           | 1                                                                                                                                                  |                      |              |              |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                    |                      |              |              |      |      |      |      |
|                   | Q1                                                                                                                                                 | Q2                   | Q3           | Q4           |      |      |      |      |
|                   | Decode                                                                                                                                             | No operation         | Process Data | No operation |      |      |      |      |

**Example:** CLRWDT

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

# PIC18F85J11 FAMILY

## COMF Complement f

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

Operands:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

Operation:  $\bar{f} \rightarrow \text{dest}$

Status Affected: N, Z

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0001 | 11da | ffff | ffff |
|------|------|------|------|

Description:

The contents of register 'f' are complemented. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See

**Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:

COMF REG, 0, 0

Before Instruction

REG = 13h

After Instruction

REG = 13h

W = ECh

## CPFSEQ Compare f with W, Skip if f = W

Syntax: CPFSEQ f {,a}

Operands:  $0 \leq f \leq 255$

$a \in [0, 1]$

Operation:  $(f) - (W)$ ,  
skip if  $(f) = (W)$   
(unsigned comparison)

Status Affected: None

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0110 | 001a | ffff | ffff |
|------|------|------|------|

Description:

Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.

If 'f' = W, then the fetched instruction is discarded and a NOP is executed instead, making this 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  $f \leq 95$  (5Fh). See

**Section 23.2.3 "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 | No operation |

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 CPFSEQ REG, 0

NEQUAL :

EQUAL :

Before Instruction

PC Address = HERE

W = ?

REG = ?

After Instruction

If REG = W;

PC = Address (EQUAL)

If REG  $\neq$  W;

PC = Address (NEQUAL)



# PIC18F85J11 FAMILY

## CPFSGT Compare f with W, Skip if f > W

|                  |                                                          |      |      |      |
|------------------|----------------------------------------------------------|------|------|------|
| Syntax:          | CPFSGT f {,a}                                            |      |      |      |
| Operands:        | $0 \leq f \leq 255$<br>$a \in [0, 1]$                    |      |      |      |
| Operation:       | (f) – (W),<br>skip if (f) > (W)<br>(unsigned comparison) |      |      |      |
| Status Affected: | None                                                     |      |      |      |
| Encoding:        | 0110                                                     | 010a | ffff | ffff |
| Description:     | Compares the contents of data memory                     |      |      |      |

Compares the contents of data memory location 'f' to the contents of the W by performing an unsigned subtraction.

If the contents of 'f' are greater than the contents of WREG, then the fetched instruction is discarded and a NOP is executed instead, making this 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  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

|         |                                                                             |
|---------|-----------------------------------------------------------------------------|
| Words:  | 1                                                                           |
| Cycles: | 1(2)<br><b>Note:</b> 3 cycles if skip and followed by a 2-word instruction. |

### Q Cycle Activity:

| Q1     | Q2                | Q3           | Q4           |
|--------|-------------------|--------------|--------------|
| Decode | Read register 'f' | Process Data | No operation |

### 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    CPFSGT REG, 0
NGREATER :
GREATER :
```

### Before Instruction

PC = Address (HERE)  
W = ?

### After Instruction

If REG > W;  
PC = Address (GREATER)  
If REG ≤ W;  
PC = Address (NGREATER)

## CPFSLT Compare f with W, Skip if f < W

|                  |                                                                                         |      |      |      |      |
|------------------|-----------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:          | CPFSLT f {,a}                                                                           |      |      |      |      |
| Operands:        | $0 \leq f \leq 255$<br>$a \in [0, 1]$                                                   |      |      |      |      |
| Operation:       | $(f) - (W)$ ,<br>skip if $(f) < (W)$<br>(unsigned comparison)                           |      |      |      |      |
| Status Affected: | None                                                                                    |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>0110</td><td>000a</td><td>ffff</td><td>ffff</td></tr></table> | 0110 | 000a | ffff | ffff |
| 0110             | 000a                                                                                    | ffff | ffff |      |      |
| Description:     | Compare f with W. If f is less than W, the processor skips the next instruction.        |      |      |      |      |

Compares the contents of data memory location 'f' to the contents of W by performing an unsigned subtraction.

If the contents of 'f' are less than the contents of W, then the fetched instruction is discarded and a NOP is executed instead, making this 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).

|         |                                                                             |
|---------|-----------------------------------------------------------------------------|
| Words:  | 1                                                                           |
| Cycles: | 1(2)<br><b>Note:</b> 3 cycles if skip and followed by a 2-word instruction. |

### Q Cycle Activity:

| Q1     | Q2                | Q3           | Q4           |
|--------|-------------------|--------------|--------------|
| Decode | Read register 'f' | Process Data | No operation |

### 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    CPFSLT REG, 1
NLESS   :
LESS    :
```

### Before Instruction

PC = Address (HERE)  
W = ?

### After Instruction

If REG < W;  
PC = Address (LESS)  
If REG ≥ W;  
PC = Address (NLESS)

# PIC18F85J11 FAMILY

## DAW Decimal Adjust W Register

| Syntax:           | DAW                                                                                                                                                                                                                                                           |              |         |      |      |        |                 |              |         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|---------|------|------|--------|-----------------|--------------|---------|
| Operands:         | None                                                                                                                                                                                                                                                          |              |         |      |      |        |                 |              |         |
| Operation:        | If $[W<3:0> > 9]$ or $[DC = 1]$ , then<br>$(W<3:0>) + 6 \rightarrow W<3:0>;$<br>else<br>$(W<3:0>) \rightarrow W<3:0>$<br><br>If $[W<7:4> > 9]$ or $[C = 1]$ , then<br>$(W<7:4>) + 6 \rightarrow W<7:4>;$<br>$C = 1;$<br>else<br>$(W<7:4>) \rightarrow W<7:4>$ |              |         |      |      |        |                 |              |         |
| Status Affected:  | C                                                                                                                                                                                                                                                             |              |         |      |      |        |                 |              |         |
| Encoding:         | <table><tr><td>0000</td><td>0000</td><td>0000</td><td>0111</td></tr></table>                                                                                                                                                                                  | 0000         | 0000    | 0000 | 0111 |        |                 |              |         |
| 0000              | 0000                                                                                                                                                                                                                                                          | 0000         | 0111    |      |      |        |                 |              |         |
| Description:      | DAW adjusts the eight-bit value in W, resulting from the earlier addition of two variables (each in packed BCD format) and produces a correct packed BCD result.                                                                                              |              |         |      |      |        |                 |              |         |
| Words:            | 1                                                                                                                                                                                                                                                             |              |         |      |      |        |                 |              |         |
| Cycles:           | 1                                                                                                                                                                                                                                                             |              |         |      |      |        |                 |              |         |
| Q Cycle Activity: | <table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register W</td><td>Process Data</td><td>Write W</td></tr></table>                                                                                                     | Q1           | Q2      | Q3   | Q4   | Decode | Read register W | Process Data | Write W |
| Q1                | Q2                                                                                                                                                                                                                                                            | Q3           | Q4      |      |      |        |                 |              |         |
| Decode            | Read register W                                                                                                                                                                                                                                               | Process Data | Write W |      |      |        |                 |              |         |

### Example 1: DAW

|                    |   |     |
|--------------------|---|-----|
| Before Instruction |   |     |
| W                  | = | A5h |
| C                  | = | 0   |
| DC                 | = | 0   |
| After Instruction  |   |     |
| W                  | = | 05h |
| C                  | = | 1   |
| DC                 | = | 0   |

### Example 2:

|                    |   |     |
|--------------------|---|-----|
| Before Instruction |   |     |
| W                  | = | CEh |
| C                  | = | 0   |
| DC                 | = | 0   |
| After Instruction  |   |     |
| W                  | = | 34h |
| C                  | = | 1   |
| DC                 | = | 0   |

## DECF Decrement f

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |      |      |      |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:           | DECF f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |      |      |
| Operands:         | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |      |      |      |
| Operation:        | $(f) - 1 \rightarrow \text{dest}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |      |      |      |
| Status Affected:  | C, DC, N, OV, Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0000</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0000 | 01da | ffff | ffff |
| 0000              | 01da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ffff | ffff |      |      |
| Description:      | <p>Decrement register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"</b> for details.</p> |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |      |      |      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |      |      |      |

### Example: DECF CNT, 1, 0

|                    |   |     |
|--------------------|---|-----|
| Before Instruction |   |     |
| CNT                | = | 01h |
| Z                  | = | 0   |
| After Instruction  |   |     |
| CNT                | = | 00h |
| Z                  | = | 1   |



# PIC18F85J11 FAMILY

## GOTO Unconditional Branch

Syntax: GOTO k

Operands:  $0 \leq k \leq 1048575$

Operation:  $k \rightarrow PC<20:1>$

Status Affected: None

Encoding:

1st word ( $k<7:0>$ )

2nd word ( $k<19:8>$ )

|      |             |          |          |
|------|-------------|----------|----------|
| 1110 | 1111        | $k_7kkk$ | $kkkk_0$ |
| 1111 | $k_{19}kkk$ | $kkkk$   | $kkkk_8$ |

Description: GOTO allows an unconditional branch anywhere within entire 2-Mbyte memory range. The 20-bit value, 'k', is loaded into  $PC<20:1>$ . GOTO is always a two-cycle instruction.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1           | Q2                     | Q3           | Q4                                  |
|--------------|------------------------|--------------|-------------------------------------|
| Decode       | Read literal 'k'<7:0>, | No operation | Read literal 'k'<19:8>, Write to PC |
| No operation | No operation           | No operation | No operation                        |

Example: GOTO THERE

After Instruction

PC = Address (THERE)

## INCF Increment f

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

Operands:  $0 \leq f \leq 255$

$d \in [0, 1]$

$a \in [0, 1]$

Operation:  $(f) + 1 \rightarrow \text{dest}$

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

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0010 | 10da | ffff | ffff |
|------|------|------|------|

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

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: INCF CNT, 1, 0

Before Instruction

CNT = FFh

Z = 0

C = ?

DC = ?

After Instruction

CNT = 00h

Z = 1

C = 1

DC = 1

# PIC18F85J11 FAMILY

## INCFSZ Increment f, Skip if 0

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |      |      |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|
| Syntax:          | INCFSZ f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |      |      |
| Operands:        | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |      |      |
| Operation:       | $(f) + 1 \rightarrow \text{dest}$ ,<br>skip if result = 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |      |      |      |
| Encoding:        | 0011                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 11da | ffff | ffff |
| Description:     | <p>The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f'. (default)</p> <p>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.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |      |      |
| Cycles:          | 1(2)<br><b>Note:</b> 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    INCFSZ   CNT, 1, 0  
                   NZERO    :  
                   ZERO     :

Before Instruction  
 PC = Address (HERE)  
 After Instruction  
 CNT = CNT + 1  
 If CNT = 0;  
 PC = Address (ZERO)  
 If CNT  $\neq$  0;  
 PC = Address (NZERO)

## INFSNZ Increment f, Skip if Not 0

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      |      |  |      |      |      |      |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|--|------|------|------|------|
| Syntax:          | INFSNZ f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |      |  |      |      |      |      |
| Operands:        | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |      |  |      |      |      |      |
| Operation:       | $(f) + 1 \rightarrow \text{dest}$ ,<br>skip if result $\neq 0$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |      |      |  |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |  |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>0100</td><td>10da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |      |  | 0100 | 10da | ffff | ffff |
| 0100             | 10da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | ffff | ffff |  |      |      |      |      |
| Description:     | <p>The contents of register 'f' are incremented. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).</p> <p>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.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"</b> for details.</p> |      |      |  |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |      |  |      |      |      |      |
| Cycles:          | 1(2)<br><b>Note:</b> 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    INFSNZ   REG, 1, 0  
                   ZERO  
                   NZERO

Before Instruction  
 PC = Address (HERE)  
 After Instruction  
 REG = REG + 1  
 If REG  $\neq$  0;  
 PC = Address (NZERO)  
 If REG = 0;  
 PC = Address (ZERO)

# PIC18F85J11 FAMILY

## IORLW Inclusive OR Literal with W

Syntax: IORLW k

Operands:  $0 \leq k \leq 255$

Operation: (W) .OR. k  $\rightarrow$  W

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 1001 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are ORed with the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4         |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

Example: IORLW 35h

Before Instruction  
W = 9Ah

After Instruction  
W = BFh

## IORWF Inclusive OR W with f

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

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

Operation: (W) .OR. (f)  $\rightarrow$  dest

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0001 | 00da | ffff | ffff |
|------|------|------|------|

Description: Inclusive OR W with register 'f'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: IORWF RESULT, 0, 1

Before Instruction  
RESULT = 13h  
W = 91h

After Instruction  
RESULT = 13h  
W = 93h

# PIC18F85J11 FAMILY

| LFSR              |      | Load FSR                                                                                                                                                                   |             |              |      |                                |      |      |          |      |  |  |  |
|-------------------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------|------|--------------------------------|------|------|----------|------|--|--|--|
| Syntax:           |      | LFSR f, k                                                                                                                                                                  |             |              |      |                                |      |      |          |      |  |  |  |
| Operands:         |      | $0 \leq f \leq 2$<br>$0 \leq k \leq 4095$                                                                                                                                  |             |              |      |                                |      |      |          |      |  |  |  |
| Operation:        |      | $k \rightarrow \text{FSRf}$                                                                                                                                                |             |              |      |                                |      |      |          |      |  |  |  |
| Status Affected:  |      | None                                                                                                                                                                       |             |              |      |                                |      |      |          |      |  |  |  |
| Encoding:         |      | <table><tr><td>1110</td><td>1110</td><td>00ff</td><td><math>k_{11}kkk</math></td></tr><tr><td>1111</td><td>0000</td><td><math>k_7kkk</math></td><td>kkkk</td></tr></table> | 1110        | 1110         | 00ff | $k_{11}kkk$                    | 1111 | 0000 | $k_7kkk$ | kkkk |  |  |  |
| 1110              | 1110 | 00ff                                                                                                                                                                       | $k_{11}kkk$ |              |      |                                |      |      |          |      |  |  |  |
| 1111              | 0000 | $k_7kkk$                                                                                                                                                                   | kkkk        |              |      |                                |      |      |          |      |  |  |  |
| Description:      |      | The 12-bit literal 'k' is loaded into the File Select Register pointed to by 'f'.                                                                                          |             |              |      |                                |      |      |          |      |  |  |  |
| Words:            |      | 2                                                                                                                                                                          |             |              |      |                                |      |      |          |      |  |  |  |
| Cycles:           |      | 2                                                                                                                                                                          |             |              |      |                                |      |      |          |      |  |  |  |
| Q Cycle Activity: |      |                                                                                                                                                                            |             |              |      |                                |      |      |          |      |  |  |  |
| Q1                |      | Q2                                                                                                                                                                         |             | Q3           |      | Q4                             |      |      |          |      |  |  |  |
| Decode            |      | Read literal 'k' MSB                                                                                                                                                       |             | Process Data |      | Write literal 'k' MSB to FSRfH |      |      |          |      |  |  |  |
| Decode            |      | Read literal 'k' LSB                                                                                                                                                       |             | Process Data |      | Write literal 'k' to FSRfL     |      |      |          |      |  |  |  |

**Example:** LFSR 2, 3ABh

After Instruction

FSR2H = 03h  
FSR2L = ABh

| MOVF              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Move f            |              |         |      |      |      |      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------|---------|------|------|------|------|
| Syntax:           | MOVF f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                   |              |         |      |      |      |      |
| Operands:         | 0 ≤ f ≤ 255<br>d ∈ [0, 1]<br>a ∈ [0, 1]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |         |      |      |      |      |
| Operation:        | f → dest                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                   |              |         |      |      |      |      |
| Status Affected:  | N, Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                   |              |         |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0101</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                   |              |         | 0101 | 00da | ffff | ffff |
| 0101              | 00da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | ffff              | ffff         |         |      |      |      |      |
| Description:      | <p>The contents of register 'f' are moved to a destination dependent upon the status of 'd'. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default). Location 'f' can be anywhere in the 256-byte bank.</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever f ≤ 95 (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                   |              |         |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |              |         |      |      |      |      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |              |         |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                   |              |         |      |      |      |      |
|                   | Q1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Q2                | Q3           | Q4      |      |      |      |      |
|                   | Decode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Read register 'f' | Process Data | Write W |      |      |      |      |

**Example:** MOVF REG, 0, 0

Before Instruction

REG = 22h  
W = FFh

After Instruction

REG = 22h  
W = 22h

# PIC18F85J11 FAMILY

## MOVFF

### Move f to f

Syntax: MOVFF  $f_s, f_d$

Operands:  $0 \leq f_s \leq 4095$   
 $0 \leq f_d \leq 4095$

Operation:  $(f_s) \rightarrow f_d$

Status Affected: None

Encoding:

1st word (source)

2nd word (destin.)

|      |      |      |                   |
|------|------|------|-------------------|
| 1100 | ffff | ffff | ffff <sub>s</sub> |
| 1111 | ffff | ffff | ffff <sub>d</sub> |

Description:

The contents of source register, ' $f_s$ ', are moved to destination register, ' $f_d$ '.

Location of source, ' $f_s$ ', can be anywhere in the 4096-byte data space (000h to FFFh) and location of destination, ' $f_d$ ', can also be anywhere from 000h to FFFh.

Either source or destination can be W (a useful special situation).

MOVFF is particularly useful for transferring a data memory location to a peripheral register (such as the transmit buffer or an I/O port).

The MOVFF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1     | Q2                            | Q3           | Q4                            |
|--------|-------------------------------|--------------|-------------------------------|
| Decode | Read register ' $f$ ' (src)   | Process Data | No operation                  |
| Decode | No operation<br>No dummy read | No operation | Write register ' $f$ ' (dest) |

### Example:

MOVFF REG1, REG2

Before Instruction

REG1 = 33h  
 REG2 = 11h

After Instruction

REG1 = 33h  
 REG2 = 33h

## MOVLB

### Move Literal to Low Nibble in BSR

Syntax: MOVLW k

Operands:  $0 \leq k \leq 255$

Operation:  $k \rightarrow \text{BSR}$

Status Affected: None

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0000 | 0001 | kkkk | kkkk |
|------|------|------|------|

Description:

The eight-bit literal 'k' is loaded into the Bank Select Register (BSR). The value of BSR<7:4> always remains '0' regardless of the value of  $k_{7:k_4}$ .

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4                       |
|--------|------------------|--------------|--------------------------|
| Decode | Read literal 'k' | Process Data | Write literal 'k' to BSR |

### Example:

MOVLB 5

Before Instruction

BSR Register = 02h

After Instruction

BSR Register = 05h



# PIC18F85J11 FAMILY

## MOVLW Move Literal to W

Syntax: MOVLW k

Operands:  $0 \leq k \leq 255$

Operation:  $k \rightarrow W$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 1110 | kkkk | kkkk |
|------|------|------|------|

Description: The eight-bit literal 'k' is loaded into W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2                  | Q3              | Q4            |
|--------|---------------------|-----------------|---------------|
| Decode | Read<br>literal 'k' | Process<br>Data | Write to<br>W |

Example: MOVLW 5Ah

After Instruction  
W = 5Ah

## MOVWF Move W to f

Syntax: MOVWF f{,a}

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

Operation:  $(W) \rightarrow f$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0110 | 111a | ffff | ffff |
|------|------|------|------|

Description: Move data from W to register 'f'.  
Location 'f' can be anywhere in the 256-byte bank.

If 'a' is '0', the Access Bank is selected.  
If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2                   | Q3              | Q4                    |
|--------|----------------------|-----------------|-----------------------|
| Decode | Read<br>register 'f' | Process<br>Data | Write<br>register 'f' |

Example: MOVWF REG, 0

Before Instruction

W = 4Fh  
REG = FFh

After Instruction

W = 4Fh  
REG = 4Fh

# PIC18F85J11 FAMILY

## MULLW Multiply Literal with W

Syntax: MULLW k

Operands:  $0 \leq k \leq 255$

Operation:  $(W) \times k \rightarrow \text{PRODH:PRODL}$

Status Affected: None

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0000 | 1101 | kkkk | kkkk |
|------|------|------|------|

Description: An unsigned multiplication is carried out between the contents of W and the 8-bit literal 'k'. The 16-bit result is placed in the PRODH:PRODL register pair. PRODH contains the high byte. W is unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4                           |
|--------|------------------|--------------|------------------------------|
| Decode | Read literal 'k' | Process Data | Write registers PRODH: PRODL |

Example: MULLW 0C4h

Before Instruction

|       |   |     |
|-------|---|-----|
| W     | = | E2h |
| PRODH | = | ?   |
| PRODL | = | ?   |

After Instruction

|       |   |     |
|-------|---|-----|
| W     | = | E2h |
| PRODH | = | ADh |
| PRODL | = | 08h |

## MULWF Multiply W with f

Syntax: MULWF f {,a}

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

Operation:  $(W) \times (f) \rightarrow \text{PRODH:PRODL}$

Status Affected: None

Encoding:

|      |      |      |      |
|------|------|------|------|
| 0000 | 001a | ffff | ffff |
|------|------|------|------|

Description: An unsigned multiplication is carried out between the contents of W and the register file location, 'f'. The 16-bit result is stored in the PRODH:PRODL register pair. PRODH contains the high byte. Both W and 'f' are unchanged.

None of the Status flags are affected.

Note that neither Overflow nor Carry is possible in this operation. A Zero result is possible but not detected.

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2                | Q3           | Q4                           |
|--------|-------------------|--------------|------------------------------|
| Decode | Read register 'f' | Process Data | Write registers PRODH: PRODL |

Example: MULWF REG, 1

Before Instruction

|       |   |     |
|-------|---|-----|
| W     | = | C4h |
| REG   | = | B5h |
| PRODH | = | ?   |
| PRODL | = | ?   |

After Instruction

|       |   |     |
|-------|---|-----|
| W     | = | C4h |
| REG   | = | B5h |
| PRODH | = | 8Ah |
| PRODL | = | 94h |

# PIC18F85J11 FAMILY

## NEGF Negate f

Syntax: NEGF f {,a}

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

Operation:  $(\bar{f}) + 1 \rightarrow f$

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0110 | 110a | ffff | ffff |
|------|------|------|------|

Description: Location 'f' is negated using two's complement. The result is placed in the data memory location 'f'.

If 'a' is '0', the Access Bank is selected.  
 If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: NEGF REG, 1

Before Instruction

REG = 0011 1010 [3Ah]

After Instruction

REG = 1100 0110 [C6h]

## NOP No Operation

Syntax: NOP

Operands: None

Operation: No operation

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0000 |
| 1111 | xxxx | xxxx | xxxx |

Description: No operation.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example:

None.

# PIC18F85J11 FAMILY

## POP Pop Top of Return Stack

|                  |                                                                                                                                                                                                                                                                                        |      |      |      |      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:          | POP                                                                                                                                                                                                                                                                                    |      |      |      |      |
| Operands:        | None                                                                                                                                                                                                                                                                                   |      |      |      |      |
| Operation:       | (TOS) → bit bucket                                                                                                                                                                                                                                                                     |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                                   |      |      |      |      |
| Encoding:        | <table><tr><td>0000</td><td>0000</td><td>0000</td><td>0110</td></tr></table>                                                                                                                                                                                                           | 0000 | 0000 | 0000 | 0110 |
| 0000             | 0000                                                                                                                                                                                                                                                                                   | 0000 | 0110 |      |      |
| Description:     | <p>The TOS value is pulled off the return stack and is discarded. The TOS value then becomes the previous value that was pushed onto the return stack.</p> <p>This instruction is provided to enable the user to properly manage the return stack to incorporate a software stack.</p> |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                      |      |      |      |      |
| Cycles:          | 1                                                                                                                                                                                                                                                                                      |      |      |      |      |

### Q Cycle Activity:

| Q1     | Q2           | Q3            | Q4           |
|--------|--------------|---------------|--------------|
| Decode | No operation | POP TOS value | No operation |

|                      |      |         |
|----------------------|------|---------|
| <u>Example:</u>      | POP  |         |
|                      | GOTO | NEW     |
| Before Instruction   |      |         |
| TOS                  | =    | 0031A2h |
| Stack (1 level down) | =    | 014332h |
| After Instruction    |      |         |
| TOS                  | =    | 014332h |
| PC                   | =    | NEW     |

## PUSH Push Top of Return Stack

|                  |                                                                                                                                                                                                                                     |      |      |      |      |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:          | PUSH                                                                                                                                                                                                                                |      |      |      |      |
| Operands:        | None                                                                                                                                                                                                                                |      |      |      |      |
| Operation:       | (PC + 2) → TOS                                                                                                                                                                                                                      |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                |      |      |      |      |
| Encoding:        | <table><tr><td>0000</td><td>0000</td><td>0000</td><td>0101</td></tr></table>                                                                                                                                                        | 0000 | 0000 | 0000 | 0101 |
| 0000             | 0000                                                                                                                                                                                                                                | 0000 | 0101 |      |      |
| Description:     | <p>The PC + 2 is pushed onto the top of the return stack. The previous TOS value is pushed down on the stack. This instruction allows implementing a software stack by modifying TOS and then pushing it onto the return stack.</p> |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                   |      |      |      |      |
| Cycles:          | 1                                                                                                                                                                                                                                   |      |      |      |      |

### Q Cycle Activity:

| Q1     | Q2                                  | Q3              | Q4              |
|--------|-------------------------------------|-----------------|-----------------|
| Decode | PUSH<br>PC + 2 onto<br>return stack | No<br>operation | No<br>operation |

|                      |      |       |
|----------------------|------|-------|
| <u>Example:</u>      | PUSH |       |
| Before Instruction   |      |       |
| TOS                  | =    | 345Ah |
| PC                   | =    | 0124h |
| After Instruction    |      |       |
| PC                   | =    | 0126h |
| TOS                  | =    | 0126h |
| Stack (1 level down) | =    | 345Ah |

# PIC18F85J11 FAMILY

## RCALL Relative Call

Syntax: RCALL n

Operands:  $-1024 \leq n \leq 1023$

Operation:  $(PC) + 2 \rightarrow TOS$ ,  
 $(PC) + 2 + 2n \rightarrow PC$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1101 | 1nnn | nnnn | nnnn |
|------|------|------|------|

Description: Subroutine call with a jump up to 1K from the current location. First, return address  $(PC + 2)$  is pushed onto the stack. Then, add the 2's complement number, '2n', to the PC. Since the PC will have incremented to fetch the next instruction, the new address will be  $PC + 2 + 2n$ . This instruction is a two-cycle instruction.

Words: 1

Cycles: 2

Q Cycle Activity:

| Q1           | Q2                                   | Q3           | Q4           |
|--------------|--------------------------------------|--------------|--------------|
| Decode       | Read literal 'n'<br>PUSH PC to stack | Process Data | Write to PC  |
| No operation | No operation                         | No operation | No operation |

Example:            HERE        RCALL Jump

Before Instruction

PC = Address (HERE)

After Instruction

PC = Address (Jump)

TOS = Address (HERE + 2)

## RESET Reset

Syntax: RESET

Operands: None

Operation: Reset all registers and flags that are affected by a MCLR Reset.

Status Affected: All

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 0000 | 1111 | 1111 |
|------|------|------|------|

Description: This instruction provides a way to execute a MCLR Reset in software.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2          | Q3           | Q4           |
|--------|-------------|--------------|--------------|
| Decode | Start Reset | No operation | No operation |

Example:            RESET

After Instruction

Registers = Reset Value

Flags\* = Reset Value

# PIC18F85J11 FAMILY

## RETFIE Return from Interrupt

**Syntax:** RETFIE {s}

**Operands:**  $s \in [0, 1]$

**Operation:** (TOS) → PC,  
 $1 \rightarrow \text{GIE/GIEH or PEIE/GIEL};$   
 if  $s = 1$ ,  
 (WS) → W,  
 (STATUS) → STATUS,  
 (BSRS) → BSR,  
 PCLATU, PCLATH are unchanged

**Status Affected:** GIE/GIEH, PEIE/GIEL.

**Encoding:**

|      |      |      |      |
|------|------|------|------|
| 0000 | 0000 | 0001 | 000s |
|------|------|------|------|

**Description:** Return from interrupt. Stack is popped and Top-of-Stack (TOS) is loaded into the PC. Interrupts are enabled by setting either the high or low priority global interrupt enable bit. If 's' = 1, the contents of the shadow registers, WS, STATUS and BSR, are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default).

**Words:** 1

**Cycles:** 2

Q Cycle Activity:

| Q1           | Q2           | Q3           | Q4                                    |
|--------------|--------------|--------------|---------------------------------------|
| Decode       | No operation | No operation | POP PC from stack<br>Set GIEH or GIEL |
| No operation | No operation | No operation | No operation                          |

**Example:** RETFIE 1

After Interrupt

|                     |   |         |
|---------------------|---|---------|
| PC                  | = | TOS     |
| W                   | = | WS      |
| BSR                 | = | BSRS    |
| STATUS              | = | STATUSS |
| GIE/GIEH, PEIE/GIEL | = | 1       |

## RETLW Return Literal to W

**Syntax:** RETLW k

**Operands:**  $0 \leq k \leq 255$

**Operation:**  $k \rightarrow W$ ,  
 (TOS) → PC,  
 PCLATU, PCLATH are unchanged

**Status Affected:** None

**Encoding:**

|      |      |      |      |
|------|------|------|------|
| 0000 | 1100 | kkkk | kkkk |
|------|------|------|------|

**Description:** W is loaded with the eight-bit literal 'k'. The program counter is loaded from the top of the stack (the return address). The high address latch (PCLATH) remains unchanged.

**Words:** 1

**Cycles:** 2

Q Cycle Activity:

| Q1           | Q2               | Q3           | Q4                               |
|--------------|------------------|--------------|----------------------------------|
| Decode       | Read literal 'k' | Process Data | POP PC from stack,<br>write to W |
| No operation | No operation     | No operation | No operation                     |

**Example:**

```
CALL TABLE ; W contains table
              ; offset value
              ; W now has
              ; table value

:
TABLE
  ADDWF PCL ; W = offset
  RETLW k0 ; Begin table
  RETLW k1 ;
:
:
  RETLW kn ; End of table
```

Before Instruction

W = 07h

After Instruction

W = value of kn

# PIC18F85J11 FAMILY

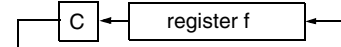
| RETURN            | Return from Subroutine                                                                                                                                                                                                                                                                                           |      |      |      |      |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|------|
| Syntax:           | RETURN {s}                                                                                                                                                                                                                                                                                                       |      |      |      |      |
| Operands:         | s ∈ [0, 1]                                                                                                                                                                                                                                                                                                       |      |      |      |      |
| Operation:        | (TOS) → PC;<br>if s = 1,<br>(WS) → W,<br>(STATUS) → STATUS,<br>(BSRS) → BSR,<br>PCLATU, PCLATH are unchanged                                                                                                                                                                                                     |      |      |      |      |
| Status Affected:  | None                                                                                                                                                                                                                                                                                                             |      |      |      |      |
| Encoding:         | <table><tr><td>0000</td><td>0000</td><td>0001</td><td>001s</td></tr></table>                                                                                                                                                                                                                                     | 0000 | 0000 | 0001 | 001s |
| 0000              | 0000                                                                                                                                                                                                                                                                                                             | 0001 | 001s |      |      |
| Description:      | Return from subroutine. The stack is popped and the top of the stack (TOS) is loaded into the program counter. If 's'= 1, the contents of the shadow registers, WS, STATUS and BSR, are loaded into their corresponding registers, W, STATUS and BSR. If 's' = 0, no update of these registers occurs (default). |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                |      |      |      |      |
| Cycles:           | 2                                                                                                                                                                                                                                                                                                                |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                  |      |      |      |      |

| Q1           | Q2           | Q3           | Q4                |
|--------------|--------------|--------------|-------------------|
| Decode       | No operation | Process Data | POP PC from stack |
| No operation | No operation | No operation | No operation      |

**Example:** RETURN

After Instruction:  
PC = TOS

| RLCF             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Rotate Left f through Carry |      |  |  |      |      |      |      |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|------|--|--|------|------|------|------|
| Syntax:          | RLCF f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                             |      |  |  |      |      |      |      |
| Operands:        | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                             |      |  |  |      |      |      |      |
| Operation:       | $(f<n>) \rightarrow \text{dest}<n + 1>$ ,<br>$(f<7>) \rightarrow C$ ,<br>$(C) \rightarrow \text{dest}<0>$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                             |      |  |  |      |      |      |      |
| Status Affected: | C, N, Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                             |      |  |  |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>0011</td><td>01da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                             |      |  |  | 0011 | 01da | ffff | ffff |
| 0011             | 01da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | ffff                        | ffff |  |  |      |      |      |      |
| Description:     | <p>The contents of register 'f' are rotated one bit to the left through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                             |      |  |  |      |      |      |      |



Words: 1  
Cycles: 1  
Q Cycle Activity:

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

**Example:** RLCF REG, 0, 0

Before Instruction  
REG = 1110 0110  
C = 0  
After Instruction  
REG = 1110 0110  
W = 1100 1100  
C = 1

# PIC18F85J11 FAMILY

## RLNCF Rotate Left f (No Carry)

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

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

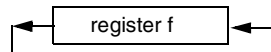
Operation:  $(f<n>) \rightarrow \text{dest}<n+1>$ ,  
 $(f<7>) \rightarrow \text{dest}<0>$

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0100 | 01da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are rotated one bit to the left. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is stored back in register 'f' (default).  
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).  
 If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** RLNCF REG, 1, 0

Before Instruction

REG = 1010 1011

After Instruction

REG = 0101 0111

## RRCF Rotate Right f through Carry

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

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

Operation:  $(f<n>) \rightarrow \text{dest}<n-1>$ ,  
 $(f<0>) \rightarrow C$ ,  
 $(C) \rightarrow \text{dest}<7>$

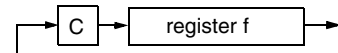
Status Affected: C, N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0011 | 00da | ffff | ffff |
|------|------|------|------|

Description: The contents of register 'f' are rotated one bit to the right through the Carry flag. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).  
 If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.



Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** RRCF REG, 0, 0

Before Instruction

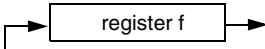
REG = 1110 0110  
 C = 0

After Instruction

REG = 1110 0110  
 W = 0111 0011  
 C = 0



# PIC18F85J11 FAMILY

| RRNCF             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Rotate Right f (No Carry) |                      |  |  |      |      |      |      |        |                   |              |                      |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|----------------------|--|--|------|------|------|------|--------|-------------------|--------------|----------------------|
| Syntax:           | RRNCF f {,d {,a}}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Operands:         | $0 \leq f \leq 255$<br>$d \in [0, 1]$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Operation:        | $(f<n>) \rightarrow \text{dest}<n-1>$ ,<br>$(f<0>) \rightarrow \text{dest}<7>$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Status Affected:  | N, Z                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Encoding:         | <table border="1"><tr><td>0100</td><td>00da</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                           |                      |  |  | 0100 | 00da | ffff | ffff |        |                   |              |                      |
| 0100              | 00da                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ffff                      | ffff                 |  |  |      |      |      |      |        |                   |              |                      |
| Description:      | <p>The contents of register 'f' are rotated one bit to the right. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed back in register 'f' (default).</p> <p>If 'a' is '0', the Access Bank will be selected, overriding the BSR value. If 'a' is '1', then the bank will be selected as per the BSR value (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> <div></div> |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                           |                      |  |  |      |      |      |      |        |                   |              |                      |
| Q Cycle Activity: | <table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read register 'f'</td><td>Process Data</td><td>Write to destination</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                           |                      |  |  | Q1   | Q2   | Q3   | Q4   | Decode | Read register 'f' | Process Data | Write to destination |
| Q1                | Q2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Q3                        | Q4                   |  |  |      |      |      |      |        |                   |              |                      |
| Decode            | Read register 'f'                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Process Data              | Write to destination |  |  |      |      |      |      |        |                   |              |                      |

**Example 1:** RRNCF REG, 1, 0

Before Instruction  
 REG = 1101 0111  
 After Instruction  
 REG = 1110 1011

**Example 2:** RRNCF REG, 0, 0

Before Instruction  
 W = ?  
 REG = 1101 0111  
 After Instruction  
 W = 1110 1011  
 REG = 1101 0111

| SETF              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Set f             |              |                    |  |      |      |      |      |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|--------------|--------------------|--|------|------|------|------|
| Syntax:           | SETF f {,a}                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                   |              |                    |  |      |      |      |      |
| Operands:         | $0 \leq f \leq 255$<br>$a \in [0, 1]$                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                   |              |                    |  |      |      |      |      |
| Operation:        | FFh $\rightarrow$ f                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                   |              |                    |  |      |      |      |      |
| Status Affected:  | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |              |                    |  |      |      |      |      |
| Encoding:         | <table border="1"><tr><td>0110</td><td>100a</td><td>ffff</td><td>ffff</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                   |                   |              |                    |  | 0110 | 100a | ffff | ffff |
| 0110              | 100a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | ffff              | ffff         |                    |  |      |      |      |      |
| Description:      | <p>The contents of the specified register are set to FFh.</p> <p>If 'a' is '0', the Access Bank is selected.</p> <p>If 'a' is '1', the BSR is used to select the GPR bank (default).</p> <p>If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever <math>f \leq 95</math> (5Fh). See <b>Section 23.2.3 “Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode”</b> for details.</p> |                   |              |                    |  |      |      |      |      |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                   |              |                    |  |      |      |      |      |
| Cycles:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                   |              |                    |  |      |      |      |      |
| Q Cycle Activity: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                   |              |                    |  |      |      |      |      |
|                   | Q1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Q2                | Q3           | Q4                 |  |      |      |      |      |
|                   | Decode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Read register 'f' | Process Data | Write register 'f' |  |      |      |      |      |

**Example:** SETF REG, 1

Before Instruction  
 REG = 5Ah  
 After Instruction  
 REG = FFh

# PIC18F85J11 FAMILY

## SLEEP Enter Sleep Mode

Syntax: SLEEP

Operands: None

Operation: 00h → WDT,  
0 → WDT postscaler,  
1 →  $\overline{TO}$ ,  
0 →  $\overline{PD}$

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 0000 | 0000 | 0011 |
|------|------|------|------|

Description: The Power-Down status bit ( $\overline{PD}$ ) is cleared. The Time-out status bit ( $\overline{TO}$ ) is set. The Watchdog Timer and its postscaler are cleared.

The processor is put into Sleep mode with the oscillator stopped.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2           | Q3           | Q4          |
|--------|--------------|--------------|-------------|
| Decode | No operation | Process Data | Go to Sleep |

**Example:** SLEEP

Before Instruction

$\overline{TO}$  = ?

$\overline{PD}$  = ?

After Instruction

$\overline{TO}$  = 1 †

$\overline{PD}$  = 0

† If WDT causes wake-up, this bit is cleared.

## SUBFWB Subtract f from W with Borrow

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

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

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

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0101 | 01da | ffff | ffff |
|------|------|------|------|

Description: Subtract register 'f' and Carry flag (borrow) from W (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example 1:** SUBFWB REG, 1, 0

Before Instruction

REG = 3

W = 2

C = 1

After Instruction

REG = FF

W = 2

C = 0

Z = 0

N = 1 ; result is negative

**Example 2:** SUBFWB REG, 0, 0

Before Instruction

REG = 2

W = 5

C = 1

After Instruction

REG = 2

W = 3

C = 1

Z = 0

N = 0 ; result is positive

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

Before Instruction

REG = 1

W = 2

C = 0

After Instruction

REG = 0

W = 2

C = 1

Z = 1 ; result is zero

N = 0

# PIC18F85J11 FAMILY

## SUBLW Subtract W from Literal

Syntax: SUBLW k

Operands:  $0 \leq k \leq 255$

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

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 1000 | kkkk | kkkk |
|------|------|------|------|

Description: W is subtracted from the eight-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4         |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

**Example 1:** SUBLW 02h

Before Instruction

W = 01h

C = ?

After Instruction

W = 01h

C = 1 ; result is positive

Z = 0

N = 0

**Example 2:** SUBLW 02h

Before Instruction

W = 02h

C = ?

After Instruction

W = 00h

C = 1 ; result is zero

Z = 1

N = 0

**Example 3:** SUBLW 02h

Before Instruction

W = 03h

C = ?

After Instruction

W = FFh ; (2's complement)

C = 0 ; result is negative

Z = 0

N = 1

## SUBWF Subtract W from f

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

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

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

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

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0101 | 11da | ffff | ffff |
|------|------|------|------|

Description: Subtract W from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example 1:** SUBWF REG, 1, 0

Before Instruction

REG = 3

W = 2

C = ?

After Instruction

REG = 1

W = 2

C = 1 ; result is positive

Z = 0

N = 0

**Example 2:** SUBWF REG, 0, 0

Before Instruction

REG = 2

W = 2

C = ?

After Instruction

REG = 2

W = 0

C = 1 ; result is zero

Z = 1

N = 0

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

Before Instruction

REG = 1

W = 2

C = ?

After Instruction

REG = FFh ; (2's complement)

W = 2

C = 0 ; result is negative

Z = 0

N = 1

# PIC18F85J11 FAMILY

## SUBWFB Subtract W from f with Borrow

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

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

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

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

Encoding: 

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

Description: Subtract W and the Carry flag (borrow) from register 'f' (2's complement method). If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

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

Before Instruction

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

After Instruction

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

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

Before Instruction

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

After Instruction

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

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

Before Instruction

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

After Instruction

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

## SWAPF Swap f

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

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

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

Status Affected: None

Encoding: 

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

Description: The upper and lower nibbles of register 'f' are exchanged. If 'd' is '0', the result is placed in W. If 'd' is '1', the result is placed in register 'f' (default).

If 'a' is '0', the Access Bank is selected. If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See **Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

**Example:** SWAPF REG, 1, 0

Before Instruction

REG = 53h

After Instruction

REG = 35h

# PIC18F85J11 FAMILY

## TBLRD Table Read

**Syntax:** TBLRD (\*; \*+; \*-; +\*)

**Operands:** None

**Operation:** if TBLRD \*,  
(Prog Mem (TBLPTR)) → TABLAT;  
TBLPTR – No Change  
if TBLRD \*+,  
(Prog Mem (TBLPTR)) → TABLAT;  
(TBLPTR) + 1 → TBLPTR  
if TBLRD \*-,  
(Prog Mem (TBLPTR)) → TABLAT;  
(TBLPTR) – 1 → TBLPTR  
if TBLRD +\*,  
(TBLPTR) + 1 → TBLPTR;  
(Prog Mem (TBLPTR)) → TABLAT

**Status Affected:** None

|           |      |      |      |                                           |
|-----------|------|------|------|-------------------------------------------|
| Encoding: | 0000 | 0000 | 0000 | 10nn<br>nn=0 *<br>=1 *+<br>=2 *-<br>=3 +* |
|-----------|------|------|------|-------------------------------------------|

**Description:** This instruction is used to read the contents of Program Memory (P.M.). To address the program memory, a pointer called Table Pointer (TBLPTR) is used.

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range.

TBLPTR<0> = 0:Least Significant Byte of program memory word

TBLPTR<0> = 1:Most Significant Byte of program memory word

The TBLRD instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

**Words:** 1

**Cycles:** 2

**Q Cycle Activity:**

| Q1           | Q2                                 | Q3           | Q4                          |
|--------------|------------------------------------|--------------|-----------------------------|
| Decode       | No operation                       | No operation | No operation                |
| No operation | No operation (Read Program Memory) | No operation | No operation (Write TABLAT) |

## TBLRD Table Read (Continued)

**Example 1:** TBLRD \*+ ;

Before Instruction

|                 |   |         |
|-----------------|---|---------|
| TABLAT          | = | 55h     |
| TBLPTR          | = | 00A356h |
| MEMORY(00A356h) | = | 34h     |

After Instruction

|        |   |         |
|--------|---|---------|
| TABLAT | = | 34h     |
| TBLPTR | = | 00A357h |

**Example 2:** TBLRD +\* ;

Before Instruction

|                 |   |         |
|-----------------|---|---------|
| TABLAT          | = | AAh     |
| TBLPTR          | = | 01A357h |
| MEMORY(01A357h) | = | 12h     |
| MEMORY(01A358h) | = | 34h     |

After Instruction

|        |   |         |
|--------|---|---------|
| TABLAT | = | 34h     |
| TBLPTR | = | 01A358h |

# PIC18F85J11 FAMILY

## TBLWT Table Write

**Syntax:** TBLWT (\*, \*+, \*-; +\*)

**Operands:** None

**Operation:** if TBLWT\*,  
(TABLAT) → Holding Register;  
TBLPTR – No Change  
if TBLWT\*+,  
(TABLAT) → Holding Register;  
(TBLPTR) + 1 → TBLPTR  
if TBLWT\*-,  
(TABLAT) → Holding Register;  
(TBLPTR) – 1 → TBLPTR  
if TBLWT\*+,  
(TBLPTR) + 1 → TBLPTR;  
(TABLAT) → Holding Register

**Status Affected:** None

|           |      |      |      |                                           |
|-----------|------|------|------|-------------------------------------------|
| Encoding: | 0000 | 0000 | 0000 | 11nn<br>nn=0 *<br>=1 *+<br>=2 *-<br>=3 +* |
|-----------|------|------|------|-------------------------------------------|

**Description:** This instruction uses the 3 LSBs of TBLPTR to determine which of the 8 holding registers the TABLAT is written to. The holding registers are used to program the contents of Program Memory (P.M.). (Refer to **Section 5.0 “Memory Organization”** for additional details on programming Flash memory.)

The TBLPTR (a 21-bit pointer) points to each byte in the program memory. TBLPTR has a 2-Mbyte address range. The LSB of the TBLPTR selects which byte of the program memory location to access.

TBLPTR<0> = 0: Least Significant Byte of program memory word

TBLPTR<0> = 1: Most Significant Byte of program memory word

The TBLWT instruction can modify the value of TBLPTR as follows:

- no change
- post-increment
- post-decrement
- pre-increment

**Words:** 1

**Cycles:** 2

**Q Cycle Activity:**

| Q1           | Q2                         | Q3           | Q4                                       |
|--------------|----------------------------|--------------|------------------------------------------|
| Decode       | No operation               | No operation | No operation                             |
| No operation | No operation (Read TABLAT) | No operation | No operation (Write to Holding Register) |

## TBLWT Table Write (Continued)

**Example 1:** TBLWT \*+;

Before Instruction

|                            |   |         |
|----------------------------|---|---------|
| TABLAT                     | = | 55h     |
| TBLPTR                     | = | 00A356h |
| HOLDING REGISTER (00A356h) | = | FFh     |

After Instructions (table write completion)

|                            |   |         |
|----------------------------|---|---------|
| TABLAT                     | = | 55h     |
| TBLPTR                     | = | 00A357h |
| HOLDING REGISTER (00A356h) | = | 55h     |

**Example 2:** TBLWT \*+;

Before Instruction

|                            |   |         |
|----------------------------|---|---------|
| TABLAT                     | = | 34h     |
| TBLPTR                     | = | 01389Ah |
| HOLDING REGISTER (01389Ah) | = | FFh     |
| HOLDING REGISTER (01389Bh) | = | FFh     |

After Instruction (table write completion)

|                            |   |         |
|----------------------------|---|---------|
| TABLAT                     | = | 34h     |
| TBLPTR                     | = | 01389Bh |
| HOLDING REGISTER (01389Ah) | = | FFh     |
| HOLDING REGISTER (01389Bh) | = | 34h     |

# PIC18F85J11 FAMILY

## TSTFSZ Test f, Skip if 0

Syntax: TSTFSZ f {,a}

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

Operation: skip if  $f = 0$

Status Affected: None

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0110 | 011a | ffff | ffff |
|------|------|------|------|

Description: If 'f' = 0, the next instruction fetched during the current instruction execution is discarded and a NOP is executed, making this 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  $f \leq 95$  (5Fh). See **Section 23.2.3 "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 | No operation |

