Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

PICkit™ 3 Starter Kit User's Guide

ISBN: 9781620766972

Table of Contents

Preface

NOTICE TO CUSTOMERS

All documentation becomes dated, and this manual is no exception. Microchip tools and documentation are constantly evolving to meet customer needs, so some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site (www.microchip.com) to obtain the latest documentation available.

Documents are identified with a "DS" number. This number is located on the bottom of each page, in front of the page number. The numbering convention for the DS number is "DSXXXXXA", where "XXXXX" is the document number and "A" is the revision level of the document.

For the most up-to-date information on development tools, see the MPLAB® IDE online help. Select the Help menu, and then Topics to open a list of available online help files.

INTRODUCTION

This chapter contains general information that will be useful to know before using the PICkit™ 3 Starter Kit User's Guide. Items discussed in this chapter include:

- Document Layout
- Conventions Used in this Guide
- Warranty Registration
- Recommended Reading
- The Microchip Web Site
- Development Systems Customer Change Notification Service
- Customer Support
- Document Revision History

DOCUMENT LAYOUT

This document describes how to use the PICkit™ 3 Starter Kit User's Guide as a development tool to emulate and debug firmware on a target board. The manual layout is as follows:

- **Section Chapter 1. "Overview"**
- **Section Chapter 2. "PIC® MCU Architecture"**
- **Section Chapter 3. "Lessons"**
- **Appendix A. "Block Diagram and MPLAB® X Shortcuts"**

CONVENTIONS USED IN THIS GUIDE

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

<table>
<tr><th>Description</th><th>Represents</th><th>Examples</th></tr>
<tr><td colspan="3">Arial font:</td></tr>
<tr><td rowspan="2">Italic characters</td><td>Referenced books</td><td>MPLAB® IDE User’s Guide</td></tr>
<tr><td>Emphasized text</td><td>...is the only compiler...</td></tr>
<tr><td rowspan="3">Initial caps</td><td>A window</td><td>the Output window</td></tr>
<tr><td>A dialog</td><td>the Settings dialog</td></tr>
<tr><td>A menu selection</td><td>select Enable Programmer</td></tr>
<tr><td>Quotes</td><td>A field name in a window or dialog</td><td>“Save project before build”</td></tr>
<tr><td>Underlined, italic text with right angle bracket</td><td>A menu path</td><td><u>File>Save</u></td></tr>
<tr><td rowspan="2">Bold characters</td><td>A dialog button</td><td>Click OK</td></tr>
<tr><td>A tab</td><td>Click the Power tab</td></tr>
<tr><td>N‘Rnnnn</td><td>A number in verilog format, where N is the total number of digits, R is the radix and n is a digit.</td><td>4‘b0010, 2‘hF1</td></tr>
<tr><td>Text in angle brackets < ></td><td>A key on the keyboard</td><td>Press <Enter>, <F1></td></tr>
<tr><td colspan="3">Courier New font:</td></tr>
<tr><td rowspan="7">Plain Courier New</td><td>Sample source code</td><td><code>#define START</code></td></tr>
<tr><td>Filenames</td><td><code>autoexec.bat</code></td></tr>
<tr><td>File paths</td><td><code>c:\mcc18\h</code></td></tr>
<tr><td>Keywords</td><td><code>_asm, _endasm, static</code></td></tr>
<tr><td>Command-line options</td><td><code>-Opa+, -Opa-</code></td></tr>
<tr><td>Bit values</td><td><code>0, 1</code></td></tr>
<tr><td>Constants</td><td><code>0xFF, ‘A’</code></td></tr>
<tr><td>Italic Courier New</td><td>A variable argument</td><td><code>file.o</code>, where <code>file</code> can be any valid filename</td></tr>
<tr><td>Square brackets [ ]</td><td>Optional arguments</td><td><code>mcc18 [options] file [options]</code></td></tr>
<tr><td>Curly brackets and pipe character: { | }</td><td>Choice of mutually exclusive arguments; an OR selection</td><td><code>errorlevel {0|1}</code></td></tr>
<tr><td rowspan="2">Ellipses...</td><td>Replaces repeated text</td><td><code>var_name [, var_name...]</code></td></tr>
<tr><td>Represents code supplied by user</td><td><code>void main (void)
{ ...
}</code></td></tr>
</table>

WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in the Warranty Registration Card entitles users to receive new product updates. Interim software releases are available at the Microchip web site.

