MPLAB® ICE 4000
IN-CIRCUIT EMULATOR
USER'S GUIDE

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.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, 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, MXDEV, MXLAB, PICMASTER, 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, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel and Total Endurance 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.

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

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, 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.

MPLAB® ICE 4000
USER'S GUIDE

Table of Contents

MPLAB® ICE 4000 User's Guide

Table of Contents

NOTES:

MPLAB® ICE 4000
USER'S GUIDE

Preface

INTRODUCTION

The general information discussed here can help you when using the MPLAB ICE 4000 emulator. Items discussed in this chapter include:

- About This Guide
- Warranty Registration
- Recommended Reading
- Troubleshooting
- The Microchip Web Site
- Development Systems Customer Change Notification Service
- Customer Support

ABOUT THIS GUIDE

Document Layout

This document describes how to use MPLAB ICE 4000 as a development tool to emulate and debug firmware on a target board. The manual layout is as follows:

- **Chapter 1: Overview** – What MPLAB ICE 4000 is and how it can help you develop your application.
- **Chapter 2: Installation** – How to install MPLAB ICE 4000 hardware and MPLAB IDE v6.xx software.
- **Chapter 3: General Set Up** – Setting up MPLAB ICE 4000 for use with MPLAB IDE.
- **Chapter 4: Basic Features** – A description of the basic features of MPLAB ICE 4000, (i.e., run, halt, reset, single step, etc.).
- **Chapter 5: External Memory Usage** – A description of PIC18F8XXX external memory modes and how they are supported on the emulator.
- **Chapter 6: Complex and Internal Triggers** – A description of complex triggers and dsPIC® internal triggers. Complex trigger examples are also given.
- **Chapter 7: Code Coverage, Trace Memory, Real-Time Reads** – A description of code coverage, emulator trace and real-time reads.
- **Chapter 8: Emulator Function Summary** – A summary of emulator functions available in MPLAB IDE when MPLAB ICE 4000 is chosen as the debug tool.
- **Appendix A: Troubleshooting** – How to solve common problems with MPLAB ICE 4000 operation.
- **Appendix B: Pod Electrical Specification** – The electrical specifications and description of the emulator pod.
- **Glossary** – A glossary of terms used.

MPLAB® ICE 4000 User's Guide

Conventions Used in this Guide

This manual uses the following documentation conventions:

DOCUMENTATION CONVENTIONS

Description	Represents	Examples
Main Document (Arial font):		
Italic characters	Referenced books	*MPLAB IDE User's Guide*
	Emphasized text	...is the *only* compiler...
Interface References (Arial font):		
Initial caps	A window, dialog or menu selection	Configuration Bits window, Settings dialog, Enable Programmer
Quotes	A field name in a window or dialog	"Save files before running the debugger"
Underlined, italic text with right arrow	A menu selection path	*File>Save*
Bold characters	A dialog button or tab	**OK** button, **Power** tab
Characters in angle brackets < >	A key on the keyboard	<Tab>, <Ctrl-C>
Code References (Courier font):		
Plain characters	File names and paths	`c:\autoexec.bat`
	Bit values	`0, 1`
	Sample code	`#define START`
Square brackets [ ]	Optional arguments	`mpasmwin [main.asm]`
Curly brackets and pipe character: { | }	Choice of mutually exclusive arguments An OR selection	`errorlevel {0|1}`
Italic characters	A variable argument; it can be either a type of data (in lower case characters) or a specific example (in uppercase characters).	`pic30-gcc` *`filename`*
Ellipses...	Replaces repeated instances of text	`list ["list_option..., "list_option"]`
0x*nnnn*	A hexadecimal number where *n* is a hexadecimal digit	`0xFFFF, 0x007A, 0x1A`
'*bnnnn*	A binary number where *n* is a digit	'b00100, 'b10

Documentation Updates

All documentation becomes dated, and this user's guide is no exception. Since Microchip tools and documentation are constantly evolving to meet customer needs, some actual dialogs and/or tool descriptions may differ from those in this document. Please refer to our web site to obtain the latest documentation available.

Documentation Numbering Conventions

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 = The document number.

 A = The revision level of the document.

WARRANTY REGISTRATION

Please complete the enclosed Warranty Registration Card and mail it promptly. Sending in your Warranty Registration Card entitles you 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 MPLAB ICE 4000. Other useful documents are listed below. The following Microchip documents are available and recommended as supplemental reference resources.

Readme for MPLAB ICE 4000

For the latest information on using MPLAB ICE 4000, read the "`Readme for MPLAB ICE 4000.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.

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.

MPLAB ICE 4000 Processor Module and Device Adapter Specification (DS51298)

Consult this document for information on the different processor modules and device adaptors available for use with the MPLAB ICE 4000 pod.

MPLAB ICE Transition Socket Specification (DS51194)

Consult this document for information on transition sockets available for use with MPLAB ICE 2000/4000 device adaptors.

TROUBLESHOOTING

See **Appendix A. "Troubleshooting"** for information on common problems.

THE MICROCHIP WEB SITE

Microchip provides online support on the Microchip World Wide Web (WWW) site. The web site is used by Microchip as a means to make files and information easily available to customers. To view the site, you must have access to the Internet and a web browser such as Netscape® Navigator or Microsoft® Internet Explorer.

The Microchip web site is available by using your favorite Internet browser:

www.microchip.com

The web site provides a variety of services. Users may download files for the latest development tools, data sheets, application notes, user's guides, articles and sample programs. A variety of information specific to the business of Microchip is also available, including listings of Microchip sales offices, distributors and factory representatives.

Technical Support

- Frequently Asked Questions (FAQ)
- Online Discussion Groups – Conferences for products, Development Systems, technical information and more
- Microchip Consultant Program Member Listing
- Links to other useful web sites related to Microchip products

Product/Design Support

- Design Tips
- Device Errata

Other available information

- Latest Microchip Press Releases
- Listing of seminars and events
- Job Postings
- Investor Information

DEVELOPMENT SYSTEMS CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip started the customer notification service to help our customers keep current on Microchip products with the least amount of effort. Once you subscribe, you will receive email notification whenever we change, update, revise or have errata related to your specified product family or development tool of interest.

Go to the Microchip web page (www.microchip.com) and click on Customer Change Notification under Support. Follow the instructions to register.

The Development Systems product group categories are:

- **Compilers** – The latest information on Microchip C compilers and other language tools. These include the 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.
- **Emulators** – The latest information on Microchip in-circuit emulators.This includes the MPLAB ICE 2000 and MPLAB ICE 4000.
- **In-Circuit Debuggers** – The latest information on the Microchip in-circuit debugger, MPLAB ICD 2.
- **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 ICD 2, PRO MATE II and MPLAB PM3 device programmers and the PICSTART Plus development programmer.

Preface

CUSTOMER SUPPORT

Microchip customers can receive assistance through several channels.

Hotline

There is a Systems Information and Upgrade Line. This line provides customers a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits.

The Hotline Numbers are:

1-800-755-2345 for U.S. and most of Canada.

1-480-792-7302 for the rest of the world.

In The Field

Customers should call their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. See the back cover for a listing of sales offices and locations.

Corporate Applications

Corporate Applications Engineers (CAEs) may be contacted at (480) 792-7627. You will need an active internet connection to obtain a "ticket" for assistance.

NOTES:

Chapter 1. Overview

1.1   INTRODUCTION

An overview of the MPLAB ICE 4000 system is given.

1.2   HIGHLIGHTS

This chapter discusses:

• MPLAB ICE 4000 Defined
• How MPLAB ICE 4000 Helps You
• MPLAB ICE 4000 Kit Components

1.3   MPLAB ICE 4000 DEFINED

MPLAB ICE 4000 is an In-Circuit Emulator (ICE) designed to emulate PIC18X microcontroller (MCU) devices and dsPIC digital signal controller (DSC) devices. It uses the latest emulation processors to provide full-speed emulation and visibility into both the instruction and the data paths during execution.

MPLAB ICE 4000 performs basic functions such as run, halt, single step, and software breakpoints, plus advanced features such as instruction address data monitoring, instruction data trace, complex triggering and code coverage, and extended memory access.

MPLAB ICE 4000 support is integrated into MPLAB IDE v6.xx, Microchip's 32-bit Integrated Development Environment (IDE). The MPLAB IDE desktop provides an environment for developing and debugging your application.

This document covers the basic setup and operation of the MPLAB ICE 4000 emulator, but it does not cover all functions of MPLAB IDE. Refer to the *MPLAB IDE v6.xx Quick Start* (DS51281) and the on-line help for MPLAB IDE v6.xx to get a full understanding of the features and debug capabilities of the MPLAB IDE.

1.4   HOW MPLAB ICE 4000 HELPS YOU

MPLAB ICE 4000 allows you to:

• Debug your application on your own hardware in real time.
• Debug with both hardware and software breakpoints.
• Measure timing between events using the stopwatch or complex trigger.
• Set breakpoints based on internal and/or external signals.
• Monitor internal file registers.
• Emulate full speed (depending on the device).
• Select the oscillator source in software.
• Program the application clock speed.
• Trace data bus activity and time stamp events.
• Set complex triggers based on program and data bus events, and external inputs.

1.5   MPLAB ICE 4000 KIT COMPONENTS

The components of the MPLAB ICE 4000 emulator kit are listed below.

1. MPLAB IDE v6.xx Quick Start (DS51281)
2. CD-ROM with MPLAB IDE software and on-line documentation
3. USB cable to connect the emulator pod to a PC
4. Emulator pod
5. Power supply and cable
6. Emulator stand
7. Processor module flex circuit cable
8. Logic probes

Additional hardware that may be ordered separately:

1. Processor module
2. Device adapter
3. Transition socket

Chapter 2. Installation

2.1   INTRODUCTION

An overview of the MPLAB ICE 4000 system components is given, as well as an explanation of how to install the system hardware and software.

2.2   HIGHLIGHTS

This chapter contains:

• MPLAB ICE 4000 System Components
• Driver and Software Installation
• Hardware Setup
• Applying Power to the System Components
• Applying Power to the System Components – Low Voltage Emulation
• Software Setup
• Removing Power From the System Components

2.3   MPLAB ICE 4000 SYSTEM COMPONENTS

The MPLAB ICE 4000 system consists of these items (Figure 2-1):

• Emulator pod
• Host-to-pod USB cable to connect a host PC to the emulator pod
• Power supply cable
• Processor module
• Flex circuit cable to connect the processor module to the device adapter
• Device adapter
• Transition socket to connect the device adapter to the target system
• Logic probe connector

FIGURE 2-1:   MPLAB ICE 4000 EMULATOR SYSTEM
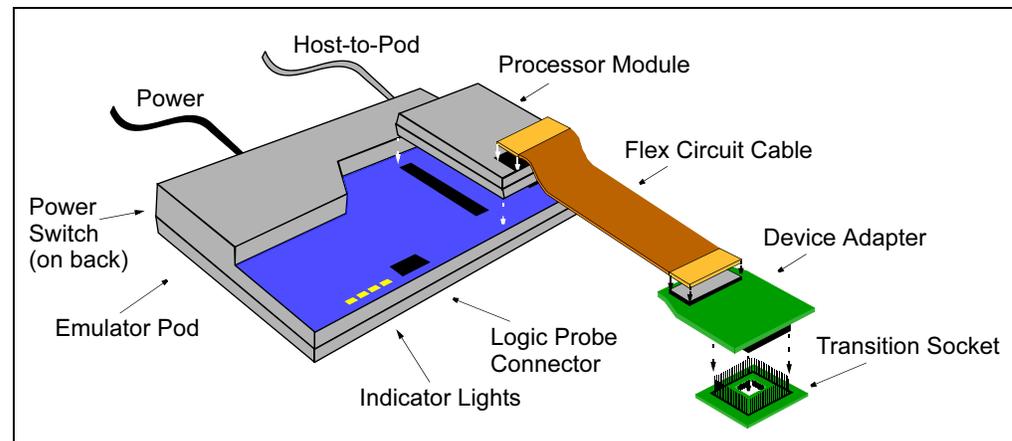
The emulator pod connects to the PC through a USB port using the provided cable. The pod contains the hardware necessary to perform the common emulator functions, such as trace, break and emulate.

The processor module inserts into two slots on top of the emulator pod. It contains the hardware necessary to emulate a specific device or family of devices. For more information on processor modules, see the *MPLAB ICE 4000 Processor Module and Device Adapter Specification* (DS51298).