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    TSTFSZ  CNT, 1
NZERO   :
ZERO    :
```

Before Instruction

PC = Address (HERE)

After Instruction

If CNT = 00h,

PC = Address (ZERO)

If CNT ≠ 00h,

PC = Address (NZERO)

## XORLW Exclusive OR Literal with W

Syntax: XORLW k

Operands:  $0 \leq k \leq 255$

Operation: (W) .XOR. k → W

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0000 | 1010 | kkkk | kkkk |
|------|------|------|------|

Description: The contents of W are XORed with the 8-bit literal 'k'. The result is placed in W.

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2               | Q3           | Q4         |
|--------|------------------|--------------|------------|
| Decode | Read literal 'k' | Process Data | Write to W |

**Example:** XORLW 0AFh

Before Instruction

W = B5h

After Instruction

W = 1Ah

# PIC18F85J11 FAMILY

## XORWF Exclusive OR W with f

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

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

Operation: (W) .XOR. (f)  $\rightarrow$  dest

Status Affected: N, Z

Encoding: 

|      |      |      |      |
|------|------|------|------|
| 0001 | 10da | ffff | ffff |
|------|------|------|------|

Description: Exclusive OR the contents of W with register 'f'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in the register 'f' (default).

If 'a' is '0', the Access Bank is selected.  
If 'a' is '1', the BSR is used to select the GPR bank (default).

If 'a' is '0' and the extended instruction set is enabled, this instruction operates in Indexed Literal Offset Addressing mode whenever  $f \leq 95$  (5Fh). See

**Section 23.2.3 "Byte-Oriented and Bit-Oriented Instructions in Indexed Literal Offset Mode"** for details.

Words: 1

Cycles: 1

Q Cycle Activity:

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

Example: XORWF REG, 1, 0

Before Instruction

REG = AFh  
W = B5h

After Instruction

REG = 1Ah  
W = B5h



## 23.2 Extended Instruction Set

In addition to the standard 75 instructions of the PIC18 instruction set, the PIC18F85J11 family of devices also provides an optional extension to the core CPU functionality. The added features include eight additional instructions that augment Indirect and Indexed Addressing operations and the implementation of Indexed Literal Offset Addressing for many of the standard PIC18 instructions.

The additional features of the extended instruction set are enabled by default on unprogrammed devices. Users must properly set or clear the XINST Configuration bit during programming to enable or disable these features.

The instructions in the extended set can all be classified as literal operations, which either manipulate the File Select Registers, or use them for Indexed Addressing. Two of the instructions, `ADDFSR` and `SUBFSR`, each have an additional special instantiation for using FSR2. These versions (`ADDULNK` and `SUBULNK`) allow for automatic return after execution.

The extended instructions are specifically implemented to optimize re-entrant program code (that is, code that is recursive or that uses a software stack) written in high-level languages, particularly C. Among other things, they allow users working in high-level languages to perform certain operations on data structures more efficiently. These include:

- Dynamic allocation and deallocation of software stack space when entering and leaving subroutines
- Function Pointer invocation
- Software Stack Pointer manipulation
- Manipulation of variables located in a software stack

A summary of the instructions in the extended instruction set is provided in Table 23-3. Detailed descriptions are provided in **Section 23.2.2 “Extended Instruction Set”**. The opcode field descriptions in Table 23-1 (page 286) apply to both the standard and extended PIC18 instruction sets.

**Note:** The instruction set extension and the Indexed Literal Offset Addressing mode were designed for optimizing applications written in C. The user may likely never use these instructions directly in the assembler. The syntax for these commands is provided as a reference for users who may be reviewing code that has been generated by a compiler.

### 23.2.1 EXTENDED INSTRUCTION SYNTAX

Most of the extended instructions use indexed arguments, using one of the File Select Registers and some offset to specify a source or destination register. When an argument for an instruction serves as part of Indexed Addressing, it is enclosed in square brackets (“[ ]”). This is done to indicate that the argument is used as an index or offset. The MPASM™ Assembler will flag an error if it determines that an index or offset value is not bracketed.

When the extended instruction set is enabled, brackets are also used to indicate index arguments in byte-oriented and bit-oriented instructions. This is in addition to other changes in their syntax. For more details, see **Section 23.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**.

**Note:** In the past, square brackets have been used to denote optional arguments in the PIC18 and earlier instruction sets. In this text, and going forward, optional arguments are denoted by braces (“{ }”).

**TABLE 23-3: EXTENSIONS TO THE PIC18 INSTRUCTION SET**

| Mnemonic,<br>Operands                        | Description                                                                             | Cycles | 16-Bit Instruction Word |      |      |      | Status<br>Affected |
|----------------------------------------------|-----------------------------------------------------------------------------------------|--------|-------------------------|------|------|------|--------------------|
|                                              |                                                                                         |        | MSb                     |      | LSb  |      |                    |
| ADDFSR    f, k                               | Add Literal to FSR                                                                      | 1      | 1110                    | 1000 | ffkk | kkkk | None               |
| ADDULNK    k                                 | Add Literal to FSR2 and Return                                                          | 2      | 1110                    | 1000 | 11kk | kkkk | None               |
| CALLW                                        | Call Subroutine using WREG                                                              | 2      | 0000                    | 0000 | 0001 | 0100 | None               |
| MOVSF        z <sub>s</sub> , f <sub>d</sub> | Move    z <sub>s</sub> (source) to    1st word<br>f <sub>d</sub> (destination) 2nd word | 2      | 1110                    | 1011 | 0zzz | zzzz | None               |
| MOVSS        z <sub>s</sub> , z <sub>d</sub> | Move    z <sub>s</sub> (source) to    1st word<br>z <sub>d</sub> (destination) 2nd word | 2      | 1110                    | 1011 | 1zzz | zzzz | None               |
| PUSHL        k                               | Store Literal at FSR2, Decrement FSR2                                                   | 1      | 1110                    | 1010 | kkkk | kkkk | None               |
| SUBFSR       f, k                            | Subtract Literal from FSR                                                               | 1      | 1110                    | 1001 | ffkk | kkkk | None               |
| SUBULNK    k                                 | Subtract Literal from FSR2 and Return                                                   | 2      | 1110                    | 1001 | 11kk | kkkk | None               |

# PIC18F85J11 FAMILY

## 23.2.2 EXTENDED INSTRUCTION SET

### ADDFSR Add Literal to FSR

|                   |                                                                             |                     |                 |                 |
|-------------------|-----------------------------------------------------------------------------|---------------------|-----------------|-----------------|
| Syntax:           | ADDFSR f, k                                                                 |                     |                 |                 |
| Operands:         | $0 \leq k \leq 63$<br>$f \in [0, 1, 2]$                                     |                     |                 |                 |
| Operation:        | $FSR(f) + k \rightarrow FSR(f)$                                             |                     |                 |                 |
| Status Affected:  | None                                                                        |                     |                 |                 |
| Encoding:         | 1110                                                                        | 1000                | ffkk            | kkkk            |
| Description:      | The 6-bit literal 'k' is added to the contents of the FSR specified by 'f'. |                     |                 |                 |
| Words:            | 1                                                                           |                     |                 |                 |
| Cycles:           | 1                                                                           |                     |                 |                 |
| Q Cycle Activity: |                                                                             |                     |                 |                 |
|                   | Q1                                                                          | Q2                  | Q3              | Q4              |
|                   | Decode                                                                      | Read<br>literal 'k' | Process<br>Data | Write to<br>FSR |

**Example:**           ADDFSR 2, 23h

Before Instruction  
FSR2 = 03FFh  
After Instruction  
FSR2 = 0422h

### ADDULNK Add Literal to FSR2 and Return

| Syntax:           | ADDULNK k                                                                                                                                                                                                                                                                                                                                                             |              |              |      |      |        |                  |              |              |              |              |              |              |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------|------|------|--------|------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Operands:         | $0 \leq k \leq 63$                                                                                                                                                                                                                                                                                                                                                    |              |              |      |      |        |                  |              |              |              |              |              |              |
| Operation:        | $FSR2 + k \rightarrow FSR2$ ,<br>(TOS) $\rightarrow$ PC                                                                                                                                                                                                                                                                                                               |              |              |      |      |        |                  |              |              |              |              |              |              |
| Status Affected:  | None                                                                                                                                                                                                                                                                                                                                                                  |              |              |      |      |        |                  |              |              |              |              |              |              |
| Encoding:         | <table><tr><td>1110</td><td>1000</td><td>11kk</td><td>kkkk</td></tr></table>                                                                                                                                                                                                                                                                                          | 1110         | 1000         | 11kk | kkkk |        |                  |              |              |              |              |              |              |
| 1110              | 1000                                                                                                                                                                                                                                                                                                                                                                  | 11kk         | kkkk         |      |      |        |                  |              |              |              |              |              |              |
| Description:      | <p>The 6-bit literal 'k' is added to the contents of FSR2. A RETURN is then executed by loading the PC with the TOS.</p> <p>The instruction takes two cycles to execute; a NOP is performed during the second cycle.</p> <p>This may be thought of as a special case of the ADDFSR instruction, where <math>f = 3</math> (binary '11'); it operates only on FSR2.</p> |              |              |      |      |        |                  |              |              |              |              |              |              |
| Words:            | 1                                                                                                                                                                                                                                                                                                                                                                     |              |              |      |      |        |                  |              |              |              |              |              |              |
| Cycles:           | 2                                                                                                                                                                                                                                                                                                                                                                     |              |              |      |      |        |                  |              |              |              |              |              |              |
| Q Cycle Activity: | <table><tr><th>Q1</th><th>Q2</th><th>Q3</th><th>Q4</th></tr><tr><td>Decode</td><td>Read literal 'k'</td><td>Process Data</td><td>Write to FSR</td></tr><tr><td>No Operation</td><td>No Operation</td><td>No Operation</td><td>No Operation</td></tr></table>                                                                                                          | Q1           | Q2           | Q3   | Q4   | Decode | Read literal 'k' | Process Data | Write to FSR | No Operation | No Operation | No Operation | No Operation |
| Q1                | Q2                                                                                                                                                                                                                                                                                                                                                                    | Q3           | Q4           |      |      |        |                  |              |              |              |              |              |              |
| Decode            | Read literal 'k'                                                                                                                                                                                                                                                                                                                                                      | Process Data | Write to FSR |      |      |        |                  |              |              |              |              |              |              |
| No Operation      | No Operation                                                                                                                                                                                                                                                                                                                                                          | No Operation | No Operation |      |      |        |                  |              |              |              |              |              |              |

**Example:**           ADDULNK 23h

Before Instruction  
FSR2 = 03FFh  
PC = 0100h  
After Instruction  
FSR2 = 0422h  
PC = (TOS)

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

# PIC18F85J11 FAMILY

## CALLW Subroutine Call Using WREG

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                       |      |      |  |      |      |      |      |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|--|------|------|------|------|
| Syntax:          | CALLW                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |      |  |      |      |      |      |
| Operands:        | None                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |  |      |      |      |      |
| Operation:       | (PC + 2) → TOS,<br>(W) → PCL,<br>(PCLATH) → PCH,<br>(PCLATU) → PCU                                                                                                                                                                                                                                                                                                                                                                    |      |      |  |      |      |      |      |
| Status Affected: | None                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |      |  |      |      |      |      |
| Encoding:        | <table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>0100</td></tr></table>                                                                                                                                                                                                                                                                                                                                               |      |      |  | 0000 | 0000 | 0001 | 0100 |
| 0000             | 0000                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0001 | 0100 |  |      |      |      |      |
| Description      | <p>First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a <code>NOP</code> instruction while the new next instruction is fetched.</p> <p>Unlike <code>CALL</code>, there is no option to update W, STATUS or BSR.</p> |      |      |  |      |      |      |      |
| Words:           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |      |  |      |      |      |      |
| Cycles:          | 2                                                                                                                                                                                                                                                                                                                                                                                                                                     |      |      |  |      |      |      |      |

Q Cycle Activity:

| Q1           | Q2           | Q3               | Q4           |
|--------------|--------------|------------------|--------------|
| Decode       | Read WREG    | Push PC to stack | No operation |
| No operation | No operation | No operation     | No operation |

**Example:**                      HERE              CALLW

Before Instruction

PC        =    address (HERE)  
PCLATH =    10h  
PCLATU =    00h  
W        =    06h

After Instruction

PC        =    001006h  
TOS      =    address (HERE + 2)  
PCLATH =    10h  
PCLATU =    00h  
W        =    06h

## MOVSF Move Indexed to f

|                    |                                                       |      |      |                   |
|--------------------|-------------------------------------------------------|------|------|-------------------|
| Syntax:            | MOVSF [z <sub>s</sub> ], f <sub>d</sub>               |      |      |                   |
| Operands:          | 0 ≤ z <sub>s</sub> ≤ 127<br>0 ≤ f <sub>d</sub> ≤ 4095 |      |      |                   |
| Operation:         | ((FSR2) + z <sub>s</sub> ) → f <sub>d</sub>           |      |      |                   |
| Status Affected:   | None                                                  |      |      |                   |
| Encoding:          |                                                       |      |      |                   |
| 1st word (source)  | 1110                                                  | 1011 | 0zzz | zzzz <sub>s</sub> |
| 2nd word (destin.) | 1111                                                  | ffff | ffff | ffff <sub>d</sub> |

**Description:** The contents of the source register are moved to destination register, 'f<sub>d</sub>'. The actual address of the source register is determined by adding the 7-bit literal offset, 'z<sub>s</sub>', in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal, 'f<sub>d</sub>', in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).

The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1     | Q2                            | Q3                    | Q4                        |
|--------|-------------------------------|-----------------------|---------------------------|
| Decode | Determine source addr         | Determine source addr | Read source reg           |
| Decode | No operation<br>No dummy read | No operation          | Write register 'f' (dest) |

**Example:**                      MOVSF    [05h], REG2

Before Instruction

FSR2        =    80h  
Contents of 85h    =    33h  
REG2        =    11h

After Instruction

FSR2        =    80h  
Contents of 85h    =    33h  
REG2        =    33h

# PIC18F85J11 FAMILY

## MOVSS Move Indexed to Indexed

|                   |                                                |      |      |                   |
|-------------------|------------------------------------------------|------|------|-------------------|
| Syntax:           | MOVSS [z <sub>s</sub> ], [z <sub>d</sub> ]     |      |      |                   |
| Operands:         | $0 \leq z_s \leq 127$<br>$0 \leq z_d \leq 127$ |      |      |                   |
| Operation:        | $((FSR2) + z_s) \rightarrow ((FSR2) + z_d)$    |      |      |                   |
| Status Affected:  | None                                           |      |      |                   |
| Encoding:         |                                                |      |      |                   |
| 1st word (source) | 1110                                           | 1011 | 1zzz | zzzz <sub>s</sub> |
| 2nd word (dest.)  | 1111                                           | xxxx | xzzz | zzzz <sub>d</sub> |

Description

The contents of the source register are moved to the destination register. The addresses of the source and destination registers are determined by adding the 7-bit literal offsets, 'z<sub>s</sub>' or 'z<sub>d</sub>', respectively, to the value of FSR2. Both registers can be located anywhere in the 4096-byte data memory space (000h to FFFh).

The MOVSS instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.

If the resultant source address points to an Indirect Addressing register, the value returned will be 00h. If the resultant destination address points to an Indirect Addressing register, the instruction will execute as a NOP.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1     | Q2                    | Q3                    | Q4                |
|--------|-----------------------|-----------------------|-------------------|
| Decode | Determine source addr | Determine source addr | Read source reg   |
| Decode | Determine dest addr   | Determine dest addr   | Write to dest reg |

**Example:** MOVSS [05h], [06h]

Before Instruction

FSR2 = 80h  
 Contents of 85h = 33h  
 Contents of 86h = 11h

After Instruction

FSR2 = 80h  
 Contents of 85h = 33h  
 Contents of 86h = 33h

## PUSHL Store Literal at FSR2, Decrement FSR2

|                  |                                                                                                                                                                                                                |      |      |      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------|------|
| Syntax:          | PUSHL k                                                                                                                                                                                                        |      |      |      |
| Operands:        | $0 \leq k \leq 255$                                                                                                                                                                                            |      |      |      |
| Operation:       | $k \rightarrow (FSR2),$<br>$FSR2 - 1 \rightarrow FSR2$                                                                                                                                                         |      |      |      |
| Status Affected: | None                                                                                                                                                                                                           |      |      |      |
| Encoding:        | 1111                                                                                                                                                                                                           | 1010 | kkkk | kkkk |
| Description:     | <p>The 8-bit literal 'k' is written to the data memory address specified by FSR2. FSR2 is decremented by 1 after the operation.</p> <p>This instruction allows users to push values onto a software stack.</p> |      |      |      |

Words: 1

Cycles: 1

Q Cycle Activity:

| Q1     | Q2       | Q3           | Q4                   |
|--------|----------|--------------|----------------------|
| Decode | Read 'k' | Process data | Write to destination |

**Example:** PUSHL 08h

Before Instruction

FSR2H:FSR2L = 01ECh  
 Memory (01ECh) = 00h

After Instruction

FSR2H:FSR2L = 01EBh  
 Memory (01ECh) = 08h

# PIC18F85J11 FAMILY

## SUBFSR Subtract Literal from FSR

Syntax: SUBFSR f, k  
 Operands:  $0 \leq k \leq 63$   
 $f \in [0, 1, 2]$   
 Operation:  $FSRf - k \rightarrow FSRf$   
 Status Affected: None  
 Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1110 | 1001 | ffkk | kkkk |
|------|------|------|------|

  
 Description: The 6-bit literal, 'k', is subtracted from the contents of the FSR specified by 'f'.  
 Words: 1  
 Cycles: 1  
 Q Cycle Activity:

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

**Example:** SUBFSR 2, 23h

Before Instruction  
 FSR2 = 03FFh  
 After Instruction  
 FSR2 = 03DCh

## SUBULNK Subtract Literal from FSR2 and Return

Syntax: SUBULNK k  
 Operands:  $0 \leq k \leq 63$   
 Operation:  $FSR2 - k \rightarrow FSR2$ ,  
 (TOS)  $\rightarrow PC$   
 Status Affected: None  
 Encoding: 

|      |      |      |      |
|------|------|------|------|
| 1110 | 1001 | 11kk | kkkk |
|------|------|------|------|

  
 Description: The 6-bit literal, 'k', is subtracted from the contents of the FSR2. A RETURN is then executed by loading the PC with the TOS.

The instruction takes two cycles to execute; a NOP is performed during the second cycle.

This may be thought of as a special case of the SUBFSR instruction, where  $f = 3$  (binary '11'); it operates only on FSR2.

Words: 1  
 Cycles: 2  
 Q Cycle Activity:

| Q1           | Q2                | Q3           | Q4                   |
|--------------|-------------------|--------------|----------------------|
| Decode       | Read register 'f' | Process Data | Write to destination |
| No Operation | No Operation      | No Operation | No Operation         |

**Example:** SUBULNK 23h

Before Instruction  
 FSR2 = 03FFh  
 PC = 0100h  
 After Instruction  
 FSR2 = 03DCh  
 PC = (TOS)

# PIC18F85J11 FAMILY

## 23.2.3 BYTE-ORIENTED AND BIT-ORIENTED INSTRUCTIONS IN INDEXED LITERAL OFFSET MODE

**Note:** Enabling the PIC18 instruction set extension may cause legacy applications to behave erratically or fail entirely.

In addition to eight new commands in the extended set, enabling the extended instruction set also enables Indexed Literal Offset Addressing (**Section 5.6.1 “Indexed Addressing with Literal Offset”**). This has a significant impact on the way that many commands of the standard PIC18 instruction set are interpreted.

When the extended set is disabled, addresses embedded in opcodes are treated as literal memory locations: either as a location in the Access Bank ( $a = 0$ ) or in a GPR bank designated by the BSR ( $a = 1$ ). When the extended instruction set is enabled and  $a = 0$ , however, a File Select Register argument of 5Fh or less is interpreted as an offset from the pointer value in FSR2 and not as a literal address. For practical purposes, this means that all instructions that use the Access RAM bit as an argument – that is, all byte-oriented and bit-oriented instructions, or almost half of the core PIC18 instructions – may behave differently when the extended instruction set is enabled.

When the content of FSR2 is 00h, the boundaries of the Access RAM are essentially remapped to their original values. This may be useful in creating backward-compatible code. If this technique is used, it may be necessary to save the value of FSR2 and restore it when moving back and forth between C and assembly routines in order to preserve the Stack Pointer. Users must also keep in mind the syntax requirements of the extended instruction set (see **Section 23.2.3.1 “Extended Instruction Syntax with Standard PIC18 Commands”**).

Although the Indexed Literal Offset mode can be very useful for dynamic stack and pointer manipulation, it can also be very annoying if a simple arithmetic operation is carried out on the wrong register. Users who are accustomed to the PIC18 programming must keep in mind that, when the extended instruction set is enabled, register addresses of 5Fh or less are used for Indexed Literal Offset Addressing.

Representative examples of typical byte-oriented and bit-oriented instructions in the Indexed Literal Offset mode are provided on the following page to show how execution is affected. The operand conditions shown in the examples are applicable to all instructions of these types.

## 23.2.3.1 Extended Instruction Syntax with Standard PIC18 Commands

When the extended instruction set is enabled, the File Select Register argument, ‘f’, in the standard, byte-oriented and bit-oriented commands is replaced with the literal offset value, ‘k’. As already noted, this occurs only when ‘f’ is less than or equal to 5Fh. When an offset value is used, it must be indicated by square brackets (“[ ]”). As with the extended instructions, the use of brackets indicates to the compiler that the value is to be interpreted as an index or an offset. Omitting the brackets, or using a value greater than 5Fh within the brackets, will generate an error in the MPASM Assembler.

If the index argument is properly bracketed for Indexed Literal Offset Addressing, the Access RAM argument is never specified; it will automatically be assumed to be ‘0’. This is in contrast to standard operation (extended instruction set disabled) when ‘a’ is set on the basis of the target address. Declaring the Access RAM bit in this mode will also generate an error in the MPASM Assembler.

The destination argument, ‘d’, functions as before.

In the latest versions of the MPASM Assembler, language support for the extended instruction set must be explicitly invoked. This is done with either the command line option, `/Y`, or the PE directive in the source listing.

## 23.2.4 CONSIDERATIONS WHEN ENABLING THE EXTENDED INSTRUCTION SET

It is important to note that the extensions to the instruction set may not be beneficial to all users. In particular, users who are not writing code that uses a software stack may not benefit from using the extensions to the instruction set.

Additionally, the Indexed Literal Offset Addressing mode may create issues with legacy applications written to the PIC18 assembler. This is because instructions in the legacy code may attempt to address registers in the Access Bank below 5Fh. Since these addresses are interpreted as literal offsets to FSR2 when the instruction set extension is enabled, the application may read or write to the wrong data addresses.

When porting an application to the PIC18F85J11 family, it is very important to consider the type of code. A large, re-entrant application that is written in C and would benefit from efficient compilation will do well when using the instruction set extensions. Legacy applications that heavily use the Access Bank will most likely not benefit from using the extended instruction set.

# PIC18F85J11 FAMILY

## ADDWF ADD W to Indexed (Indexed Literal Offset mode)

**Syntax:** ADDWF [k] {,d}

**Operands:**  $0 \leq k \leq 95$   
 $d \in [0, 1]$

**Operation:**  $(W) + ((FSR2) + k) \rightarrow \text{dest}$

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

**Encoding:**

|      |      |      |      |
|------|------|------|------|
| 0010 | 01d0 | kkkk | kkkk |
|------|------|------|------|

**Description:** The contents of W are added to the contents of the register indicated by FSR2, offset by the value, 'k'.  
 If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).

**Words:** 1

**Cycles:** 1

**Q Cycle Activity:**

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

**Example:** ADDWF [OFST], 0

**Before Instruction**

|                   |   |       |
|-------------------|---|-------|
| W                 | = | 17h   |
| OFST              | = | 2Ch   |
| FSR2              | = | 0A00h |
| Contents of 0A2Ch | = | 20h   |

**After Instruction**

|                   |   |     |
|-------------------|---|-----|
| W                 | = | 37h |
| Contents of 0A2Ch | = | 20h |

## BSF Bit Set Indexed (Indexed Literal Offset mode)

**Syntax:** BSF [k], b

**Operands:**  $0 \leq f \leq 95$   
 $0 \leq b \leq 7$

**Operation:**  $1 \rightarrow ((FSR2) + k) \langle b \rangle$

**Status Affected:** None

**Encoding:**

|      |      |      |      |
|------|------|------|------|
| 1000 | bbb0 | kkkk | kkkk |
|------|------|------|------|

**Description:** Bit 'b' of the register indicated by FSR2, offset by the value, 'k', is set.

**Words:** 1

**Cycles:** 1

**Q Cycle Activity:**

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

**Example:** BSF [FLAG\_OFST], 7

**Before Instruction**

|                   |   |       |
|-------------------|---|-------|
| FLAG_OFST         | = | 0Ah   |
| FSR2              | = | 0A00h |
| Contents of 0A0Ah | = | 55h   |

**After Instruction**

|                   |   |     |
|-------------------|---|-----|
| Contents of 0A0Ah | = | D5h |
|-------------------|---|-----|

## SETF Set Indexed (Indexed Literal Offset mode)

**Syntax:** SETF [k]

**Operands:**  $0 \leq k \leq 95$

**Operation:**  $FFh \rightarrow ((FSR2) + k)$

**Status Affected:** None

**Encoding:**

|      |      |      |      |
|------|------|------|------|
| 0110 | 1000 | kkkk | kkkk |
|------|------|------|------|

**Description:** The contents of the register indicated by FSR2, offset by 'k', are set to FFh.

**Words:** 1

**Cycles:** 1

**Q Cycle Activity:**

| Q1     | Q2       | Q3           | Q4             |
|--------|----------|--------------|----------------|
| Decode | Read 'k' | Process Data | Write register |

**Example:** SETF [OFST]

**Before Instruction**

|                   |   |       |
|-------------------|---|-------|
| OFST              | = | 2Ch   |
| FSR2              | = | 0A00h |
| Contents of 0A2Ch | = | 00h   |

**After Instruction**

|                   |   |     |
|-------------------|---|-----|
| Contents of 0A2Ch | = | FFh |
|-------------------|---|-----|

# PIC18F85J11 FAMILY

---

## 23.2.5 SPECIAL CONSIDERATIONS WITH MICROCHIP MPLAB® IDE TOOLS

The latest versions of Microchip's software tools have been designed to fully support the extended instruction set for the PIC18F85J11 family. This includes the MPLAB C18 C Compiler, MPASM assembly language and MPLAB Integrated Development Environment (IDE).

When selecting a target device for software development, MPLAB IDE will automatically set default Configuration bits for that device. The default setting for the XINST Configuration bit is '0', disabling the extended instruction set and Indexed Literal Offset Addressing. For proper execution of applications developed to take advantage of the extended instruction set, XINST must be set during programming.

To develop software for the extended instruction set, the user must enable support for the instructions and the Indexed Addressing mode in their language tool(s). Depending on the environment being used, this may be done in several ways:

- A menu option or dialog box within the environment that allows the user to configure the language tool and its settings for the project
- A command line option
- A directive in the source code

These options vary between different compilers, assemblers and development environments. Users are encouraged to review the documentation accompanying their development systems for the appropriate information.



## 24.0 DEVELOPMENT SUPPORT

The PIC® microcontrollers are supported with a full range of hardware and software development tools:

- Integrated Development Environment
  - MPLAB® IDE Software
- Assemblers/Compilers/Linkers
  - MPASM™ Assembler
  - MPLAB C18 and MPLAB C30 C Compilers
  - MPLINK™ Object Linker/  
MPLIB™ Object Librarian
  - MPLAB ASM30 Assembler/Linker/Library
- Simulators
  - MPLAB SIM Software Simulator
- Emulators
  - MPLAB ICE 2000 In-Circuit Emulator
  - MPLAB REAL ICE™ In-Circuit Emulator
- In-Circuit Debugger
  - MPLAB ICD 2
- Device Programmers
  - PICSTART® Plus Development Programmer
  - MPLAB PM3 Device Programmer
  - PICKit™ 2 Development Programmer
- Low-Cost Demonstration and Development Boards and Evaluation Kits

## 24.1 MPLAB Integrated Development Environment Software

The MPLAB IDE software brings an ease of software development previously unseen in the 8/16-bit microcontroller market. The MPLAB IDE is a Windows® operating system-based application that contains:

- A single graphical interface to all debugging tools
  - Simulator
  - Programmer (sold separately)
  - Emulator (sold separately)
  - In-Circuit Debugger (sold separately)
- A full-featured editor with color-coded context
- A multiple project manager
- Customizable data windows with direct edit of contents
- High-level source code debugging
- Visual device initializer for easy register initialization
- Mouse over variable inspection
- Drag and drop variables from source to watch windows
- Extensive on-line help
- Integration of select third party tools, such as HI-TECH Software C Compilers and IAR C Compilers

The MPLAB IDE allows you to:

- Edit your source files (either assembly or C)
- One touch assemble (or compile) and download to PIC MCU emulator and simulator tools (automatically updates all project information)
- Debug using:
  - Source files (assembly or C)
  - Mixed assembly and C
  - Machine code

MPLAB IDE supports multiple debugging tools in a single development paradigm, from the cost-effective simulators, through low-cost in-circuit debuggers, to full-featured emulators. This eliminates the learning curve when upgrading to tools with increased flexibility and power.

# PIC18F85J11 FAMILY

---

## 24.2 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for all PIC MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multi-purpose source files
- Directives that allow complete control over the assembly process

## 24.3 MPLAB C18 and MPLAB C30 C Compilers

The MPLAB C18 and MPLAB C30 Code Development Systems are complete ANSI C compilers for Microchip's PIC18 and PIC24 families of microcontrollers and the dsPIC30 and dsPIC33 family of digital signal controllers. These compilers provide powerful integration capabilities, superior code optimization and ease of use not found with other compilers.

For easy source level debugging, the compilers provide symbol information that is optimized to the MPLAB IDE debugger.

## 24.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler and the MPLAB C18 C Compiler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

## 24.5 MPLAB ASM30 Assembler, Linker and Librarian

MPLAB ASM30 Assembler produces relocatable machine code from symbolic assembly language for dsPIC30F devices. MPLAB C30 C Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire dsPIC30F instruction set
- Support for fixed-point and floating-point data
- Command line interface
- Rich directive set
- Flexible macro language
- MPLAB IDE compatibility

## 24.6 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C18 and MPLAB C30 C Compilers, and the MPASM and MPLAB ASM30 Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 24.7 MPLAB ICE 2000 High-Performance In-Circuit Emulator

The MPLAB ICE 2000 In-Circuit Emulator is intended to provide the product development engineer with a complete microcontroller design tool set for PIC microcontrollers. Software control of the MPLAB ICE 2000 In-Circuit Emulator is advanced by the MPLAB Integrated Development Environment, which allows editing, building, downloading and source debugging from a single environment.

The MPLAB ICE 2000 is a full-featured emulator system with enhanced trace, trigger and data monitoring features. Interchangeable processor modules allow the system to be easily reconfigured for emulation of different processors. The architecture of the MPLAB ICE 2000 In-Circuit Emulator allows expansion to support new PIC microcontrollers.

The MPLAB ICE 2000 In-Circuit Emulator system has been designed as a real-time emulation system with advanced features that are typically found on more expensive development tools. The PC platform and Microsoft® Windows® 32-bit operating system were chosen to best make these features available in a simple, unified application.

## 24.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC® and MCU devices. It debugs and programs PIC® and dsPIC® Flash microcontrollers with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The MPLAB REAL ICE probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with the popular MPLAB ICD 2 system (RJ11) or with the new high speed, noise tolerant, low-voltage differential signal (LVDS) interconnection (CAT5).

MPLAB REAL ICE is field upgradeable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added, such as software breakpoints and assembly code trace. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, real-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 24.9 MPLAB ICD 2 In-Circuit Debugger

Microchip's In-Circuit Debugger, MPLAB ICD 2, is a powerful, low-cost, run-time development tool, connecting to the host PC via an RS-232 or high-speed USB interface. This tool is based on the Flash PIC MCUs and can be used to develop for these and other PIC MCUs and dsPIC DSCs. The MPLAB ICD 2 utilizes the in-circuit debugging capability built into the Flash devices. This feature, along with Microchip's In-Circuit Serial Programming™ (ICSP™) protocol, offers cost-effective, in-circuit Flash debugging from the graphical user interface of the MPLAB Integrated Development Environment. This enables a designer to develop and debug source code by setting breakpoints, single stepping and watching variables, and CPU status and peripheral registers. Running at full speed enables testing hardware and applications in real time. MPLAB ICD 2 also serves as a development programmer for selected PIC devices.

## 24.10 MPLAB PM3 Device Programmer

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

# PIC18F85J11 FAMILY

---

## 24.11 PICSTART Plus Development Programmer

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

## 24.12 PICkit 2 Development Programmer

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

## 24.13 Demonstration, Development and Evaluation Boards

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

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

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

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

Check the Microchip web page ([www.microchip.com](http://www.microchip.com)) and the latest *"Product Selector Guide"* (DS00148) for the complete list of demonstration, development and evaluation kits.

## 25.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings<sup>(†)</sup>

|                                                                                                                    |                       |
|--------------------------------------------------------------------------------------------------------------------|-----------------------|
| Ambient temperature under bias .....                                                                               | -40°C to +100°C       |