RECOMMENDED READING

This user's guide describes how to use the PICkit™ 3 Starter Kit User's Guide. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Readme for PICkit™ 3 Starter Kit User's Guide

For the latest information on using PICkit™ 3 Starter Kit User's Guide, read the "`Readme for PICkit™ 3 Starter Kit Board User's Guide.txt`" file (an ASCII text file) in the Readmes subdirectory of the MPLAB IDE installation directory. The Readme file contains update information and known issues that may not be included in this user's guide.

PIC16(L)F1825/29 Data Sheet (DS41440)

This data sheet summarizes the features of the PIC16F1829.

PIC18(L)F1XK22 Data Sheet (DS41365)

This data sheet summarizes the features of the PIC18F14K22.

Readme Files

For the latest information on using other tools, read the tool-specific Readme files in the Readmes subdirectory of the MPLAB IDE installation directory. The Readme files contain update information and known issues that may not be included in this user's guide.

THE MICROCHIP WEB SITE

Microchip provides online support via our web site at 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 (FAQs), 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

DEVELOPMENT SYSTEMS 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, click on Customer Change Notification and follow the registration instructions.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the HI-TECH C® C16, MPLAB C18 and MPLAB C30 C compilers; MPASM™ and MPLAB ASM30 assemblers; MPLINK™ and MPLAB LINK30 object linkers; and MPLIB™ and MPLAB LIB30 object librarians.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 2, MPLAB ICD 3, PICkit™ 3.
- **MPLAB® IDE** – The latest information on Microchip MPLAB IDE, the Windows® Integrated Development Environment for development systems tools. This list is focused on the MPLAB IDE, MPLAB SIM simulator, MPLAB IDE Project Manager and general editing and debugging features.
- **Programmers** – The latest information on Microchip programmers. These include the MPLAB PM3 device programmers and PICkit™ 3 development programmers.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

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

DOCUMENT REVISION HISTORY

Revision A (October 2012)

- Initial Release of this Document.

Revision B (November 2012)

- Revised Sections 3.5.3, 3.5.4.1.1, 3.11.3.2, Table 3-15.

NOTES:

Chapter 1. Overview

1.1 INTRODUCTION

This chapter introduces the hardware that is included in the kit, as well as a quick start to downloading and installing the accompanying software.

1.2 HIGHLIGHTS

This chapter discusses:

- What's New
- Included Items
- The Low Pin Count (LPC) Board Hardware
- Software Overview
- Running the Demonstrations

1.3 WHAT'S NEW

This kit is an update to the PICkit™ 2 Starter Kit. Modifications to the previous LPC board (DM164120-1) were made so that the full functionality of the code can be debugged without the need of a debug header. The software has also been rewritten to accommodate new technologies. The following is a list of new features:

1. Software is in both the 'C' and assembler language
2. Extension of the number of lessons and modules covered
3. MPLAB® X support as well as the older MPLAB® 8
4. New PIC16 enhanced mid-range and PIC18 routines
5. Uses the universal XC8 compiler

The following is a list of hardware changes to the LPC board:

1. Potentiometer connected to RA4 (formerly to RA0)
2. Switch connected to RA2 (formerly to RA3)

This new LPC board is still backwards compatible. Bridging the old pins to the new pins will restore functionality.

1.4 INCLUDED ITEMS

1. 1x PICkit 3 Programmer
2. 1x Micro USB cable
3. 1x LPC Board (Part Number : DM164130-9)
4. 1x PIC16F1829-I/P
5. 1x PIC18F14K22 -I/P

The 13 lessons can be downloaded from the web.

The PIC16F1829 is a new enhanced mid-range device, which supports more features than the older mid-range PIC16 parts.

The software associated with the kit supports the PIC16F1829 and PIC18F14K22. The software is intended to run on these two devices, although the software can be easily ported to other devices.

1.5 THE LOW PIN COUNT BOARD

Support for 18-pin devices requires some board modifications. 14- and 20-pin PIC devices will have full access to all of the human interface devices. If an 8-pin part is used, then the LEDs will have to be bridged to the necessary pins on the PIC MCU. The switch and potentiometer are already connected to pins that are supported by an 8-pin device. The board provides holes next to the LEDs that can be easily soldered to in order to create any desired hardware changes.

The board is programmable by an In-Circuit Serial Programmer™ (ICSP™), such as a PICkit™ programmer. The board should be supplied with 5V. Figure 1-1 shows the LPC Demo Board.