The device adapter is connected to the processor module by the flex circuit cable. Device adapters are interchangeable assemblies that allow the emulator to interface to a target application system. Device adapters also have control logic that allows the target application to provide a clock source and power to the processor module. For more information on processor modules, see the *MPLAB ICE 4000 Processor Module and Device Adapter Specification* (DS51298).

The transition socket is connected to the device adapter. Transition sockets are available in various styles to allow a common device adapter to be connected to one of the supported surface mount package styles. For more information on transition sockets, see the *MPLAB ICE Transition Socket Specification* (DS51194).

The logic probes may be connected into the logic probe connector on the emulator pod.

2.4   DRIVER AND SOFTWARE INSTALLATION

CAUTION
Do not allow Windows® OS to pick a communications driver, **i.e., the emulator will not work and you will then have to uninstall the Windows driver so you may install the proper Microchip driver.** **If you have allowed the Windows driver to install, follow the directions in the file MPUsbClean.htm found in the Drivers*nn*\ICE4k_USB subdirectory of the MPLAB IDE installation directory, where *nn* is the version of Windows OS. Then return here to install the correct driver.**

1. Run the installation for the MPLAB IDE v6.xx software application on your PC. You may obtain the installation executable from the Microchip web site or from the MPLAB IDE CD-ROM available from Microchip.

2. When the MPLAB IDE installation is complete, the driver installation instructions will appear, as well as a dialog that asks you to reboot. Click **Cancel** in the dialog and follow the driver installation instructions.

    If you accidentally close these instructions, they may be found at:

    MPLAB IDE installation directory\Drivers*nn*\ICE4k_USB\Ddice4k*nn*.htm

    where *nn* represents the version of Windows OS.

3. Shut down your PC from the Start menu.

2.5   HARDWARE SETUP

CAUTION

The PC, MPLAB ICE 4000 and the target system should *NOT* be powered at this time.

---

1. Plug the host-to-pod USB cable into the PC and then the pod.

    Connect one end of the host-to-pod cable to the USB port on the PC chassis and connect the other end to the USB connector on the back of the MPLAB ICE 4000 pod.

2. Plug the power cord into the pod.

    Make certain that the emulator pod on/off switch is in the "O" or "off" position before completing this step.

    > **Note:** USB cannot power the emulator pod, i.e., MPLAB ICE 4000 must be run with the supplied external power supply.

3. Plug the power cord into an outlet.
4. Plug the processor module into the emulator pod.

    Insert the processor module firmly onto the top of the MPLAB ICE 4000 pod.

5. If a target board will be used with the MPLAB ICE 4000 system:

    a) Connect the end of the flex circuit cable marked "emulation module" to the processor module.

    b) Connect the end of the flex circuit cable marked "device adapter" to the device adapter.

CAUTION

To later remove the flex cable from either the processor module or device adapter, grip and pull up or push down on the stiffener. DO NOT twist off.

PULL UP OR PUSH DOWN

DO NOT TWIST

---
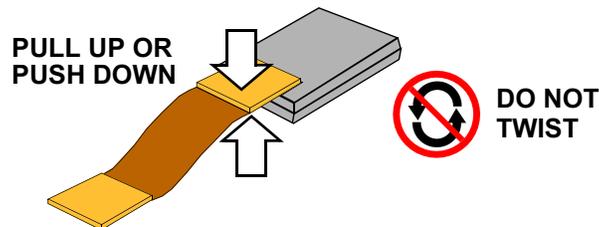
    c) Plug the device adapter into the transition socket on your target application. Make sure the target application powers the transition socket according the electrical specs for the device to be emulated.

    d) Connect the logic probes. Plug the logic probes into the logic probe connector found on the front of the emulator pod.

2.6 APPLYING POWER TO THE SYSTEM COMPONENTS

To prevent damage to any of the subsystem or target application parts, power up the system components as specified below.

CAUTION

Damage to the emulator system and/or target application may occur if these steps are not followed.

> **Note:** When power is supplied by the target system, MPLAB ICE 4000 loads the target system with up to 150 mA. An MPLAB ICE 4000 device adapter loads the system with up to 10 mA.

See **Section 2.7 "Applying Power to the System Components – Low Voltage Emulation"** before using **low voltage emulation** if this feature is desired.

1. Apply power to the PC.
2. Apply power to the emulator pod.

CAUTION

Insert the processor module BEFORE turning on the emulator pod. DO NOT insert a processor module with power applied to the pod.
---

3. Apply power to the target application circuit.

2.7 APPLYING POWER TO THE SYSTEM COMPONENTS – LOW VOLTAGE EMULATION

MPLAB ICE 4000 supports low voltage emulation (2.5V). In this configuration, **power MUST BE SUPPLIED by the target system**.

> **Note:** When power is supplied by the target system, MPLAB ICE 4000 loads the target system with up to 150 mA. An MPLAB ICE 4000 device adapter loads the system with up to 10 mA.

To prevent damage to any of the subsystem or target application parts, power up the system components as specified below.

CAUTION

Damage to the emulator system and/or target application may occur if these steps are not followed.
---

1. Check the limitations for your selected device. Some devices allow emulator pins to come up as output high, instead of input. Make certain you do not allow the emulated device to do this if your application cannot tolerate 5V.
2. Apply power to the PC.
3. Apply power to the emulator pod.

CAUTION

Insert the processor module BEFORE turning on the emulator pod. DO NOT insert a processor module with power applied to the pod.
---

4. Apply power to the target application circuit.

2.8   SOFTWARE SETUP

1. Launch MPLAB IDE v6.xx.
2. From the Configure menu, select Select Device. In this dialog, choose the device you will be emulating and click **OK**.
3. From the Debugger menu, select Select Tool and then MPLAB ICE 4000 to enable the emulator.

> **Note:**   If you cannot select the emulator as the debug tool, please see **Appendix A. "Troubleshooting"**.

Once you have selected MPLAB ICE 4000, debug options will appear on the Debugger menu.

> **Note:**   It is recommended that you **not** have a programmer enabled at the same time as the emulator. See **Appendix A. "Troubleshooting"** for more information.

4. The default port setting is USB = ICEUSB-0. If you wish to change this settings, you may do so on the Port tab of the *Debugger>Settings* dialog. See **Section 3.4 "Configuring the Communications Port"** for more information.
5. MPLAB ICE 4000 allows the emulator processor module to be powered by either the emulator pod or the target system. This is set up in MPLAB IDE as follows:
   • Emulator pod: *Debugger>Settings*, **Power** tab, "Processor Power From Emulator"
   • Target system: *Debugger>Settings*, **Power** tab, "Processor Power From Target Board"

> **Note:**   You must select "Processor Power From Target Board" when using low-voltage emulation.

See **Section 3.5 "Selecting Processor Power"** for more information. Also, refer to the *MPLAB ICE 4000 Processor Module and Device Adapter Specification* for processor module power requirements before configuring the system for target system power.

When connecting to a target application system, you may notice a voltage level on the target application, even though you have not yet applied power to the target application circuit. This is normal and is due to current leakage of protection diodes through V_{CC} of the Device Adapter. The current leakage will typically be less than 20 mA. However, if the target application is using a voltage regulator, it should be noted that some regulators require the use of an external shunt diode between V_{IN} and V_{OUT} for reverse-bias protection. Refer to the manufacture's data sheets for additional information.

MPLAB ICE 4000 should now be ready for you to use as an emulator. If you have had problems, please consult **Appendix A. "Troubleshooting"**. Otherwise, proceed to the next chapter for additional software setup considerations.

2.9   REMOVING POWER FROM THE SYSTEM COMPONENTS

To prevent damage to any of the subsystem or target application parts, power down the system components as specified below.

CAUTION
Damage to the emulator system and/or target application may occur if these steps are not followed.

1. Select _Debugger>None_.
2. Close MPLAB IDE v6.xx. Save any projects/work spaces when prompted.
3. Remove power from target application circuit.
4. Turn off the emulator pod.
5. Turn off the PC.

Chapter 3. General Set Up

3.1 INTRODUCTION

After installing MPLAB ICE 4000 and starting up the MPLAB IDE, MPLAB IDE must be set up to correctly emulate the selected processor.

3.2 HIGHLIGHTS

The steps needed to get started with MPLAB ICE 4000 are:

• Checking Configuration Bit Values
• Configuring the Communications Port
• Selecting Processor Power
• Setting Up the Processor Clock
• Setting Up Miscellaneous Hardware
• Using MPLAB IDE Projects and Work spaces

3.3 CHECKING CONFIGURATION BIT VALUES

To view the values of configuration bits for your selected device, open the Configuration Bits window by selecting *Configure>Configuration Bits*.

When you first select a device and start a project, default data sheet values for configuration bits are used. These may not be what you want for development, e.g., you may not want the watchdog timer enabled.

3.3.1 Watchdog Timer On/Off

Determine whether or not the Watchdog Timer (WDT) will be needed. During development, WDT is usually disabled. However, if a specific development requires the WDT to be enabled, remember to set its prescaler.

3.3.2 Processor Mode and External Memory

If a processor has a mode that supports external program memory (Microprocessor or Extended Microcontroller), select the mode here. Additional memory features are set up on the **Memory** tab of the Settings dialog.

PIC18C601/801 devices use external memory exclusively (they have no on-chip program memory), so there is no need for Processor mode selection bits. However, you may use the **Memory** tab to set up whether the external memory is provided by the emulator or the target.

3.3.3 Oscillator Settings

Select the oscillator you will use for development here and the frequency for the oscillator in *Debugger>Settings*, Clock tab (**Section 3.6 "Setting Up the Processor Clock"**). Make sure you only enter frequencies that are available for the selected oscillator.

3.4 CONFIGURING THE COMMUNICATIONS PORT

MPLAB ICE 4000 communicates with the PC via a USB port.

1. Select *Debugger>Settings* and click on the Port tab.
2. Select a communications port, where ICEUSB-x are the available USB ports. The number of available ports will depend on the configuration of individual PC's.

> **Note:** Before you change the communications port in software, make sure you have first changed it in hardware.

Click **Apply** to accept the setting in this tab of the Settings dialog.

3.5 SELECTING PROCESSOR POWER

MPLAB ICE 4000 allows the emulator processor chip to be powered by the emulator pod (5V) or the target system (2.5V to 5V). The emulator defaults to Processor Power from Emulator (system power) when first initialized.

For information on processor module power issues, please refer to the *MPLAB ICE 4000 Processor Module and Device Adapter Specification*.

FIGURE 3-1:   SETTINGS DIALOG – POWER TAB

3.5.1 Processor Power from Emulator (System Power)

Select the **Power** tab of the *Debugger>Settings* dialog. Under Processor Power, select From Emulator and click **Apply**.

> **Note:** The emulator system cannot provide power to a target board through a device adapter. If the device adapter is plugged into an unpowered target board, it is not unusual to see a voltage level of 1-3V at V_{CC} of the target board, caused by leakage current through V_{CC} of the device adapter. MPLAB IDE may also have difficulty initializing the emulator when power has not yet been applied to the target board.

3.5.2 Processor Power from Target Board

Select the **Power** tab of the *Debugger>Settings* dialog. Under Processor Power, select From Target Board and click **Apply**.

Refer to the *MPLAB ICE 4000 Processor Module and Device Adapter Specification* for processor module power requirements before configuring the system for target system power.

> **Note:** If you use power from the target board, make sure it is always present or the emulator will not function properly. If the device adapter is plugged into an unpowered target board, there may still be some leakage current through V_{CC} of the device adapter. MPLAB IDE may also have difficulty initializing the emulator when power has not yet been applied to the target board.

3.5.3 Low Voltage Emulation

MPLAB ICE 4000 also supports LOW VOLTAGE emulation down to 2.5V. The emulator system cannot provide any voltage level other than 5V to the emulator processor. In this configuration, power must be supplied by the target board (see **Section 3.5.2 "Processor Power from Target Board"**).

Before using low voltage, check the limitations for your selected device. Some devices allow emulator pins to come up as output high, instead of input. Make certain you do not allow the emulated device to do this if your application cannot tolerate 5V.

3.6 SETTING UP THE PROCESSOR CLOCK

MPLAB ICE 4000 can use the on-board clock with either emulator or target power, or it can use the target board clock with target board power.

3.6.1 Using the On-board Clock

MPLAB ICE 4000 has an on-board clock that can be programmed to a frequency between 32 kHz and 100 MHz. Refer to the specific device's data sheet to determine the supported frequency range.

1. Select an Oscillator Type (**Section 3.3.3 "Oscillator Settings"**).
2. Select the **Clock** tab on the _Debugger>Settings_ dialog.
3. Select the Desired Frequency magnitude (MHz, kHz or Hz) and enter the Desired Frequency. Refer to the specific device's data sheet to determine the supported frequency range for each oscillator type.