| Storage temperature .....                                                                                          | -65°C to +150°C       |
| Voltage on any digital only I/O pin or $\overline{\text{MCLR}}$ with respect to VSS (except VDD) .....             | -0.3V to 6.0V         |
| Voltage on any combined digital and analog pin with respect to VSS (except VDD and $\overline{\text{MCLR}}$ )..... | -0.3V to (VDD + 0.3V) |
| Voltage on VDDCORE with respect to VSS .....                                                                       | -0.3V to 2.75V        |
| Voltage on VDD with respect to VSS .....                                                                           | -0.3V to 3.6V         |
| Total power dissipation ( <b>Note 1</b> ) .....                                                                    | 1.0W                  |
| Maximum current out of VSS pin .....                                                                               | 300 mA                |
| Maximum current into VDD pin .....                                                                                 | 250 mA                |
| Maximum output current sunk by PORTA<7:6> and any PORTB and PORTC I/O pins.....                                    | 25 mA                 |
| Maximum output current sunk by any PORTD, PORTE and PORTJ I/O pins .....                                           | 8 mA                  |
| Maximum output current sunk by PORTA<5:0> and any PORTF, PORTG and PORTH I/O pins .....                            | 2 mA                  |
| Maximum output current sourced by PORTA<7:6> and any PORTB and PORTC I/O pins .....                                | 25 mA                 |
| Maximum output current sourced by any PORTD, PORTE and PORTJ I/O pins .....                                        | 8 mA                  |
| Maximum output current sourced by PORTA<5:0> and any PORTF, PORTG and PORTH I/O pins .....                         | 2 mA                  |
| Maximum current sunk by all ports combined .....                                                                   | 200 mA                |

**Note 1:** Power dissipation is calculated as follows:

$$P_{dis} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$$

† **NOTICE:** Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

# PIC18F85J11 FAMILY

**FIGURE 25-1: PIC18F85J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR ENABLED (INDUSTRIAL)<sup>(1)</sup>**

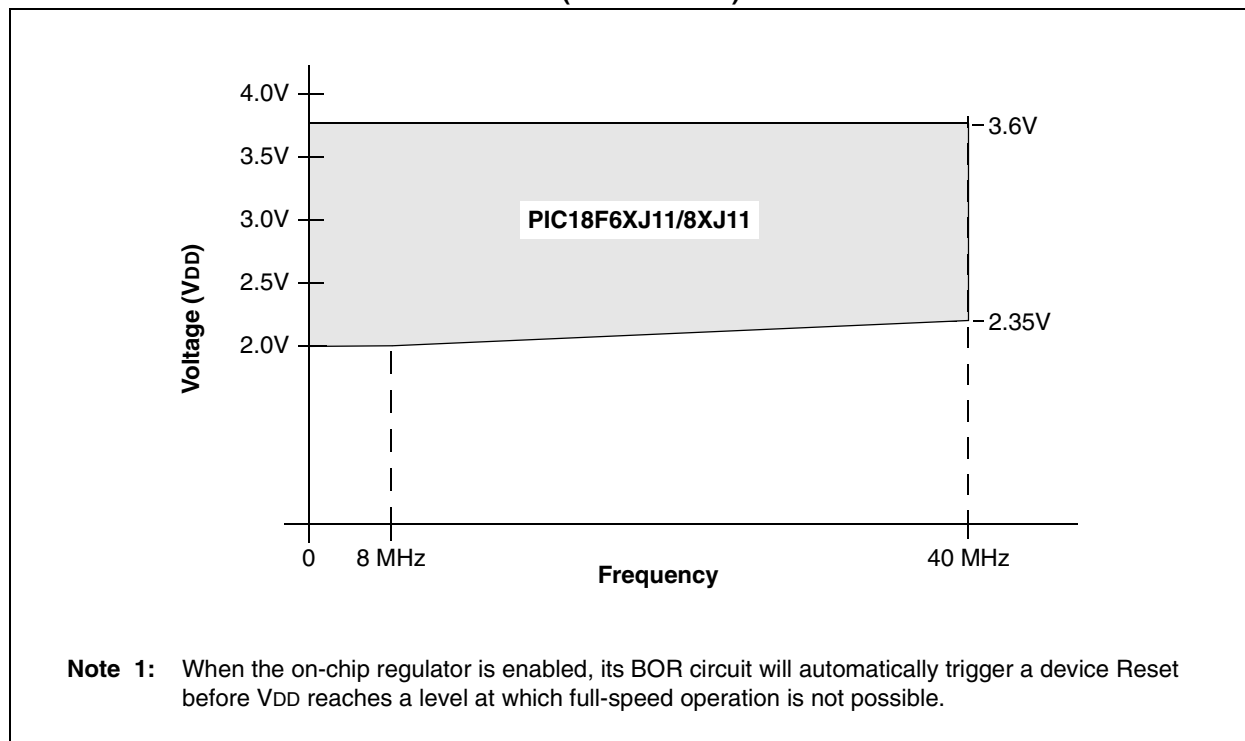
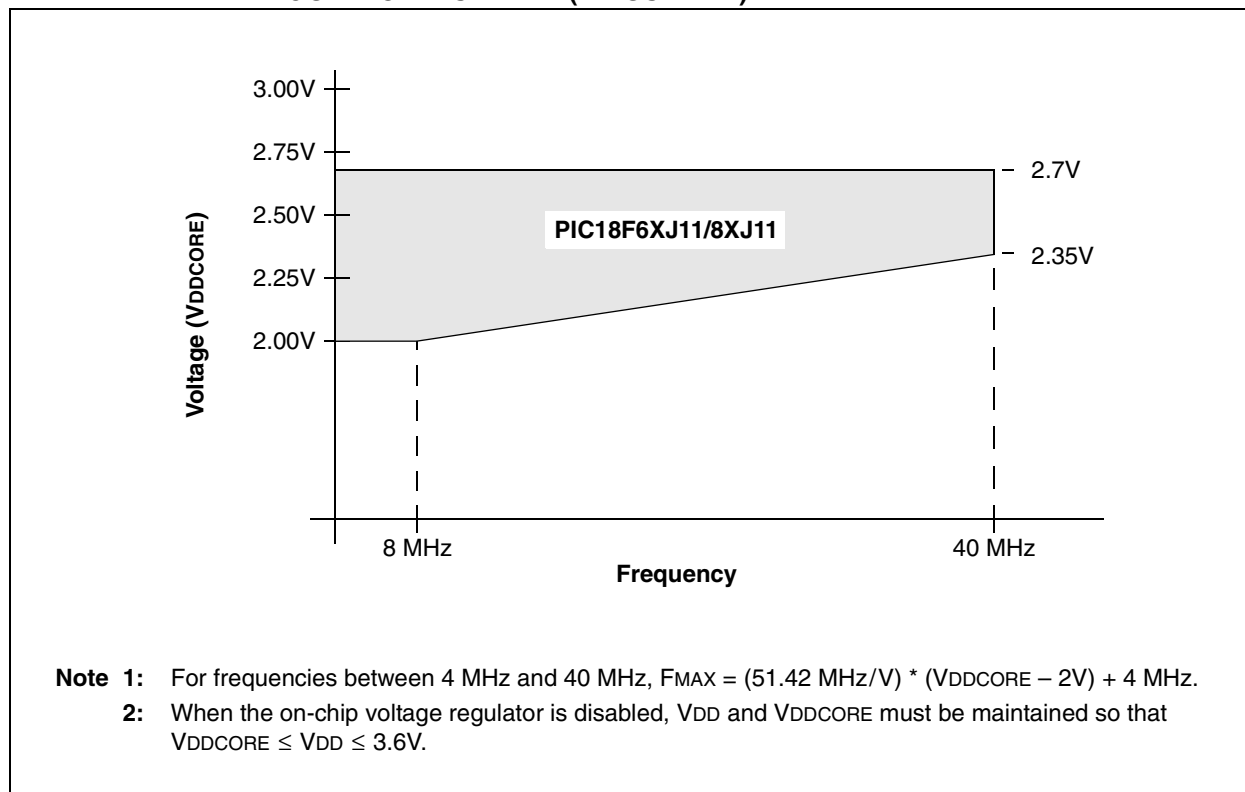


**FIGURE 25-2: PIC18F85J11 FAMILY VOLTAGE-FREQUENCY GRAPH, REGULATOR DISABLED (INDUSTRIAL)<sup>(1,2)</sup>**



# PIC18F85J11 FAMILY

## 25.1 DC Characteristics: Supply Voltage PIC18F85J11 Family (Industrial)

| PIC18F85J11 Family<br>(Industrial) |         |                                                                         | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |        |            |        |                                                           |
|------------------------------------|---------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|--------|------------|--------|-----------------------------------------------------------|
| Param No.                          | Symbol  | Characteristic                                                          | Min                                                                                                                                                     | Typ    | Max        | Units  | Conditions                                                |
| D001                               | VDD     | <b>Supply Voltage</b>                                                   | VDDCORE<br>2.0                                                                                                                                          | —<br>— | 3.6<br>3.6 | V<br>V | ENVREG tied to VSS<br>ENVREG tied to VDD                  |
| D001B                              | VDDCORE | <b>External Supply for Microcontroller Core</b>                         | 2.0                                                                                                                                                     | —      | 2.70       | V      | ENVREG tied to VSS                                        |
| D001C                              | AVDD    | <b>Analog Supply Voltage</b>                                            | VDD – 0.3                                                                                                                                               | —      | VDD + 0.3  | V      |                                                           |
| D001D                              | AVSS    | <b>Analog Ground Potential</b>                                          | VSS – 0.3                                                                                                                                               | —      | VSS + 0.3  | V      |                                                           |
| D002                               | VDR     | <b>RAM Data Retention Voltage<sup>(1)</sup></b>                         | 1.5                                                                                                                                                     | —      | —          | V      |                                                           |
| D003                               | VPOR    | <b>VDD Start Voltage</b><br>to ensure internal<br>Power-on Reset signal | —                                                                                                                                                       | —      | 0.7        | V      | See <b>Section 4.3 “Power-on Reset (POR)”</b> for details |
| D004                               | SVDD    | <b>VDD Rise Rate</b><br>to ensure internal<br>Power-on Reset signal     | 0.05                                                                                                                                                    | —      | —          | V/ms   | See <b>Section 4.3 “Power-on Reset (POR)”</b> for details |
| D005                               | VBOR    | <b>Brown-out Reset Voltage</b>                                          | —                                                                                                                                                       | 1.9    | —          | V      |                                                           |

**Note 1:** This is the limit to which VDD can be lowered in Sleep mode, or during a device Reset, without losing RAM data.

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial)

| PIC18F85J11 Family<br>(Industrial) |                                                               | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |     |               |                       |
|------------------------------------|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------|-----------------------|
| Param No.                          | Device                                                        | Typ                                                                                                                                                     | Max | Units         | Conditions            |
|                                    | <b>Power-Down Current (<math>I_{PD}</math>)<sup>(1)</sup></b> |                                                                                                                                                         |     |               |                       |
|                                    | All devices                                                   | 0.2                                                                                                                                                     | 0.9 | $\mu\text{A}$ | $-40^{\circ}\text{C}$ |
|                                    |                                                               | 0.1                                                                                                                                                     | 0.9 | $\mu\text{A}$ | $+25^{\circ}\text{C}$ |
|                                    |                                                               | 2.4                                                                                                                                                     | 5   | $\mu\text{A}$ | $+85^{\circ}\text{C}$ |
|                                    | All devices                                                   | 0.5                                                                                                                                                     | 0.9 | $\mu\text{A}$ | $-40^{\circ}\text{C}$ |
|                                    |                                                               | 0.1                                                                                                                                                     | 0.9 | $\mu\text{A}$ | $+25^{\circ}\text{C}$ |
|                                    |                                                               | 2.7                                                                                                                                                     | 5   | $\mu\text{A}$ | $+85^{\circ}\text{C}$ |
|                                    | All devices                                                   | 2.7                                                                                                                                                     | 6   | $\mu\text{A}$ | $-40^{\circ}\text{C}$ |
|                                    |                                                               | 3.5                                                                                                                                                     | 6   | $\mu\text{A}$ | $+25^{\circ}\text{C}$ |
|                                    |                                                               | 6.7                                                                                                                                                     | 12  | $\mu\text{A}$ | $+85^{\circ}\text{C}$ |
|                                    |                                                               |                                                                                                                                                         |     |               |                       |
|                                    |                                                               |                                                                                                                                                         |     |               |                       |

**Legend:** TBD = To Be Determined

- 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.  
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;  
MCLR = VDD; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of  $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to VSS).
- 5:** Voltage regulator enabled (ENVREG tied to VDD).



# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated) |      |                                   |            |                                              |                                                                             |                                                                             |
|------------------------------------|-------------------------------------|---------------------------------------------------------|------|-----------------------------------|------------|----------------------------------------------|-----------------------------------------------------------------------------|-----------------------------------------------------------------------------|
|                                    |                                     | Operating temperature                                   |      | -40°C ≤ TA ≤ +85°C for industrial |            |                                              |                                                                             |                                                                             |
| Param No.                          | Device                              | Typ                                                     | Max  | Units                             | Conditions |                                              |                                                                             |                                                                             |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                         |      |                                   |            |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 6.5                                                     | 16   | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 31 kHz<br>( <b>RC_RUN</b> mode,<br>internal oscillator<br>source)    |                                                                             |
|                                    |                                     | 7                                                       | 16   | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 9.5                                                     | 20   | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 10                                                      | 18   | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                             |                                                                             |
|                                    |                                     | 10.5                                                    | 18   | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 12.5                                                    | 24   | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 41                                                      | 100  | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                             |                                                                             |
|                                    |                                     | 52                                                      | 100  | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 71                                                      | 110  | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 359                                                     | 750  | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> |                                                                             | FOSC = 1 MHz<br>( <b>INTOSC_RUN</b> mode,<br>internal oscillator<br>source) |
|                                    |                                     | 387                                                     | 750  | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 407                                                     | 840  | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 438                                                     | 850  | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                             |                                                                             |
|                                    |                                     | 470                                                     | 850  | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 491                                                     | 910  | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 486                                                     | 900  | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                             |                                                                             |
|                                    |                                     | 526                                                     | 900  | μA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 564                                                     | 990  | μA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 0.76                                                    | 1.45 | mA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 4 MHz<br>( <b>INTOSC_RUN</b> mode,<br>internal oscillator<br>source) |                                                                             |
|                                    |                                     | 0.84                                                    | 1.45 | mA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 0.9                                                     | 1.6  | mA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 1.1                                                     | 1.63 | mA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                             |                                                                             |
|                                    |                                     | 1.18                                                    | 1.63 | mA                                | +25°C      |                                              |                                                                             |                                                                             |
|                                    |                                     | 1.24                                                    | 1.75 | mA                                | +85°C      |                                              |                                                                             |                                                                             |
|                                    | All devices                         | 1.25                                                    | 1.86 | mA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                             |                                                                             |
|                                    |                                     | 1.29                                                    | 1.86 | mA                                | +25°C      |                                              |                                                                             |                                                                             |
| 1.37                               |                                     | 1.94                                                    | mA   | +85°C                             |            |                                              |                                                                             |                                                                             |

**Legend:** TBD = To Be Determined

- 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.  
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;  
MCLR = VDD; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to VSS).
- 5:** Voltage regulator enabled (ENVREG tied to VDD).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated) |       |                                   |            |                                              |                                                                      |
|------------------------------------|-------------------------------------|---------------------------------------------------------|-------|-----------------------------------|------------|----------------------------------------------|----------------------------------------------------------------------|
|                                    |                                     | Operating temperature                                   |       | -40°C ≤ TA ≤ +85°C for industrial |            |                                              |                                                                      |
| Param No.                          | Device                              | Typ                                                     | Max   | Units                             | Conditions |                                              |                                                                      |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                         |       |                                   |            |                                              |                                                                      |
|                                    | All devices                         | 2.4                                                     | 8     | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 31 kHz<br>(RC_IDLE mode,<br>internal oscillator<br>source)    |
|                                    |                                     | 2.5                                                     | 8     | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 4.8                                                     | 12    | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 3.2                                                     | 9     | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                      |
|                                    |                                     | 3.2                                                     | 9     | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 6                                                       | 14    | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 62                                                      | 82    | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                      |
|                                    |                                     | 42                                                      | 82    | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 59                                                      | 97    | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 251                                                     | 570   | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 1 MHz<br>(INTOSC_IDLE mode,<br>internal oscillator<br>source) |
|                                    |                                     | 264                                                     | 570   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 272                                                     | 590   | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 284                                                     | 610   | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                      |
|                                    |                                     | 284                                                     | 610   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 293                                                     | 650   | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 295                                                     | 710   | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                      |
|                                    |                                     | 323                                                     | 710   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 392                                                     | 790   | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 368                                                     | 760   | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 4 MHz<br>(INTOSC_IDLE mode,<br>internal oscillator<br>source) |
|                                    |                                     | 362                                                     | 760   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 370                                                     | 800   | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 400                                                     | 850   | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                      |
|                                    |                                     | 410                                                     | 850   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 418                                                     | 900   | μA                                | +85°C      |                                              |                                                                      |
|                                    | All devices                         | 460                                                     | 950   | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                      |
|                                    |                                     | 462                                                     | 950   | μA                                | +25°C      |                                              |                                                                      |
|                                    |                                     | 486                                                     | 1,000 | μA                                | +85°C      |                                              |                                                                      |

**Legend:** TBD = To Be Determined

- 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.  
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;  
MCLR = VDD; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to VSS).
- 5:** Voltage regulator enabled (ENVREG tied to VDD).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated)      |     |       |            |                                              |                                                           |
|------------------------------------|-------------------------------------|--------------------------------------------------------------|-----|-------|------------|----------------------------------------------|-----------------------------------------------------------|
|                                    |                                     | Operating temperature      -40°C ≤ TA ≤ +85°C for industrial |     |       |            |                                              |                                                           |
| Param No.                          | Device                              | Typ                                                          | Max | Units | Conditions |                                              |                                                           |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                              |     |       |            |                                              |                                                           |
|                                    | All devices                         | 165                                                          | 490 | μA    | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 1 MHz<br>( <b>PRI_RUN</b> mode,<br>EC oscillator)  |
|                                    |                                     | 180                                                          | 490 | μA    | +25°C      |                                              |                                                           |
|                                    |                                     | 200                                                          | 490 | μA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 256                                                          | 670 | μA    | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                           |
|                                    |                                     | 260                                                          | 670 | μA    | +25°C      |                                              |                                                           |
|                                    |                                     | 280                                                          | 670 | μA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 460                                                          | 850 | μA    | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                           |
|                                    |                                     | 456                                                          | 850 | μA    | +25°C      |                                              |                                                           |
|                                    |                                     | 482                                                          | 850 | μA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 0.63                                                         | 2.2 | mA    | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 4 MHz<br>( <b>PRI_RUN</b> mode,<br>EC oscillator)  |
|                                    |                                     | 0.68                                                         | 2.2 | mA    | +25°C      |                                              |                                                           |
|                                    |                                     | 0.74                                                         | 2.2 | mA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 0.91                                                         | 2.5 | mA    | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                           |
|                                    |                                     | 1.04                                                         | 2.5 | mA    | +25°C      |                                              |                                                           |
|                                    |                                     | 1.04                                                         | 2.5 | mA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 1.32                                                         | 3.0 | mA    | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                           |
|                                    |                                     | 1.32                                                         | 3.0 | mA    | +25°C      |                                              |                                                           |
|                                    |                                     | 1.41                                                         | 3.0 | mA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 7.47                                                         | 14  | mA    | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> | FOSC = 40 MHz<br>( <b>PRI_RUN</b> mode,<br>EC oscillator) |
|                                    |                                     | 5.81                                                         | 14  | mA    | +25°C      |                                              |                                                           |
|                                    |                                     | 6.32                                                         | 13  | mA    | +85°C      |                                              |                                                           |
|                                    | All devices                         | 8.84                                                         | 18  | mA    | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                           |
|                                    |                                     | 8.66                                                         | 18  | mA    | +25°C      |                                              |                                                           |
| 7.97                               |                                     | 16                                                           | mA  | +85°C |            |                                              |                                                           |

**Legend:** TBD = To Be Determined

- 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  $V_{DD}$  or  $V_{SS}$ , 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.  
The test conditions for all  $I_{DD}$  measurements in active operation mode are:  
 $\text{OSC1}$  = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
 $\text{MCLR}$  =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of  $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to  $V_{SS}$ ).
- 5:** Voltage regulator enabled (ENVREG tied to  $V_{DD}$ ).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |      |       |            |                                              |                                                                                  |
|------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------|------------|----------------------------------------------|----------------------------------------------------------------------------------|
| Param No.                          | Device                              | Typ                                                                                                                                                     | Max  | Units | Conditions |                                              |                                                                                  |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                                                                                                                         |      |       |            |                                              |                                                                                  |
|                                    | All devices                         | 2.8                                                                                                                                                     | 3.8  | mA    | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 4 MHz,<br>16 MHz internal<br>( <b>PRI_RUN</b> mode,<br>HSPLL oscillator)  |
|                                    |                                     | 3.02                                                                                                                                                    | 3.8  | mA    | +25°C      |                                              |                                                                                  |
|                                    |                                     | 3.01                                                                                                                                                    | 4.5  | mA    | +85°C      |                                              |                                                                                  |
|                                    | All devices                         | 4.5                                                                                                                                                     | 5.4  | mA    | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> | FOSC = 4 MHz,<br>16 MHz internal<br>( <b>PRI_RUN</b> mode,<br>HSPLL oscillator)  |
|                                    |                                     | 4.8                                                                                                                                                     | 5.6  | mA    | +25°C      |                                              |                                                                                  |
|                                    |                                     | 4.54                                                                                                                                                    | 5.6  | mA    | +85°C      |                                              |                                                                                  |
|                                    | All devices                         | 5.72                                                                                                                                                    | 6.7  | mA    | -40°C      | VDD = 3.3V <sup>(5)</sup>                    | FOSC = 4 MHz,<br>16 MHz internal<br>( <b>PRI_RUN</b> mode,<br>HSPLL oscillator)  |
|                                    |                                     | 5.55                                                                                                                                                    | 6.5  | mA    | +25°C      |                                              |                                                                                  |
|                                    |                                     | 5.3                                                                                                                                                     | 6.5  | mA    | +85°C      |                                              |                                                                                  |
|                                    | All devices                         | 7.4                                                                                                                                                     | 8.5  | mA    | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> | FOSC = 10 MHz,<br>40 MHz internal<br>( <b>PRI_RUN</b> mode,<br>HSPLL oscillator) |
|                                    |                                     | 7.23                                                                                                                                                    | 8.5  | mA    | +25°C      |                                              |                                                                                  |
|                                    |                                     | 6.55                                                                                                                                                    | 7.5  | mA    | +85°C      |                                              |                                                                                  |
|                                    | All devices                         | 9.74                                                                                                                                                    | 11.6 | mA    | -40°C      | VDD = 3.3V <sup>(5)</sup>                    | FOSC = 10 MHz,<br>40 MHz internal<br>( <b>PRI_RUN</b> mode,<br>HSPLL oscillator) |
|                                    |                                     | 9.43                                                                                                                                                    | 11.6 | mA    | +25°C      |                                              |                                                                                  |
|                                    |                                     | 8.89                                                                                                                                                    | 10.5 | mA    | +85°C      |                                              |                                                                                  |

**Legend:** TBD = To Be Determined

- 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.  
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;  
MCLR = VDD; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of -10°C to +70°C. Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to VSS).
- 5:** Voltage regulator enabled (ENVREG tied to VDD).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated) |     |                                   |            |                                              |                                                    |
|------------------------------------|-------------------------------------|---------------------------------------------------------|-----|-----------------------------------|------------|----------------------------------------------|----------------------------------------------------|
|                                    |                                     | Operating temperature                                   |     | -40°C ≤ TA ≤ +85°C for industrial |            |                                              |                                                    |
| Param No.                          | Device                              | Typ                                                     | Max | Units                             | Conditions |                                              |                                                    |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                         |     |                                   |            |                                              |                                                    |
|                                    | All devices                         | 50                                                      | 120 | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 1 MHz<br>(PRI_IDLE mode,<br>EC oscillator)  |
|                                    |                                     | 51                                                      | 120 | μA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 54                                                      | 130 | μA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 223                                                     | 480 | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                    |
|                                    |                                     | 134                                                     | 300 | μA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 110                                                     | 270 | μA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 307                                                     | 550 | μA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                    |
|                                    |                                     | 254                                                     | 500 | μA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 194                                                     | 460 | μA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 307                                                     | 850 | μA                                | -40°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 4 MHz<br>(PRI_IDLE mode,<br>EC oscillator)  |
|                                    |                                     | 200                                                     | 850 | μA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 202                                                     | 800 | μA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 483                                                     | 950 | μA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                    |
|                                    |                                     | 318                                                     | 950 | μA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 343                                                     | 900 | μA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 0.52                                                    | 1.3 | mA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                    |
|                                    |                                     | 0.47                                                    | 1.2 | mA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 0.47                                                    | 1.2 | mA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 2.38                                                    | 8   | mA                                | -40°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> | FOSC = 40 MHz<br>(PRI_IDLE mode,<br>EC oscillator) |
|                                    |                                     | 2.04                                                    | 8   | mA                                | +25°C      |                                              |                                                    |
|                                    |                                     | 2.52                                                    | 9   | mA                                | +85°C      |                                              |                                                    |
|                                    | All devices                         | 3.02                                                    | 10  | mA                                | -40°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                    |
|                                    |                                     | 2.99                                                    | 10  | mA                                | +25°C      |                                              |                                                    |