FIGURE 1-1: DEMO BOARD HARDWARE LAYOUT

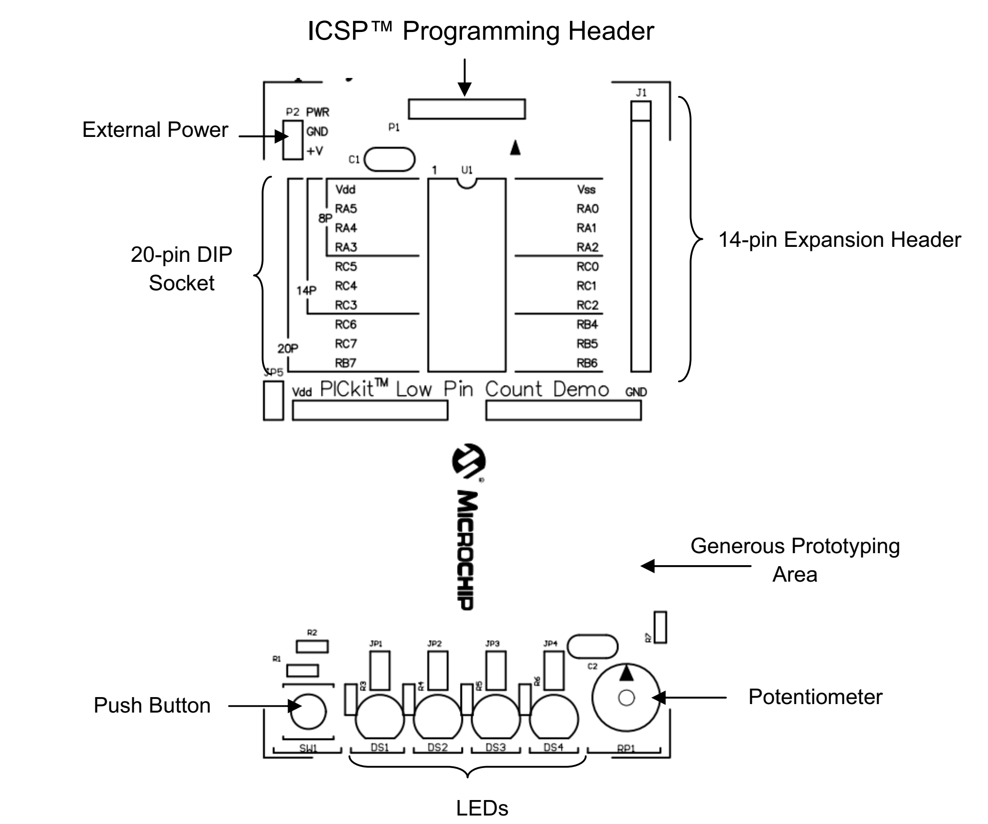


Table 1-1 lists the components that are connected to the two PIC devices that come with the board.

TABLE 1-1: PIN ASSIGNMENTS

Device	LEDs <DS4:DS1>	Switch – SW1	Potentiometer – RP1
PIC16F1829	<RC4:RC0>	RA2	RA4
PIC18F14K22	<RC4:RC0>	RA2	RA4

1.6 SOFTWARE OVERVIEW

This guide will assume that the reader has a basic knowledge of electronics. The reader does not need to have any programming experience with a PIC MCU before reading, although a basic knowledge of programming and what the difference between a bit and byte will help.

The software is written in both assembly and ‘C’ in the MPLAB X and MPLAB 8 integrated design environment (IDE). The assembly version is more complex and requires more lines of code, however it is closely tied to the PIC device’s hardware and the reader will gain a much better understanding by doing these lessons in parallel with the ‘C’ routines. The ‘C’ programming language is a higher level language assembly, hence it provides the reader with an easier to read flow of the program. Each lesson has both versions and are functionally equivalent.

It is recommended that the lessons be followed sequentially, as presented, since most of the lessons build up on one another. Each new program will introduce a new peripheral or concept. This guide is not intended to be read without following along in the code.

The PIC18 and enhanced PIC16 programs will be presented side-by-side and their differences and similarities explained.

1.7 RUNNING THE DEMONSTRATIONS

The board comes preprogrammed with a lesson. To use this program, either apply 5V to the power header (P2), or connect a programmer to the programmer header (P1) and apply 5V through the programmer in the IDE. The demo program will blink the four red LEDs in succession. Press the push button (SW1), and the sequence will reverse. Rotate the potentiometer (RP1), and the light sequence will blink at a different rate. This demo program is developed through the first seven lessons in this guide.