> **Note:** If you enter a frequency that is out of range, your system will not operate properly.

4. Click **Apply**.

The clock will be programmed to operate as close to the entered frequency as possible. Since the generated clock frequency will be slightly different than the desired clock frequency, the Actual Frequency will be displayed. The Actual Frequency will be within 0.5% of the desired frequency.

3.6.1.1 PLL

If the Oscillator mode has a HW PLL associated with it, the run time frequency will be the desired frequency. Example: To emulate a target with a 5 MHz HS crystal while using HW PLL mode, set the desired frequency to 20 MHz.

For parts (e.g., PIC18F8680) that support SW enabled PLL, please do not enter a frequency that, when multiplied by 4, would go over the maximum speed the emulator supports.

Consult the device limitations to see how PLL is emulated for your selected device.

3.6.1.2 VERIFY FREQUENCY

To verify the clock frequency, you can set up a complex trigger and then measure the trigger output pulse width (one instruction cycle) of the TRIGOUT logic probe (**Section B.7 "Logic Probes"**) in frequency and multiply by 4.

3.6.2   Using a Target Board Clock

MPLAB ICE 4000 can use the processor clock on the target board as long as target board (external) power is being used. It can determine the frequency of the target board clock and use it for displaying timing information.

1. Select the target board Oscillator Type (**Section 3.3.3 "Oscillator Settings"**).
2. Select the **Power** tab of the *Debugger>Settings* dialog and set Processor Power to From Target Board (see **Section 3.5.2 "Processor Power from Target Board"**).
3. Select the **Clock** tab on the *Debugger>Settings* dialog. Then select Use Target Board Clock.

> **Note:** If MPLAB ICE 4000 is not hooked up to a target board and you click **Use Target Board Clock**, you will get a warning dialog.

The target board clock frequency will be calculated, displayed and used for any internal time calculations. A warning is issued if the frequency is less than 32 kHz.

Because of measurement error, the calculated frequency may not be what is desired for internal time calculations. (e.g., Your crystal oscillator has a frequency of 8 MHz ± 50 ppm, but the target frequency is shown as 7.993755.) Measurement error can range from 3.9% to a fraction of a percent.

4. Click **Apply**.

3.7   SETTING UP MISCELLANEOUS HARDWARE

In addition to the settings you have already made, there are other settings that you may or may not wish to change in the *Debugger>Settings* dialog. Depending on the device you have chosen, these tabs may or may not be available and may look different for different devices.

3.7.1   Settings Dialog, Break Options Tab

Use the Break Options tab to change the global break and trace point environment options. These include Global Hardware Break Enable (for complex trigger usage) and Freeze Peripherals on Halt. If enabled in the configuration bits (*Configure>Configuration Bits*), you may set stack and watchdog timer break options.

3.7.2   Settings Dialog, Memory Tab

Some parts allow device memory to be supplemented or replaced by off-chip (external) memory. Memory modes are selected using configuration bits (*Configure>Configuration Bits*).

Devices that support Microcontroller mode only do not have a Memory tab. Devices that support Extended Microcontroller mode and/or Microprocessor mode will display the Memory tab.

> **Note:** There are several limitations concerning external memory, some of them device-dependent. Please see the limitations section of the on-line help file for more information.

3.7.3 Settings Dialog, Pins Tab

Set up processor pins, such as the $\overline{\text{MCLR}}$ pull-up resistor connection. Also set up the emulator operation to preserve user's external bus access method when halted.

3.7.4 Settings Dialog, Peripheral Tab

Set up peripheral functions, such a freeze on halt.

3.8 USING MPLAB IDE PROJECTS AND WORK SPACES

MPLAB IDE v5.xx and lower used projects to help you manage the files to build your application. MPLAB IDE v6.xx now uses projects and work spaces to aid in the development of complicated applications.

For general information on projects and work spaces, see the on-line help for MPLAB IDE v6.xx.

NOTES:

Chapter 4. Basic Features

4.1 INTRODUCTION

MPLAB ICE 4000 provides a wide variety of tools to emulate and debug an application. MPLAB ICE 4000 offers a basic set of in-circuit debugging tools, including the ability to run, halt, reset and single step the processor, plus additional tools for advanced debugging techniques.

Several basic MPLAB ICE 4000 emulator features are built-in to the MPLAB IDE software. A general description of these features is provided here, but for more detailed information, consult the MPLAB IDE documentation.

Other basic tools appear when MPLAB ICE 4000 is selected as the debug tool.

4.2 HIGHLIGHTS

MPLAB ICE 4000 basic features include the following:

• Starting and Stopping Emulation
• Viewing Processor Memory and Files
• Using Software Breakpoints
• Using Hardware Breakpoints
• Using Trigger In/Out Settings
• Using a Real-Time Watch
• Using the Stopwatch
• Monitoring Emulator States and Operations

4.3 STARTING AND STOPPING EMULATION

To debug an application in MPLAB IDE, you must either build your source code into an executable file using projects and work spaces (see MPLAB IDE on-line help for more information) or you must import an existing executable file using *File>Import*. Once you have your application in executable form, you may run, halt, step through or reset your code.

• To run your code, select either *Debugger>Run* or **Run** from the Debug toolbar.
• To halt your code, select either *Debugger>Halt* or **Halt** from the Debug toolbar.
• To step through your code, select either *Debugger>Step Into* or **Step Into** from the Debug toolbar. Be careful not to step into a Sleep instruction or you will have to perform a processor reset to resume emulation.

> **Note:** You cannot step through external memory high-level language code in the Program Memory or File window. Use the Disassembly window.

• To step over a line of code, select either *Debugger>Step Over* or **Step Over** from the Debug toolbar.
• To repeatedly step through your code, select either *Debugger>Animate* or **Animate** from the Debug toolbar.

- To perform a processor reset on your code, select either *Debugger>Reset> Processor Reset* or **Reset** from the Debug toolbar. Additional resets, such as POR/BOR, MCLR and System, may be available, depending on device.

Remember to rebuild your program when making changes so the code being executed accurately reflects the current state of your application.

4.4   VIEWING PROCESSOR MEMORY AND FILES

MPLAB IDE provides several standard windows for viewing debug and various processor memory information, selectable from the View menu. See MPLAB IDE on-line help for more information on using these windows.

- Project – view the project tree
- Output – view output from build and other tools
- Hardware Stack – view the hardware stack contents
- Program Memory – view program memory contents
- File Registers – view file register contents
- EEPROM – view the EEPROM memory contents
- Watch – view selected SFRs and symbols
- Special Function Registers – view SFR contents

In addition, you may view trace results. For more on trace, see **Chapter 7. "Code Coverage, Trace Memory, Real-Time Reads"**.

- ICE Trace – view the contents of the trace buffer

To view your source code, select *File>Open* and enter or browse for the source code file. Code in this window is color-coded according to the processor and build tool selected. To change the style of color-coding, click the right mouse button and select *Advanced>Text Mode*. To change the colors used, click the right mouse button and select Properties, Text tab. For information on using this window to edit your code, see MPLAB Editor on-line help.

4.5   USING SOFTWARE BREAKPOINTS

MPLAB ICE 4000 uses software breakpoints to halt the processor at a specific location. With a software breakpoint, execution stops before the instruction at the break location is executed.

> **Note:**   You cannot set software breakpoints in external Flash memory. Use the complex trigger to set hardware breakpoints.
> You can set software breakpoints in external RAM memory.

Some considerations when using software breakpoints:

- If a software breakpoint is set, then checksums calculated on program memory at run time will be incorrect. Programmer checksums are not affected.
- Manually resetting the time-stamp when using software breakpoints will cause the time-stamp to be incorrect.

Software breakpoints are a standard MPLAB IDE debug feature:

- MPLAB IDE Help – Using Breakpoints

4.6   USING HARDWARE BREAKPOINTS

In addition to setting software breakpoints on program memory addresses, hardware breakpoints may be set with more complex conditions. Also, hardware breakpoints can be used to capture real-time events.

With a hardware breakpoint, execution halt may *skid*, or one or more additional instructions may be executed past the set breakpoint before the processor halts.

Hardware breakpoints are set using the complex trigger. The complex trigger can be set up to require that up to four events occur before a break (or trace) occurs. The combination of these events can be specified three ways:

- Sequential
- All Events
- Any Event

In addition, the complex triggering feature along with the trace memory window can be used to perform the following functions:

- Time Between Events
- Filter Trace

Complex trigger breakpoints can then be selectively enabled and disabled. Breakpoints set in this manner are retained with the project.

Select *Debugger>Complex Triggers and Code Coverage* and click on the Complex Trigger Settings tab.

For more information on complex triggering, as well as complex triggering examples, see **Chapter 6. "Complex and Internal Triggers"**.

4.7   USING TRIGGER IN/OUT SETTINGS

MPLAB ICE 4000 offers the following external input and output options:

- Generate a single pulse of nonspecific duration, either on the final trigger event or on a filtered trace event. Use the positive edge to trigger other equipment.
- Break on a specified edge of an external trigger input.
- Freeze the trace buffer on the rising edge of the external trigger input.

These options are set through the Trigger In/Out Settings dialog (via *Debugger>Complex Triggers and Code Coverage*, Trigger In/Out Settings tab). Trigger In/Out Settings can be used with the logic probes (**Section B.7 "Logic Probes"**).

MPLAB® ICE 4000 User's Guide

FIGURE 4-1:   **TRIGGER IN/OUT SETTINGS TAB**
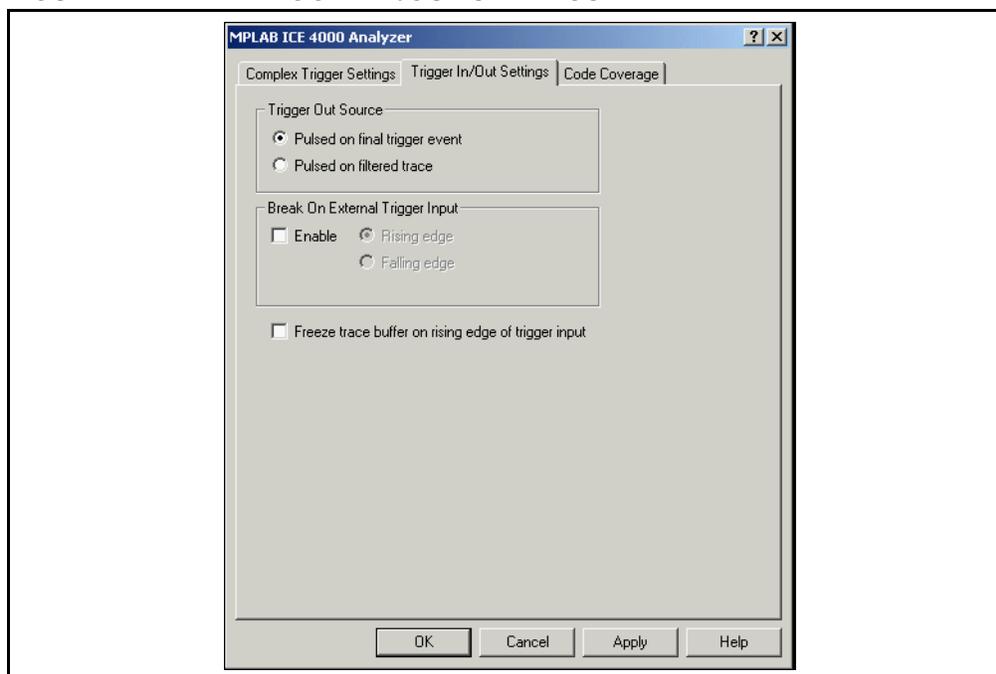
4.7.1   Trigger Out Source (TRGOUT)

Select either "Pulsed on final trigger event" or "Pulsed on filtered trace". If you select "Pulsed on final trigger event", the pulse will only occur on the final event. If you want a repeating pulse every time an event occurs, select "Pulsed on filtered trace".

4.7.2   Break On External Trigger Input (TRGIN)

Select Enable break on external trigger input, and then specify the following:

- Break on a specified rising or falling edge of an external trigger input.
- Freeze trace buffer on rising edge of trigger input (TRGIN)

Select Freeze trace buffer on rising edge of trigger input if you wish to freeze the trace buffer on the rising edge of the external trigger input.

4.8   USING A REAL-TIME WATCH

Watch window variables will be updated real-time if bus access (read/write) occurs that affects those variables. However, increments or decrements of the variables will not cause an updates.