| 4.23                               |                                     | 11                                                      | mA  | +85°C                             |            |                                              |                                                    |

**Legend:** TBD = To Be Determined

- 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  $V_{DD}$  or  $V_{SS}$ , 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.  
The test conditions for all  $I_{DD}$  measurements in active operation mode are:  
 $\text{OSC1}$  = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
 $\text{MCLR}$  =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of  $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to  $V_{SS}$ ).
- 5:** Voltage regulator enabled (ENVREG tied to  $V_{DD}$ ).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                     | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature      -40°C ≤ TA ≤ +85°C for industrial |     |       |            |                                              |                                                                             |
|------------------------------------|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------|-----|-------|------------|----------------------------------------------|-----------------------------------------------------------------------------|
| Param No.                          | Device                              | Typ                                                                                                                     | Max | Units | Conditions |                                              |                                                                             |
|                                    | Supply Current (IDD) <sup>(2)</sup> |                                                                                                                         |     |       |            |                                              |                                                                             |
|                                    | All devices                         | 10.5                                                                                                                    | 22  | μA    | -10°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 32 kHz <sup>(3)</sup><br>( <b>SEC_RUN</b> mode,<br>Timer1 as clock)  |
|                                    |                                     | 13.4                                                                                                                    | 28  | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 17.6                                                                                                                    | 40  | μA    | +70°C      |                                              |                                                                             |
|                                    | All devices                         | 13.2                                                                                                                    | 30  | μA    | -10°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                             |
|                                    |                                     | 16.2                                                                                                                    | 35  | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 20.7                                                                                                                    | 50  | μA    | +70°C      |                                              |                                                                             |
|                                    | All devices                         | 39                                                                                                                      | 120 | μA    | -10°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                             |
|                                    |                                     | 58                                                                                                                      | 150 | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 75                                                                                                                      | 190 | μA    | +70°C      |                                              |                                                                             |
|                                    | All devices                         | 5.7                                                                                                                     | 15  | μA    | -10°C      | VDD = 2.0V,<br>VDDCORE = 2.0V <sup>(4)</sup> | FOSC = 32 kHz <sup>(3)</sup><br>( <b>SEC_IDLE</b> mode,<br>Timer1 as clock) |
|                                    |                                     | 8.9                                                                                                                     | 20  | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 12.8                                                                                                                    | 26  | μA    | +70°C      |                                              |                                                                             |
|                                    | All devices                         | 6.6                                                                                                                     | 17  | μA    | -10°C      | VDD = 2.5V,<br>VDDCORE = 2.5V <sup>(4)</sup> |                                                                             |
|                                    |                                     | 9.7                                                                                                                     | 24  | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 13.7                                                                                                                    | 30  | μA    | +70°C      |                                              |                                                                             |
|                                    | All devices                         | 39                                                                                                                      | 115 | μA    | -10°C      | VDD = 3.3V <sup>(5)</sup>                    |                                                                             |
|                                    |                                     | 52.8                                                                                                                    | 145 | μA    | +25°C      |                                              |                                                                             |
|                                    |                                     | 72.7                                                                                                                    | 185 | μA    | +70°C      |                                              |                                                                             |

**Legend:** TBD = To Be Determined

- 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  $V_{DD}$  or  $V_{SS}$ , 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.  
The test conditions for all  $I_{DD}$  measurements in active operation mode are:  
 $\text{OSC1}$  = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
 $\text{MCLR}$  =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of  $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to  $V_{SS}$ ).
- 5:** Voltage regulator enabled (ENVREG tied to  $V_{DD}$ ).

# PIC18F85J11 FAMILY

## 25.2 DC Characteristics: Power-Down and Supply Current PIC18F85J11 Family (Industrial) (Continued)

| PIC18F85J11 Family<br>(Industrial) |                                                                                           | Standard Operating Conditions (unless otherwise stated) |                               |                                   |                |                                                                      |                                                                      |
|------------------------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------|-----------------------------------|----------------|----------------------------------------------------------------------|----------------------------------------------------------------------|
|                                    |                                                                                           | Operating temperature                                   |                               | -40°C ≤ TA ≤ +85°C for industrial |                |                                                                      |                                                                      |
| Param No.                          | Device                                                                                    | Typ                                                     | Max                           | Units                             | Conditions     |                                                                      |                                                                      |
| D022<br>(ΔI <sub>WDT</sub> )       | Module Differential Currents (ΔI <sub>WDT</sub> , ΔI <sub>OSCB</sub> , ΔI <sub>AD</sub> ) | Watchdog Timer                                          | 1.6                           | 4                                 | μA             | -40°C                                                                | V <sub>DD</sub> = 2.0V,<br>V <sub>DDCORE</sub> = 2.0V <sup>(4)</sup> |
|                                    |                                                                                           |                                                         | 1.7                           | 4                                 | μA             | +25°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | 1.6                           | 4                                 | μA             | +85°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | 2.5                           | 5                                 | μA             | -40°C                                                                | V <sub>DD</sub> = 2.5V,<br>V <sub>DDCORE</sub> = 2.5V <sup>(4)</sup> |
|                                    |                                                                                           |                                                         | 2.5                           | 5                                 | μA             | +25°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | 2.3                           | 5                                 | μA             | +85°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | 3.8                           | 6                                 | μA             | -40°C                                                                | V <sub>DD</sub> = 3.3V <sup>(5)</sup>                                |
|                                    |                                                                                           |                                                         | 2.6                           | 6                                 | μA             | +25°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | 2.4                           | 6                                 | μA             | +85°C                                                                |                                                                      |
|                                    |                                                                                           |                                                         | D025<br>(ΔI <sub>OSCB</sub> ) | Timer1 Oscillator                 | 6.6            | 12.5                                                                 | μA                                                                   |
| 7.9                                | 12.5                                                                                      | μA                                                      |                               |                                   | +25°C          |                                                                      |                                                                      |
| 11.5                               | 18                                                                                        | μA                                                      |                               |                                   | +85°C          |                                                                      |                                                                      |
| 7.2                                | 12.5                                                                                      | μA                                                      |                               |                                   | -40°C          | V <sub>DD</sub> = 2.5V,<br>V <sub>DDCORE</sub> = 2.5V <sup>(4)</sup> | 32 kHz on Timer1 <sup>(3)</sup>                                      |
| 8.1                                | 12.5                                                                                      | μA                                                      |                               |                                   | +25°C          |                                                                      |                                                                      |
| 11.9                               | 18.5                                                                                      | μA                                                      |                               |                                   | +85°C          |                                                                      |                                                                      |
| 7                                  | 12.5                                                                                      | μA                                                      |                               |                                   | -40°C          | V <sub>DD</sub> = 3.3V <sup>(5)</sup>                                | 32 kHz on Timer1 <sup>(3)</sup>                                      |
| 9                                  | 12.5                                                                                      | μA                                                      |                               |                                   | +25°C          |                                                                      |                                                                      |
| 11                                 | 18.5                                                                                      | μA                                                      | +85°C                         |                                   |                |                                                                      |                                                                      |
| D026<br>(ΔI <sub>AD</sub> )        | A/D Converter                                                                             | 1                                                       | 1.5                           | μA                                | -40°C to +85°C | V <sub>DD</sub> = 2.0V,<br>V <sub>DDCORE</sub> = 2.0V <sup>(4)</sup> | A/D on, not converting                                               |
|                                    |                                                                                           | 1                                                       | 1.5                           | μA                                | -40°C to +85°C | V <sub>DD</sub> = 2.5V,<br>V <sub>DDCORE</sub> = 2.5V <sup>(4)</sup> |                                                                      |
|                                    |                                                                                           | 1                                                       | 1.5                           | μA                                | -40°C to +85°C | V <sub>DD</sub> = 3.3V <sup>(5)</sup>                                |                                                                      |

**Legend:** TBD = To Be Determined

- 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  $V_{DD}$  or  $V_{SS}$ , 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.  
The test conditions for all  $I_{DD}$  measurements in active operation mode are:  
OSC1 = external square wave, from rail-to-rail; all I/O pins tri-stated, pulled to  $V_{DD}$ ;  
MCLR =  $V_{DD}$ ; WDT enabled/disabled as specified.
- 3:** Standard, low-cost 32 kHz crystals have an operating temperature range of  $-10^{\circ}\text{C}$  to  $+70^{\circ}\text{C}$ . Extended temperature crystals are available at a much higher cost.
- 4:** Voltage regulator disabled (ENVREG tied to  $V_{SS}$ ).
- 5:** Voltage regulator enabled (ENVREG tied to  $V_{DD}$ ).

# PIC18F85J11 FAMILY

## 25.3 DC Characteristics: PIC18F85J11 Family (Industrial)

| DC CHARACTERISTICS |                                      |                                                                                                                                                                                                                                | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial                                     |                                                                                                                                     |                                           |                                                                                                                                                                                                 |
|--------------------|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Param No.          | Symbol                               | Characteristic                                                                                                                                                                                                                 | Min                                                                                                                                                                                         | Max                                                                                                                                 | Units                                     | Conditions                                                                                                                                                                                      |
|                    | V <sub>IL</sub>                      | <b>Input Low Voltage</b><br>All I/O ports:<br>with TTL buffer<br>with Schmitt Trigger buffer<br>MCLR<br>OSC1<br>OSC1<br>T13CKI                                                                                                 | V <sub>SS</sub><br>V <sub>SS</sub><br>V <sub>SS</sub><br>V <sub>SS</sub><br>V <sub>SS</sub><br>V <sub>SS</sub>                                                                              | 0.15 V <sub>DD</sub><br>0.2 V <sub>DD</sub><br>0.2 V <sub>DD</sub><br>0.3 V <sub>DD</sub><br>0.2 V <sub>DD</sub><br>0.3             | V<br>V<br>V<br>V<br>V<br>V                | HS, HSPLL modes<br>EC, ECPLL modes <sup>(1)</sup>                                                                                                                                               |
|                    | V <sub>IH</sub>                      | <b>Input High Voltage</b><br>I/O ports with analog functions:<br>with TTL buffer<br>with Schmitt Trigger buffer<br>Digital only I/O ports:<br>with TTL buffer<br>with Schmitt Trigger buffer<br>MCLR<br>OSC1<br>OSC1<br>T13CKI | 0.25 V <sub>DD</sub> + 0.8V<br>0.8 V <sub>DD</sub><br>0.25 V <sub>DD</sub> + 0.8V<br>2.0<br>0.8 V <sub>DD</sub><br>0.8 V <sub>DD</sub><br>0.7 V <sub>DD</sub><br>0.8 V <sub>DD</sub><br>1.6 | V <sub>DD</sub><br>V <sub>DD</sub><br>5.5<br>5.5<br>5.5<br>V <sub>DD</sub><br>V <sub>DD</sub><br>V <sub>DD</sub><br>V <sub>DD</sub> | V<br>V<br>V<br>V<br>V<br>V<br>V<br>V<br>V | V <sub>DD</sub> < 3.3V<br><br>V <sub>DD</sub> < 3.3V<br>3.3V ≤ V <sub>DD</sub> ≤ 3.6V<br><br>HS, HSPLL modes<br>EC, ECPLL modes                                                                 |
|                    | I <sub>IL</sub>                      | <b>Input Leakage Current<sup>(1)</sup></b><br>I/O ports<br>MCLR<br>OSC1                                                                                                                                                        | —<br>—<br>—                                                                                                                                                                                 | ±1<br>±1<br>±1                                                                                                                      | μA<br>μA<br>μA                            | V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> ,<br>pin at high-impedance<br>V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub><br>V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> |
|                    | I <sub>PU</sub><br>I <sub>PURB</sub> | <b>Weak Pull-up Current</b><br>PORTB weak pull-up current                                                                                                                                                                      | 30                                                                                                                                                                                          | 240                                                                                                                                 | μA                                        | V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub>                                                                                                                                      |

**Note 1:** Negative current is defined as current sourced by the pin.



# PIC18F85J11 FAMILY

## 25.3 DC Characteristics: PIC18F85J11 Family (Industrial) (Continued)

| DC CHARACTERISTICS                             |        |                                                        | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |     |       |                                                                                                    |
|------------------------------------------------|--------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------|----------------------------------------------------------------------------------------------------|
| Param No.                                      | Symbol | Characteristic                                         | Min                                                                                                                                                     | Max | Units | Conditions                                                                                         |
| D080                                           | VOL    | <b>Output Low Voltage</b><br>I/O ports:                |                                                                                                                                                         |     |       |                                                                                                    |
|                                                |        | PORTA, PORTF, PORTG, PORTH                             | —                                                                                                                                                       | 0.4 | V     | $I_{OL} = 2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$   |
|                                                |        | PORTD, PORTE, PORTJ                                    | —                                                                                                                                                       | 0.4 | V     | $I_{OL} = 3.4\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ |
|                                                |        | PORTB, PORTC                                           | —                                                                                                                                                       | 0.4 | V     | $I_{OL} = 3.4\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ |
| D083                                           |        | OSC2/CLKO (EC, ECPLL modes)                            | —                                                                                                                                                       | 0.4 | V     | $I_{OL} = 1.6\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ |
| D090                                           | VOH    | <b>Output High Voltage<sup>(1)</sup></b><br>I/O ports: |                                                                                                                                                         |     |       |                                                                                                    |
|                                                |        | PORTA, PORTF, PORTG, PORTH                             | 2.4                                                                                                                                                     | —   | V     | $I_{OH} = -2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$  |
|                                                |        | PORTD, PORTE, PORTJ                                    | 2.4                                                                                                                                                     | —   | V     | $I_{OH} = -2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$  |
|                                                |        | PORTB, PORTC                                           | 2.4                                                                                                                                                     | —   | V     | $I_{OH} = -2\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$  |
| D092                                           |        | OSC2/CLKO (INTOSC, EC, ECPLL modes)                    | 2.4                                                                                                                                                     | —   | V     | $I_{OH} = -1\text{ mA}$ , $V_{DD} = 3.3\text{V}$ , $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$  |
| <b>Capacitive Loading Specs on Output Pins</b> |        |                                                        |                                                                                                                                                         |     |       |                                                                                                    |
| D100                                           | COSC2  | OSC2 pin                                               | —                                                                                                                                                       | 15  | pF    | In HS mode when external clock is used to drive OSC1                                               |
| D101                                           | CIO    | All I/O pins and OSC2                                  | —                                                                                                                                                       | 50  | pF    | To meet the AC Timing Specifications                                                               |
| D102                                           | CB     | SCLx, SDAx                                             | —                                                                                                                                                       | 400 | pF    | I <sup>2</sup> C™ Specification                                                                    |

**Note 1:** Negative current is defined as current sourced by the pin.

# PIC18F85J11 FAMILY

**TABLE 25-1: MEMORY PROGRAMMING REQUIREMENTS**

| DC CHARACTERISTICS          |       |                                   | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ for industrial |      |     |       |                                                |
|-----------------------------|-------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----|-------|------------------------------------------------|
| Param No.                   | Sym   | Characteristic                    | Min                                                                                                                                                     | Typ† | Max | Units | Conditions                                     |
| <b>Program Flash Memory</b> |       |                                   |                                                                                                                                                         |      |     |       |                                                |
| D130                        | EP    | Cell Endurance                    | 100                                                                                                                                                     | 1k   | —   | E/W   | $-40^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ |
| D131                        | VPR   | VDD for Read                      | V <sub>MIN</sub>                                                                                                                                        | —    | 3.6 | V     | V <sub>MIN</sub> = Minimum operating voltage   |
| D132B                       | VPEW  | VDD for Self-Timed Write          | V <sub>MIN</sub>                                                                                                                                        | —    | 3.6 | V     | V <sub>MIN</sub> = Minimum operating voltage   |
| D133A                       | TIW   | Self-Timed Write Cycle Time       | —                                                                                                                                                       | 2.8  | —   | ms    |                                                |
| D134                        | TRETD | Characteristic Retention          | 20                                                                                                                                                      | —    | —   | Year  | Provided no other specifications are violated  |
| D135                        | IDDP  | Supply Current during Programming | —                                                                                                                                                       | 3    | 7   | mA    |                                                |
| D1xxx                       | TWE   | Writes per Erase Cycle            | —                                                                                                                                                       | —    | 1   |       | Per one physical word address                  |

† Data in “Typ” column is at 3.3V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

# PIC18F85J11 FAMILY

**TABLE 25-2: COMPARATOR SPECIFICATIONS**

| Operating Conditions: $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (unless otherwise stated) |                    |                                         |     |      |                 |       |          |
|--------------------------------------------------------------------------------------------------------------------------|--------------------|-----------------------------------------|-----|------|-----------------|-------|----------|
| Param No.                                                                                                                | Sym                | Characteristics                         | Min | Typ  | Max             | Units | Comments |
| D300                                                                                                                     | V <sub>IOFF</sub>  | Input Offset Voltage                    | —   | ±5.0 | ±10             | mV    |          |
| D301                                                                                                                     | V <sub>ICM</sub>   | Input Common Mode Voltage*              | 0   | —    | $AV_{DD} - 1.5$ | V     |          |
| D302                                                                                                                     | CMRR               | Common Mode Rejection Ratio*            | 55  | —    | —               | dB    |          |
| 300                                                                                                                      | T <sub>RESP</sub>  | Response Time <sup>(1)</sup>            | —   | 150  | 400             | ns    |          |
| 301                                                                                                                      | T <sub>MC2OV</sub> | Comparator Mode Change to Output Valid* | —   | —    | 10              | μs    |          |

\* These parameters are characterized but not tested.

**Note 1:** Response time measured with one comparator input at  $(AV_{DD} - 1.5)/2$ , while the other input transitions from  $V_{SS}$  to  $V_{DD}$ .

**TABLE 25-3: VOLTAGE REFERENCE SPECIFICATIONS**

| Operating Conditions: $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (unless otherwise stated) |                  |                              |             |     |             |       |          |
|--------------------------------------------------------------------------------------------------------------------------|------------------|------------------------------|-------------|-----|-------------|-------|----------|
| Param No.                                                                                                                | Sym              | Characteristics              | Min         | Typ | Max         | Units | Comments |
| D310                                                                                                                     | V <sub>RES</sub> | Resolution                   | $V_{DD}/24$ | —   | $V_{DD}/32$ | LSb   |          |
| D311                                                                                                                     | V <sub>RAA</sub> | Absolute Accuracy            | —           | —   | 1/2         | LSb   |          |
| D312                                                                                                                     | V <sub>RUR</sub> | Unit Resistor Value (R)      | —           | 2k  | —           | Ω     |          |
| 310                                                                                                                      | T <sub>SET</sub> | Settling Time <sup>(1)</sup> | —           | —   | 10          | μs    |          |

**Note 1:** Settling time measured while  $CVRR = 1$  and  $CVR3:CVR0$  bits transition from '0000' to '1111'.

**TABLE 25-4: INTERNAL VOLTAGE REGULATOR SPECIFICATIONS**

| Operating Conditions: $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (unless otherwise stated) |                    |                                  |     |     |     |       |                           |
|-------------------------------------------------------------------------------------------|--------------------|----------------------------------|-----|-----|-----|-------|---------------------------|
| Param No.                                                                                 | Sym                | Characteristics                  | Min | Typ | Max | Units | Comments                  |
|                                                                                           | V <sub>RGOUT</sub> | Regulator Output Voltage*        | —   | 2.5 | —   | V     |                           |
|                                                                                           | CEFC               | External Filter Capacitor Value* | 4.7 | 10  | —   | μF    | Capacitor must be low-ESR |

\* These parameters are characterized but not tested. Parameter numbers not yet assigned for these specifications.

# PIC18F85J11 FAMILY

## 25.4 AC (Timing) Characteristics

### 25.4.1 TIMING PARAMETER SYMBOLOGY

The timing parameter symbols have been created following one of the following formats:

1. TppS2ppS
2. TppS
3. Tcc:ST (I<sup>2</sup>C specifications only)
4. Ts (I<sup>2</sup>C specifications only)

|   |           |   |      |
|---|-----------|---|------|
| T |           | T |      |
| F | Frequency | T | Time |

Lowercase letters (pp) and their meanings:

|    |                 |     |                                    |
|----|-----------------|-----|------------------------------------|
| pp |                 | osc | OSC1                               |
| cc | CCP1            | rd  | $\overline{RD}$                    |
| ck | CLKO            | rw  | $\overline{RD}$ or $\overline{WR}$ |
| cs | $\overline{CS}$ | sc  | SCK                                |
| di | SDI             | ss  | $\overline{SS}$                    |
| do | SDO             | t0  | T0CKI                              |
| dt | Data in         | t1  | T13CKI                             |
| io | I/O port        | wr  | $\overline{WR}$                    |
| mc | MCLR            |     |                                    |

Uppercase letters and their meanings:

|                       |                          |      |                |
|-----------------------|--------------------------|------|----------------|
| S                     |                          | P    | Period         |
| F                     | Fall                     | R    | Rise           |
| H                     | High                     | V    | Valid          |
| I                     | Invalid (High-impedance) | Z    | High-impedance |
| L                     | Low                      |      |                |
| I <sup>2</sup> C only |                          | High | High           |
| AA                    | output access            | Low  | Low            |
| BUF                   | Bus free                 |      |                |

Tcc:ST (I<sup>2</sup>C specifications only)

|     |                 |     |                |
|-----|-----------------|-----|----------------|
| CC  |                 | SU  | Setup          |
| HD  | Hold            |     |                |
| ST  |                 | STO | Stop condition |
| DAT | DATA input hold |     |                |
| STA | Start condition |     |                |

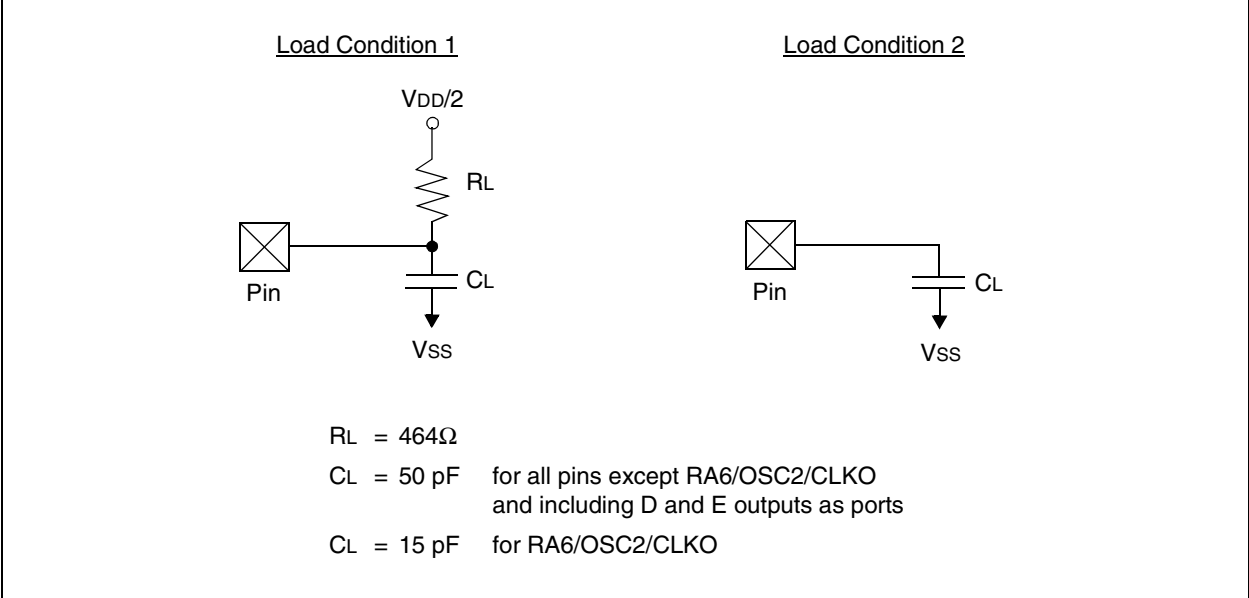
25.4.2 TIMING CONDITIONS

The temperature and voltages specified in Table 25-5 apply to all timing specifications unless otherwise noted. Figure 25-3 specifies the load conditions for the timing specifications.

TABLE 25-5: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

|                    |                                                         |                                                |
|--------------------|---------------------------------------------------------|------------------------------------------------|
| AC CHARACTERISTICS | Standard Operating Conditions (unless otherwise stated) |                                                |
|                    | Operating temperature                                   | -40°C ≤ TA ≤ +85°C for industrial              |
|                    | Operating voltage VDD range                             | as described in Section 25.1 and Section 25.3. |

FIGURE 25-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS



# PIC18F85J11 FAMILY

## 25.4.3 TIMING DIAGRAMS AND SPECIFICATIONS

**FIGURE 25-4: EXTERNAL CLOCK TIMING (ALL MODES EXCEPT PLL)**

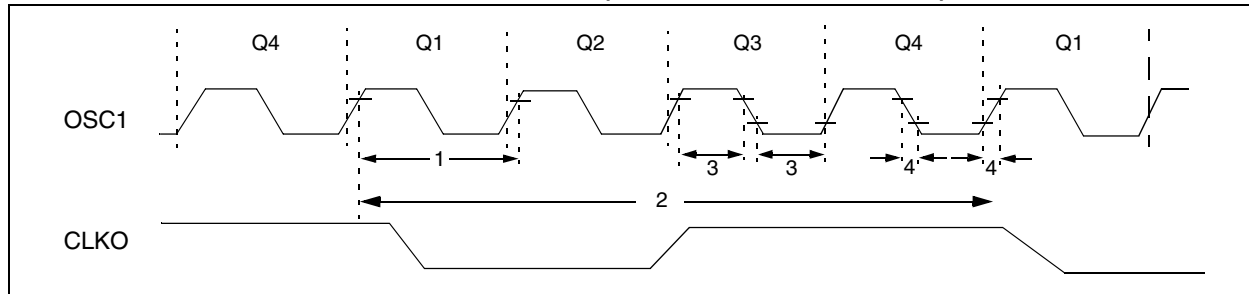


**TABLE 25-6: EXTERNAL CLOCK TIMING REQUIREMENTS**

| Param. No. | Symbol     | Characteristic                             | Min | Max | Units | Conditions               |
|------------|------------|--------------------------------------------|-----|-----|-------|--------------------------|
| 1A         | FOSC       | External CLKI Frequency <sup>(1)</sup>     | DC  | 40  | MHz   | ECPLL Oscillator mode    |
|            |            | Oscillator Frequency <sup>(1)</sup>        | DC  | 40  | MHz   | HSPLL Oscillator mode    |
| 1          | TOSC       | External CLKI Period <sup>(1)</sup>        | 25  | —   | ns    | EC Oscillator mode       |
|            |            | Oscillator Period <sup>(1)</sup>           | 25  | 250 | ns    | HS Oscillator mode       |
| 2          | Tcy        | Instruction Cycle Time <sup>(1)</sup>      | 100 | —   | ns    | Tcy = 4/FOSC, Industrial |
| 3          | TosL, TosH | External Clock in (OSC1) High or Low Time  | 10  | —   | ns    | EC Oscillator mode       |
| 4          | TosR, TosF | External Clock in (OSC1) Rise or Fall Time | —   | 7.5 | ns    | EC Oscillator mode       |

**Note 1:** Instruction cycle period (Tcy) equals four times the input oscillator time base period for all configurations except PLL. All specified values are based on characterization data for that particular oscillator type under standard operating conditions with the device executing code. Exceeding these specified limits may result in an unstable oscillator operation and/or higher than expected current consumption. All devices are tested to operate at “min.” values with an external clock applied to the OSC1/CLKI pin. When an external clock input is used, the “max.” cycle time limit is “DC” (no clock) for all devices.

# PIC18F85J11 FAMILY

**TABLE 25-7: PLL CLOCK TIMING SPECIFICATIONS (V<sub>DD</sub> = 2.15V TO 3.6V)**

| Param No. | Sym             | Characteristic                | Min | Typ† | Max | Units | Conditions   |
|-----------|-----------------|-------------------------------|-----|------|-----|-------|--------------|
| F10       | FOSC            | Oscillator Frequency Range    | 4   | —    | 10  | MHz   | HS mode only |
| F11       | FSYS            | On-Chip VCO System Frequency  | 16  | —    | 40  | MHz   | HS mode only |
| F12       | t <sub>rc</sub> | PLL Start-up Time (Lock Time) | —   | —    | 2   | ms    |              |
| F13       | ΔCLK            | CLKO Stability (Jitter)       | -2  | —    | +2  | %     |              |

† Data in “Typ” column is at 3.3V, 25°C, unless otherwise stated. These parameters are for design guidance only and are not tested.

**TABLE 25-8: INTERNAL RC ACCURACY (INTOSC AND INTRC SOURCES)**

| PIC18F85J11 Family<br>(Industrial) |                                                                                                       | Standard Operating Conditions (unless otherwise stated)<br>Operating temperature -40°C ≤ TA ≤ +85°C for industrial |      |        |       |                |                |
|------------------------------------|-------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|------|--------|-------|----------------|----------------|
| Param No.                          | Device                                                                                                | Min                                                                                                                | Typ  | Max    | Units | Conditions     |                |
|                                    | INTOSC Accuracy @ Freq = 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, 31 kHz <sup>(1)</sup> |                                                                                                                    |      |        |       |                |                |
|                                    | All devices                                                                                           | -2                                                                                                                 | +/-1 | 2      | %     | +25°C          | VDD = 2.7-3.3V |
|                                    |                                                                                                       | -5                                                                                                                 | —    | 5      | %     | -10°C to +85°C | VDD = 2.0-3.3V |
|                                    |                                                                                                       | -10                                                                                                                | +/-1 | 10     | %     | -40°C to +85°C | VDD = 2.0-3.3V |
|                                    | INTRC Accuracy @ Freq = 31 kHz <sup>(1)</sup>                                                         |                                                                                                                    |      |        |       |                |                |