NOTES:

Chapter 2. PIC® MCU Architecture

2.1 INTRODUCTION

This chapter describes the architecture of the enhanced mid-range PIC16F1829 (DS41440), as well as the PIC18 (DS41365).

2.2 CORE BASICS

Enhanced PIC16 and PIC18 devices use a modified Harvard architecture, meaning the code memory and data memory are independent. This allows faster execution because code instructions and data can be accessed simultaneously. The subsequent instruction is fetched while decoding and executing the current instruction. In Figure 2-1 and Figure 2-3, the reader should notice the separate lines for data bus and program bus. This guide will cover nearly all of the registers and modules as seen in the following figures. The following block diagrams should be referenced while each lesson is being performed in order to understand the interactions.

FIGURE 2-1: SIMPLIFIED ENHANCED MID-RANGE PIC® MCU BLOCK DIAGRAM

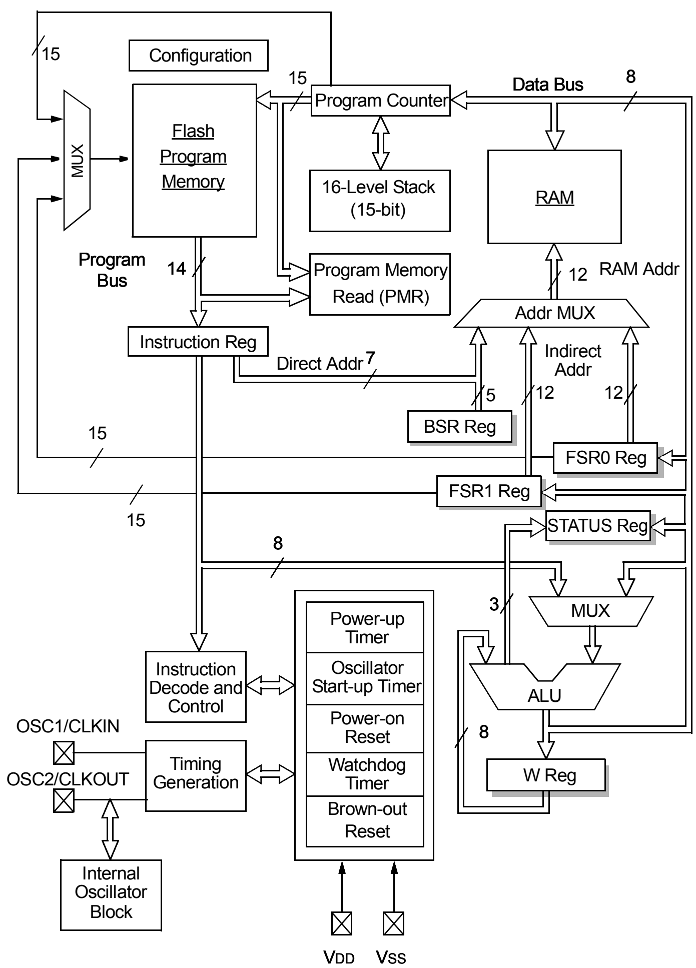
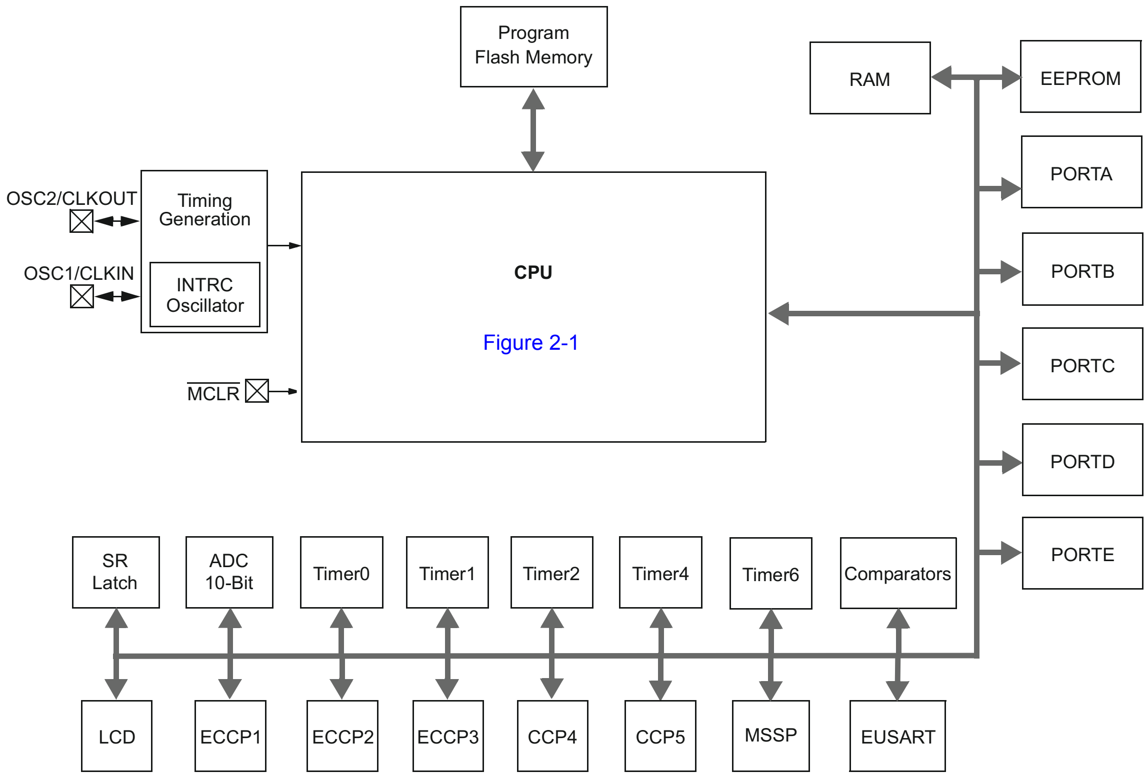
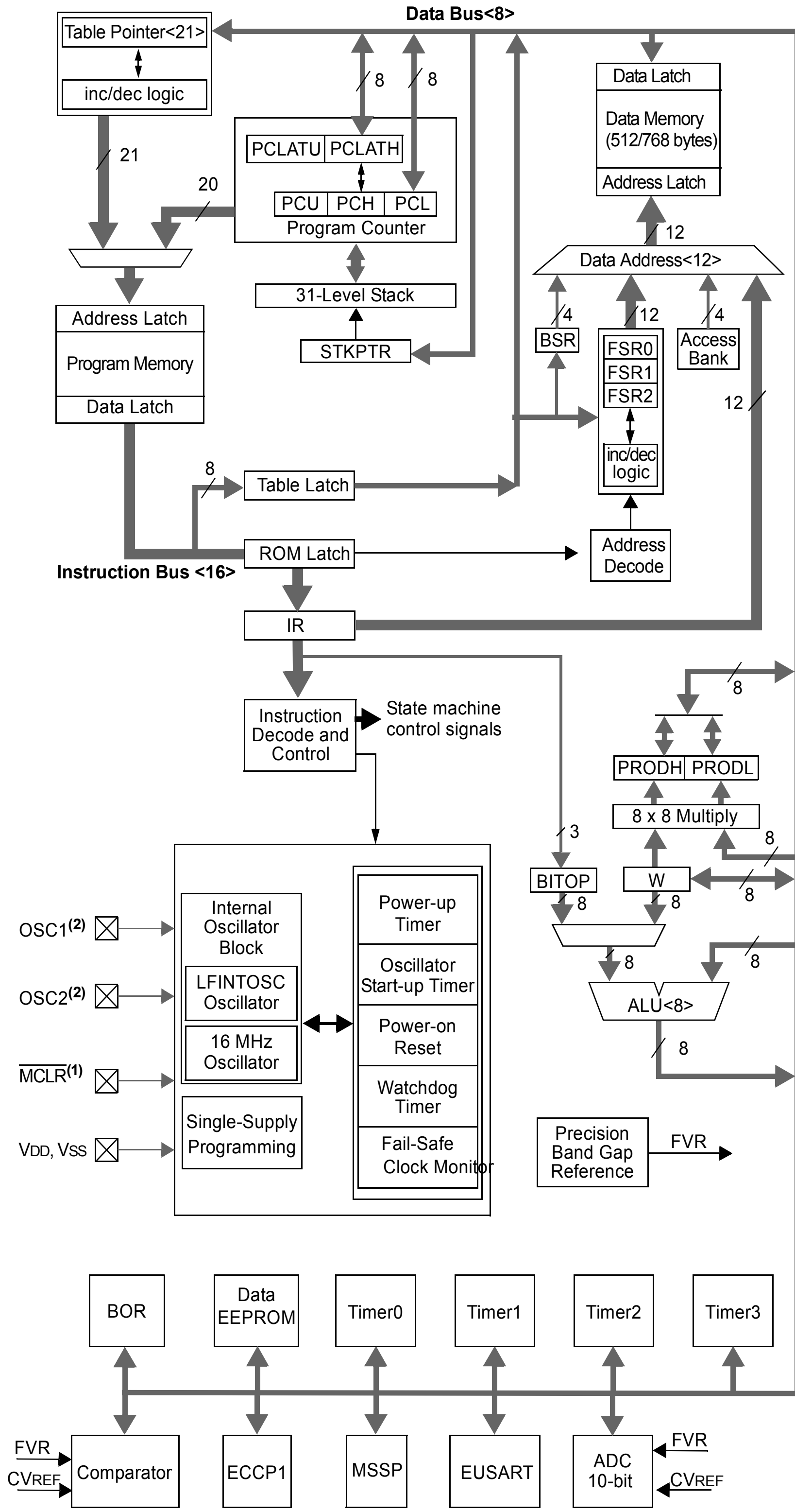