For more information on watch windows, please refer to documentation for MPLAB IDE v6.xx or greater.

4.9   USING THE STOPWATCH

MPLAB ICE 4000 provides a stopwatch to use with emulation (see dialog below).

- **System Clock** – The running time, in seconds, based on the PC system clock. You may reset the time immediately by clicking **Reset Now**, or on a halt by checking Clear on Reset (When Halted).
- **Processor** – Displays the frequency for the selected device (Processor Frequency) in MHz versus the calculated (average) Instruction Cycle rate. For the actual instruction cycles run, see the ICE trace window.

FIGURE 4-2:   STOPWATCH DIALOG
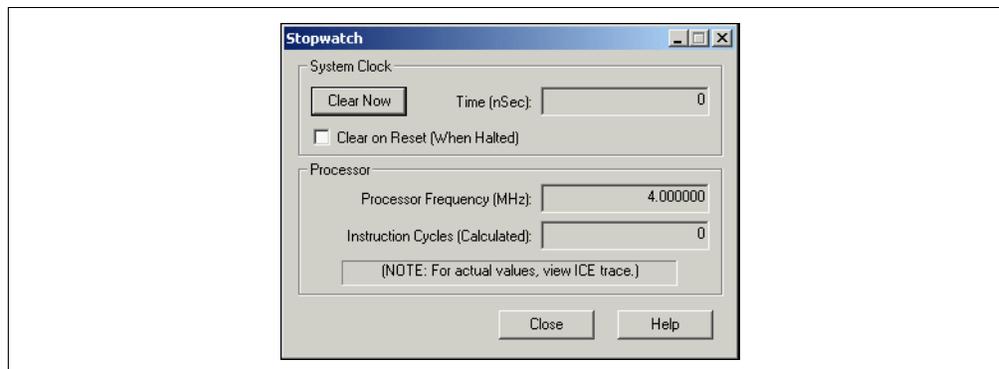
4.10   MONITORING EMULATOR STATES AND OPERATIONS

Select *Debugger>Settings*, **View** tab and click **Show State and Operation Gauge** to open an emulator monitor window.

FIGURE 4-3:   ICE STATE AND OPERATIONS GAUGE WINDOW
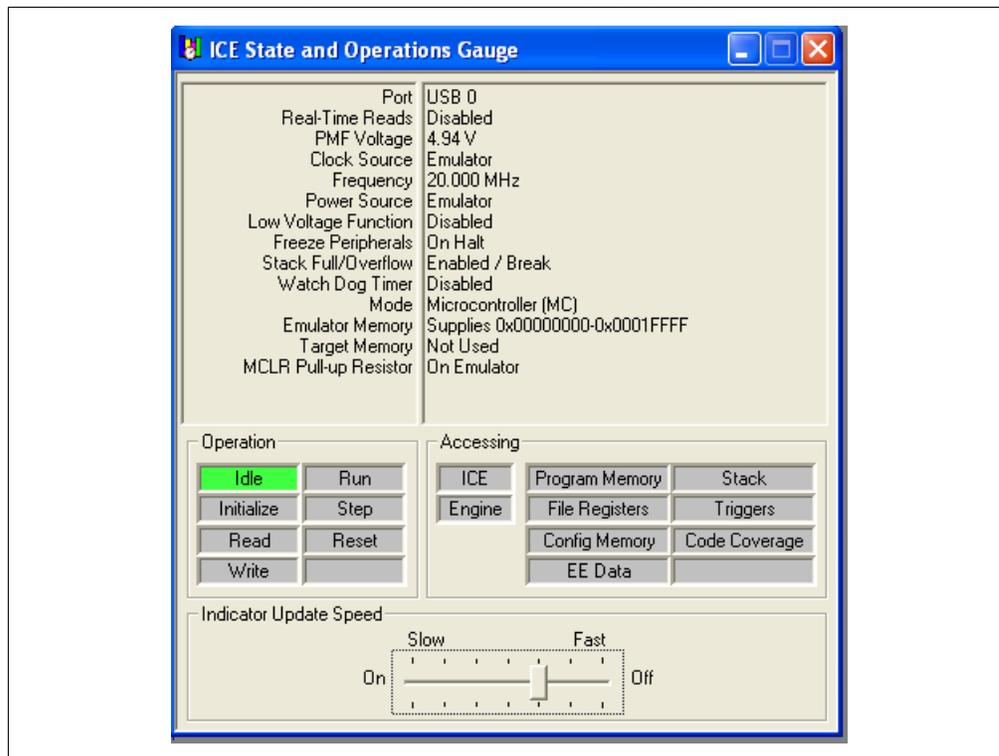
MPLAB® ICE 4000 User's Guide

View basic emulator data in the top portion of the window. Depending on the device selected for emulation, more or less items may be displayed.

TABLE 4-1:   EMULATOR DATA DEFINITIONS

Item	Definition	Change Using
Port	Type of communications port between the emulator and the PC	*Debugger>Settings*, **Port** tab
Real-Time Reads*	Enabled or disabled	*Debugger>Settings*, **View** tab
PMF Voltage	Current operating voltage of the processor module	See Low Voltage Function
Clock Source	Emulator or target	*Configure>Configuration Bits*, Oscillator *Debugger>Settings*, **Clock** tab
Frequency	Current operating frequency	*Configure>Configuration Bits*, Oscillator *Debugger>Settings*, **Clock** tab
Power Source	Emulator or target	*Debugger>Settings*, **Power** tab
Low Voltage Function*	Enabled or disabled	*Configure>Configuration Bits*, Low Voltage Program
Freeze Peripherals	On halt or Not Used	*Debugger>Settings*, **Break Options** tab
Stack Full/Overflow	Enabled or disabled, break or reset	*Configure>Configuration Bits*, Stack Overflow *Debugger>Settings*, **Break Options** tab
Watchdog Timer	Enabled or disabled, break or reset	*Configure>Configuration Bits*, Watchdog Timer *Debugger>Settings*, **Break Options** tab
Mode*	Microcontroller, Extended Microcontroller or Microprocessor	*Configure>Configuration Bits*, Processor Mode *Debugger>Settings*, **Memory** tab
Emulator Memory*	Address range	*Configure>Configuration Bits*, Processor Mode *Debugger>Settings*, **Memory** tab
Target Memory*	Address range or Not Used	*Configure>Configuration Bits*, Processor Mode *Debugger>Settings*, **Memory** tab
MCHP Pull-up Resistor*	On emulator or on target board	*Debugger>Settings*, **Pins** tab

* Availability dependent on device emulated.

Operation and **Accessing** portions of the window will display the current operation being performed by the emulator and what, if any, memory or functions are being accessed during this operation. Items will change from gray to green (highlighted) when active.

You may select how quickly the window is updated with information using the **Indicator Update Speed** bar.

Chapter 5. External Memory Usage

5.1   INTRODUCTION

Some Microchip devices allow the extension or replacement of program memory resources with external (off-chip) memory devices. Of these devices, the MPLAB ICE 4000 supports PIC18F8XXX devices (but not PIC17CXXX devices.)

How the emulator functions for these modes is discussed here.

5.2   HIGHLIGHTS

Topics covered in the chapter are:

• PIC18F8XXX Program Memory Modes
• Emulating PIC18F8XXX Program Memory Modes
• MPLAB IDE and External Memory

5.3   PIC18F8XXX PROGRAM MEMORY MODES

PIC18F8XXX devices are capable of operating in any one of several Program Memory modes, using combinations of on-chip and external program memory. Available Program Memory modes are as follows:

• The **Microprocessor** mode permits access only to external program memory; the contents of the on-chip Flash memory are ignored. The program counter permits access to a linear program memory space and defines the amount of accessible program memory (see **Section 5.3.1 "Program Counter"**.)

• The **Microprocessor with Boot Block** mode accesses on-chip Flash memory from address 000000h to the end of the boot block. Above this, external program memory is accessed all the way up to the program counter accessible limit (see **Section 5.3.1 "Program Counter"**.) Program execution automatically switches between the two memories, as required.

• The **Microcontroller** mode accesses only on-chip Flash memory. Attempts to read above the physical limit of the on-chip Flash causes a read of all '0's (a NOP instruction.)

• 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 Flash memory; above this, the device accesses external program memory up to the program counter accessible limit (see **Section 5.3.1 "Program Counter"**.) As with Boot Block mode, execution automatically switches between the two memories, as required.

In all modes, the device has complete access to data RAM and EEPROM. For more information, consult the device data sheet section "Memory Organization".

MPLAB® ICE 4000 User's Guide

FIGURE 5-1: **MEMORY MAPS FOR PROGRAM MEMORY MODES**

Boot – End of boot block.
Boundary – End of on-chip program memory.
xxxxxxh – End of accessible program memory. Defined by size of program counter.
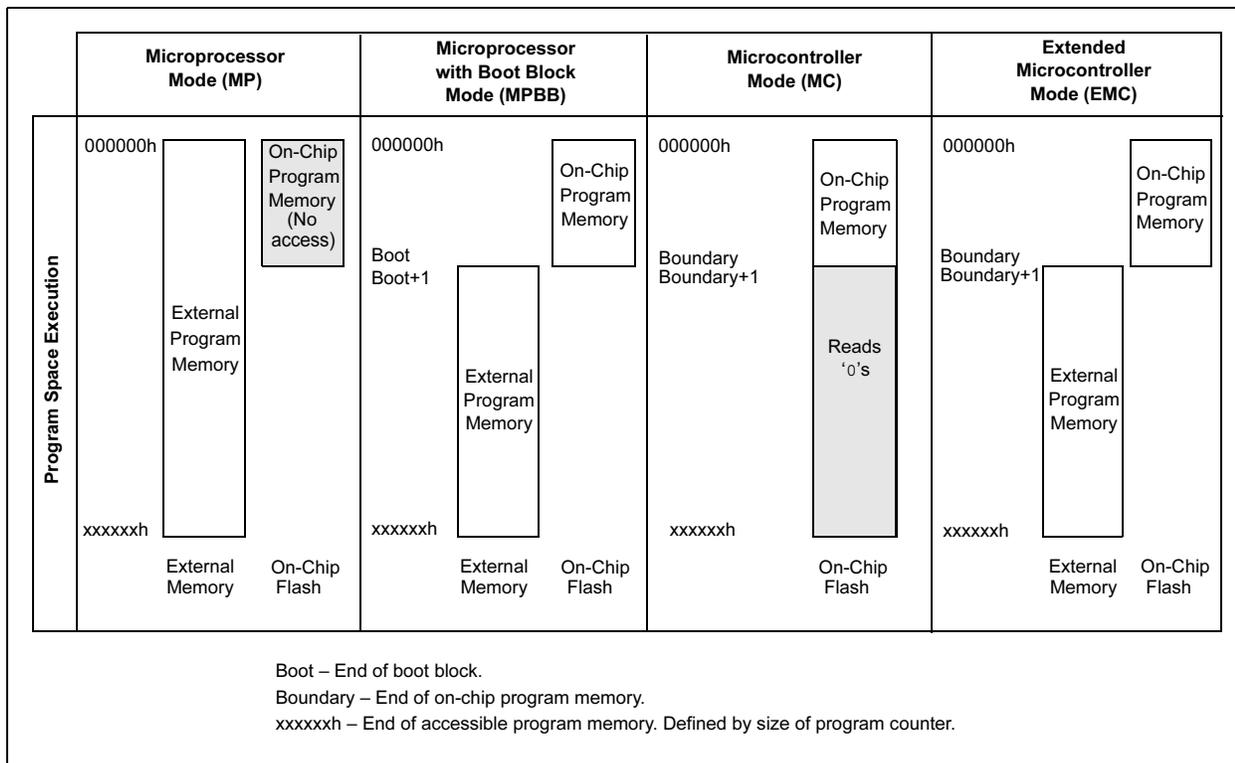
5.3.1 Program Counter

The size of the program counter will determine how much program memory can be accessed, i.e., a 21-bit program counter permits access to a 2-Mbyte (1-Mword) program memory space (on-chip, off-chip or a combination of both types of program memory.)

5.3.2 Configuration Bits

A Program Memory mode is set by using configuration bits. Depending on the device, the Processor Mode Select bits are located in a configuration register which is programmed when the device is programmed. For more information, consult the device data sheet section "Special Features of the CPU".

5.3.3 External Memory Interface

The External Memory Interface is a feature of a PIC18F8XXX device that allows the microcontroller to access external memory devices (such as Flash, EPROM, SRAM, etc.) as program or data memory.