|                                    | All devices                                                                                           | 26.562                                                                                                             | —    | 35.938 | kHz   | -40°C to +85°C | VDD = 2.0-3.3V |

**Legend:** TBD = To Be Determined

**Note 1:** The accuracy specification of the 31 kHz clock is determined by which source is providing it at a given time. When INTSRC (OSCTUNE<7>) is ‘1’, use the INTOSC accuracy specification. When INTSRC is ‘0’, use the INTRC accuracy specification.

# PIC18F85J11 FAMILY

**FIGURE 25-5: CLKO AND I/O TIMING**

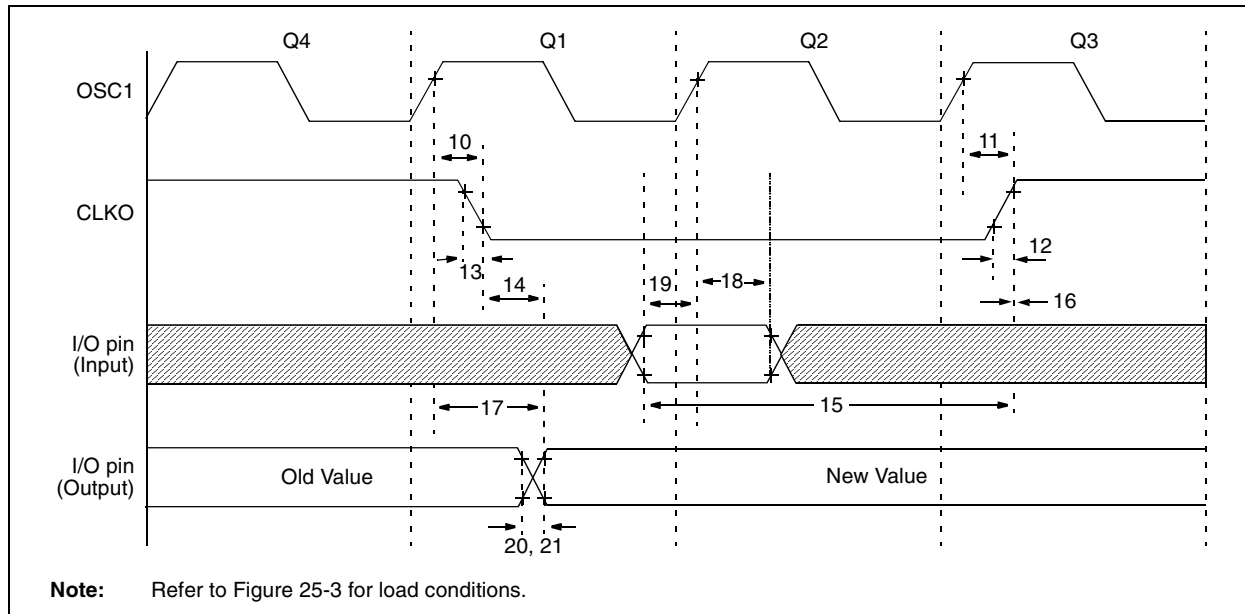


**TABLE 25-9: CLKO AND I/O TIMING REQUIREMENTS**

| Param No. | Symbol   | Characteristic                                                      | Min                | Typ | Max               | Units | Conditions |
|-----------|----------|---------------------------------------------------------------------|--------------------|-----|-------------------|-------|------------|
| 10        | TosH2ckL | OSC1 $\uparrow$ to CLKO $\downarrow$                                | —                  | 75  | 200               | ns    | (Note 1)   |
| 11        | TosH2ckH | OSC1 $\uparrow$ to CLKO $\uparrow$                                  | —                  | 75  | 200               | ns    | (Note 1)   |
| 12        | TckR     | CLKO Rise Time                                                      | —                  | 15  | 30                | ns    | (Note 1)   |
| 13        | TckF     | CLKO Fall Time                                                      | —                  | 15  | 30                | ns    | (Note 1)   |
| 14        | TckL2ioV | CLKO $\downarrow$ to Port Out Valid                                 | —                  | —   | $0.5 T_{CY} + 20$ | ns    |            |
| 15        | TioV2ckH | Port In Valid before CLKO $\uparrow$                                | $0.25 T_{CY} + 25$ | —   | —                 | ns    |            |
| 16        | TckH2ioI | Port In Hold after CLKO $\uparrow$                                  | 0                  | —   | —                 | ns    |            |
| 17        | TosH2ioV | OSC1 $\uparrow$ (Q1 cycle) to Port Out Valid                        | —                  | 50  | 150               | ns    |            |
| 18        | TosH2ioI | OSC1 $\uparrow$ (Q2 cycle) to Port Input Invalid (I/O in hold time) | 100                | —   | —                 | ns    |            |
| 19        | TioV2osH | Port Input Valid to OSC1 $\uparrow$ (I/O in setup time)             | 0                  | —   | —                 | ns    |            |
| 20        | TioR     | Port Output Rise Time                                               | —                  | —   | 6                 | ns    |            |
| 21        | TioF     | Port Output Fall Time                                               | —                  | —   | 5                 | ns    |            |
| 22†       | TINP     | INTx Pin High or Low Time                                           | $T_{CY}$           | —   | —                 | ns    |            |
| 23†       | TRBP     | RB7:RB4 Change INTx High or Low Time                                | $T_{CY}$           | —   | —                 | ns    |            |

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

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



# PIC18F85J11 FAMILY

**FIGURE 25-6: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER AND POWER-UP TIMER TIMING**

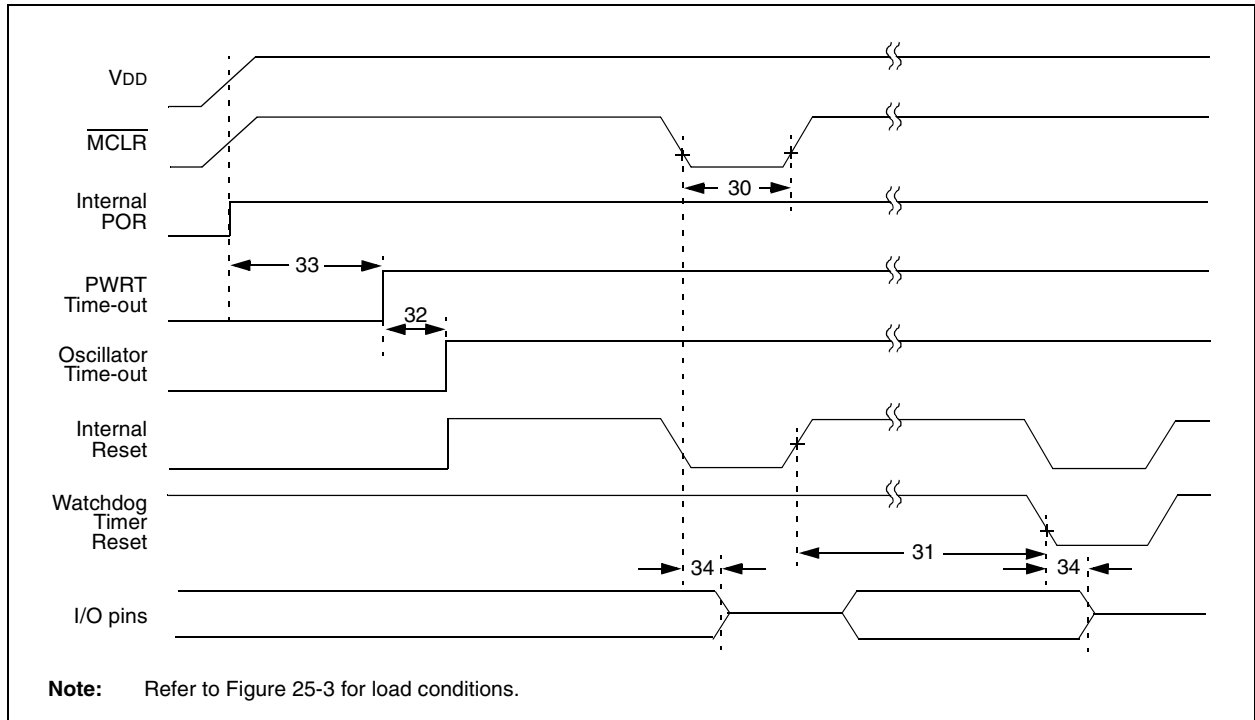


**TABLE 25-10: RESET, WATCHDOG TIMER, OSCILLATOR START-UP TIMER, POWER-UP TIMER AND BROWN-OUT RESET REQUIREMENTS**

| Param No. | Symbol | Characteristic                                           | Min       | Typ    | Max       | Units | Conditions                                 |
|-----------|--------|----------------------------------------------------------|-----------|--------|-----------|-------|--------------------------------------------|
| 30        | TMCL   | MCLR Pulse Width (low)                                   | 2 Tcy     | 10 Tcy | —         |       | (Note 1)                                   |
| 31        | TWDT   | Watchdog Timer Time-out Period (no postscaler)           | 3.4       | 4.0    | 4.6       | ms    |                                            |
| 32        | TOST   | Oscillator Start-up Timer Period                         | 1024 TOSC | —      | 1024 TOSC |       | TOSC = OSC1 period                         |
| 33        | TPWRT  | Power-up Timer Period                                    | 45.8      | 65.5   | 85.2      | ms    |                                            |
| 34        | TIOZ   | I/O High-Impedance from MCLR Low or Watchdog Timer Reset | —         | 2      | —         | μs    |                                            |
| 38        | TCSD   | CPU Start-up Time                                        | —         | 10     | —         | μs    | Voltage regulator enabled and put to Sleep |
|           |        |                                                          | —         | 200    | —         | μs    |                                            |
| 39        | TIOBST | Time for INTOSC to Stabilize                             | —         | 1      | —         | μs    |                                            |

**Note 1:** To ensure device Reset, MCLR must be low for at least 2 Tcy or 400 μs, whichever is lower.

# PIC18F85J11 FAMILY

**FIGURE 25-7: TIMER0 AND TIMER1 EXTERNAL CLOCK TIMINGS**

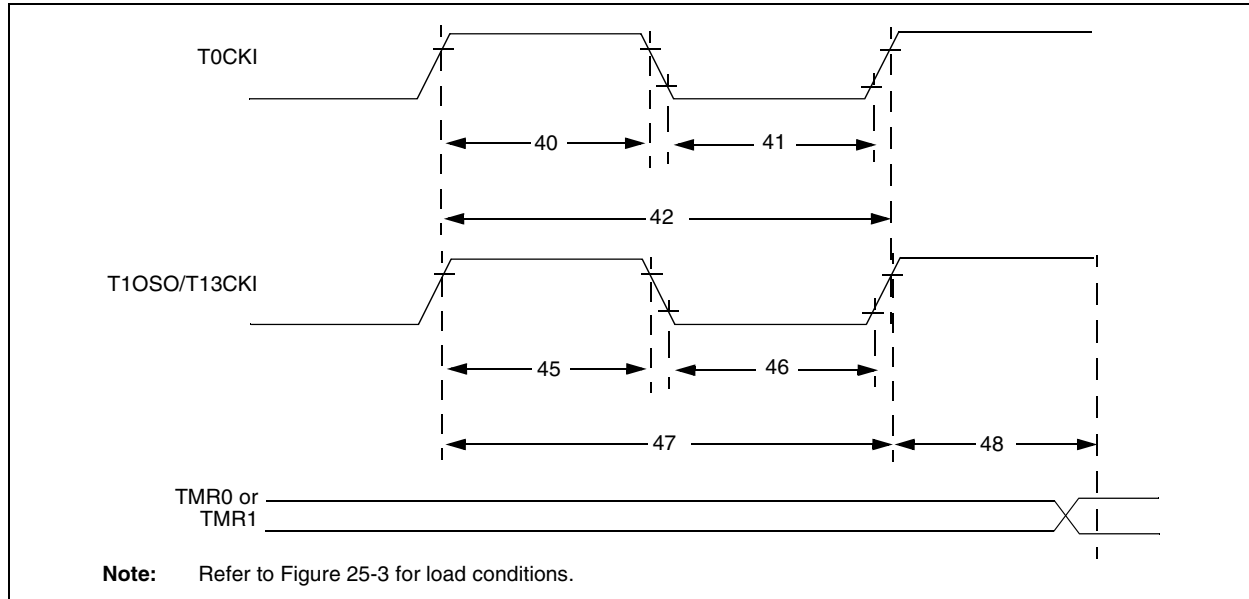


**TABLE 25-11: TIMER0 AND TIMER1 EXTERNAL CLOCK REQUIREMENTS**

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

**FIGURE 25-8: CAPTURE/COMPARE/PWM TIMINGS (CCP1, CCP2 MODULES)**

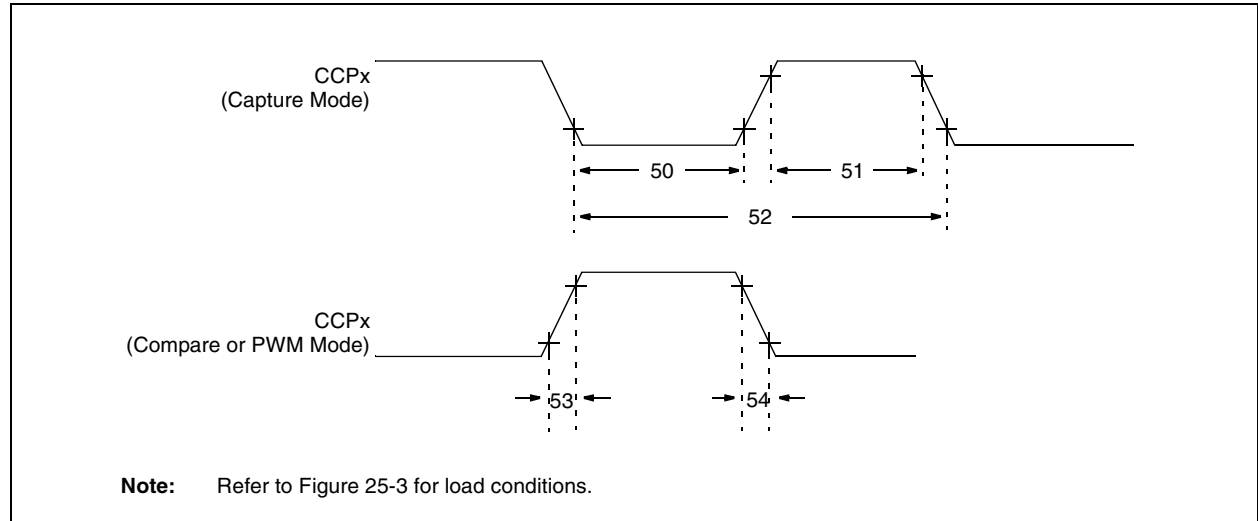


**TABLE 25-12: CAPTURE/COMPARE/PWM REQUIREMENTS (CCP1, CCP2 MODULES)**

| Param No. | Symbol | Characteristic        |                | Min                       | Max | Units | Conditions                      |
|-----------|--------|-----------------------|----------------|---------------------------|-----|-------|---------------------------------|
| 50        | TccL   | CCPx Input Low Time   | No prescaler   | $0.5 T_{CY} + 20$         | —   | ns    |                                 |
|           |        |                       | With prescaler | 10                        | —   | ns    |                                 |
| 51        | TccH   | CCPx Input High Time  | No prescaler   | $0.5 T_{CY} + 20$         | —   | ns    |                                 |
|           |        |                       | With prescaler | 10                        | —   | ns    |                                 |
| 52        | TccP   | CCPx Input Period     |                | $\frac{3 T_{CY} + 40}{N}$ | —   | ns    | N = prescale value (1, 4 or 16) |
| 53        | TccR   | CCPx Output Fall Time |                | —                         | 25  | ns    |                                 |
| 54        | TccF   | CCPx Output Fall Time |                | —                         | 25  | ns    |                                 |

# PIC18F85J11 FAMILY

**FIGURE 25-9: EXAMPLE SPI MASTER MODE TIMING (CKE = 0)**

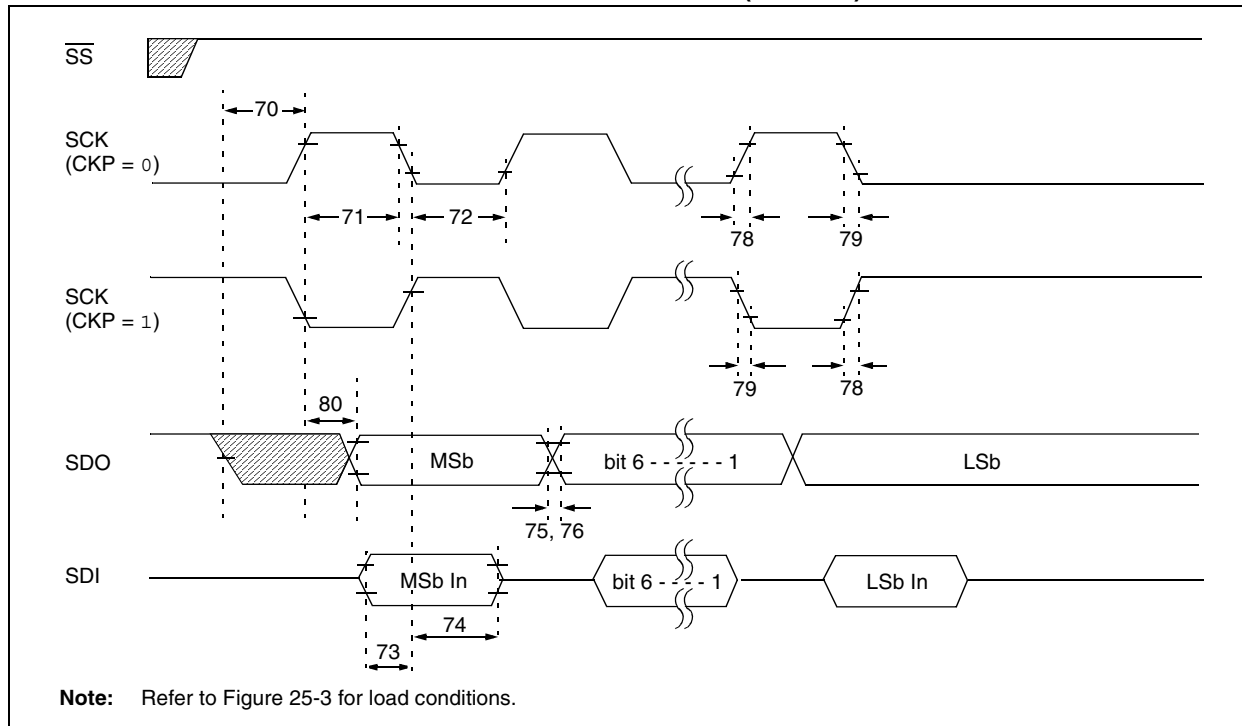


**TABLE 25-13: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 0)**

| Param No. | Symbol             | Characteristic                                            | Min          | Max           | Units | Conditions  |
|-----------|--------------------|-----------------------------------------------------------|--------------|---------------|-------|-------------|
| 70        | TssL2sCH, TssL2sCL | SS ↓ to SCK ↓ or SCK ↑ Input                              | Tcy          | —             | ns    |             |
| 71        | Tsch               | SCK Input High Time                                       | Continuous   | 1.25 Tcy + 30 | —     | ns          |
| 71A       |                    | (Slave mode)                                              | Single byte  | 40            | —     | ns (Note 1) |
| 72        | Tscl               | SCK Input Low Time                                        | Continuous   | 1.25 Tcy + 30 | —     | ns          |
| 72A       |                    | (Slave mode)                                              | Single byte  | 40            | —     | ns (Note 1) |
| 73        | TdIV2sCH, TdIV2sCL | Setup Time of SDI Data Input to SCK Edge                  | 100          | —             | ns    |             |
| 73A       | Tb2B               | Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2 | 1.5 Tcy + 40 | —             | ns    | (Note 2)    |
| 74        | Tsch2dIL, Tscl2dIL | Hold Time of SDI Data Input to SCK Edge                   | 100          | —             | ns    |             |
| 75        | TdoR               | SDO Data Output Rise Time                                 | —            | 25            | ns    |             |
| 76        | TdoF               | SDO Data Output Fall Time                                 | —            | 25            | ns    |             |
| 78        | Tscr               | SCK Output Rise Time (Master mode)                        | —            | 25            | ns    |             |
| 79        | TscF               | SCK Output Fall Time (Master mode)                        | —            | 25            | ns    |             |
| 80        | Tsch2doV, Tscl2doV | SDO Data Output Valid after SCK Edge                      | —            | 50            | ns    |             |

**Note 1:** Requires the use of Parameter #73A.

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

**FIGURE 25-10: EXAMPLE SPI MASTER MODE TIMING (CKE = 1)**

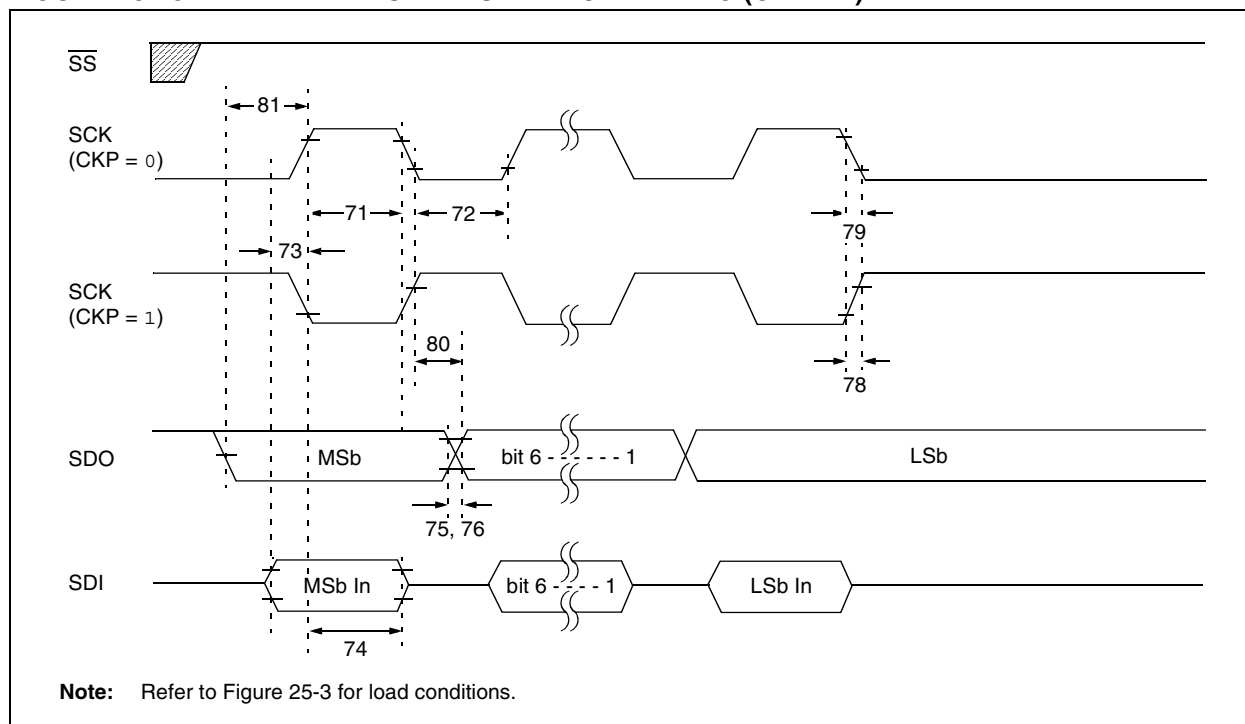


**TABLE 25-14: EXAMPLE SPI MODE REQUIREMENTS (MASTER MODE, CKE = 1)**

| Param. No. | Symbol             | Characteristic                                            |             | Min           | Max | Units | Conditions |
|------------|--------------------|-----------------------------------------------------------|-------------|---------------|-----|-------|------------|
| 71         | Tsch               | SCK Input High Time (Slave mode)                          | Continuous  | 1.25 Tcy + 30 | —   | ns    |            |
| 71A        |                    |                                                           | Single byte | 40            | —   | ns    | (Note 1)   |
| 72         | Tscl               | SCK Input Low Time (Slave mode)                           | Continuous  | 1.25 Tcy + 30 | —   | ns    |            |
| 72A        |                    |                                                           | Single byte | 40            | —   | ns    | (Note 1)   |
| 73         | TdIV2sch, TdIV2scl | Setup Time of SDI Data Input to SCK Edge                  |             | 100           | —   | ns    |            |
| 73A        | Tb2B               | Last Clock Edge of Byte 1 to the 1st Clock Edge of Byte 2 |             | 1.5 Tcy + 40  | —   | ns    | (Note 2)   |
| 74         | Tsch2dIL, Tscl2dIL | Hold Time of SDI Data Input to SCK Edge                   |             | 100           | —   | ns    |            |
| 75         | TdoR               | SDO Data Output Rise Time                                 |             | —             | 25  | ns    |            |
| 76         | TdoF               | SDO Data Output Fall Time                                 |             | —             | 25  | ns    |            |
| 78         | TscR               | SCK Output Rise Time (Master mode)                        |             | —             | 25  | ns    |            |
| 79         | TscF               | SCK Output Fall Time (Master mode)                        |             | —             | 25  | ns    |            |
| 80         | Tsch2doV, Tscl2doV | SDO Data Output Valid after SCK Edge                      |             | —             | 50  | ns    |            |
| 81         | TdoV2sch, TdoV2scl | SDO Data Output Setup to SCK Edge                         |             | Tcy           | —   | ns    |            |

**Note 1:** Requires the use of Parameter #73A.

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

# PIC18F85J11 FAMILY

**FIGURE 25-11: EXAMPLE SPI SLAVE MODE TIMING (CKE = 0)**

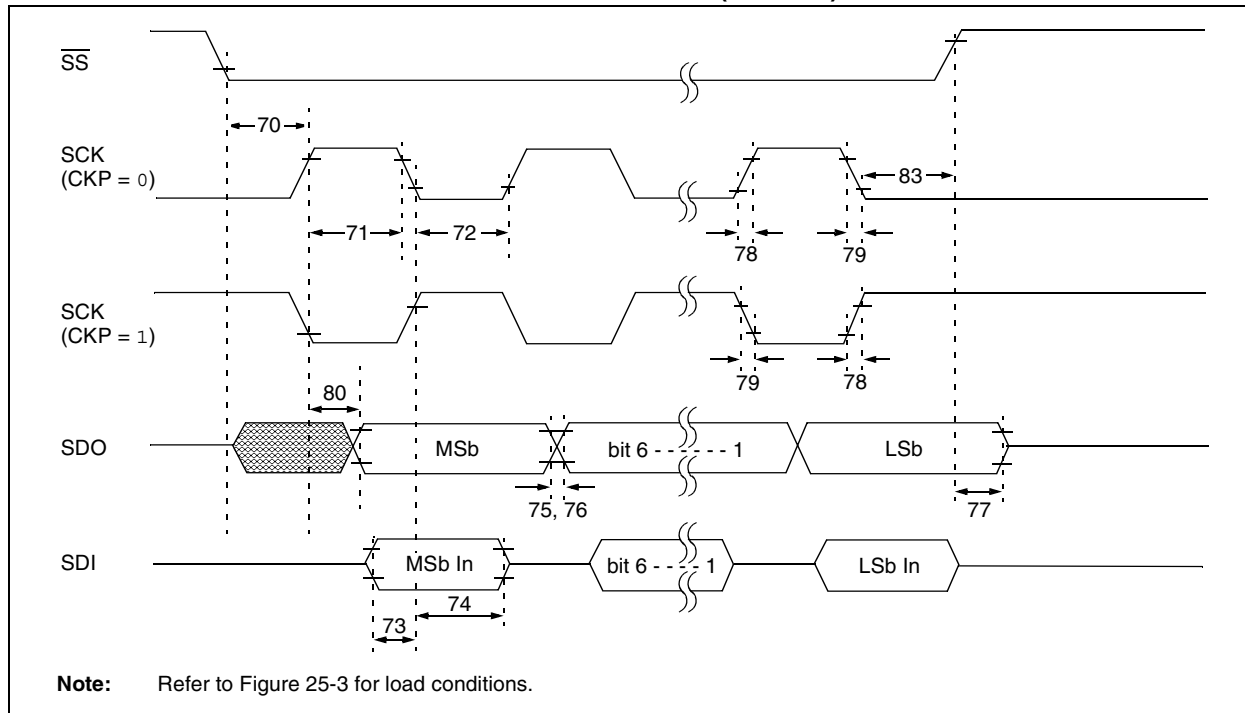


**TABLE 25-15: EXAMPLE SPI MODE REQUIREMENTS (SLAVE MODE TIMING, CKE = 0)**

| Param No. | Symbol             | Characteristic                                                         | Min          | Max           | Units | Conditions |
|-----------|--------------------|------------------------------------------------------------------------|--------------|---------------|-------|------------|
| 70        | TssL2scH, TssL2scL | $\overline{SS} \downarrow$ to SCK $\downarrow$ or SCK $\uparrow$ Input | 3 Tcy        | —             | ns    |            |
| 70A       | TssL2WB            | $\overline{SS}$ to Write to SSPBUF                                     | 3 Tcy        | —             | ns    |            |
| 71        | Tsch               | SCK Input High Time (Slave mode)                                       | Continuous   | 1.25 Tcy + 30 | ns    |            |
| 71A       |                    |                                                                        | Single Byte  | 40            | ns    | (Note 1)   |
| 72        | Tscl               | SCK Input Low Time (Slave mode)                                        | Continuous   | 1.25 Tcy + 30 | ns    |            |
| 72A       |                    |                                                                        | Single Byte  | 40            | ns    | (Note 1)   |
| 73        | TdIV2scH, TdIV2scL | Setup Time of SDI Data Input to SCK Edge                               | 100          | —             | ns    |            |
| 73A       | Tb2B               | Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2            | 1.5 Tcy + 40 | —             | ns    | (Note 2)   |
| 74        | Tsch2diL, TscL2diL | Hold Time of SDI Data Input to SCK Edge                                | 100          | —             | ns    |            |
| 75        | TdoR               | SDO Data Output Rise Time                                              | —            | 25            | ns    |            |
| 76        | TdoF               | SDO Data Output Fall Time                                              | —            | 25            | ns    |            |
| 77        | TssH2doZ           | $\overline{SS} \uparrow$ to SDO Output High-Impedance                  | 10           | 50            | ns    |            |
| 78        | Tscr               | SCK Output Rise Time (Master mode)                                     | —            | 25            | ns    |            |
| 79        | Tscf               | SCK Output Fall Time (Master mode)                                     | —            | 25            | ns    |            |
| 80        | Tsch2doV, TscL2doV | SDO Data Output Valid after SCK Edge                                   | —            | 50            | ns    |            |
| 83        | Tsch2ssH, TscL2ssH | $\overline{SS} \uparrow$ after SCK Edge                                | 1.5 Tcy + 40 | —             | ns    |            |

**Note 1:** Requires the use of Parameter #73A.

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

**FIGURE 25-12: EXAMPLE SPI SLAVE MODE TIMING (CKE = 1)**

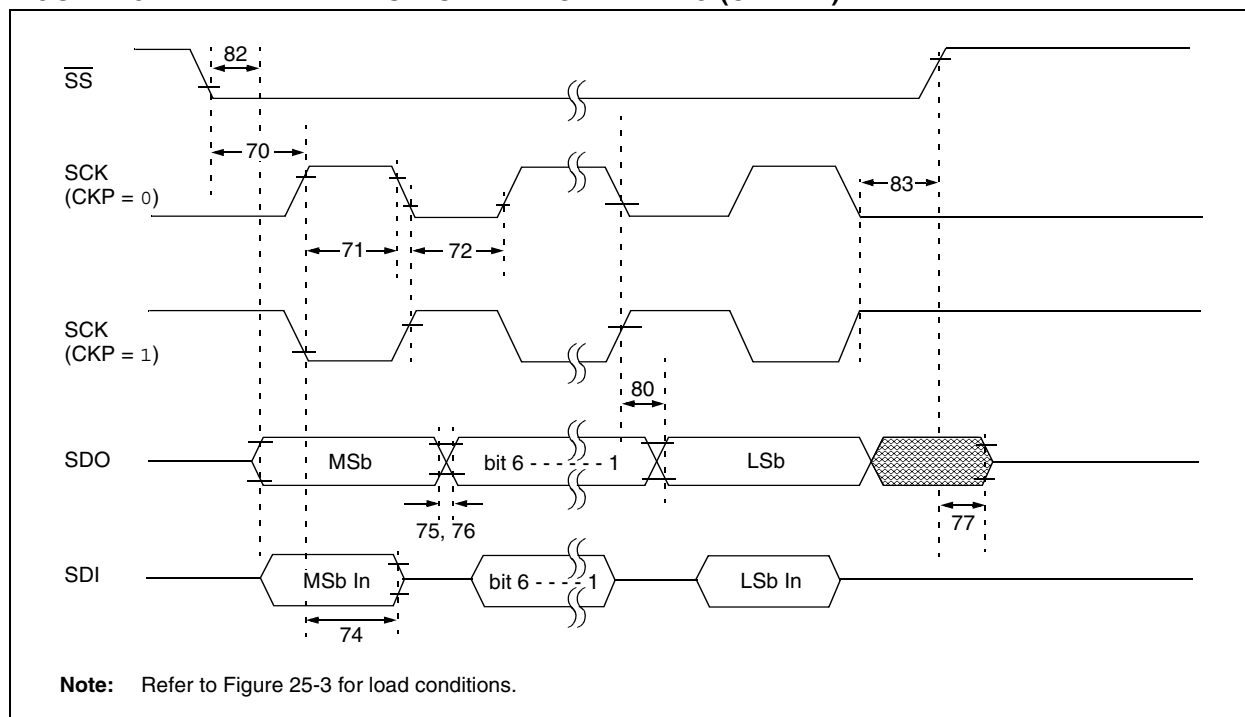


**TABLE 25-16: EXAMPLE SPI SLAVE MODE REQUIREMENTS (CKE = 1)**

| Param No. | Symbol             | Characteristic                                              | Min           | Max | Units | Conditions |
|-----------|--------------------|-------------------------------------------------------------|---------------|-----|-------|------------|
| 70        | TssL2sCH, TssL2sCL | SS ↓ to SCK ↓ or SCK ↑ Input                                | 3 Tcy         | —   | ns    |            |
| 70A       | TssL2WB            | SS to Write to SSPBUF                                       | 3 Tcy         | —   | ns    |            |
| 71        | TsCH               | SCK Input High Time                                         | 1.25 Tcy + 30 | —   | ns    |            |
| 71A       |                    | (Slave mode)                                                |               |     |       |            |
|           |                    | Continuous                                                  | 40            | —   | ns    | (Note 1)   |
|           |                    | Single Byte                                                 | 40            | —   | ns    | (Note 1)   |
| 72        | TsCL               | SCK Input Low Time                                          | 1.25 Tcy + 30 | —   | ns    |            |
| 72A       |                    | (Slave mode)                                                |               |     |       |            |
|           |                    | Continuous                                                  | 40            | —   | ns    | (Note 1)   |
|           |                    | Single Byte                                                 | 40            | —   | ns    | (Note 1)   |
| 73A       | Tb2B               | Last Clock Edge of Byte 1 to the First Clock Edge of Byte 2 | 1.5 Tcy + 40  | —   | ns    | (Note 2)   |
| 74        | TsCH2dIL, TsCL2dIL | Hold Time of SDI Data Input to SCK Edge                     | 100           | —   | ns    |            |
| 75        | TdOR               | SDO Data Output Rise Time                                   | —             | 25  | ns    |            |
| 76        | TdOF               | SDO Data Output Fall Time                                   | —             | 25  | ns    |            |
| 77        | TssH2dOZ           | SS ↑ to SDO Output High-Impedance                           | 10            | 50  | ns    |            |
| 78        | TsCR               | SCK Output Rise Time (Master mode)                          | —             | 25  | ns    |            |
| 79        | TsCF               | SCK Output Fall Time (Master mode)                          | —             | 25  | ns    |            |
| 80        | TsCH2dOV, TsCL2dOV | SDO Data Output Valid after SCK Edge                        | —             | 50  | ns    |            |
| 82        | TssL2dOV           | SDO Data Output Valid after SS ↓ Edge                       | —             | 50  | ns    |            |
| 83        | TsCH2ssH, TsCL2ssH | SS ↑ after SCK Edge                                         | 1.5 Tcy + 40  | —   | ns    |            |

**Note 1:** Requires the use of Parameter #73A.

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

# PIC18F85J11 FAMILY

FIGURE 25-13: I<sup>2</sup>C™ BUS START/STOP BITS TIMING

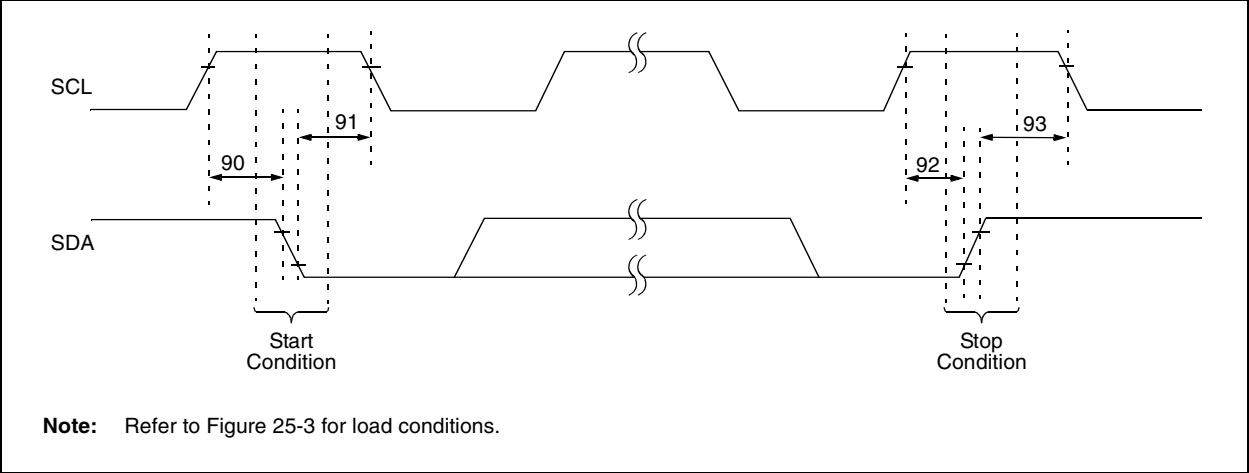


TABLE 25-17: I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS (SLAVE MODE)

| Param. No. | Symbol  | Characteristic             |              | Min  | Max | Units | Conditions                                            |
|------------|---------|----------------------------|--------------|------|-----|-------|-------------------------------------------------------|
| 90         | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4700 | —   | ns    | Only relevant for Repeated Start condition            |
|            |         |                            | 400 kHz mode | 600  | —   |       |                                                       |
| 91         | THD:STA | Start Condition Hold Time  | 100 kHz mode | 4000 | —   | ns    | After this period, the first clock pulse is generated |
|            |         |                            | 400 kHz mode | 600  | —   |       |                                                       |
| 92         | TSU:STO | Stop Condition Setup Time  | 100 kHz mode | 4700 | —   | ns    |                                                       |
|            |         |                            | 400 kHz mode | 600  | —   |       |                                                       |
| 93         | THD:STO | Stop Condition Hold Time   | 100 kHz mode | 4000 | —   | ns    |                                                       |
|            |         |                            | 400 kHz mode | 600  | —   |       |                                                       |



# PIC18F85J11 FAMILY

**FIGURE 25-14: I<sup>2</sup>C™ BUS DATA TIMING**

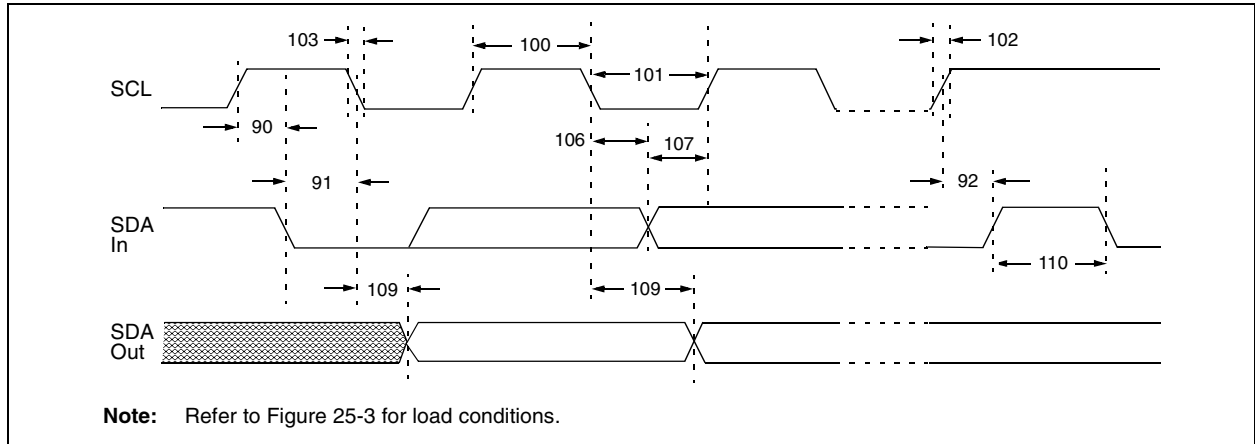


**TABLE 25-18: I<sup>2</sup>C™ BUS DATA REQUIREMENTS (SLAVE MODE)**

| Param. No. | Symbol  | Characteristic             | Min          | Max         | Units | Conditions                                    |
|------------|---------|----------------------------|--------------|-------------|-------|-----------------------------------------------|
| 100        | THIGH   | Clock High Time            | 100 kHz mode | 4.0         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 0.6         | —     | μs                                            |
|            |         |                            | MSSP module  | 1.5 Tcy     | —     |                                               |
| 101        | TLOW    | Clock Low Time             | 100 kHz mode | 4.7         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 1.3         | —     | μs                                            |
|            |         |                            | MSSP module  | 1.5 Tcy     | —     |                                               |
| 102        | TR      | SDA and SCL Rise Time      | 100 kHz mode | —           | 1000  | ns                                            |
|            |         |                            | 400 kHz mode | 20 + 0.1 Cb | 300   | ns<br>Cb is specified to be from 10 to 400 pF |
| 103        | TF      | SDA and SCL Fall Time      | 100 kHz mode | —           | 300   | ns                                            |
|            |         |                            | 400 kHz mode | 20 + 0.1 Cb | 300   | ns<br>Cb is specified to be from 10 to 400 pF |
| 90         | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 0.6         | —     | μs                                            |
| 91         | THD:STA | Start Condition Hold Time  | 100 kHz mode | 4.0         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 0.6         | —     | μs                                            |
| 106        | THD:DAT | Data Input Hold Time       | 100 kHz mode | 0           | —     | ns                                            |
|            |         |                            | 400 kHz mode | 0           | 0.9   | μs                                            |
| 107        | TSU:DAT | Data Input Setup Time      | 100 kHz mode | 250         | —     | ns                                            |
|            |         |                            | 400 kHz mode | 100         | —     | ns                                            |
| 92         | TSU:STO | Stop Condition Setup Time  | 100 kHz mode | 4.7         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 0.6         | —     | μs                                            |
| 109        | TAA     | Output Valid from Clock    | 100 kHz mode | —           | 3500  | ns                                            |
|            |         |                            | 400 kHz mode | —           | —     | ns                                            |
| 110        | TBUF    | Bus Free Time              | 100 kHz mode | 4.7         | —     | μs                                            |
|            |         |                            | 400 kHz mode | 1.3         | —     | μs                                            |
| D102       | CB      | Bus Capacitive Loading     | —            | 400         | pF    |                                               |

**Note 1:** As a transmitter, the device must provide this internal minimum delay time to bridge the undefined region (min. 300 ns) of the falling edge of SCL to avoid unintended generation of Start or Stop conditions.

**2:** A Fast mode I<sup>2</sup>C™ bus device can be used in a Standard mode I<sup>2</sup>C bus system, but the requirement, TSU:DAT ≥ 250 ns, must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, TR max. + TSU:DAT = 1000 + 250 = 1250 ns (according to the Standard mode I<sup>2</sup>C bus specification), before the SCL line is released.

# PIC18F85J11 FAMILY

**FIGURE 25-15: MSSP I<sup>2</sup>C™ BUS START/STOP BITS TIMING WAVEFORMS**

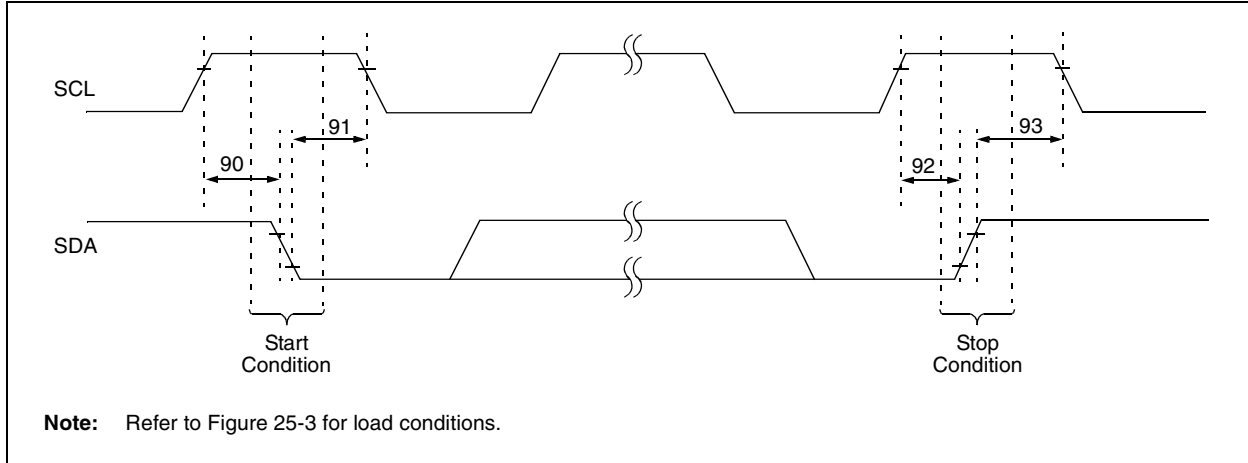
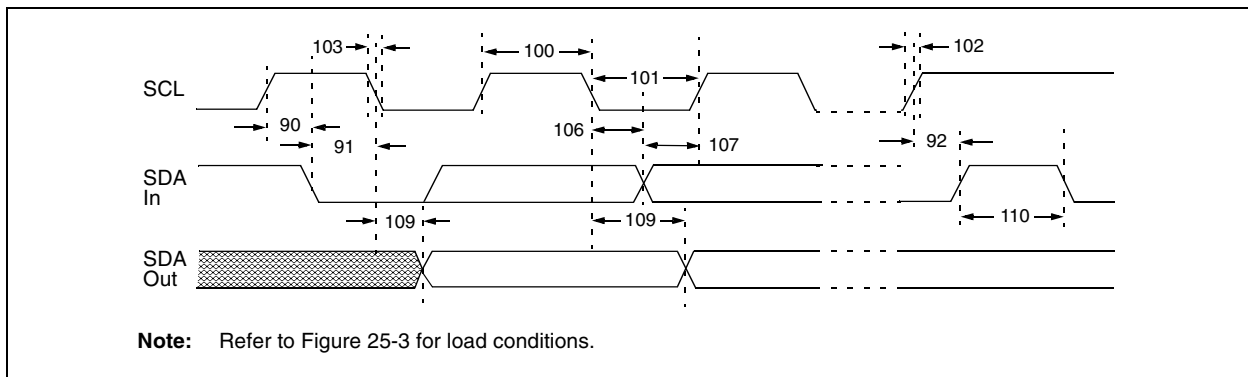


**TABLE 25-19: MSSP I<sup>2</sup>C™ BUS START/STOP BITS REQUIREMENTS**

| Param. No. | Symbol  | Characteristic             | Min                       | Max                   | Units | Conditions                                                  |
|------------|---------|----------------------------|---------------------------|-----------------------|-------|-------------------------------------------------------------|
| 90         | TSU:STA | Start Condition Setup Time | 100 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     | ns<br>Only relevant for Repeated Start condition            |
|            |         |                            | 400 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
| 91         | THD:STA | Start Condition Hold Time  | 100 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     | ns<br>After this period, the first clock pulse is generated |
|            |         |                            | 400 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
| 92         | TSU:STO | Stop Condition Setup Time  | 100 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     | ns                                                          |
|            |         |                            | 400 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
| 93         | THD:STO | Stop Condition Hold Time   | 100 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     | ns                                                          |
|            |         |                            | 400 kHz mode              | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | $2(T_{OSC})(BRG + 1)$ | —     |                                                             |

**Note 1:** Maximum pin capacitance = 10 pF for all I<sup>2</sup>C™ pins.

**FIGURE 25-16: MSSP I<sup>2</sup>C™ BUS DATA TIMING**



# PIC18F85J11 FAMILY

**TABLE 25-20: MSSP I<sup>2</sup>C™ BUS DATA REQUIREMENTS**

| Param. No. | Symbol  | Characteristic             | Min                       | Max              | Units | Conditions |
|------------|---------|----------------------------|---------------------------|------------------|-------|------------|
| 100        | THIGH   | Clock High Time            | 100 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 400 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | 2(Tosc)(BRG + 1) | —     | ms         |
| 101        | TLOW    | Clock Low Time             | 100 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 400 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | 2(Tosc)(BRG + 1) | —     | ms         |
| 102        | TR      | SDA and SCL Rise Time      | 100 kHz mode              | —                | 1000  | ns         |
|            |         |                            | 400 kHz mode              | 20 + 0.1 CB      | 300   | ns         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | —                | 300   | ns         |
| 103        | TF      | SDA and SCL Fall Time      | 100 kHz mode              | —                | 300   | ns         |
|            |         |                            | 400 kHz mode              | 20 + 0.1 CB      | 300   | ns         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | —                | 100   | ns         |
| 90         | TSU:STA | Start Condition Setup Time | 100 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 400 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | 2(Tosc)(BRG + 1) | —     | ms         |
| 91         | THD:STA | Start Condition Hold Time  | 100 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 400 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | 2(Tosc)(BRG + 1) | —     | ms         |
| 106        | THD:DAT | Data Input Hold Time       | 100 kHz mode              | 0                | —     | ns         |
|            |         |                            | 400 kHz mode              | 0                | 0.9   | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | TBD              | —     | ns         |
| 107        | TSU:DAT | Data Input Setup Time      | 100 kHz mode              | 250              | —     | ns         |
|            |         |                            | 400 kHz mode              | 100              | —     | ns         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | TBD              | —     | ns         |
| 92         | TSU:STO | Stop Condition Setup Time  | 100 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 400 kHz mode              | 2(Tosc)(BRG + 1) | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | 2(Tosc)(BRG + 1) | —     | ms         |
| 109        | TAA     | Output Valid from Clock    | 100 kHz mode              | —                | 3500  | ns         |
|            |         |                            | 400 kHz mode              | —                | 1000  | ns         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | —                | —     | ns         |
| 110        | TBUF    | Bus Free Time              | 100 kHz mode              | 4.7              | —     | ms         |
|            |         |                            | 400 kHz mode              | 1.3              | —     | ms         |
|            |         |                            | 1 MHz mode <sup>(1)</sup> | TBD              | —     | ms         |
| D102       | CB      | Bus Capacitive Loading     | —                         | 400              | pF    |            |

**Legend:** TBD = To Be Determined

**Note 1:** Maximum pin capacitance = 10 pF for all I<sup>2</sup>C™ pins.

- 2:** A Fast mode I<sup>2</sup>C bus device can be used in a Standard mode I<sup>2</sup>C bus system, but parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line, parameter #102 + parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCL line is released.

# PIC18F85J11 FAMILY

**FIGURE 25-17: EUSART/AUSART SYNCHRONOUS TRANSMISSION (MASTER/SLAVE) TIMING**

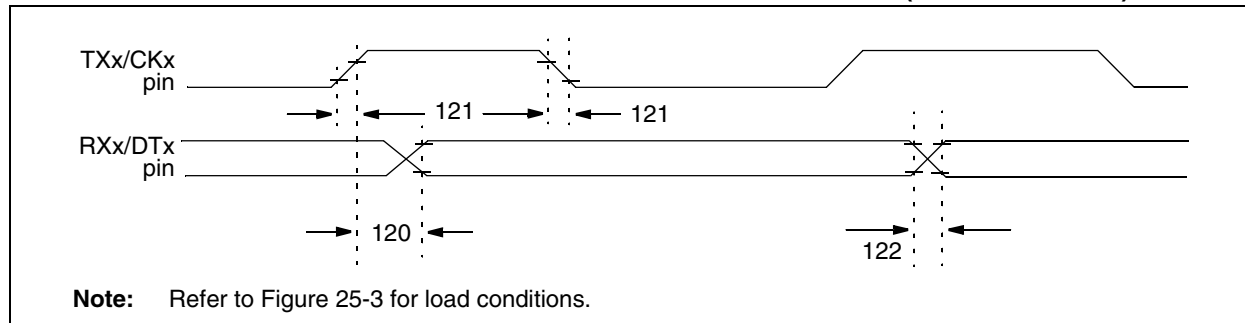


**TABLE 25-21: EUSART/AUSART SYNCHRONOUS TRANSMISSION REQUIREMENTS**

| Param No. | Symbol   | Characteristic                                               | Min | Max | Units | Conditions |
|-----------|----------|--------------------------------------------------------------|-----|-----|-------|------------|
| 120       | TCKH2DTV | SYNC XMIT (MASTER and SLAVE)<br>Clock High to Data Out Valid | —   | 40  | ns    |            |
| 121       | TCKRF    | Clock Out Rise Time and Fall Time (Master mode)              | —   | 20  | ns    |            |
| 122       | TDTRF    | Data Out Rise Time and Fall Time                             | —   | 20  | ns    |            |

**FIGURE 25-18: EUSART/AUSART SYNCHRONOUS RECEIVE (MASTER/SLAVE) TIMING**

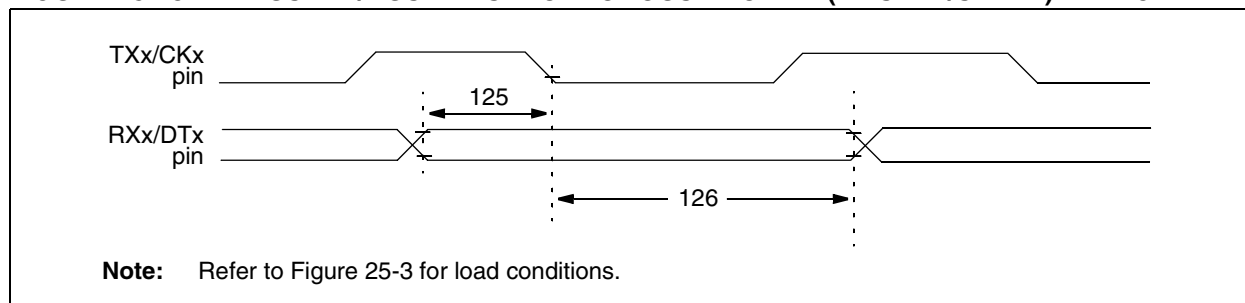


**TABLE 25-22: EUSART/AUSART SYNCHRONOUS RECEIVE REQUIREMENTS**

| Param. No. | Symbol   | Characteristic                                                        | Min | Max | Units | Conditions |
|------------|----------|-----------------------------------------------------------------------|-----|-----|-------|------------|
| 125        | TDTV2CKL | SYNC RCV (MASTER and SLAVE)<br>Data Hold before CKx ↓ (DTx hold time) | 10  | —   | ns    |            |
| 126        | TCKL2DTL | Data Hold after CKx ↓ (DTx hold time)                                 | 15  | —   | ns    |            |

# PIC18F85J11 FAMILY

**TABLE 25-23: A/D CONVERTER CHARACTERISTICS: PIC18F85J11 FAMILY (INDUSTRIAL)**

| Param No. | Symbol           | Characteristic                                       | Min                       | Typ | Max             | Units      | Conditions                         |
|-----------|------------------|------------------------------------------------------|---------------------------|-----|-----------------|------------|------------------------------------|
| A01       | NR               | Resolution                                           | —                         | —   | 10              | bits       |                                    |
| A03       | EIL              | Integral Linearity Error                             | —                         | —   | $<\pm 1$        | LSb        | $\Delta V_{REF} \geq 3.0V$         |
| A04       | EDL              | Differential Linearity Error                         | —                         | —   | $<\pm 1$        | LSb        | $\Delta V_{REF} \geq 3.0V$         |
| A06       | EOFF             | Offset Error                                         | —                         | —   | $<\pm 3$        | LSb        | $\Delta V_{REF} \geq 3.0V$         |
| A07       | EGN              | Gain Error                                           | —                         | —   | $<\pm 3$        | LSb        | $\Delta V_{REF} \geq 3.0V$         |
| A10       | —                | Monotonicity                                         | Guaranteed <sup>(1)</sup> |     |                 | —          | $V_{SS} \leq V_{AIN} \leq V_{REF}$ |
| A20       | $\Delta V_{REF}$ | Reference Voltage Range<br>( $V_{REFH} - V_{REFL}$ ) | 2.0                       | —   | —               | V          | $V_{DD} < 3.0V$                    |
|           |                  |                                                      | 3                         | —   | —               | V          | $V_{DD} \geq 3.0V$                 |
| A21       | $V_{REFH}$       | Reference Voltage High                               | $V_{SS}$                  | —   | $V_{REFH}$      | V          |                                    |
| A22       | $V_{REFL}$       | Reference Voltage Low                                | $V_{SS} - 0.3V$           | —   | $V_{DD} - 3.0V$ | V          |                                    |
| A25       | $V_{AIN}$        | Analog Input Voltage                                 | $V_{REFL}$                | —   | $V_{REFH}$      | V          |                                    |
| A30       | $Z_{AIN}$        | Recommended Impedance of<br>Analog Voltage Source    | —                         | —   | 2.5             | k $\Omega$ |                                    |
| A50       | $I_{REF}$        | $V_{REF}$ Input Current <sup>(2)</sup>               | —                         | —   | 5               | $\mu A$    | During $V_{AIN}$ acquisition       |
|           |                  |                                                      | —                         | —   | 150             | $\mu A$    | 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:**  $V_{REFH}$  current is from RA3/AN3/ $V_{REF+}$  pin or  $V_{DD}$ , whichever is selected as the  $V_{REFH}$  source.  
 $V_{REFL}$  current is from RA2/AN2/ $V_{REF-}$  pin or  $V_{SS}$ , whichever is selected as the  $V_{REFL}$  source.

# PIC18F85J11 FAMILY

**FIGURE 25-19: A/D CONVERSION TIMING**

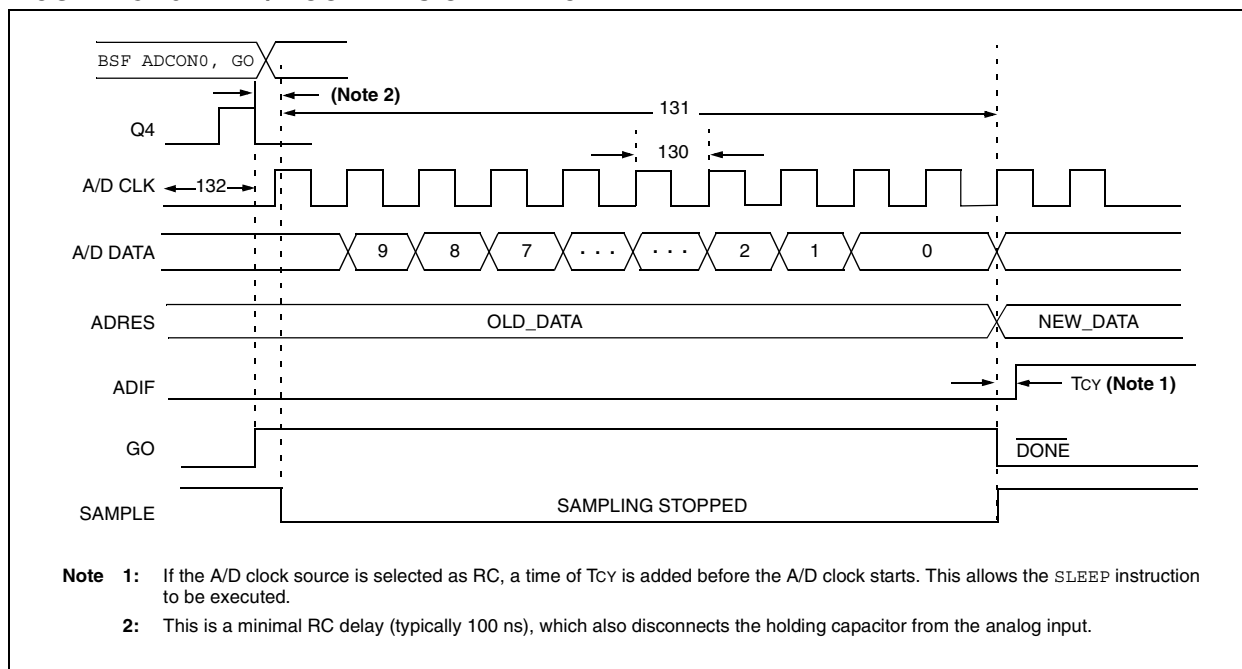


**TABLE 25-24: A/D CONVERSION REQUIREMENTS**

| Param No. | Symbol | Characteristic                                                  | Min | Max                 | Units   | Conditions                   |
|-----------|--------|-----------------------------------------------------------------|-----|---------------------|---------|------------------------------|
| 130       | TAD    | A/D Clock Period                                                | 0.7 | 25.0 <sup>(1)</sup> | $\mu$ s | TOSC based, VREF $\geq$ 3.0V |
|           |        |                                                                 | TBD | 1                   | $\mu$ s | A/D RC mode                  |
| 131       | TCNV   | Conversion Time (not including acquisition time) <sup>(2)</sup> | 11  | 12                  | TAD     |                              |
| 132       | TACQ   | Acquisition Time <sup>(3)</sup>                                 | 1.4 | —                   | $\mu$ s | -40°C to +85°C               |
| 135       | TSWC   | Switching Time from Convert $\rightarrow$ Sample                | —   | (Note 4)            |         |                              |
| TBD       | TDIS   | Discharge Time                                                  | 0.2 | —                   | $\mu$ s |                              |

**Legend:** TBD = To Be Determined

- Note 1:** The time of the A/D clock period is dependent on the device frequency and the TAD clock divider.
- Note 2:** ADRES registers may be read on the following  $T_{CY}$  cycle.
- Note 3:** The time for the holding capacitor to acquire the “New” input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD). The source impedance ( $R_s$ ) on the input channels is 50 $\Omega$ .
- Note 4:** On the following cycle of the device clock.

## 26.0 DC AND AC CHARACTERISTICS GRAPHS AND TABLES

Graphs and tables are not available at this time.

# PIC18F85J11 FAMILY

---

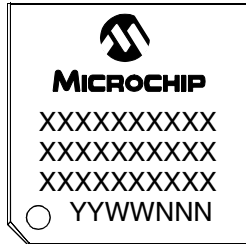
NOTES:



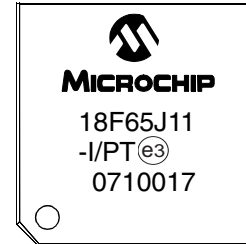
## 27.0 PACKAGING INFORMATION

### 27.1 Package Marking Information

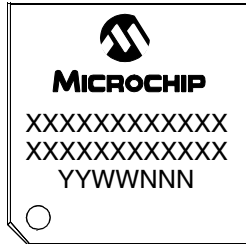
64-Lead TQFP



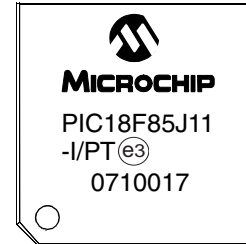
Example



80-Lead TQFP



Example



|                |        |                                                                                                                  |
|----------------|--------|------------------------------------------------------------------------------------------------------------------|
| <b>Legend:</b> | XX...X | Customer-specific information                                                                                    |
|                | Y      | Year code (last digit of calendar year)                                                                          |
|                | YY     | Year code (last 2 digits of calendar year)                                                                       |
|                | WW     | Week code (week of January 1 is week '01')                                                                       |
|                | NNN    | Alphanumeric traceability code                                                                                   |
|                | (e3)   | Pb-free JEDEC designator for Matte Tin (Sn)                                                                      |
|                | *      | This package is Pb-free. The Pb-free JEDEC designator (e3) can be found on the outer packaging for this package. |

**Note:** In the event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

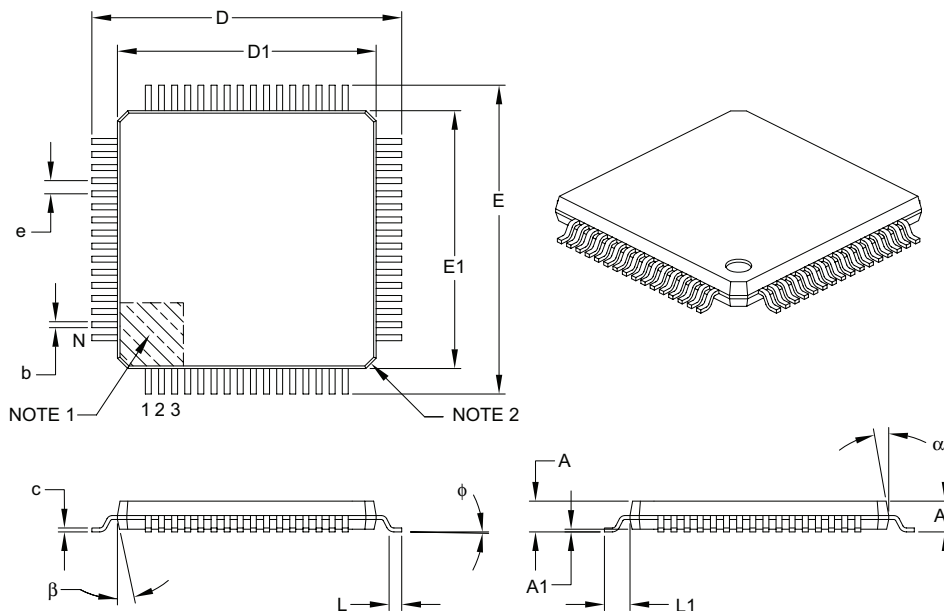
# PIC18F85J11 FAMILY

## 27.2 Package Details

The following sections give the technical details of the packages.

### 64-Lead Plastic Thin Quad Flatpack (PT) – 10x10x1 mm Body, 2.00 mm Footprint [TQFP]

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



| Dimension Limits         | Units | MILLIMETERS |      |      |
|--------------------------|-------|-------------|------|------|
|                          |       | MIN         | NOM  | MAX  |
| Number of Leads          | N     | 64          |      |      |
| Lead Pitch               | e     | 0.50 BSC    |      |      |
| Overall Height           | A     | –           | –    | 1.20 |
| Molded Package Thickness | A2    | 0.95        | 1.00 | 1.05 |
| Standoff                 | A1    | 0.05        | –    | 0.15 |
| Foot Length              | L     | 0.45        | 0.60 | 0.75 |
| Footprint                | L1    | 1.00 REF    |      |      |
| Foot Angle               | φ     | 0°          | 3.5° | 7°   |
| Overall Width            | E     | 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     | 0.17        | 0.22 | 0.27 |
| Mold Draft Angle Top     | α     | 11°         | 12°  | 13°  |
| Mold Draft Angle Bottom  | β     | 11°         | 12°  | 13°  |

#### Notes:

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

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

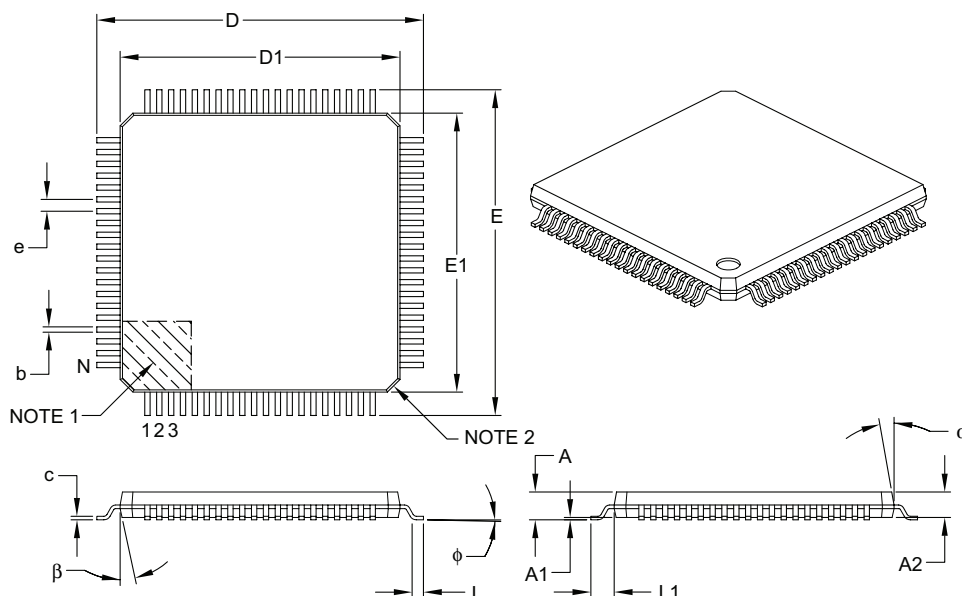
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

# PIC18F85J11 FAMILY

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

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



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

### Notes:

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

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

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

Microchip Technology Drawing C04-092B

# PIC18F85J11 FAMILY

---

NOTES:

# PIC18F85J11 FAMILY

## APPENDIX A: REVISION HISTORY

### Revision A (October 2006)

Original data sheet for PIC18F85J11 family devices.

### Revision B (March 2007)

Updated power-down and supply current electrical characteristics and package detail drawings.

### Revision C (April 2007)

Updated electrical characteristics.

## APPENDIX B: MIGRATION BETWEEN HIGH-END DEVICE FAMILIES

Devices in the PIC18F85J11 and PIC18F8722 families are very similar in their functions and feature sets. However, there are some potentially important differences which should be considered when migrating an application across device families to achieve a new design goal. These are summarized in Table B-1. The areas of difference which could have a major impact on migration are discussed in greater detail later in this section.

**TABLE B-1: NOTABLE DIFFERENCES BETWEEN PIC18F85J11 AND PIC18F8722 FAMILIES**

| Characteristic                      | PIC18F85J11 Family                             | PIC18F8722 Family                                      |
|-------------------------------------|------------------------------------------------|--------------------------------------------------------|
| Operating Frequency                 | 40 MHz @ 2.15V                                 | 40 MHz @ 4.2V                                          |
| Supply Voltage                      | 2.0V-3.6V, dual voltage requirement            | 2.0V-5.5V                                              |
| Operating Current                   | Low                                            | Lower                                                  |
| Program Memory Endurance            | 1,000 write/erase cycles (typical)             | 100,000 write/erase cycles (typical)                   |
| I/O Sink/Source at 25 mA            | PORTB and PORTC only                           | All ports                                              |
| Input Voltage Tolerance on I/O pins | 5.5V on digital only pins                      | VDD on all I/O pins                                    |
| I/O                                 | 68 (RF0 is not available)                      | 70                                                     |
| Pull-ups                            | PORTB, PORTD, PORTE and PORTJ                  | PORTB                                                  |
| Oscillator Options                  | Limited options (EC, HS, PLL, flexible INTRC)  | More options (EC, HS, XT, LP, RC, PLL, flexible INTRC) |
| Program Memory Retention            | 20 years (minimum)                             | 40 years (minimum)                                     |
| Self-Writes to Program Memory       | Available                                      | Available                                              |
| Programming Time (normalized)       | 156 $\mu$ s/byte (10 ms/64-byte block)         | 15.6 $\mu$ s/byte (1 ms/64)                            |
| Programming Entry                   | Low voltage, key sequence                      | VPP and LVP                                            |
| Code Protection                     | Single block, all or nothing                   | Multiple code protection blocks                        |
| Configuration Words                 | Stored in last 4 words of program memory space | Stored in configuration space, starting at 300000h     |
| Power-up Timer                      | Always on                                      | Configurable                                           |
| Data EEPROM                         | Use self-programming                           | Available                                              |
| BOR                                 | Simple BOR with voltage regulator              | Programmable BOR                                       |
| LVD                                 | Simple LVD with voltage regulator              | Available                                              |
| A/D Channels                        | 12                                             | 16                                                     |
| A/D Calibration                     | Required                                       | Not required                                           |
| Microprocessor mode (EMB)           | Self-calibration feature                       | Available                                              |
| External Memory Addressing          | Address shifting available                     | Address shifting not available                         |
| In-Circuit Emulation                | Not available                                  | Available                                              |

# PIC18F85J11 FAMILY

---