FIGURE 2-2: SIMPLIFIED ENHANCED MID-RANGE PIC® MCU DATA BLOCK DIAGRAM



Note 1: See applicable chapters for more information on peripherals.

FIGURE 2-3: SIMPLIFIED PIC18 BLOCK DIAGRAM



Note 1: RA3 is only available when $\overline{MCLR}$ functionality is disabled.

2: OSC1/CLKIN and OSC2/CLKOUT are only available in select oscillator modes and when these pins are not being used as digital I/O.

2.3 DATA/PROGRAM BUS

The data bus is connected to the outside world via port pins, as well as all of the peripheral registers (timers, ADC, PWM). The program bus connects to the Flash memory where the program is stored. This is where assembled code is programmed to.

2.4 ACCUMULATOR

There is only one accumulator – the working register (WREG). The accumulator handles all data bus related tasks, such as mathematical operations. The ALU only deals with 8-bit sized data – hence the categorical names of 8/16/32-bit micros.

2.5 INSTRUCTIONS

Instructions tell what the PIC device should do, whether it is shifting a few bits or jumping to a new line in code. They form the very essence of each program in program memory. All enhanced mid-range PIC devices have only 49 instructions. The PIC18 has 75 available instructions. Since there are very few instructions needed to learn, the PIC device can be referred to as a "reduced instruction set computing", or RISC, processor.

Each instruction will be explained in detail as they are introduced in each lesson. For now, the basis of what makes up each instruction will be explained.

One instruction cycle consists of four clock cycles. This means that if the PIC MCU is running at 4 MHz, each instruction will take one microsecond, as seen in Equation 2-1.

EQUATION 2-1: INSTRUCTION TIME

$$T(clock\ cycle) = \frac{1}{FOSC}$$

$$4 * T = \frac{4}{FOSC} = \frac{4}{4\ MHz} = 1\ \mu s$$

All instructions are executed in a single instruction cycle, unless a conditional test is true, or the program counter (PC) is changed. In these cases, the execution takes two instruction cycles, with the additional instruction cycle executed as a `NOP` (do nothing), see Example 2-1.

EXAMPLE 2-1:

```
BTFSS       PORTA, RA0
```

This takes two instruction cycles only if pin RA0 is set (active-high), since the skip operation affects the PC.

The PIC18 has a larger word size than the enhanced PIC16 architecture. The PIC18 has a 16-bit wide word containing the operation code (opcode) and all required operands. The enhanced PIC16 has a 14-bit wide word. An opcode is interpreted by the processor and is unique to each instruction.

The opcodes are broken into four formats:

1. Byte oriented
2. Bit oriented
3. Literal
4. Control

2.6 BYTE

All byte instructions on the enhanced PIC16 contain a 6-bit opcode, 7-bit file address, and a destination bit. All PIC18 byte instructions contain a 6-bit opcode, 8-bit file address, a destination bit, and a RAM access bit.The sum of all the bit field sizes confirms that the PIC16 enhanced core does indeed have a 14-bit wide word size for instructions. Likewise, the same can be seen for the PIC18 for its 16-bit wide word length.

The RAM access bit (a) on the PIC18 is set when the user wishes to use the Bank Select Register (BSR) for manually selecting the bank. The PIC16 user will always need to make sure that they are in the correct bank by using the ‘`banksel`’ directive. This is explained in the first few lessons.