Using the MEMCON register, the following may be configured:

• External bus enable and disable
• 16-Bit Mode – Word Write mode, Byte Select mode or Byte Write mode
• Wait – Table read/write bus cycle wait counts (0-3 TCY)

For more information, see the External Memory Interface section of the device data sheet.

5.4   EMULATING PIC18F8XXX PROGRAM MEMORY MODES

Emulating a device that uses external memory requires the following steps:

1. Setting Configuration Bits
2. Setting External Memory
3. Setting Memory Options

5.4.1   Setting Configuration Bits

To set the values of configuration bits for your selected device, open the Configuration Bits window by selecting *Configure>Configuration Bits*. In the Category column, find Processor mode and select the mode.

> **Note:** Configuration bits may also be set in code using `__config`. Refer to the device data sheet and header file (`.inc` or `.h`) for more information.

5.4.2   Setting External Memory

To set up MPLAB IDE and MPLAB ICE 4000 to use external memory, select *Configure>External Memory*. Then check "Use External Memory" and enter a range.

5.4.3   Setting Memory Options

To set up memory options, open the Settings dialog (*Debugger>Settings*) and select the **Memory** tab (**Section 8.17 "Settings Dialog, Memory Tab"**).

The Processor mode currently selected in the configuration bits will be reflected under "Mode". If the mode selected supports external memory, functions on this tab will be active.

When emulating a Program Memory mode that makes use of external memory, two choices are available:

- **External Program Memory Supplied by Emulator** – Use emulator memory for both on-chip and off-chip (external) memory during development. This has the advantage of speeding up program load time after a build. It has the disadvantage of not actually using the target external memory.
- **External Program Memory On Target Board** – Use the emulator as a device would be used, with the emulator memory representing only on-chip memory. This has the advantage of testing the application as it will actually be run, with external memory. It has the disadvantage of longer upload/download times, as commands, writes and reads to external memory will take time.

  In order for the MPLAB ICE 4000 to load your code into external program memory, the target system must provide SRAM or DRAM. If the target system uses non-volatile memory, such as Flash, the emulator will not be able to load code into external memory. For non-volatile memory, you must provide an alternate method of loading program memory. However, the emulator can upload non-volatile memory and run from non-volatile memory.

5.4.3.1   EXTERNAL PROGRAM MEMORY SUPPLIED BY EMULATOR

To use only emulator memory, select "Supplied by Emulator" under "External (Off-Chip) Program Memory". The amount of emulator-supplied memory will display in "Emulator Supplied" under "ICE Memory Mapping". "Target Supplied" should say "Not Used".

If you still need to access a range of target memory to control external circuits, set up a "Memory Mapped Peripheral Range". Check the box "Enable Banked Access Mode" and enter a target memory start and end address. This may either be a range within the emulator range, or a range above this but below the program counter maximum (see **Section 5.3.1 "Program Counter"**.)

FIGURE 5-2: MEMORY MAPPED PERIPHERAL RANGE
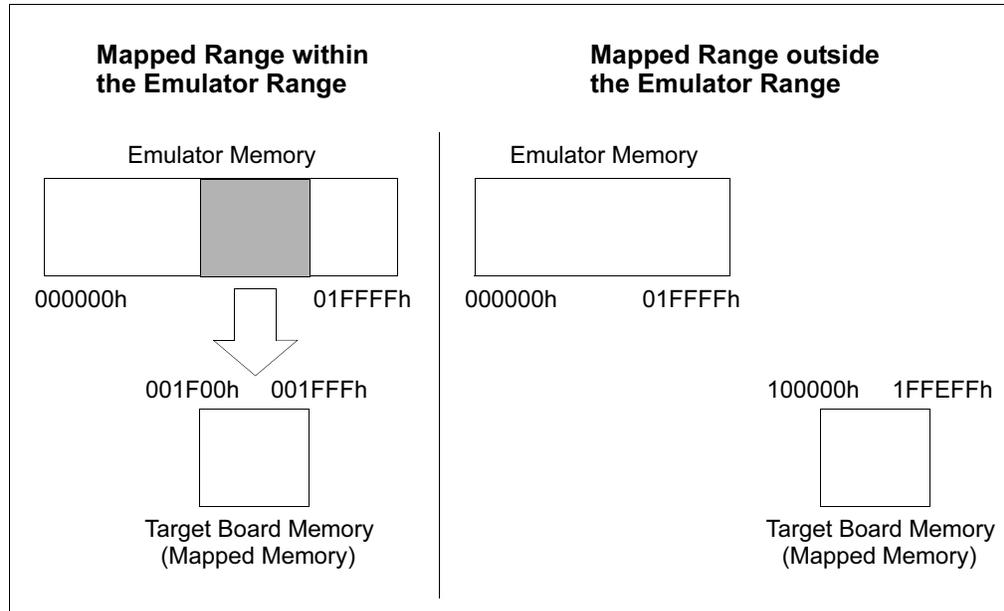
The amount of emulator- and target-supplied memory will display in "Emulator Supplied" and "Target Supplied" under "ICE Memory Mapping", respectively.

In order to communicate with the target board memory, you will need to set up the external memory interface under "Target 16-Bit Access Mode". Select the access mode (Word Write, Byte Select or Byte Write) and a table read/write Wait time in cycles. For more information, see the MEMCON register under the External Memory Interface section of the device data sheet.

5.4.3.2   EXTERNAL PROGRAM MEMORY ON TARGET BOARD

To use a combination of emulator and target memory, or all target memory, select "On Target Board" under "External (Off-Chip) Program Memory". The amount of emulator- and target-supplied memory will display in "Emulator Supplied" and "Target Supplied" under "ICE Memory Mapping", respectively.

In order to communicate with the target board memory, you will need to set up the external memory interface under "Target 16-Bit Access Mode". Select the access mode (Word Write, Byte Select or Byte Write) and a table read/write Wait time in cycles. For more information, see the MEMCON register under the External Memory Interface section of the device data sheet.

5.4.3.3   PIC18C601/801 CHIP SELECT SETUP

For PIC18C601/801 devices, you should specify values for the CSEL2 and CSELIO registers. See the device data sheet for more on these registers.

5.5   MPLAB IDE AND EXTERNAL MEMORY

Using External Memory modes with MPLAB ICE 4000 will impact the following features of MPLAB IDE.

5.5.1   Viewing Program Memory

To view the program memory, select _View>Program Memory_. The data for emulator program memory is always displayed. The data for external (target board) memory will be displayed as long as "Use External Memory" has been specified in the External Memory dialog (_Configure>External Memory_). To refresh the external memory display, select _Debugger>Upload Program Memory from ICE_ after you run, single step or trace. If uploading takes too long, you may wish to change the amount of external memory used in the External Memory dialog.

5.5.2   Stepping Through Code

If your external Flash memory contains code that is written in a high-level language, such as C, you cannot step through this code using the File (Editor) window. You must use the Program Memory window.

5.5.3   Setting Breakpoints

Software breakpoints may be set for emulator memory or external RAM memory. Hardware breakpoints, set using the complex trigger, may be used for either emulator or any type of external memory.

NOTES:

Chapter 6. Complex and Internal Triggers

6.1 INTRODUCTION

In addition to the break and trigger functions discussed in **Chapter 4. "Basic Features"**, MPLAB ICE 4000 has more advanced triggering features. These include complex triggers for all devices and internal triggers for dsPIC devices.

6.2 HIGHLIGHTS

This chapter covers:

• Complex Triggers
• Internal Triggers

6.3 COMPLEX TRIGGERS

MPLAB ICE 4000 has a highly flexible and powerful triggering mechanism. A trigger is a combination of events that can cause:

• a hardware breakpoint, and/or
• a trace memory capture.

An event is a description of the state of the system captured during one cycle.

In addition, an external signal can be generated when the trigger occurs. This is useful for synchronizing other analyzers or equipment to MPLAB ICE 4000.

Complex trigger functions may be set using the Complex Trigger Settings tab of the MPLAB ICE 4000 Analyzer dialog.

> **Note:** You must enable the Global Hardware Break Enable (*Debugger>Settings*, Break Options tab) to use complex triggering.

• Complex Trigger Settings
• Complex Trigger Settings Syntax
• Trigger Type Selection
• Memory Selection
• Complex Triggering Examples

6.4 COMPLEX TRIGGER SETTINGS

Complex triggers can be specified by selecting *Debugger>Complex Triggers and Code Coverage*, Complex Trigger Settings tab. The Complex Trigger Settings tab will contain different items depending on your selection of:

1. Trigger Type – Sequential, All Events, Any Event, Time Between Events or Filter Trace
2. Memory – Program Memory or Data Memory

Figure 6-1 shows what the tab will look like for a sequential event and program memory selection. This is the typical layout of the Complex Trigger Settings tab.

FIGURE 6-1:   COMPLEX TRIGGER – PROGRAM MEMORY, SEQUENTIAL
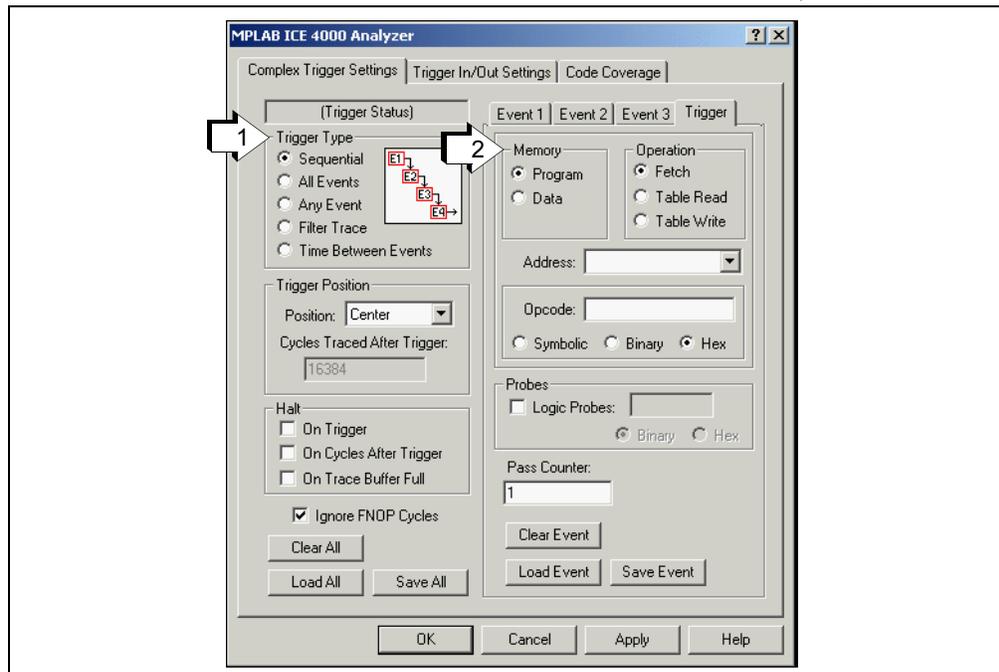
For information on trigger type (event) selection, see **Section 6.6 "Trigger Type Selection"**.

Trace-related trigger information may be entered in the following items:

• **Trigger Position** – Used to position the trigger location in the trace memory window, indicating the number of cycles captured by the trace memory window after the trigger occurs (Cycles Traced After Trigger). The approximate cycle that generated the trigger will be indicated with an arrow.

• **Halt On Trigger** – The trigger point will generate a hardware breakpoint, halting the processor, and will be the last entry in the trace memory window. To capture traces without halting the target, unselect this option.

• **Halt On Cycles After Trigger** – The trigger point will not generate a hardware breakpoint until the Cycles Traced After Trigger is reached. Then the processor will be halted. To capture traces without halting the target, unselect this option.

• **Halt On Trace Buffer Full** – A trigger will cause the trace buffer to stop before overwriting old data. When the trace buffer is full (64K-1 or 65535 entries), a hardware breakpoint will halt the processor.

> **Note:**   The number of downloaded lines specified in the Configure Trace tab does not affect the number of cycles collected, and Trigger Position still applies.

• **Ignore FNOP Cycles** – Specifies whether or not forced NOP cycles are to be considered in the event processing. A forced NOP cycle is the second cycle of a two-cycle instruction. Since the PICmicro® MCU architecture is pipelined, it prefetches the next instruction in the physical address space while it is executing the current instruction. However, if the current instruction changes the program counter, this prefetched instruction is explicitly ignored, causing a forced NOP cycle.

Suppose a project has the following source code:

```
RoutineA
  <code for RoutineA>
  RETLW 0
RoutineB
  <code for RoutineB>
  RETLW 0
```

When the `RETLW` statement of `RoutineA` is executed, a prefetch of the next instruction in the address space (the first instruction in `RoutineB`) is performed.