## B.1 Power Requirement Differences

The most significant difference between the PIC18F85J11 and PIC18F8722 device families is the power requirements. PIC18F85J11 devices are designed on a smaller process; this results in lower maximum voltage and higher leakage current.

The operating voltage range for PIC18F85J11 devices is 2.0V to 3.6V. In addition, these devices have split power requirements: one for the core logic and one for the I/O. One of the VDD pins is separated for the core logic supply, VDDCORE. This pin has specific voltage and capacitor requirements as described in **Section 25.0 “Electrical Characteristics”**.

The current specifications for PIC18F85J11 devices are yet to be determined.

## B.2 Pin Differences

There are several differences in the pinout between the PIC18F85J11 and the PIC18F8722 families:

- Input voltage tolerance
- Output current capabilities
- Available I/O

Pins on the PIC18F85J11 that have digital only input capability will tolerate voltages up to 5.5V and are thus tolerant to voltages above VDD. Table 10-1 in **Section 10.1 “I/O Port Pin Capabilities”** contains the complete list.

In addition to input differences, there are output differences as well. PIC18F85J11 devices have three classes of pin output current capability: high, medium and low. Not all I/O pins can source or sink equal levels of current. Only PORTB and PORTC support the 25 mA source/sink capability that is supported by all output pins on the PIC18F8722. Table 10-2 in **Section 10.1 “I/O Port Pin Capabilities”** contains the complete list of output capabilities.

There are additional differences in how some pin functions are implemented on PIC18F85J11 devices. First, the MCLR pin is dedicated only to MCLR and cannot be configured as an input (RG5). Finally, RF0 does not exist on PIC18F85J11 devices.

All of these pin differences (including power pin differences) should be accounted for when making a conversion between PIC18F8722 and PIC18F85J11 devices.

## B.3 Oscillator Differences

PIC18F8722 and PIC18F85J11 family devices share a similar range of oscillator options. The major difference is that PIC18F85J11 family devices support a smaller number of primary (external) oscillator options, namely HS and EC Oscillator modes.

While both device families have an internal PLL that can be used with the primary oscillators, the PLL for the PIC18F85J11 family is not enabled as a device configuration option. Instead, it must be enabled in software.

The clocking differences should be considered when making a conversion between the PIC18F8722 and PIC18F85J11 device families.

## B.4 Peripherals

Peripherals must also be considered when making a conversion between the PIC18F85J11 and the PIC18F8722:

- **External Memory Bus:** The External Memory Bus (EMB) on the PIC18F85J11 does not support Microcontroller mode; however, it does support external address offset.
- **A/D Converter:** There are only 12 channels on PIC18F85J11 devices. The converters for these devices also require a calibration step prior to normal operation.
- **Data EEPROM:** PIC18F85J11 devices do not have this module.
- **BOR:** PIC18F85J11 devices do not have a programmable BOR. Simple Brown-out Reset capability is provided through the use of the internal voltage regulator.
- **LVD:** PIC18F85J11 devices do not have a separate programmable LVD module. Simple, Low-Voltage Detection capability with a configurable interrupt is provided through the use of the internal voltage regulator.

## INDEX

### A

|                                                                                                  |          |
|--------------------------------------------------------------------------------------------------|----------|
| A/D .....                                                                                        | 251      |
| A/D Converter Interrupt, Configuring .....                                                       | 255      |
| Acquisition Requirements .....                                                                   | 256      |
| ADCAL Bit .....                                                                                  | 259      |
| ADCON0 Register .....                                                                            | 251      |
| ADCON1 Register .....                                                                            | 251      |
| ADCON2 Register .....                                                                            | 251      |
| ADRESH Register .....                                                                            | 251, 254 |
| ADRESL Register .....                                                                            | 251      |
| Analog Port Pins, Configuring .....                                                              | 257      |
| Associated Registers .....                                                                       | 259      |
| Automatic Acquisition Time .....                                                                 | 257      |
| Configuring the Module .....                                                                     | 255      |
| Conversion Clock (TAD) .....                                                                     | 257      |
| Conversion Requirements .....                                                                    | 372      |
| Conversion Status (GO/DONE Bit) .....                                                            | 254      |
| Conversions .....                                                                                | 258      |
| Converter Calibration .....                                                                      | 259      |
| Converter Characteristics .....                                                                  | 371      |
| Operation in Power-Managed Modes .....                                                           | 259      |
| Special Event Trigger (CCP2) .....                                                               | 258      |
| Use of the CCP2 Trigger .....                                                                    | 258      |
| Absolute Maximum Ratings .....                                                                   | 339      |
| AC (Timing) Characteristics .....                                                                | 354      |
| Load Conditions for Device Timing .....                                                          |          |
| Specifications .....                                                                             | 355      |
| Parameter Symbolology .....                                                                      | 354      |
| Temperature and Voltage .....                                                                    |          |
| Specifications .....                                                                             | 355      |
| Timing Conditions .....                                                                          | 355      |
| ACKSTAT .....                                                                                    | 207      |
| ACKSTAT Status Flag .....                                                                        | 207      |
| ADCAL Bit .....                                                                                  | 259      |
| ADCON0 Register .....                                                                            | 251      |
| GO/DONE Bit .....                                                                                | 254      |
| ADCON1 Register .....                                                                            | 251      |
| ADCON2 Register .....                                                                            | 251      |
| ADDFSR .....                                                                                     | 328      |
| ADDLW .....                                                                                      | 291      |
| Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART). <i>See</i> AUSART. |          |
| ADDULNK .....                                                                                    | 328      |
| ADDWF .....                                                                                      | 291      |
| ADDWFC .....                                                                                     | 292      |
| ADRESH Register .....                                                                            | 251      |
| ADRESL Register .....                                                                            | 251, 254 |
| Analog-to-Digital Converter. <i>See</i> A/D.                                                     |          |
| ANDLW .....                                                                                      | 292      |
| ANDWF .....                                                                                      | 293      |
| Assembler .....                                                                                  |          |
| MPASM Assembler .....                                                                            | 336      |
| AUSART .....                                                                                     |          |
| Asynchronous Mode .....                                                                          | 242      |
| Associated Registers, Receive .....                                                              | 245      |
| Associated Registers, Transmit .....                                                             | 243      |
| Receiver .....                                                                                   | 244      |
| Setting up 9-Bit Mode with .....                                                                 |          |
| Address Detect .....                                                                             | 244      |
| Transmitter .....                                                                                | 242      |

|                                            |     |
|--------------------------------------------|-----|
| Baud Rate Generator (BRG) .....            | 240 |
| Associated Registers .....                 | 240 |
| Baud Rate Error, Calculating .....         | 240 |
| Baud Rates, Asynchronous Modes .....       | 241 |
| High Baud Rate Select (BRGH Bit) .....     | 240 |
| Operation in Power-Managed Modes .....     | 240 |
| Sampling .....                             | 240 |
| Control Registers .....                    | 237 |
| Synchronous Master Mode .....              | 246 |
| Associated Registers, Receive .....        | 248 |
| Associated Registers, Transmit .....       | 247 |
| Reception .....                            | 248 |
| Transmission .....                         | 246 |
| Synchronous Slave Mode .....               | 249 |
| Associated Registers, Receive .....        | 250 |
| Associated Registers, Transmit .....       | 249 |
| Reception .....                            | 250 |
| Transmission .....                         | 249 |
| Auto-Wake-up on Sync Break Character ..... | 230 |

### B

|                                                 |     |
|-------------------------------------------------|-----|
| Baud Rate Generator .....                       | 203 |
| BC .....                                        | 293 |
| BCF .....                                       | 294 |
| BF .....                                        | 207 |
| BF Status Flag .....                            | 207 |
| Block Diagrams .....                            |     |
| 16-Bit Byte Select Mode .....                   | 99  |
| 16-Bit Byte Write Mode .....                    | 97  |
| 16-Bit Word Write Mode .....                    | 98  |
| 8-Bit Multiplexed Mode .....                    | 101 |
| A/D .....                                       | 254 |
| Analog Input Model .....                        | 255 |
| AUSART Receive .....                            | 244 |
| AUSART Transmit .....                           | 242 |
| Baud Rate Generator .....                       | 203 |
| Capture Mode Operation .....                    | 166 |
| Comparator Analog Input Model .....             | 265 |
| Comparator I/O Operating Modes .....            | 262 |
| Comparator Output .....                         | 264 |
| Comparator Voltage Reference .....              | 268 |
| Comparator Voltage Reference .....              |     |
| Output Buffer Example .....                     | 269 |
| Compare Mode Operation .....                    | 167 |
| Connections for On-Chip Voltage Regulator ..... | 279 |
| Device Clock .....                              | 29  |
| EUSART Receive .....                            | 228 |
| EUSART Transmit .....                           | 226 |
| External Power-on Reset Circuit .....           |     |
| (Slow VDD Power-up) .....                       | 47  |
| Fail-Safe Clock Monitor .....                   | 282 |
| Generic I/O Port Operation .....                | 123 |
| Interrupt Logic .....                           | 108 |
| MSSP (I <sup>2</sup> C Master Mode) .....       | 201 |
| MSSP (I <sup>2</sup> C Mode) .....              | 182 |
| MSSP (SPI Mode) .....                           | 173 |
| On-Chip Reset Circuit .....                     | 45  |
| PIC18F6XJ11 .....                               | 10  |
| PIC18F8XJ11 .....                               | 11  |
| PLL .....                                       | 34  |

# PIC18F85J11 FAMILY

|                                             |     |
|---------------------------------------------|-----|
| PORTD and PORTE (Parallel Slave Port) ..... | 144 |
| PWM Operation (Simplified) .....            | 169 |
| Reads From Flash Program Memory .....       | 87  |
| Single Comparator .....                     | 263 |
| Table Read Operation .....                  | 83  |
| Table Write Operation .....                 | 84  |
| Table Writes to Flash Program Memory .....  | 89  |
| Timer0 in 16-Bit Mode .....                 | 148 |
| Timer0 in 8-Bit Mode .....                  | 148 |
| Timer1 (16-Bit Read/Write Mode) .....       | 152 |
| Timer1 (8-Bit Mode) .....                   | 152 |
| Timer2 .....                                | 158 |
| Timer3 (16-Bit Read/Write Mode) .....       | 160 |
| Timer3 (8-Bit Mode) .....                   | 160 |
| Watchdog Timer .....                        | 278 |
| BN .....                                    | 294 |
| BNC .....                                   | 295 |
| BNN .....                                   | 295 |
| BNOV .....                                  | 296 |
| BNZ .....                                   | 296 |
| BOR. <i>See</i> Brown-out Reset.            |     |
| BOV .....                                   | 299 |
| BRA .....                                   | 297 |
| BRG. <i>See</i> Baud Rate Generator.        |     |
| BRGH Bit                                    |     |
| TXSTA1 Register .....                       | 221 |
| TXSTA2 Register .....                       | 240 |
| Brown-out Reset (BOR) .....                 | 47  |
| and On-Chip Voltage Regulator .....         | 280 |
| Detecting .....                             | 47  |
| BSF .....                                   | 297 |
| BTFSC .....                                 | 298 |
| BTFSS .....                                 | 298 |
| BTG .....                                   | 299 |
| BZ .....                                    | 300 |
| <b>C</b>                                    |     |
| C Compilers                                 |     |
| MPLAB C18 .....                             | 336 |
| MPLAB C30 .....                             | 336 |
| CALL .....                                  | 300 |
| CALLW .....                                 | 329 |
| Capture (CCP Module) .....                  | 166 |
| Associated Registers .....                  | 168 |
| CCPR2H:CCPR2L Registers .....               | 166 |
| CCPx Pin Configuration .....                | 166 |
| Software Interrupt .....                    | 166 |
| Timer1/Timer3 Mode Selection .....          | 166 |
| Capture/Compare/PWM (CCP) .....             | 163 |
| Capture Mode. <i>See</i> Capture.           |     |
| CCPRxH Register .....                       | 164 |
| CCPRxL Register .....                       | 164 |
| CCPx Mode and Timer Resources .....         | 164 |
| Compare Mode. <i>See</i> Compare.           |     |
| Configuration .....                         | 164 |
| Interaction of CCP1 and CCP2 for            |     |
| Timer Resources .....                       | 165 |
| Interconnect Configurations .....           | 164 |
| Clock Sources .....                         | 31  |
| Default System Clock on Reset .....         | 32  |
| Selection Using OSCCON Register .....       | 32  |
| CLRF .....                                  | 301 |
| CLRWDT .....                                | 301 |

|                                              |          |
|----------------------------------------------|----------|
| Code Examples                                |          |
| 16 x 16 Signed Multiply Routine .....        | 106      |
| 16 x 16 Unsigned Multiply Routine .....      | 106      |
| 8 x 8 Signed Multiply Routine .....          | 105      |
| 8 x 8 Unsigned Multiply Routine .....        | 105      |
| Changing Between Capture Prescalers .....    | 166      |
| Computed GOTO Using an Offset Value .....    | 63       |
| Erasing a Flash Program Memory Row .....     | 88       |
| Fast Register Stack .....                    | 63       |
| How to Clear RAM (Bank 1) Using              |          |
| Indirect Addressing .....                    | 76       |
| Implementing a Real-Time Clock Using         |          |
| a Timer1 Interrupt Service .....             | 155      |
| Initializing PORTA .....                     | 124      |
| Initializing PORTB .....                     | 126      |
| Initializing PORTC .....                     | 128      |
| Initializing PORTD .....                     | 131      |
| Initializing PORTE .....                     | 134      |
| Initializing PORTF .....                     | 137      |
| Initializing PORTG .....                     | 139      |
| Initializing PORTH .....                     | 141      |
| Initializing PORTJ .....                     | 142      |
| Loading the SSPBUF (SSPSR) Register .....    | 176      |
| Reading a Flash Program Memory Word .....    | 87       |
| Saving STATUS, WREG and BSR                  |          |
| Registers in RAM .....                       | 122      |
| Writing to Flash Program Memory .....        | 90       |
| Code Protection .....                        | 271      |
| COMF .....                                   | 302      |
| Comparator .....                             | 261      |
| Analog Input Connection Considerations ..... | 265      |
| Associated Registers .....                   | 265      |
| Configuration .....                          | 262      |
| Effects of a Reset .....                     | 264      |
| Interrupts .....                             | 264      |
| Operation .....                              | 263      |
| Operation During Sleep .....                 | 264      |
| Outputs .....                                | 263      |
| Reference .....                              | 263      |
| External Signal .....                        | 263      |
| Internal Signal .....                        | 263      |
| Response Time .....                          | 263      |
| Comparator Specifications .....              | 353      |
| Comparator Voltage Reference .....           | 267      |
| Accuracy and Error .....                     | 268      |
| Associated Registers .....                   | 269      |
| Configuring .....                            | 267      |
| Connection Considerations .....              | 268      |
| Effects of a Reset .....                     | 268      |
| Operation During Sleep .....                 | 268      |
| Compare (CCP Module) .....                   | 167      |
| Associated Registers .....                   | 168      |
| CCPR2 Register .....                         | 167      |
| CCPx Pin Configuration .....                 | 167      |
| Software Interrupt .....                     | 167      |
| Special Event Trigger .....                  | 161, 167 |
| Timer1/Timer3 Mode Selection .....           | 167      |
| Compare (CCP2 Module)                        |          |
| Special Event Trigger .....                  | 258      |
| Computed GOTO .....                          | 63       |
| Configuration Bits .....                     | 271      |
| Configuration Bits, Device IDs               |          |
| Associated Registers .....                   | 272      |



# PIC18F85J11 FAMILY

|                                            |     |
|--------------------------------------------|-----|
| Configuration Register Protection .....    | 284 |
| Core Features .....                        |     |
| Easy Migration .....                       | 8   |
| Extended Instruction Set .....             | 7   |
| External Memory Bus .....                  | 7   |
| Memory Options .....                       | 7   |
| nanoWatt Technology .....                  | 7   |
| Oscillator Options and Features .....      | 7   |
| CPFSEQ .....                               | 302 |
| CPFSGT .....                               | 303 |
| CPFSLT .....                               | 303 |
| Crystal Oscillator/Ceramic Resonator ..... | 33  |
| Customer Change Notification Service ..... | 391 |
| Customer Notification Service .....        | 391 |
| Customer Support .....                     | 391 |

## D

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| Data Addressing Modes .....                                                   | 76  |
| Comparing Addressing Modes with the<br>Extended Instruction Set Enabled ..... | 80  |
| Direct .....                                                                  | 76  |
| Indexed Literal Offset .....                                                  | 79  |
| BSR .....                                                                     | 81  |
| Instructions Affected .....                                                   | 79  |
| Mapping Access Bank .....                                                     | 81  |
| Indirect .....                                                                | 76  |
| Inherent and Literal .....                                                    | 76  |
| Data Memory .....                                                             | 66  |
| Access Bank .....                                                             | 69  |
| Bank Select Register (BSR) .....                                              | 66  |
| Extended Instruction Set .....                                                | 79  |
| General Purpose Registers .....                                               | 69  |
| Memory Maps .....                                                             |     |
| PIC18FX3J11/X4J11 Devices .....                                               | 67  |
| PIC18FX5J11 Devices .....                                                     | 68  |
| Special Function Registers .....                                              | 70  |
| Special Function Registers .....                                              | 70  |
| DAW .....                                                                     | 304 |
| DC and AC Characteristics .....                                               |     |
| Graphs and Tables .....                                                       | 373 |
| DC Characteristics .....                                                      |     |
| PIC18F85J11 Family .....                                                      | 350 |
| Power-Down and Supply Current .....                                           | 342 |
| Supply Voltage .....                                                          | 341 |
| DCFSNZ .....                                                                  | 305 |
| DECF .....                                                                    | 304 |
| DECFSZ .....                                                                  | 305 |
| Default System Clock .....                                                    | 32  |
| Details on Individual Family Members .....                                    | 8   |
| Development Support .....                                                     | 335 |
| Device Overview .....                                                         | 7   |
| Features (64-Pin Devices) .....                                               | 9   |
| Features (80-Pin Devices) .....                                               | 9   |
| Direct Addressing .....                                                       | 77  |

## E

|                                                                                                  |     |
|--------------------------------------------------------------------------------------------------|-----|
| Effect on Standard PIC18 Instructions .....                                                      | 332 |
| Effects of Power-Managed Modes on<br>Various Clock Sources .....                                 | 36  |
| Electrical Characteristics .....                                                                 | 339 |
| Enhanced Universal Synchronous Asynchronous<br>Receiver Transmitter (EUSART). <i>See</i> EUSART. |     |
| ENVREG Pin .....                                                                                 | 279 |

## Equations

|                                                            |     |
|------------------------------------------------------------|-----|
| A/D Acquisition Time .....                                 | 256 |
| A/D Minimum Charging Time .....                            | 256 |
| Calculating the Minimum Required<br>Acquisition Time ..... | 256 |

|              |   |
|--------------|---|
| Errata ..... | 5 |
|--------------|---|

## EUSART

|                                                    |     |
|----------------------------------------------------|-----|
| Asynchronous Mode .....                            | 226 |
| Associated Registers, Receive .....                | 229 |
| Associated Registers, Transmit .....               | 227 |
| Auto-Wake-up on Sync Break .....                   | 230 |
| Break Character Sequence .....                     | 231 |
| Receiver .....                                     | 228 |
| Setting up 9-Bit Mode with<br>Address Detect ..... | 228 |
| Transmitter .....                                  | 226 |
| Baud Rate Generator (BRG) .....                    | 221 |
| Associated Registers .....                         | 221 |
| Auto-Baud Rate Detect .....                        | 224 |
| Baud Rate Error, Calculating .....                 | 221 |
| Baud Rates, Asynchronous Modes .....               | 222 |
| High Baud Rate Select (BRGH Bit) .....             | 221 |
| Operation in Power-Managed Modes .....             | 221 |
| Sampling .....                                     | 221 |
| Control Registers .....                            | 217 |
| Synchronous Master Mode .....                      | 232 |
| Associated Registers, Receive .....                | 234 |
| Associated Registers, Transmit .....               | 233 |
| Reception .....                                    | 234 |
| Transmission .....                                 | 232 |
| Synchronous Slave Mode .....                       | 235 |
| Associated Registers, Receive .....                | 236 |
| Associated Registers, Transmit .....               | 235 |
| Reception .....                                    | 236 |
| Transmission .....                                 | 235 |
| Extended Instruction Set .....                     | 327 |
| ADDFSR .....                                       | 328 |
| ADDULNK .....                                      | 328 |
| CALLW .....                                        | 329 |
| Considerations when Enabling .....                 | 332 |
| MOVSF .....                                        | 329 |
| MOVSS .....                                        | 330 |
| PUSHL .....                                        | 330 |
| SUBFSR .....                                       | 331 |
| SUBULNK .....                                      | 331 |
| Syntax .....                                       | 327 |
| Use with MPLAB IDE Tools .....                     | 334 |
| Extended Microcontroller Mode .....                | 96  |
| External Memory Bus .....                          | 93  |
| 16-Bit Byte Select Mode .....                      | 99  |
| 16-Bit Byte Write Mode .....                       | 97  |
| 16-Bit Data Width Modes .....                      | 96  |
| 16-Bit Mode Timing .....                           | 100 |
| 16-Bit Word Write Mode .....                       | 98  |
| 21-Bit Addressing .....                            | 95  |
| 8-Bit Data Width Mode .....                        | 101 |
| 8-Bit Mode Timing .....                            | 102 |
| Address and Data Line Usage (table) .....          | 95  |
| Address and Data Width .....                       | 95  |
| Address Shifting .....                             | 95  |
| and Program Memory Modes .....                     | 96  |
| Control .....                                      | 94  |
| I/O Port Functions .....                           | 93  |

# PIC18F85J11 FAMILY

|                                           |          |                                                   |          |
|-------------------------------------------|----------|---------------------------------------------------|----------|
| Operation in Power-Managed Modes .....    | 103      | Clock Synchronization and the CKP Bit .....       | 197      |
| Wait States .....                         | 96       | Effects of a Reset .....                          | 211      |
| Weak Pull-ups on Port Pins .....          | 96       | General Call Address Support .....                | 200      |
| External Oscillator Modes .....           | 33       | I <sup>2</sup> C Clock Rate w/BRG .....           | 203      |
| EC Modes .....                            | 34       | Master Mode .....                                 | 201      |
| HS Modes .....                            | 33       | Baud Rate Generator .....                         | 203      |
| <b>F</b>                                  |          | Operation .....                                   | 202      |
| Fail-Safe Clock Monitor .....             | 271, 282 | Reception .....                                   | 207      |
| Exiting Fail-Safe Operation .....         | 283      | Repeated Start Condition Timing .....             | 206      |
| Interrupts in Power-Managed Modes .....   | 283      | Start Condition Timing .....                      | 205      |
| POR or Wake-up From Sleep .....           | 283      | Transmission .....                                | 207      |
| WDT During Oscillator Failure .....       | 282      | Multi-Master Communication, Bus Collision         |          |
| Fast Register Stack .....                 | 63       | and Arbitration .....                             | 211      |
| Firmware Instructions .....               | 285      | Multi-Master Mode .....                           | 211      |
| Flash Configuration Words .....           | 271      | Operation .....                                   | 187      |
| Mapping .....                             | 271      | Read/Write Bit Information (R/W Bit) .....        | 187, 189 |
| Flash Program Memory .....                | 83       | Registers .....                                   | 182      |
| Associated Registers .....                | 91       | Serial Clock (SCK/SCL) .....                      | 189      |
| Control Registers .....                   | 84       | Slave Mode .....                                  | 187      |
| EECON1 and EECON2 .....                   | 84       | Addressing .....                                  | 187      |
| TABLAT (Table Latch) Register .....       | 86       | Addressing Masking .....                          | 188      |
| TBLPTR (Table Pointer) Register .....     | 86       | Reception .....                                   | 189      |
| Erase Sequence .....                      | 88       | Transmission .....                                | 189      |
| Erasing .....                             | 88       | Sleep Operation .....                             | 211      |
| Operation During Code-Protect .....       | 91       | Stop Condition Timing .....                       | 210      |
| Reading .....                             | 87       | INCF .....                                        | 306      |
| Table Pointer                             |          | INCFSZ .....                                      | 307      |
| Boundaries Based on Operation .....       | 86       | In-Circuit Debugger .....                         | 284      |
| Table Pointer Boundaries .....            | 86       | In-Circuit Serial Programming (ICSP) .....        | 271, 284 |
| Table Reads and Table Writes .....        | 83       | Indexed Literal Offset Addressing                 |          |
| Write Sequence .....                      | 89       | and Standard PIC18 Instructions .....             | 332      |
| Writing .....                             | 89       | Indexed Literal Offset Mode .....                 | 332      |
| Unexpected Termination .....              | 91       | Indirect Addressing .....                         | 77       |
| Write Verify .....                        | 91       | INFSNZ .....                                      | 307      |
| FSCM. See Fail-Safe Clock Monitor.        |          | Initialization Conditions for all Registers ..... | 51–55    |
| <b>G</b>                                  |          | Instruction Cycle .....                           | 64       |
| GOTO .....                                | 306      | Clocking Scheme .....                             | 64       |
| <b>H</b>                                  |          | Flow/Pipelining .....                             | 64       |
| Hardware Multiplier .....                 | 105      | Instruction Set .....                             | 285      |
| Introduction .....                        | 105      | ADDLW .....                                       | 291      |
| Operation .....                           | 105      | ADDWF .....                                       | 291      |
| Performance Comparison .....              | 105      | ADDWF (Indexed Literal Offset Mode) .....         | 333      |
| <b>I</b>                                  |          | ADDWFC .....                                      | 292      |
| I/O Ports .....                           | 123      | ANDLW .....                                       | 292      |
| Input Voltage Considerations .....        | 123      | ANDWF .....                                       | 293      |
| Open-Drain Outputs .....                  | 124      | BC .....                                          | 293      |
| Output Pin Drive .....                    | 123      | BCF .....                                         | 294      |
| Pin Capabilities .....                    | 123      | BN .....                                          | 294      |
| Pull-up Configuration .....               | 124      | BNC .....                                         | 295      |
| I <sup>2</sup> C Mode (MSSP) .....        | 182      | BNN .....                                         | 295      |
| Acknowledge Sequence Timing .....         | 210      | BN OV .....                                       | 296      |
| Associated Registers .....                | 216      | BNZ .....                                         | 296      |
| Baud Rate Generator .....                 | 203      | BOV .....                                         | 299      |
| Bus Collision                             |          | BRA .....                                         | 297      |
| During a Repeated Start Condition .....   | 214      | BSF .....                                         | 297      |
| During a Stop Condition .....             | 215      | BSF (Indexed Literal Offset Mode) .....           | 333      |
| Clock Arbitration .....                   | 204      | BTFSC .....                                       | 298      |
| Clock Stretching .....                    | 196      | BTFSS .....                                       | 298      |
| 10-Bit Slave Receive Mode (SEN = 1) ..... | 196      | BTG .....                                         | 299      |
| 10-Bit Slave Transmit Mode .....          | 196      | BZ .....                                          | 300      |
| 7-Bit Slave Receive Mode (SEN = 1) .....  | 196      | CALL .....                                        | 300      |
| 7-Bit Slave Transmit Mode .....           | 196      | CLRF .....                                        | 301      |
|                                           |          | CLRWDT .....                                      | 301      |
|                                           |          | COMF .....                                        | 302      |
|                                           |          | CPFSEQ .....                                      | 302      |

# PIC18F85J11 FAMILY

|                                                             |     |                                                         |          |
|-------------------------------------------------------------|-----|---------------------------------------------------------|----------|
| CPFSGT .....                                                | 303 | Interrupt Sources .....                                 | 271      |
| CPFSLT .....                                                | 303 | A/D Conversion Complete .....                           | 255      |
| DAW .....                                                   | 304 | Capture Complete (CCP) .....                            | 166      |
| DCFSNZ .....                                                | 305 | Compare Complete (CCP) .....                            | 167      |
| DECf .....                                                  | 304 | Interrupt-on-Change (RB7:RB4) .....                     | 126      |
| DECFSZ .....                                                | 305 | TMR1 Overflow .....                                     | 151      |
| General Format .....                                        | 287 | TMR2 to PR2 Match (PWM) .....                           | 169      |
| GOTO .....                                                  | 306 | TMR3 Overflow .....                                     | 159      |
| INCF .....                                                  | 306 | Interrupts .....                                        | 107      |
| INCFSZ .....                                                | 307 | During, Context Saving .....                            | 122      |
| INFSNZ .....                                                | 307 | INTx Pin .....                                          | 122      |
| IORLW .....                                                 | 308 | PORTB, Interrupt-on-Change .....                        | 122      |
| IORWF .....                                                 | 308 | TMR0 .....                                              | 122      |
| LFSR .....                                                  | 309 | Interrupts, Flag Bits .....                             |          |
| MOVf .....                                                  | 309 | Interrupt-on-Change (RB7:RB4)                           |          |
| MOVFF .....                                                 | 310 | Flag (RBIF Bit) .....                                   | 126      |
| MOVLB .....                                                 | 310 | INTOSC, INTRC. <i>See</i> Internal Oscillator Block.    |          |
| MOVLW .....                                                 | 311 | IORLW .....                                             | 308      |
| MOVWF .....                                                 | 311 | IORWF .....                                             | 308      |
| MULLW .....                                                 | 312 | IPR Registers .....                                     | 118      |
| MULWF .....                                                 | 312 | <b>L</b>                                                |          |
| NEGF .....                                                  | 313 | LFSR .....                                              | 309      |
| NOP .....                                                   | 313 | <b>M</b>                                                |          |
| Opcode Field Descriptions .....                             | 286 | Master Clear ( $\overline{\text{MCLR}}$ ) .....         | 47       |
| PIC18F85J11 Family (table) .....                            | 288 | Master Synchronous Serial Port (MSSP). <i>See</i> MSSP. |          |
| POP .....                                                   | 314 | Memory Organization .....                               | 57       |
| PUSH .....                                                  | 314 | Data Memory .....                                       | 66       |
| RCALL .....                                                 | 315 | Program Memory .....                                    | 57       |
| RESET .....                                                 | 315 | Memory Programming Requirements .....                   | 352      |
| RETFIE .....                                                | 316 | Microchip Internet Web Site .....                       | 391      |
| RETLW .....                                                 | 316 | Microcontroller Mode .....                              | 96       |
| RETURN .....                                                | 317 | Migration Between High-End Device Families .....        | 379      |
| RLCF .....                                                  | 317 | Oscillator Differences .....                            | 380      |
| RLNCF .....                                                 | 318 | Peripherals .....                                       | 380      |
| RRCF .....                                                  | 318 | Pin Differences .....                                   | 380      |
| RRNCF .....                                                 | 319 | Power Requirement Differences .....                     | 380      |
| SETF .....                                                  | 319 | MOVf .....                                              | 309      |
| SETF (Indexed Literal Offset Mode) .....                    | 333 | MOVFF .....                                             | 310      |
| SLEEP .....                                                 | 320 | MOVLB .....                                             | 310      |
| Standard Instructions .....                                 | 285 | MOVLW .....                                             | 311      |
| SUBFWB .....                                                | 320 | MOVsf .....                                             | 329      |
| SUBLW .....                                                 | 321 | MOVSS .....                                             | 330      |
| SUBWF .....                                                 | 321 | MOVWF .....                                             | 311      |
| SUBWFB .....                                                | 322 | MPLAB ASM30 Assembler, Linker, Librarian .....          | 336      |
| SWAPF .....                                                 | 322 | MPLAB ICD 2 In-Circuit Debugger .....                   | 337      |
| TBLRD .....                                                 | 323 | MPLAB ICE 2000 High-Performance                         |          |
| TBLWT .....                                                 | 324 | Universal In-Circuit Emulator .....                     | 337      |
| TSTFSZ .....                                                | 325 | MPLAB Integrated Development                            |          |
| XORLW .....                                                 | 325 | Environment Software .....                              | 335      |
| XORWF .....                                                 | 326 | MPLAB PM3 Device Programmer .....                       | 337      |
| INTCON Register                                             |     | MPLAB REAL ICE In-Circuit Emulator System .....         | 337      |
| RBIF Bit .....                                              | 126 | MPLINK Object Linker/MPLIB Object Librarian .....       | 336      |
| INTCON Registers .....                                      | 109 | MSSP                                                    |          |
| Inter-Integrated Circuit. <i>See</i> I <sup>2</sup> C Mode. |     | ACK Pulse .....                                         | 187, 189 |
| Internal Oscillator Block .....                             | 35  | Control Registers (general) .....                       | 173      |
| Adjustment .....                                            | 35  | Module Overview .....                                   | 173      |