The destination bit (d) specifies whether the result will be stored in WREG or back in the original file register. When ‘d’ is zero, the result is placed in the WREG resister. Otherwise, the result is placed in the file register.

The file register (f) specifies which register to use. This can be a Special Function Register (SFR) or General Purpose Register (GPR).

EXAMPLE 2-2:

```
ADDWF data,f
```

This adds the contents of WREG and data, with the result being saved back to the file register data.

The PIC18 can move data from one file register directly to another file register, circumventing the WREG. All file moves in the enhanced PIC16 architecture must go through the WREG.

2.7 BIT

Bit instructions operate on a specific bit within a file register. These instructions may set or clear a specific bit within a file register. They may also be used to test a specific bit within a file register. All bit instructions on the enhanced PIC16 contain a 4-bit opcode, 7-bit file address, and a 3-bit bit address. All PIC18 byte instructions contain a 4-bit opcode, 8-bit file address, 3-bit bit address and a RAM access bit.

EXAMPLE 2-3:

```
BSF PORTA,RA0
```

This sets pin RA0 in the PORTA register.

2.8 LITERAL

Literal operations contain the data operand within the instruction. Both architectures use an 8-bit intermediate value. The rest of the bits are reserved for the opcode.

EXAMPLE 2-4:

```
MOVLW   ‘A’
```

This moves the ASCII value of ‘A’ (`0x41`) into WREG.

2.9 CONTROL

Instructions that dictate what address the PC will select in program memory are called control instructions. This would include `call`, `goto`, and `branch`. Each has a unique word length. Please refer to the "**Instruction Set Summary**" chapter in any PIC device data sheet for more information.

FIGURE 2-4: ENHANCED PIC16 GENERAL FORMAT FOR INSTRUCTIONS

Byte-oriented file register operations — **Example Instruction**

13 – 8	7	6 – 0
OPCODE	d	f (FILE #)

`ADDWF MYREG, W`

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13 – 10	9 – 7	6 – 0
OPCODE	b (BIT #)	f (FILE #)

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13 – 8	7 – 0
OPCODE	k (literal)

`BSF MYREG, BIT`

k = 8-bit immediate value

`CALL` and `GOTO` instructions only

13 – 11	10 – 0
OPCODE	k (literal)

`MOVLW 0x45`

k = 11-bit immediate value

`MOVLP` instruction only

13 – 7	6 – 0
OPCODE	k (literal)

`CALL LABEL`

k = 7-bit immediate value

`MOVLB` instruction only

13 – 5	4 – 0
OPCODE	k (literal)

`MOVLP 15`

k = 5-bit immediate value

`BRA` instruction only

13 – 9	8 – 0
OPCODE	k (literal)

`MOVLB 3`

k = 9-bit immediate value

`FSR` Offset instructions

13 – 7	6	5 – 0
OPCODE	n	k (literal)

`BRA LABEL`

n = appropriate FSR
k = 6-bit immediate value

`FSR` Increment instructions

13 – 3	2	1 – 0
OPCODE	n	m (mode)