This prefetched instruction will not be executed, but the program memory address does appear on the bus. If a trigger is set at program memory address `RoutineB`, the prefetch done during the execution of the `RETLW` in `RoutineA` will cause the trigger to fire. To prevent this, check the Ignore FNOP Cycles check box. Two points to consider when using this check box are:

Depending on the processor module, the trigger may skid two additional cycles.

- **Clear All** – Clears the current trigger information in all tabs.
- **Load All** – Opens the Load All Trigger Definitions dialog, allowing you to load a `*.trg` file with trigger information for all tabs of the dialog.
- **Save All** – Opens the Save All Trigger Definitions dialog, allowing you to save as a `*.trg` file trigger information for all tabs of the dialog.

For information on memory access selection, see **Section 6.7 "Memory Selection"**. Additional memory information that you may enter is:

- **Address** (Optional) – A single Event may specify one or more addresses. This can be either a Program Memory or Data Memory address.
- **Opcode** or **Value** (Optional) – The actual value of an opcode, the data for a table read/write operation, or the value of a file register. Also select whether the opcode/value is expressed as Symbolic, Binary or Hex(adecimal).

Other triggering information that you may enter is:

- **Probes** (Optional) – A value on the external logic probe inputs. Also select whether the value is expressed as Binary or Hex(adecimal).
- **Pass Counter** or **Captured Events** – A Pass Counter counts the number of times the event must occur before proceeding to the next event. Pass Counters are available only with Sequential or Time Between Events triggers. A Captured Event counts the number of times a captured event must occur before proceeding to the next event. You may select an infinite number of events with a check box. Captured Events are available only with the Filter Trace trigger.
- **Clear Event** – Clears the current trigger information in the active tab.
- **Load Event** – Opens the Load Current Trigger Level dialog, allowing you to load a `*.evt` file with trigger information for the active tab of the dialog.

> **Note:** You may also load an MPLAB ICE 2000 file (`*.trl`), but you can only save it as an MPLAB ICE 4000 file (`*.evt`).

- **Save Event** – Opens the Save Current Trigger Level dialog, allowing you to save as a `*.evt` file trigger information for the active tab of the dialog.

There are several buttons on the Complex Trigger Settings tab with the following functions.

- **OK** – Accepts the current setting in the tab and closes the dialog.
- **Cancel** – Closes the dialog without accepting the current settings.
- **Apply** – Accepts the current setting in the tab without closing the dialog.
- **Help** – Brings up the on-line help file to walk you through setting up a complex trigger.

6.5 COMPLEX TRIGGER SETTINGS SYNTAX

The following types of address specifications can be entered in one event level:

TABLE 6-1: COMPLEX TRIGGER ADDRESS SYNTAX

Description	Syntax	Example
Individual address	`<addr>`	`Delay`
Address plus/ minus offset	`<addr>+<offset>` `<addr>-<offset>`	`Delay+5` `DelayEnd-2`
Address range	`<addr1>:<addr2>`	`Delay:EndDelay` `Delay:Delay+5`

Two or more individual addresses or ranges	`<addr1>;<addr2>`
	`<range1>;<range2>`
		`Delay; EndDelay`
		`StartA:EndA;`
		`StartB:EndB`

Everything outside an address range(s)	`!(<addr1>:<addr2>)`	`!(Delay:EndDelay)` `!(Main:Main + 40)`

Each specified address can be either an absolute (numeric) address or a defined symbol. Absolute addresses must be entered in hex, with a leading numeric digit.

The value/opcode field may be entered two ways. If Symbolic is selected, either hex constants or symbols can be used as described by the complex trigger address syntax above. If hex or binary is selected, a single constant value, with or without "don't cares," can be entered. A "don't care" state may be specified using either 'x' or 'X'. If the radix is binary, the "don't care" applies to a single bit. If the radix is hex, the "don't care" applies to four bits.

The logic probe input value may be entered in either binary or hex, just like the value field.

Pass counters, number of captured events and trigger position values are all entered in decimal.

TABLE 6-2: COMPLEX TRIGGER SYNTAX EXAMPLES

To Specify	Select Event Type	Place	In Field	Value Type
Program Memory Address 0xAE26	Program Memory	`0AE26`	Address	—
Data Memory Address 21	Data Memory	`21`	Address	—
All 14-Bit RETLW instructions	Program Memory	`34xx`	Value	Hex
All 14-Bit RETLW instructions	Program Memory	`3400:34FF`	Value	Symbolic
Bit 4 high, bit 0 low	Data Memory	`xxx1xxx0`	Value	Binary

6.6 TRIGGER TYPE SELECTION

Up to four events can be required before the trace or break occurs. The combination of these events can be specified three ways:

• Sequential
• All Events
• Any Event

In addition, the complex triggering feature along with the trace memory window can be used to perform the following functions:

• Filter Trace
• Time Between Events

6.6.1 Sequential

Figure 6-2 shows the Trigger subtab of the Complex Trigger Settings tab for Sequential event selection.

1. Select Sequential if each event must occur in sequence to generate the trigger.
2. Click on a tab to enter information about each event (**Event 1**, **Event 2**, **Event 3**) and the trigger event (**Trigger**).

FIGURE 6-2: COMPLEX TRIGGER – SEQUENTIAL EVENT SELECTION
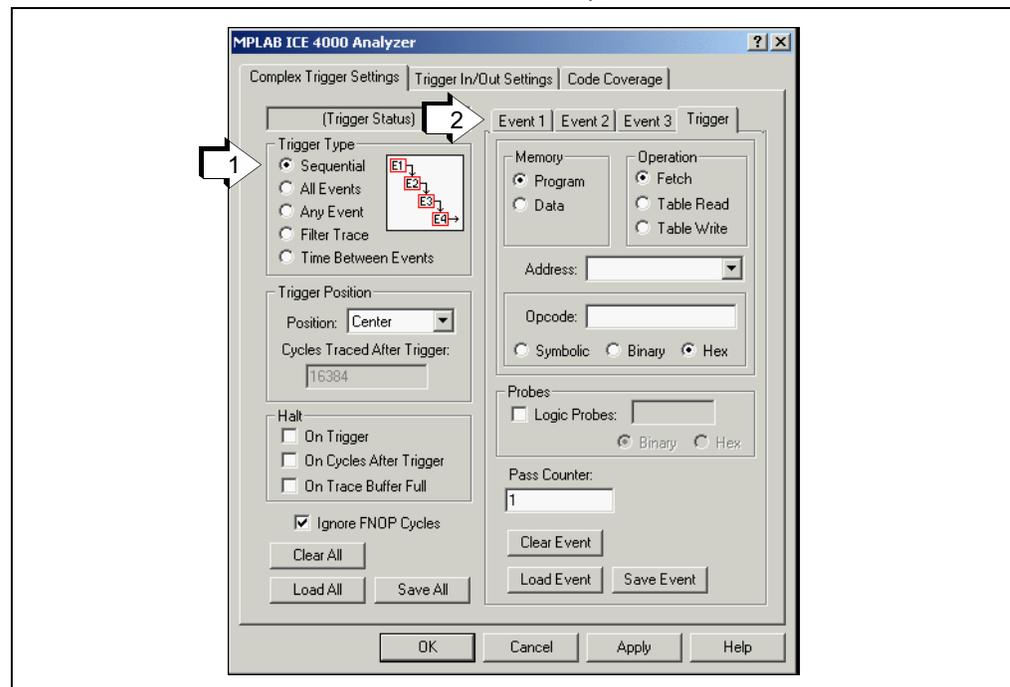

Keep in mind that for sequential triggers, the emulator will wait until **Event 1** is satisfied before proceeding to **Event 2**, and on down to the final **Trigger** event. If only one trigger event is specified and it is placed in **Event 1**, extra instruction cycles will be executed while the emulator determines that **Event 2, Event 3** and the **Trigger** event have been satisfied. Therefore, always right-justify the trigger specification; that is, use the right-most event tabs first.

6.6.2   All Events

Figure 6-3 shows an Event subtab of the Complex Trigger Settings tab for All Events selection.

1. Select All Events if each event must occur, in any order, to generate the trigger. The trigger will be generated when the final event occurs.
2. Click on a tab to enter information about each event (**Event 1**, **Event 2**, **Event 3**, **Event 4**).

**FIGURE 6-3:   COMPLEX TRIGGER – ALL EVENTS SELECTION**
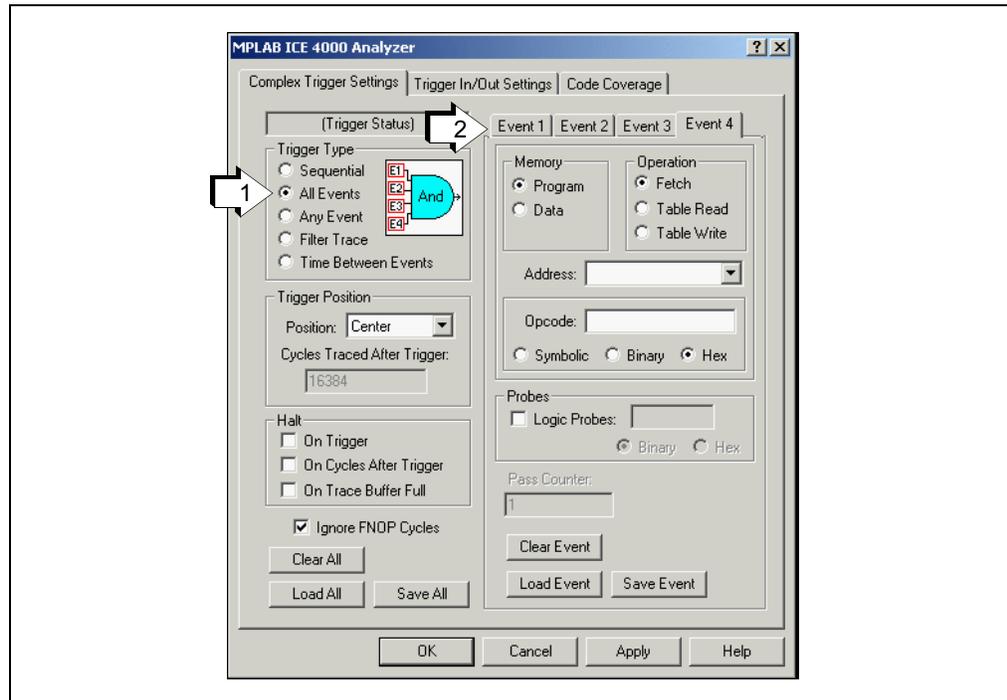
Pass Counter does not apply (grayed out) for this event selection as event timing is not necessary for triggering.

6.6.3 Any Event

Figure 6-4 shows an Event subtab of the Complex Trigger Settings tab for Any Event selection.

1. Select Any Event if any single event will generate the trigger.
2. Click on a tab to enter information about each event (**Event 1**, **Event 2**, **Event 3**, **Event 4**).

**FIGURE 6-4:   COMPLEX TRIGGER – ANY EVENT SELECTION**
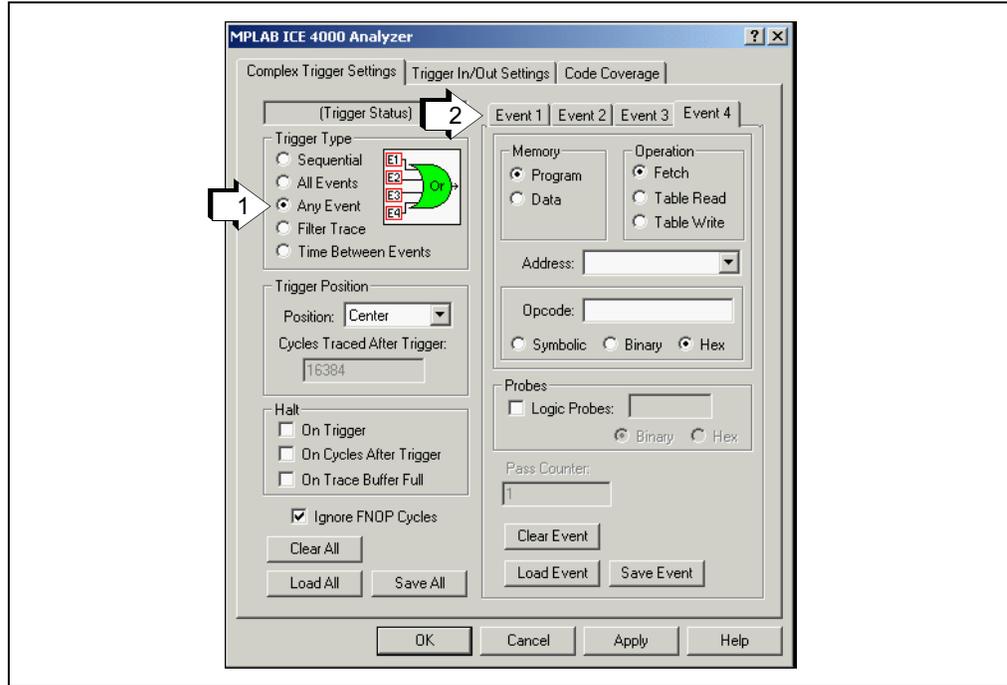
Pass Counter does not apply (grayed out) for this event selection as event timing is not necessary for triggering.