| INTOSC Frequency Drift .....                                | 35  | SPI Master/Slave Connection .....                       | 177      |
| INTOSC Output Frequency .....                               | 35  | SSPBUF Register .....                                   | 178      |
| OSC1, OSC2 Pin Configuration .....                          | 35  | SSPSR Register .....                                    | 178      |
| Internal RC Oscillator                                      |     | MULLW .....                                             | 312      |
| Use with WDT .....                                          | 278 | MULWF .....                                             | 312      |
| Internal Voltage Regulator Specifications .....             | 353 |                                                         |          |
| Internet Address .....                                      | 391 |                                                         |          |

# PIC18F85J11 FAMILY

## N

|            |     |
|------------|-----|
| NEGF ..... | 313 |
| NOP .....  | 313 |

## O

|                                       |          |
|---------------------------------------|----------|
| Oscillator Configuration .....        | 29       |
| EC .....                              | 29       |
| ECPLL .....                           | 29       |
| HS .....                              | 29       |
| HSPLL .....                           | 29       |
| Internal Oscillator Block .....       | 35       |
| INTOSC .....                          | 29       |
| INTRC .....                           | 29       |
| Oscillator Selection .....            | 271      |
| Oscillator Start-up Timer (OST) ..... | 36       |
| Oscillator Switching .....            | 31       |
| Oscillator Transitions .....          | 32       |
| Oscillator, Timer1 .....              | 151, 161 |
| Oscillator, Timer3 .....              | 159      |

## P

|                                            |        |
|--------------------------------------------|--------|
| Packaging .....                            | 375    |
| Details .....                              | 376    |
| Marking .....                              | 375    |
| Parallel Slave Port (PSP) .....            | 144    |
| Associated Registers .....                 | 146    |
| RE0/RD Pin .....                           | 144    |
| RE1/WR Pin .....                           | 144    |
| RE2/CS Pin .....                           | 144    |
| PICSTART Plus Development Programmer ..... | 338    |
| PIE Registers .....                        | 115    |
| Pin Functions .....                        |        |
| AVDD .....                                 | 18, 27 |
| AVss .....                                 | 18, 27 |
| ENVREG .....                               | 18, 27 |
| MCLR .....                                 | 12, 19 |
| RA0/AN0 .....                              | 12, 19 |
| RA1/AN1 .....                              | 12, 19 |
| RA2/AN2/VREF- .....                        | 12, 19 |
| RA3/AN3/VREF+ .....                        | 12, 19 |
| RA4/T0CKI .....                            | 12, 19 |
| RA5/AN4 .....                              | 12, 19 |
| RA6/OSC2/CLKO .....                        | 12, 19 |
| RA7/OSC1/CLKI .....                        | 12, 19 |
| RB0/INT0 .....                             | 13, 20 |
| RB1/INT1 .....                             | 13, 20 |
| RB2/INT2 .....                             | 13, 20 |
| RB3/INT3 .....                             | 13     |
| RB3/INT3/CCP2 .....                        | 20     |
| RB4/KBI0 .....                             | 13, 20 |
| RB5/KBI1 .....                             | 13, 20 |
| RB6/KBI2/PGC .....                         | 13, 20 |
| RB7/KBI3/PGD .....                         | 13, 20 |
| RC0/T1OSO/T13CKI .....                     | 14, 21 |
| RC1/T1OSI/CCP2 .....                       | 14, 21 |
| RC2/CCP1 .....                             | 14, 21 |
| RC3/SCK/SCL .....                          | 14, 21 |
| RC4/SDI/SDA .....                          | 14, 21 |
| RC5/SDO .....                              | 14, 21 |
| RC6/TX1/CK1 .....                          | 14, 21 |
| RC7/RX1/DT1 .....                          | 14, 21 |
| RD0/AD0/PSP0 .....                         | 22     |
| RD0/PSP0 .....                             | 15     |
| RD1/AD1/PSP1 .....                         | 22     |
| RD1/PSP1 .....                             | 15     |

|                               |        |
|-------------------------------|--------|
| RD2/AD2/PSP2 .....            | 22     |
| RD2/PSP2 .....                | 15     |
| RD3/AD3/PSP3 .....            | 22     |
| RD3/PSP3 .....                | 15     |
| RD4/AD4/PSP4 .....            | 22     |
| RD4/PSP4 .....                | 15     |
| RD5/AD5/PSP5 .....            | 22     |
| RD5/PSP5 .....                | 15     |
| RD6/AD6/PSP6 .....            | 22     |
| RD6/PSP6 .....                | 15     |
| RD7/AD7/PSP7 .....            | 22     |
| RD7/PSP7 .....                | 15     |
| RE0/RD .....                  | 16     |
| RE0/RD/AD8 .....              | 23     |
| RE1/WR .....                  | 16     |
| RE1/WR/AD9 .....              | 23     |
| RE2/AD10/CS .....             | 23     |
| RE2/CS .....                  | 16     |
| RE3 .....                     | 16     |
| RE3/AD11 .....                | 23     |
| RE4 .....                     | 16     |
| RE4/AD12 .....                | 23     |
| RE5 .....                     | 16     |
| RE5/AD13 .....                | 23     |
| RE6 .....                     | 16     |
| RE6/AD14 .....                | 23     |
| RE7/AD15/CCP2 .....           | 23     |
| RE7/CCP2 .....                | 16     |
| RF1/AN6/C2OUT .....           | 17, 24 |
| RF2/AN7/C1OUT .....           | 17, 24 |
| RF3/AN8 .....                 | 17, 24 |
| RF4/AN9 .....                 | 17, 24 |
| RF5/AN10/CVREF .....          | 17, 24 |
| RF6/AN11 .....                | 17, 24 |
| RF7/AN5/SS .....              | 17, 24 |
| RG0 .....                     | 18, 25 |
| RG1/TX2/CK2 .....             | 18, 25 |
| RG2/RX2/DT2 .....             | 18, 25 |
| RG3 .....                     | 18, 25 |
| RG4 .....                     | 18, 25 |
| RH0/A16 .....                 | 26     |
| RH1/A17 .....                 | 26     |
| RH2/A18 .....                 | 26     |
| RH3/A19 .....                 | 26     |
| RH4 .....                     | 26     |
| RH5 .....                     | 26     |
| RH6 .....                     | 26     |
| RH7 .....                     | 26     |
| RJ0/ALE .....                 | 27     |
| RJ1/OE .....                  | 27     |
| RJ2/WRL .....                 | 27     |
| RJ3/WRH .....                 | 27     |
| RJ4/BA0 .....                 | 27     |
| RJ5/CE .....                  | 27     |
| RJ6/LB .....                  | 27     |
| RJ7/UB .....                  | 27     |
| VDD .....                     | 18, 27 |
| VDDCORE/VCAP .....            | 18, 27 |
| Vss .....                     | 18, 27 |
| Pinout I/O Descriptions ..... |        |
| PIC18F6XJ11 .....             | 12     |
| PIC18F8XJ11 .....             | 19     |
| PIR Registers .....           | 112    |

# PIC18F85J11 FAMILY

|                                               |     |                                                |        |
|-----------------------------------------------|-----|------------------------------------------------|--------|
| PLL .....                                     | 34  | Idle Modes .....                               | 41     |
| ECPLL Oscillator Mode .....                   | 34  | PRI_IDLE .....                                 | 42     |
| HSPLL Oscillator Mode .....                   | 34  | RC_IDLE .....                                  | 43     |
| POP .....                                     | 314 | SEC_IDLE .....                                 | 42     |
| POR. See Power-on Reset.                      |     | Multiple Sleep Commands .....                  | 38     |
| PORTA                                         |     | Run Modes .....                                | 38     |
| Associated Registers .....                    | 125 | PRI_RUN .....                                  | 38     |
| LATA Register .....                           | 124 | RC_RUN .....                                   | 40     |
| PORTA Register .....                          | 124 | SEC_RUN .....                                  | 38     |
| TRISA Register .....                          | 124 | Selecting .....                                | 37     |
| PORTB                                         |     | Sleep Mode .....                               | 41     |
| Associated Registers .....                    | 127 | Summary (table) .....                          | 37     |
| LATB Register .....                           | 126 | Power-on Reset (POR) .....                     | 47     |
| PORTB Register .....                          | 126 | Power-up Delays .....                          | 36     |
| RB7:RB4 Interrupt-on-Change                   |     | Power-up Timer (PWRT) .....                    | 36, 48 |
| Flag (RBIF Bit) .....                         | 126 | Time-out Sequence .....                        | 48     |
| TRISB Register .....                          | 126 | Prescaler, Capture .....                       | 166    |
| PORTC                                         |     | Prescaler, Timer0 .....                        | 149    |
| Associated Registers .....                    | 130 | Prescaler, Timer2 .....                        | 170    |
| LATC Register .....                           | 128 | PRI_IDLE Mode .....                            | 42     |
| PORTC Register .....                          | 128 | PRI_RUN Mode .....                             | 38     |
| RC3/SCK/SCL Pin .....                         | 189 | Program Counter .....                          | 61     |
| TRISC Register .....                          | 128 | PCL, PCH and PCU Registers .....               | 61     |
| PORTD .....                                   | 144 | PCLATH and PCLATU Registers .....              | 61     |
| Associated Registers .....                    | 133 | Program Memory                                 |        |
| LATD Register .....                           | 131 | Extended Instruction Set .....                 | 78     |
| PORTD Register .....                          | 131 | Flash Configuration Words .....                | 58     |
| TRISD Register .....                          | 131 | Hard Memory Vectors .....                      | 58     |
| PORTE                                         |     | Instructions .....                             | 65     |
| Associated Registers .....                    | 136 | Two-Word .....                                 | 65     |
| LATE Register .....                           | 134 | Interrupt Vector .....                         | 58     |
| PORTE Register .....                          | 134 | Look-up Tables .....                           | 63     |
| RE0/RD Pin .....                              | 144 | Memory Maps .....                              | 57     |
| RE1/WR Pin .....                              | 144 | Hard Vectors and                               |        |
| RE2/CS Pin .....                              | 144 | Configuration Words .....                      | 58     |
| TRISE Register .....                          | 134 | Modes .....                                    | 59     |
| PORTF                                         |     | Extended Microcontroller .....                 | 59     |
| Associated Registers .....                    | 138 | Extended Microcontroller                       |        |
| LATF Register .....                           | 137 | (Address Shifting) .....                       | 60     |
| PORTF Register .....                          | 137 | Memory Access (table) .....                    | 60     |
| TRISF Register .....                          | 137 | Microcontroller .....                          | 59     |
| PORTG                                         |     | Reset Vector .....                             | 58     |
| Associated Registers .....                    | 140 | Program Memory Modes                           |        |
| LATG Register .....                           | 139 | Operation of the External Memory Bus .....     | 96     |
| PORTG Register .....                          | 139 | Program Verification and Code Protection ..... | 284    |
| TRISG Register .....                          | 139 | Programming, Device Instructions .....         | 285    |
| PORTH                                         |     | PSP. See Parallel Slave Port.                  |        |
| Associated Registers .....                    | 141 | PSPMODE Bit (PSPCON Register) .....            | 144    |
| LATH Register .....                           | 141 | Pulse-Width Modulation. See PWM (CCP Module).  |        |
| PORTH Register .....                          | 141 | PUSH .....                                     | 314    |
| TRISH Register .....                          | 141 | PUSH and POP Instructions .....                | 62     |
| PORTJ                                         |     | PUSHL .....                                    | 330    |
| Associated Registers .....                    | 143 | PWM (CCP Module)                               |        |
| LATJ Register .....                           | 142 | Associated Registers .....                     | 171    |
| PORTJ Register .....                          | 142 | Duty Cycle .....                               | 170    |
| TRISJ Register .....                          | 142 | Example Frequencies/Resolutions .....          | 170    |
| Power-Managed Modes .....                     | 37  | Period .....                                   | 169    |
| and SPI Operation .....                       | 181 | Setup for PWM Operation .....                  | 171    |
| Clock Sources .....                           | 37  | TMR2 to PR2 Match .....                        | 169    |
| Clock Transitions and Status Indicators ..... | 38  |                                                |        |
| Entering .....                                | 37  | <b>Q</b>                                       |        |
| Exiting Idle and Sleep Modes .....            | 43  | Q Clock .....                                  | 170    |
| By Interrupt .....                            | 43  |                                                |        |
| By Reset .....                                | 43  |                                                |        |
| By WDT Time-out .....                         | 43  |                                                |        |
| Without an Oscillator Start-up Delay .....    | 43  |                                                |        |

# PIC18F85J11 FAMILY

## R

RAM. *See* Data Memory.

|                    |     |
|--------------------|-----|
| RC_IDLE Mode ..... | 43  |
| RC_RUN Mode .....  | 40  |
| RCALL .....        | 315 |

### RCON Register

|                                        |       |
|----------------------------------------|-------|
| Bit Status During Initialization ..... | 50    |
| Reader Response .....                  | 392   |
| Register File .....                    | 69    |
| Register File Summary .....            | 71–74 |

### Registers

|                                                                 |         |
|-----------------------------------------------------------------|---------|
| ADCON0 (A/D Control 0) .....                                    | 251     |
| ADCON1 (A/D Control 1) .....                                    | 252     |
| ADCON2 (A/D Control 2) .....                                    | 253     |
| BAUDCON1 (Baud Rate Control 1) .....                            | 220     |
| CCPxCON (CCPx Control) .....                                    | 163     |
| CMCON (Comparator Control) .....                                | 261     |
| CONFIG1H (Configuration 1 High) .....                           | 273     |
| CONFIG1L (Configuration 1 Low) .....                            | 273     |
| CONFIG2H (Configuration 2 High) .....                           | 275     |
| CONFIG2L (Configuration 2 Low) .....                            | 274     |
| CONFIG3H (Configuration 3 High) .....                           | 276     |
| CONFIG3L (Configuration 3 Low) .....                            | 59, 276 |
| CVRCON (Comparator Voltage<br>Reference Control) .....          | 267     |
| DEVID1 (Device ID Register 1) .....                             | 277     |
| DEVID2 (Device ID Register 2) .....                             | 277     |
| EECON1 (EEPROM Control 1) .....                                 | 85      |
| INTCON (Interrupt Control) .....                                | 109     |
| INTCON2 (Interrupt Control 2) .....                             | 110     |
| INTCON3 (Interrupt Control 3) .....                             | 111     |
| IPR1 (Peripheral Interrupt Priority 1) .....                    | 118     |
| IPR2 (Peripheral Interrupt Priority 2) .....                    | 119     |
| IPR3 (Peripheral Interrupt Priority 3) .....                    | 120     |
| MEMCON (External Memory Bus Control) .....                      | 94      |
| OSCCON (Oscillator Control) .....                               | 30      |
| OSCTUNE (Oscillator Tuning) .....                               | 31      |
| PIE1 (Peripheral Interrupt Enable 1) .....                      | 115     |
| PIE2 (Peripheral Interrupt Enable 2) .....                      | 116     |
| PIE3 (Peripheral Interrupt Enable 3) .....                      | 117     |
| PIR1 (Peripheral Interrupt<br>Request (Flag) 1) .....           | 112     |
| PIR2 (Peripheral Interrupt<br>Request (Flag) 2) .....           | 113     |
| PIR3 (Peripheral Interrupt<br>Request (Flag) 3) .....           | 114     |
| PSPCON (Parallel Slave Port Control) .....                      | 145     |
| RCON (Reset Control) .....                                      | 46, 121 |
| RCSTA1 (EUSART Receive<br>Status and Control) .....             | 219     |
| RCSTA2 (AUSART Receive<br>Status and Control) .....             | 239     |
| SSPCON1 (MSSP Control 1, I <sup>2</sup> C Mode) .....           | 184     |
| SSPCON1 (MSSP Control 1, SPI Mode) .....                        | 175     |
| SSPCON2 (MSSP Control 2,<br>I <sup>2</sup> C Master Mode) ..... | 185     |
| SSPCON2 (MSSP Control 2, I <sup>2</sup> C Slave Mode) .....     | 186     |
| SSPSTAT (MSSP Status, I <sup>2</sup> C Mode) .....              | 183     |
| SSPSTAT (MSSP Status, SPI Mode) .....                           | 174     |
| STATUS .....                                                    | 75      |
| STKPTR (Stack Pointer) .....                                    | 62      |
| T0CON (Timer0 Control) .....                                    | 147     |
| T1CON (Timer1 Control) .....                                    | 151     |
| T2CON (Timer2 Control) .....                                    | 157     |
| T3CON (Timer3 Control) .....                                    | 159     |

|                                                      |     |
|------------------------------------------------------|-----|
| TXSTA1 (EUSART Transmit Status<br>and Control) ..... | 218 |
| TXSTA2 (AUSART Transmit Status<br>and Control) ..... | 238 |
| WDTCON (Watchdog Timer Control) .....                | 278 |
| RESET .....                                          | 315 |
| Reset .....                                          | 45  |
| Brown-out Reset (BOR) .....                          | 45  |
| MCLR Reset, During Power-Managed Modes .....         | 45  |
| MCLR Reset, Normal Operation .....                   | 45  |
| Power-on Reset (POR) .....                           | 45  |
| RESET Instruction .....                              | 45  |
| Stack Full Reset .....                               | 45  |
| Stack Underflow Reset .....                          | 45  |
| Watchdog Timer (WDT) Reset .....                     | 45  |
| Resets .....                                         | 271 |
| Brown-out Reset (BOR) .....                          | 271 |
| Oscillator Start-up Timer (OST) .....                | 271 |
| Power-on Reset (POR) .....                           | 271 |
| Power-up Timer (PWRT) .....                          | 271 |
| RETFIE .....                                         | 316 |
| RETLW .....                                          | 316 |
| RETURN .....                                         | 317 |
| Return Address Stack .....                           | 61  |
| Return Stack Pointer (STKPTR) .....                  | 62  |
| RLCF .....                                           | 317 |
| RLNCF .....                                          | 318 |
| RRCF .....                                           | 318 |
| RRNCF .....                                          | 319 |

## S

|                                                         |     |
|---------------------------------------------------------|-----|
| SCK .....                                               | 173 |
| SDI .....                                               | 173 |
| SDO .....                                               | 173 |
| SEC_IDLE Mode .....                                     | 42  |
| SEC_RUN Mode .....                                      | 38  |
| Serial Clock, SCK .....                                 | 173 |
| Serial Data In (SDI) .....                              | 173 |
| Serial Data Out (SDO) .....                             | 173 |
| Serial Peripheral Interface. <i>See</i> SPI Mode.       |     |
| SETF .....                                              | 319 |
| Slave Select (SS) .....                                 | 173 |
| SLEEP .....                                             | 320 |
| Sleep .....                                             |     |
| OSC1 and OSC2 Pin States .....                          | 36  |
| Software Simulator (MPLAB SIM) .....                    | 336 |
| Special Event Trigger. <i>See</i> Compare (CCP Module). |     |
| Special Features of the CPU .....                       | 271 |
| SPI Mode (MSSP) .....                                   |     |
| Associated Registers .....                              | 181 |
| Bus Mode Compatibility .....                            | 181 |
| Effects of a Reset .....                                | 181 |
| Enabling SPI I/O .....                                  | 177 |
| Master Mode .....                                       | 178 |
| Master/Slave Connection .....                           | 177 |
| Operation .....                                         | 176 |
| Operation in Power-Managed Modes .....                  | 181 |
| Serial Clock .....                                      | 173 |
| Serial Data In .....                                    | 173 |
| Serial Data Out .....                                   | 173 |
| Slave Mode .....                                        | 179 |
| Slave Select .....                                      | 173 |
| Slave Select Synchronization .....                      | 179 |
| SPI Clock .....                                         | 178 |
| Typical Connection .....                                | 177 |

# PIC18F85J11 FAMILY

|                             |          |
|-----------------------------|----------|
| SS                          | 173      |
| SSPOV                       | 207      |
| SSPOV Status Flag           | 207      |
| SSPSTAT Register            |          |
| R/W Bit                     | 187, 189 |
| Stack Full/Underflow Resets | 63       |
| STATUS Register             | 75       |
| SUBFSR                      | 331      |
| SUBFWB                      | 320      |
| SUBLW                       | 321      |
| SUBLNK                      | 331      |
| SUBWF                       | 321      |
| SUBWFB                      | 322      |
| SWAPF                       | 322      |

## T

|                                                           |          |
|-----------------------------------------------------------|----------|
| Table Pointer Operations (table)                          | 86       |
| Table Reads/Table Writes                                  | 63       |
| TBLRD                                                     | 323      |
| TBLWT                                                     | 324      |
| Timer0                                                    | 147      |
| Associated Registers                                      | 149      |
| Clock Source Select (T0CS Bit)                            | 148      |
| Interrupt                                                 | 149      |
| Operation                                                 | 148      |
| Prescaler                                                 | 149      |
| Switching Assignment                                      | 149      |
| Prescaler Assignment (PSA Bit)                            | 149      |
| Prescaler Select (T0PS2:T0PS0 Bits)                       | 149      |
| Prescaler. <i>See</i> Prescaler, Timer0.                  |          |
| Reads and Writes in 16-Bit Mode                           | 148      |
| Source Edge Select (T0SE Bit)                             | 148      |
| Timer1                                                    | 151      |
| 16-Bit Read/Write Mode                                    | 153      |
| Associated Registers                                      | 155      |
| Interrupt                                                 | 154      |
| Operation                                                 | 152      |
| Oscillator                                                | 151, 153 |
| Layout Considerations                                     | 154      |
| Oscillator, as Secondary Clock                            | 31       |
| Overflow Interrupt                                        | 151      |
| Resetting, Using the CCPx                                 |          |
| Special Event Trigger                                     | 154      |
| TMR1H Register                                            | 151      |
| TMR1L Register                                            | 151      |
| Use as a Clock Source                                     | 153      |
| Use as a Real-Time Clock                                  | 154      |
| Timer2                                                    | 157      |
| Associated Registers                                      | 158      |
| Interrupt                                                 | 158      |
| Operation                                                 | 157      |
| Output                                                    | 158      |
| PR2 Register                                              | 169      |
| TMR2 to PR2 Match Interrupt                               | 169      |
| Timer3                                                    | 159      |
| 16-Bit Read/Write Mode                                    | 161      |
| Associated Registers                                      | 161      |
| Interrupt                                                 | 161      |
| Operation                                                 | 160      |
| Oscillator                                                | 159, 161 |
| Overflow Interrupt                                        | 159      |
| Special Event Trigger (CCP)                               | 161      |
| TMR3H Register                                            | 159      |
| TMR3L Register                                            | 159      |
| Timing Diagrams                                           |          |
| A/D Conversion                                            | 372      |
| Acknowledge Sequence                                      | 210      |
| Asynchronous Reception                                    | 229, 245 |
| Asynchronous Transmission                                 | 227, 243 |
| Asynchronous Transmission                                 |          |
| (Back-to-Back)                                            | 227, 243 |
| Automatic Baud Rate Calculation                           | 225      |
| Auto-Wake-up Bit (WUE) During                             |          |
| Normal Operation                                          | 230      |
| Auto-Wake-up Bit (WUE) During Sleep                       | 230      |
| Baud Rate Generator with Clock Arbitration                | 204      |
| BRG Overflow Sequence                                     | 225      |
| BRG Reset Due to SDA Arbitration During                   |          |
| Start Condition                                           | 213      |
| Bus Collision During a Repeated Start                     |          |
| Condition (Case 1)                                        | 214      |
| Bus Collision During a Repeated Start                     |          |
| Condition (Case 2)                                        | 214      |
| Bus Collision During a Start                              |          |
| Condition (SCL = 0)                                       | 213      |
| Bus Collision During a Stop                               |          |
| Condition (Case 1)                                        | 215      |
| Bus Collision During a Stop                               |          |
| Condition (Case 2)                                        | 215      |
| Bus Collision During a Start                              |          |
| Condition (SDA Only)                                      | 212      |
| Bus Collision for Transmit and Acknowledge                | 211      |
| Capture/Compare/PWM (CCP1, CCP2)                          | 361      |
| CLKO and I/O                                              | 358      |
| Clock Synchronization                                     | 197      |
| Clock/Instruction Cycle                                   | 64       |
| EUSART/AUSART Synchronous Receive                         |          |
| (Master/Slave)                                            | 370      |
| EUSART/AUSART Synchronous Transmission                    |          |
| (Master/Slave)                                            | 370      |
| Example SPI Master Mode (CKE = 0)                         | 362      |
| Example SPI Master Mode (CKE = 1)                         | 363      |
| Example SPI Slave Mode (CKE = 0)                          | 364      |
| Example SPI Slave Mode (CKE = 1)                          | 365      |
| External Clock (All Modes Except PLL)                     | 356      |
| External Memory Bus for Sleep (Extended                   |          |
| Microcontroller Mode)                                     | 100, 102 |
| External Memory Bus for TBLRD (Extended                   |          |
| Microcontroller Mode)                                     | 100, 102 |
| Fail-Safe Clock Monitor                                   | 283      |
| First Start Bit Timing                                    | 205      |
| I <sup>2</sup> C Bus Data                                 | 367      |
| I <sup>2</sup> C Bus Start/Stop Bits                      | 366      |
| I <sup>2</sup> C Master Mode (7 or 10-Bit Transmission)   | 208      |
| I <sup>2</sup> C Master Mode (7-Bit Reception)            | 209      |
| I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 0)   | 193      |
| I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 0,   |          |
| ADMSK = 01001)                                            | 194      |
| I <sup>2</sup> C Slave Mode (10-Bit Reception, SEN = 1)   | 199      |
| I <sup>2</sup> C Slave Mode (10-Bit Transmission)         | 195      |
| I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 0)    | 190      |
| I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 0,    |          |
| ADMSK = 01011)                                            | 191      |
| I <sup>2</sup> C Slave Mode (7-Bit Reception, SEN = 1)    | 198      |
| I <sup>2</sup> C Slave Mode (7-Bit Transmission)          | 192      |
| I <sup>2</sup> C Slave Mode General Call Address Sequence |          |
| (7 or 10-Bit Address Mode)                                | 200      |

# PIC18F85J11 FAMILY

|                                                                                                 |          |
|-------------------------------------------------------------------------------------------------|----------|
| I <sup>2</sup> C Stop Condition Receive or Transmit Mode .....                                  | 210      |
| MSSP I <sup>2</sup> C Bus Data .....                                                            | 368      |
| MSSP I <sup>2</sup> C Bus Start/Stop Bits .....                                                 | 368      |
| Parallel Slave Port (PSP) Read .....                                                            | 146      |
| Parallel Slave Port (PSP) Write .....                                                           | 145      |
| PWM Output .....                                                                                | 169      |
| Repeated Start Condition .....                                                                  | 206      |
| Reset, Watchdog Timer (WDT), Oscillator Start-up<br>Timer (OST) and Power-up Timer (PWRT) ..... | 359      |
| Send Break Character Sequence .....                                                             | 231      |
| Slave Synchronization .....                                                                     | 179      |
| Slow Rise Time (MCLR Tied to VDD,<br>VDD Rise > TPWRT) .....                                    | 49       |
| SPI Mode (Master Mode) .....                                                                    | 178      |
| SPI Mode (Slave Mode, CKE = 0) .....                                                            | 180      |
| SPI Mode (Slave Mode, CKE = 1) .....                                                            | 180      |
| Synchronous Reception<br>(Master Mode, SREN) .....                                              | 234, 248 |
| Synchronous Transmission .....                                                                  | 232, 246 |
| Synchronous Transmission<br>(Through TXEN) .....                                                | 233, 247 |
| Time-out Sequence on Power-up<br>(MCLR Not Tied to VDD), Case 1 .....                           | 49       |
| Time-out Sequence on Power-up<br>(MCLR Not Tied to VDD), Case 2 .....                           | 49       |
| Time-out Sequence on Power-up<br>(MCLR Tied to VDD, VDD Rise < TPWRT) .....                     | 48       |
| Timer0 and Timer1 External Clock .....                                                          | 360      |
| Transition for Entry to Idle Mode .....                                                         | 42       |
| Transition for Entry to SEC_RUN Mode .....                                                      | 39       |
| Transition for Entry to Sleep Mode .....                                                        | 41       |
| Transition for Two-Speed Start-up<br>(INTRC to HSPLL) .....                                     | 281      |
| Transition for Wake From Idle to Run Mode .....                                                 | 42       |
| Transition for Wake From Sleep (HSPLL) .....                                                    | 41       |
| Transition From RC_RUN Mode to<br>PRI_RUN Mode .....                                            | 40       |
| Transition From SEC_RUN Mode to<br>PRI_RUN Mode (HSPLL) .....                                   | 39       |
| Transition to RC_RUN Mode .....                                                                 | 40       |
| Timing Diagrams and Specifications                                                              |          |
| Capture/Compare/PWM Requirements<br>(CCP1, CCP2) .....                                          | 361      |
| CLKO and I/O Requirements .....                                                                 | 358      |
| EUSART/AUSART Synchronous Receive<br>Requirements .....                                         | 370      |
| EUSART/AUSART Synchronous Transmission<br>Requirements .....                                    | 370      |

|                                                                                                               |          |
|---------------------------------------------------------------------------------------------------------------|----------|
| Example SPI Mode Requirements<br>(Master Mode, CKE = 0) .....                                                 | 362      |
| Example SPI Mode Requirements<br>(Master Mode, CKE = 1) .....                                                 | 363      |
| Example SPI Mode Requirements<br>(Slave Mode, CKE = 0) .....                                                  | 364      |
| Example SPI Slave Mode<br>Requirements (CKE = 1) .....                                                        | 365      |
| External Clock Requirements .....                                                                             | 356      |
| I <sup>2</sup> C Bus Data Requirements (Slave Mode) .....                                                     | 367      |
| I <sup>2</sup> C Bus Start/Stop Bits Requirements<br>(Slave Mode) .....                                       | 366      |
| Internal RC Accuracy .....                                                                                    | 357      |
| MSSP I <sup>2</sup> C Bus Data Requirements .....                                                             | 369      |
| MSSP I <sup>2</sup> C Bus Start/Stop Bits Requirements .....                                                  | 368      |
| PLL Clock .....                                                                                               | 357      |
| Reset, Watchdog Timer, Oscillator Start-up<br>Timer, Power-up Timer and Brown-out<br>Reset Requirements ..... | 359      |
| Timer0 and Timer1 External<br>Clock Requirements .....                                                        | 360      |
| Top-of-Stack Access .....                                                                                     | 61       |
| TSTFSZ .....                                                                                                  | 325      |
| Two-Speed Start-up .....                                                                                      | 271, 281 |
| Two-Word Instructions<br>Example Cases .....                                                                  | 65       |

## V

|                                        |     |
|----------------------------------------|-----|
| VDDCORE/VCAP Pin .....                 | 279 |
| Voltage Reference Specifications ..... | 353 |
| Voltage Regulator (On-Chip) .....      | 279 |
| Brown-out Reset (BOR) .....            | 280 |
| Low-Voltage Detection (LVD) .....      | 279 |
| Operation in Sleep Mode .....          | 280 |
| Power-up Requirements .....            | 280 |

## W

|                                  |                    |
|----------------------------------|--------------------|
| Watchdog Timer (WDT) .....       | 271, 278           |
| Associated Registers .....       | 278                |
| Control Register .....           | 278                |
| Programming Considerations ..... | 278                |
| WCOL .....                       | 205, 206, 207, 210 |
| WCOL Status Flag .....           | 205, 206, 207, 210 |
| WWW Address .....                | 391                |
| WWW, On-Line Support .....       | 5                  |

## X

|             |     |
|-------------|-----|
| XORLW ..... | 325 |
| XORWF ..... | 326 |



## THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at [www.microchip.com](http://www.microchip.com). This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

## CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at [www.microchip.com](http://www.microchip.com), click on Customer Change Notification and follow the registration instructions.

## CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support
- Development Systems Information Line

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

**Technical support is available through the web site at: <http://support.microchip.com>**

# PIC18F85J11 FAMILY

---

## READER RESPONSE

It is our intention to provide you with the best documentation possible to ensure successful use of your Microchip product. If you wish to provide your comments on organization, clarity, subject matter, and ways in which our documentation can better serve you, please FAX your comments to the Technical Publications Manager at (480) 792-4150.

Please list the following information, and use this outline to provide us with your comments about this document.

To: Technical Publications Manager  
RE: Reader Response  
Total Pages Sent \_\_\_\_\_  
From: Name \_\_\_\_\_  
Company \_\_\_\_\_  
Address \_\_\_\_\_  
City / State / ZIP / Country \_\_\_\_\_  
Telephone: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_ FAX: (\_\_\_\_\_) \_\_\_\_\_ - \_\_\_\_\_

Application (optional):

Would you like a reply? \_\_\_Y \_\_\_N

Device: PIC18F85J11 Family

Literature Number: DS39774C

Questions:

1. What are the best features of this document?

---

---

2. How does this document meet your hardware and software development needs?

---

---

3. Do you find the organization of this document easy to follow? If not, why?

---

---

4. What additions to the document do you think would enhance the structure and subject?

---

---

5. What deletions from the document could be made without affecting the overall usefulness?

---

---

6. Is there any incorrect or misleading information (what and where)?

---

---

7. How would you improve this document?

---

---





## WORLDWIDE SALES AND SERVICE

### AMERICAS

#### Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
<http://support.microchip.com>  
Web Address:  
[www.microchip.com](http://www.microchip.com)

#### Atlanta

Duluth, GA  
Tel: 678-957-9614  
Fax: 678-957-1455

#### Boston

Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088

#### Chicago

Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075

#### Dallas

Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924

#### Detroit

Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

#### Kokomo

Kokomo, IN  
Tel: 765-864-8360  
Fax: 765-864-8387

#### Los Angeles

Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608

#### Santa Clara

Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

#### Toronto

Mississauga, Ontario,  
Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

### ASIA/PACIFIC

#### Asia Pacific Office

Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

#### Australia - Sydney

Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

#### China - Beijing

Tel: 86-10-8528-2100  
Fax: 86-10-8528-2104

#### China - Chengdu

Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

#### China - Fuzhou

Tel: 86-591-8750-3506  
Fax: 86-591-8750-3521

#### China - Hong Kong SAR

Tel: 852-2401-1200  
Fax: 852-2401-3431

#### China - Qingdao

Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

#### China - Shanghai

Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

#### China - Shenyang

Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

#### China - Shenzhen

Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

#### China - Shunde

Tel: 86-757-2839-5507  
Fax: 86-757-2839-5571

#### China - Wuhan

Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

#### China - Xian

Tel: 86-29-8833-7250  
Fax: 86-29-8833-7256

### ASIA/PACIFIC

#### India - Bangalore

Tel: 91-80-4182-8400  
Fax: 91-80-4182-8422

#### India - New Delhi

Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

#### India - Pune

Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

#### Japan - Yokohama

Tel: 81-45-471- 6166  
Fax: 81-45-471-6122

#### Korea - Gumi

Tel: 82-54-473-4301  
Fax: 82-54-473-4302

#### Korea - Seoul

Tel: 82-2-554-7200  
Fax: 82-2-558-5932 or  
82-2-558-5934

#### Malaysia - Penang

Tel: 60-4-646-8870  
Fax: 60-4-646-5086

#### Philippines - Manila

Tel: 63-2-634-9065  
Fax: 63-2-634-9069

#### Singapore

Tel: 65-6334-8870  
Fax: 65-6334-8850

#### Taiwan - Hsin Chu

Tel: 886-3-572-9526  
Fax: 886-3-572-6459

#### Taiwan - Kaohsiung

Tel: 886-7-536-4818  
Fax: 886-7-536-4803

#### Taiwan - Taipei

Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

#### Thailand - Bangkok

Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### EUROPE

#### Austria - Wels

Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

#### Denmark - Copenhagen

Tel: 45-4450-2828  
Fax: 45-4485-2829

#### France - Paris

Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

#### Germany - Munich

Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

#### Italy - Milan

Tel: 39-0331-742611  
Fax: 39-0331-466781

#### Netherlands - Drunen

Tel: 31-416-690399  
Fax: 31-416-690340

#### Spain - Madrid

Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

#### UK - Wokingham

Tel: 44-118-921-5869  
Fax: 44-118-921-5820

12/08/06

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Microchip:](#)

[PIC18F63J11-I/PT](#) [PIC18F63J11T-I/PT](#) [PIC18F64J11-I/PT](#) [PIC18F64J11T-I/PT](#) [PIC18F65J11-I/PT](#) [PIC18F65J11T-I/PT](#) [PIC18F83J11T-I/PT](#) [PIC18F84J11-I/PT](#) [PIC18F84J11T-I/PT](#) [PIC18F85J11-I/PT](#) [PIC18F85J11T-I/PT](#)