`ADDFSR FSR1, 3`

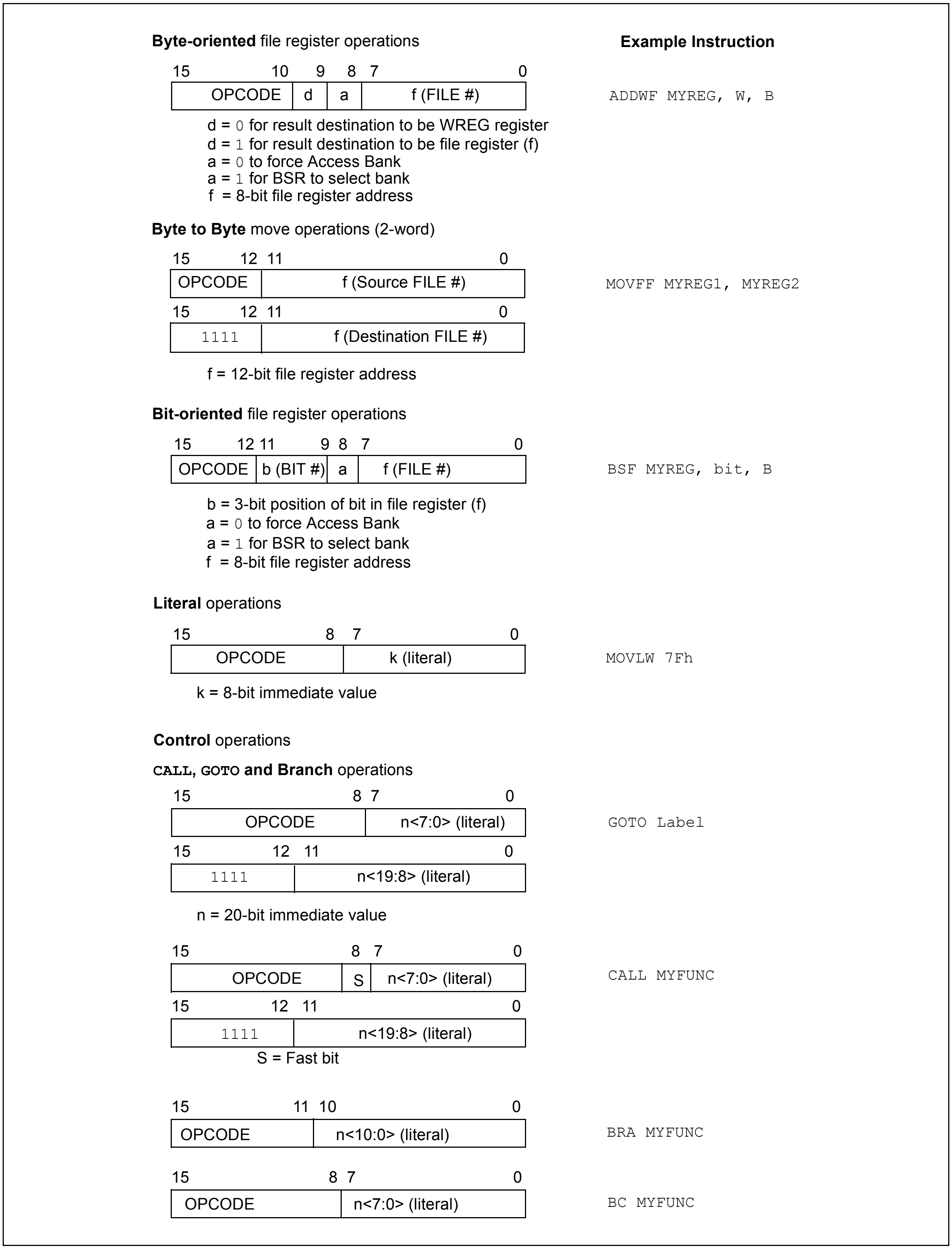
n = appropriate FSR
m = 2-bit mode value

OPCODE only

13 – 0
OPCODE

`MOVIW ++FSR0`

FIGURE 2-5: PIC18 GENERAL FORMAT FOR INSTRUCTIONS



There are some subtle differences between the block diagrams in Figure 2-1 and Figure 2-3. This document will point out a few of the important ones.

2.10 STACK LEVEL

The PIC18 has a deeper stack level of 31, whereas the enhanced core has 16. A deeper stack allows the PIC device to make more calls in the software before returning to the original address where the first call was made.

A `call` or `goto` modifies the program counter to point to a different place in code. Without these, the code would execute from the top to the bottom. The lessons will show the significance of this.

The call stack is used to save the return address before going to a new position in program memory.

As a frame of reference, some of the baseline parts (PIC10/12) devices have a call stack that is only two levels deep. It is quite a challenge to create modular code with a limited stack depth.

2.11 MEMORY ORGANIZATION

There are three sections of memory in the PIC16 enhanced mid-range and PIC18 devices:

1. Program Memory
2. Data RAM
3. Data EEPROM

2.12 PROGRAM MEMORY

There are five sections of program memory:

1. Flash Program Memory
2. Configuration Words
3. Device ID
4. Revision ID
5. User ID

2.12.1 Flash Program Memory

All enhanced mid-range and PIC18 devices use Flash memory for programming. Flash allows the PIC device to be erased and written to hundreds of thousands of times.

2.12.2 Configuration Words

There are several Configuration Word bits, or fuses, that allow different configurations at run-time. Oscillator selections, memory protection, low-voltage detection, etc., are some examples of configuration options. Each device has different configuration options. Enhanced mid-range Configuration bits are read-only during code execution. PIC18 can read all and write most Configuration bits during code execution. The Configuration bits are programmed in a special way, as seen in the lesson source files.

2.12.3 Device ID

The Device ID contains the read-only manufacture's ID for the PIC MCU. The PIC16F1829 ID is stored in DEVICEID and the PIC18F14K22 is stored in DEVID1 and DEVID2.