6.6.4 Filter Trace

Figure 6-5 shows the Filter subtab of the Complex Trigger Settings tab for Time Between Events selection. This event selection is used in conjunction with the trace memory window.

1. Select Filter Trace to specify the events that will be captured by the trace memory window.
2. To select sequential events to proceed the trace, click on a tab to enter information about an event (**Event 1**, **Event 2**, **Event 3**).
3. To perform a continuous trace, click on Infinite Events (**Filter**).

FIGURE 6-5:   COMPLEX TRIGGER – FILTER TRACE SELECTION
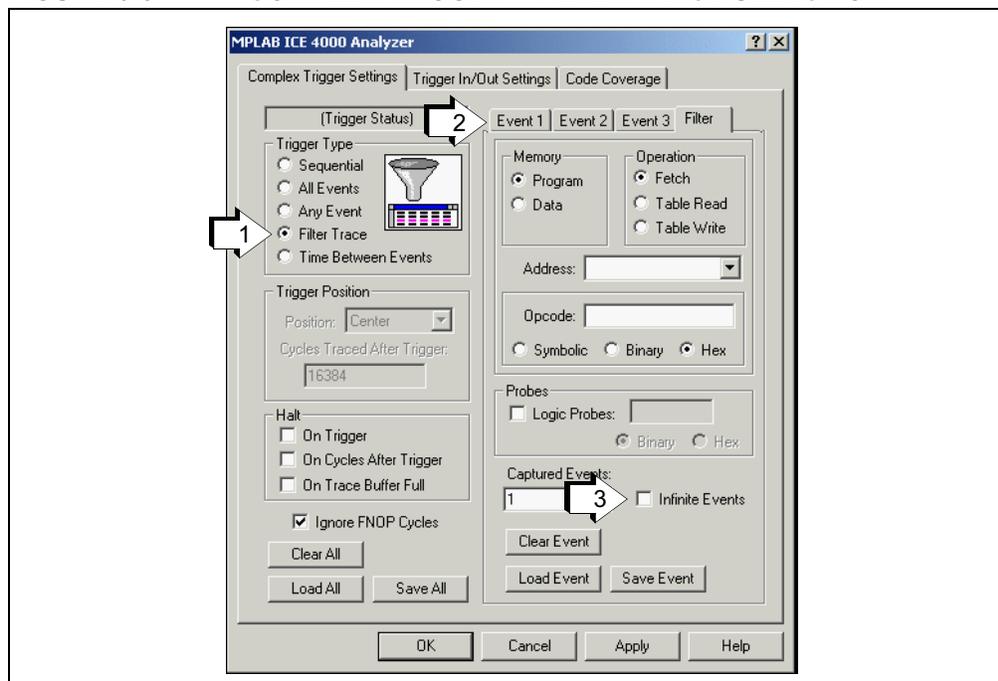
Filter Trace is used to capture only the execution of certain events in the trace buffer. Filtered traces allow you to make more efficient use of trace memory. You can focus in on sections of code and collect only those, or you can select to not collect delay loops or other repeated sections of code that are not of interest

Up to three qualifying events can be specified to occur sequentially before starting to collect the specified events. The Pass Count field on trigger event is changed to Captured Events, indicating how many of these events to record.

To perform a continuous filter trace, check the Infinite Events check box. Infinite Events is a special condition of capturing a filtered trace. As the trace buffer gets full, old data is discarded and new data is read First In, First Out (FIFO). This is useful in some special cases;

1. Capturing filtered events up until a user halt. You may wish to see the data collected up until the system was manually halted.

2. Use "Reload Data" as you are filtering events to get a trace display of the current trace buffer. Continue to select "Reload Data" to upload another capture of the current data being filtered. This is useful if you are monitoring a system in real time and just want to see what's going on without halting the system.

3. Sometimes a system misbehaves after a long period of time. Infinite events will continue collecting the data in your critical routine until the application is halted. Often the misbehaving state will get into a place in the program where no events will be collected. The trace buffer will then be a record of the data of interest before the "glitch."

4. Filtered traces can output a series of triggers on the TRGOUT connector. If you wanted to trigger an external logic analyzer every time a particular routine is executed, this can be achieved by filtering on that address and setting the appropriate triggering in the Trigger In/Out tab. Infinite Events, in this case, will allow continuous triggers to be generated on the TRGOUT connector.

Trigger Position does not apply when filtering, as is Halt on Trigger and Halt on Trace Buffer Full.

Ignore FNOP Cycles should not be specified when performing a Filter Trace; otherwise, the trace buffer will capture the cycle that executes after the cycle that caused the trigger.

> **Note:** Since Ignore FNOP Cycles should not be specified with a Filter Trace, prefetches will cause the trigger to fire.

Also, performing a Filter Trace on Data Memory access is not recommended, since the traced data will be one cycle after the trigger specification.

> **Note:** When triggering on multiple events, keep in mind that triggers on data memory accesses are skewed two cycles from the instructions that caused them.

6.6.5   Time Between Events

Figure 6-6 shows the Start Timer subtab of the Complex Trigger Settings tab for Time Between Events selection. This event selection is used in conjunction with the trace memory window.

1. Select Time Between Events to specify a starting and terminating event.
2. Click on a tab to enter information about:
   - sequential events to proceed Start Timer (**Event 1**, **Event 2**)
   - events for starting and stopping the timer (**Start Timer**, **Stop Timer**).

FIGURE 6-6:   COMPLEX TRIGGER – TIME BETWEEN EVENTS SELECTION
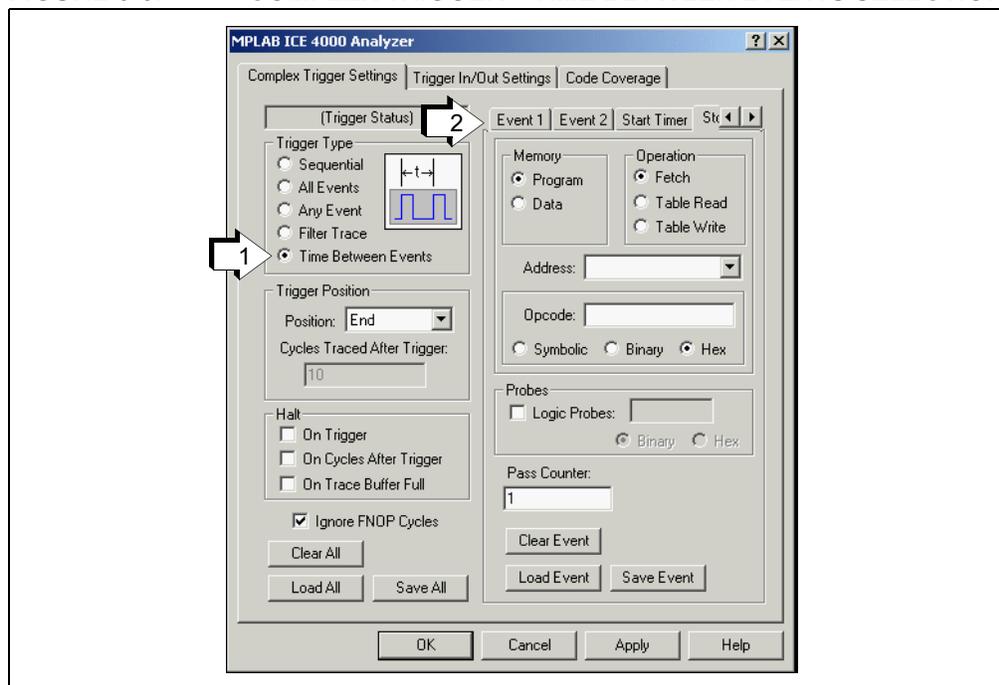
For the trigger type Time Between Events, the time stamp generator is held at zero until a specified starting event occurs. The time stamp generator then continues to increment normally until a specified stopping event occurs. The time stamp can then be used to measure the lapsed time.

Up to two events and the start event can be specified to occur sequentially before the time stamp generator starts incrementing. The trace memory display time stamp will start incrementing when the starting event occurs and will stop when the terminating event occurs. The time between the events can be determined by examining the time stamp in the trace memory window.

6.7   MEMORY SELECTION

Certain items on the subtabs of the Complex Trigger Settings tab will be different depending on the memory selection.

6.7.1   Program Memory

If Program Memory is specified, the trigger can be generated on an instruction fetch or a table read/table write operation. The item following Address will be Opcode for Program Memory selection.

> **Note:** You must proceed a hex address that begins with an alphabetic character with a zero, e.g., instead of entering EF, enter 0EF. Otherwise, the address will be assumed to be a label.

FIGURE 6-7:   COMPLEX TRIGGER – PROGRAM MEMORY SELECTION

6.7.2   Data Memory

If Data Memory is specified, the trigger can be set on a read or a write operation to file registers or special function registers. This is called data monitoring. The item following Address will be Value for Data Memory selection.

> **Note:** You must proceed a hex address that begins with an alphabetic character with a zero, e.g., instead of entering EF, enter 0EF. Otherwise, the address will be assumed to be a label.

FIGURE 6-8:   COMPLEX TRIGGER – DATA MEMORY SELECTION
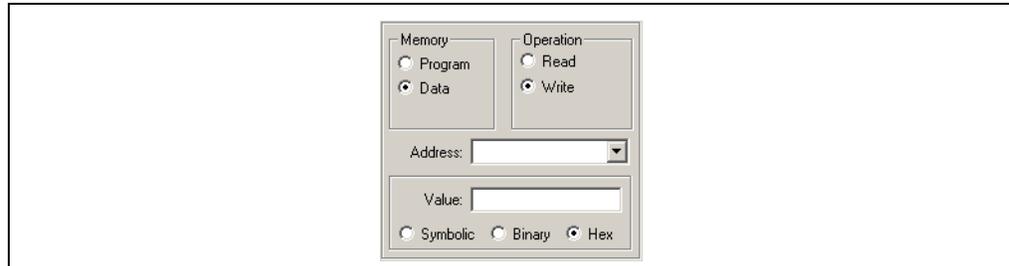
6.8   COMPLEX TRIGGERING EXAMPLES

The following examples show some of the ways that complex triggers can be used to help debug or characterize a problem.

• Sequential Example – Program Memory
• Sequential Example – Data Memory
• Time Between Events Example
• Filter Trace Example – Program Memory
• Filter Trace Example – Data Memory

6.8.1   Sequential Example – Program Memory

An application has several subroutines. A particular subroutine (RoutineA) functions correctly to begin with, but after a time, it begins to function incorrectly. This subroutine is called many times, so it would be nice to skip the executions where the subroutine functions properly and breaks just before the subroutine starts to fail. It is observed that the routine functions correctly until another after subroutine (RoutineB) is called.

This problem can be solved by a Sequential trigger with two events. The first event that must occur is the execution of `RoutineB`, so the **Event 3** tab is specified with the fetch of program memory address `RoutineB`. The trigger should fire when `RoutineA` is called, so the **Trigger** tab is specified with the fetch of program memory address `RoutineA`. The Ignore FNOP Cycles is left checked so prefetches are ignored, and Halt On Trigger is checked so the improperly executing subroutine can be stepped.

FIGURE 6-9:   SETTING THE FIRST OF TWO SEQUENTIAL EVENTS
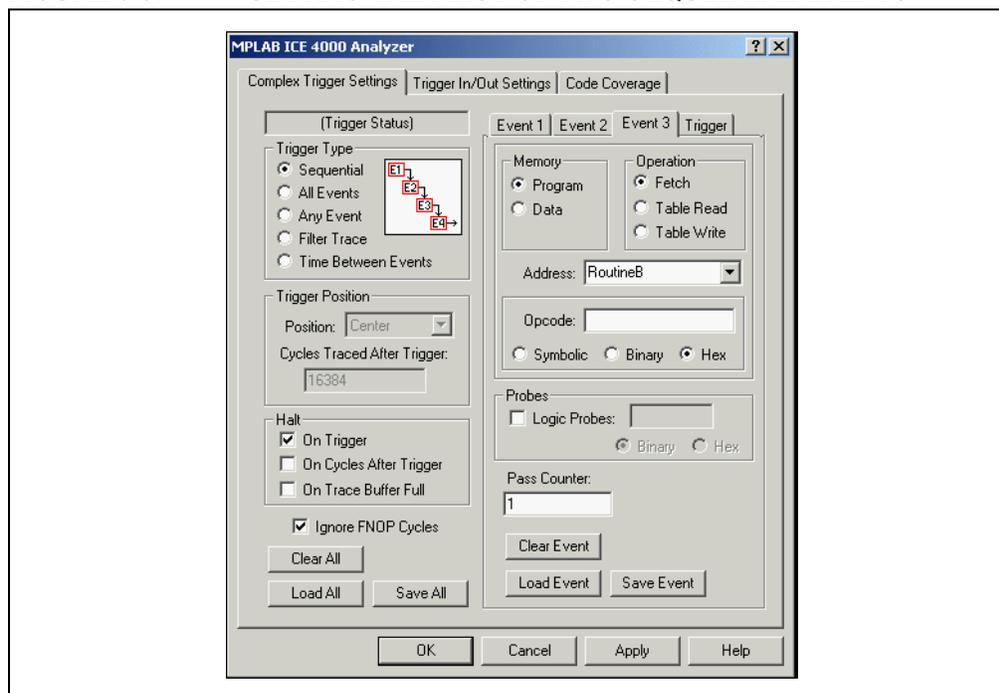
FIGURE 6-10:   SETTING THE SECOND OF TWO SEQUENTIAL EVENTS
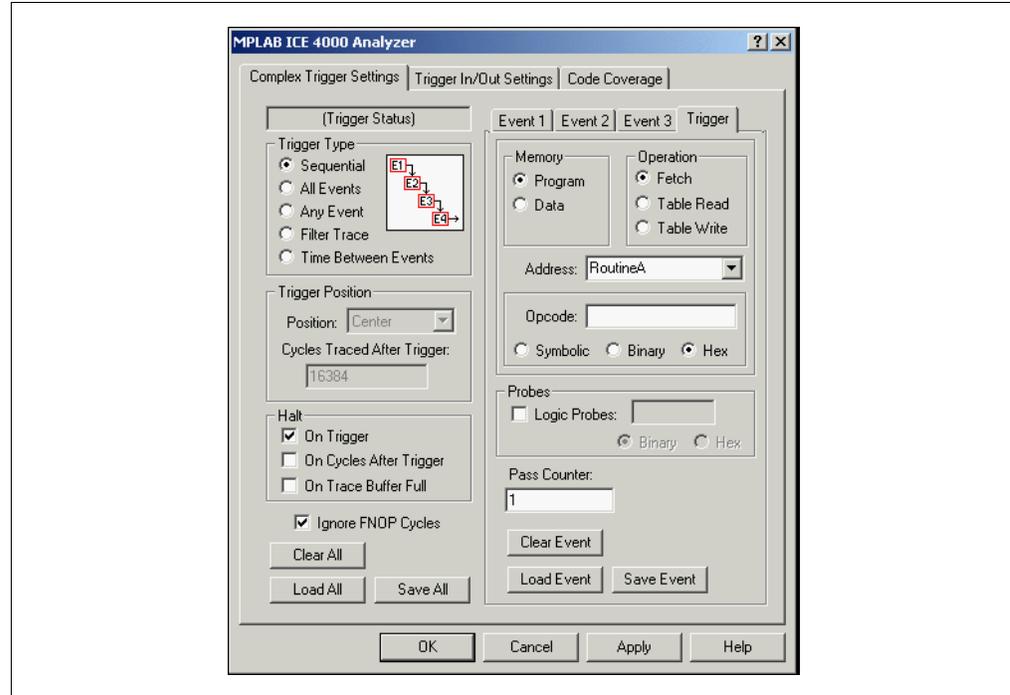
6.8.2   Sequential Example – Data Memory

A flag bit is getting erroneously set somewhere. Where is it getting set?

A trigger can also be set on data memory. Suppose that the flag bit is in RAM location `Flags`, at bit position 3. It does not matter what the value of the other bits are, but the trigger should occur when bit 3 of `Flags` becomes 1. For this, use a Sequential trigger on a Write-to-Data Memory. Specify the Address as `Flags` and the Value as `xxxx1xxx` in binary. Uncheck Ignore FNOP Cycles and check Halt-On-Trigger if desired.

FIGURE 6-11:   SEQUENTIAL EVENT – TRIGGERING ON DATA MEMORY
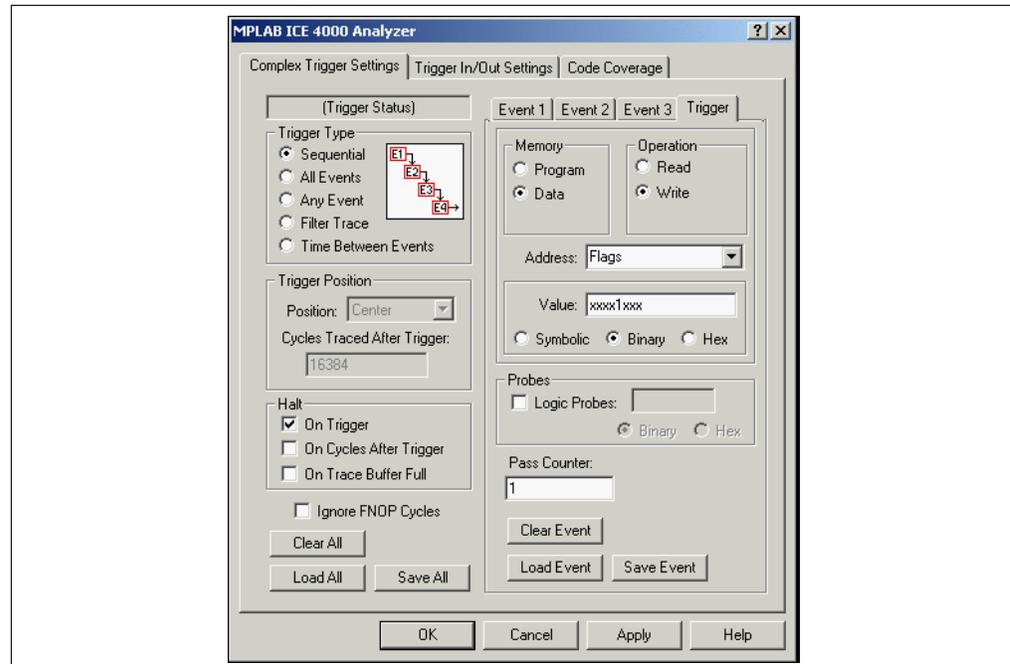
6.8.3   Time Between Events Example

An application contains a delay loop. How long is the delay?

For this problem, a Time Between Events trigger is very useful. Suppose the delay routine is called `Delay`. For debugging purposes, a label `EndDelay` has been added at the address of the delay routine's return statement. Set the **Start Timer** tab to a program memory fetch of address `Delay`, and the **Stop Timer** tab to a program memory fetch of address `EndDelay`. Check the Ignore FNOP Cycles box, and leave both halt check boxes unchecked. Apply the trigger, open the trace memory window, then run. After `EndDelay` is reached, the trace memory window will fill and the largest time stamp value will be the time between the two events.

FIGURE 6-12:   SPECIFYING THE EVENT THAT WILL START THE TIMER
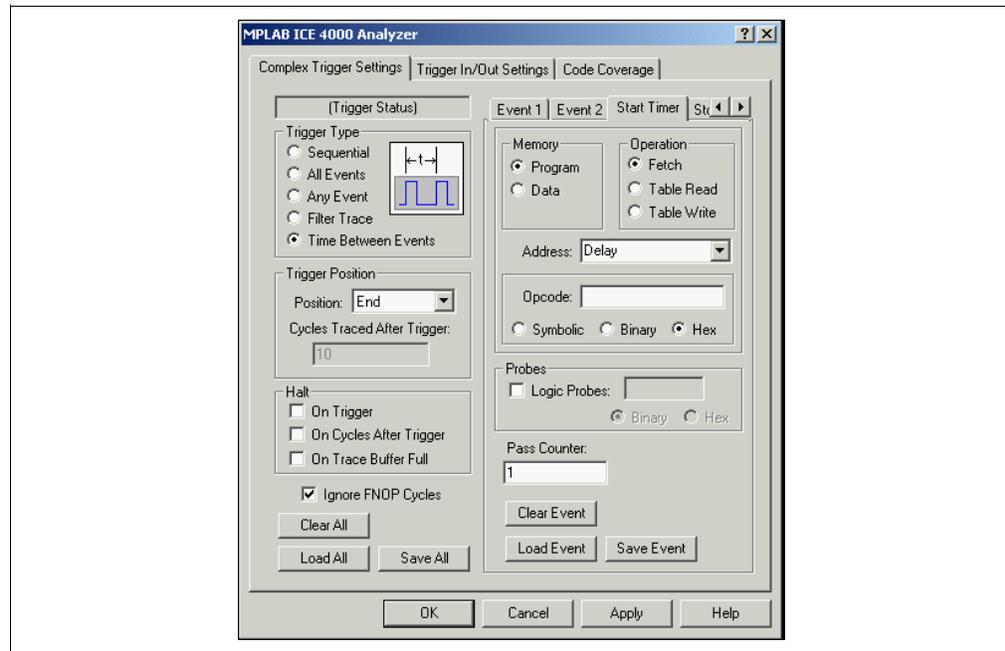
FIGURE 6-13:   SPECIFYING THE EVENT THAT WILL STOP THE TIMER
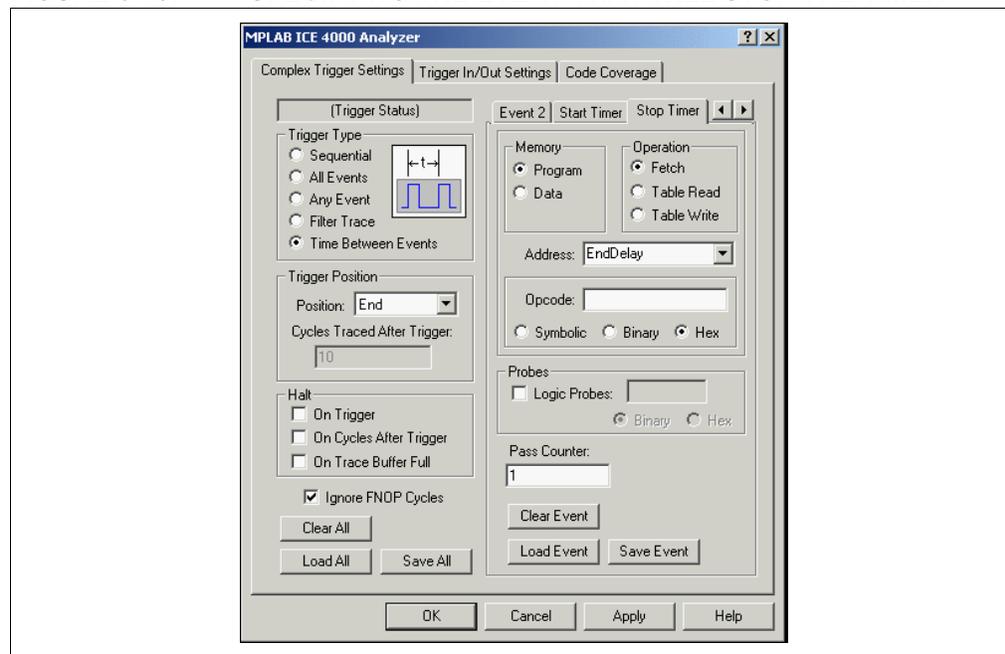
6.8.4 Filter Trace Example – Program Memory

A program has a large delay loop. How can program execution be traced without wading though thousands of delay loop cycles?

For this, use a Filter Trace. Specify the traced addresses to be everything except the program memory range where the delay loop is located. Uncheck Ignore FNOP Cycles. Check Infinite Events and use either a software breakpoint or a user halt to stop emulation or uncheck Infinite Events and enter a value for Captured Events. When viewing the trace memory window, all instances of the delay code should be gone.

FIGURE 6-14:   FILTER TRACE
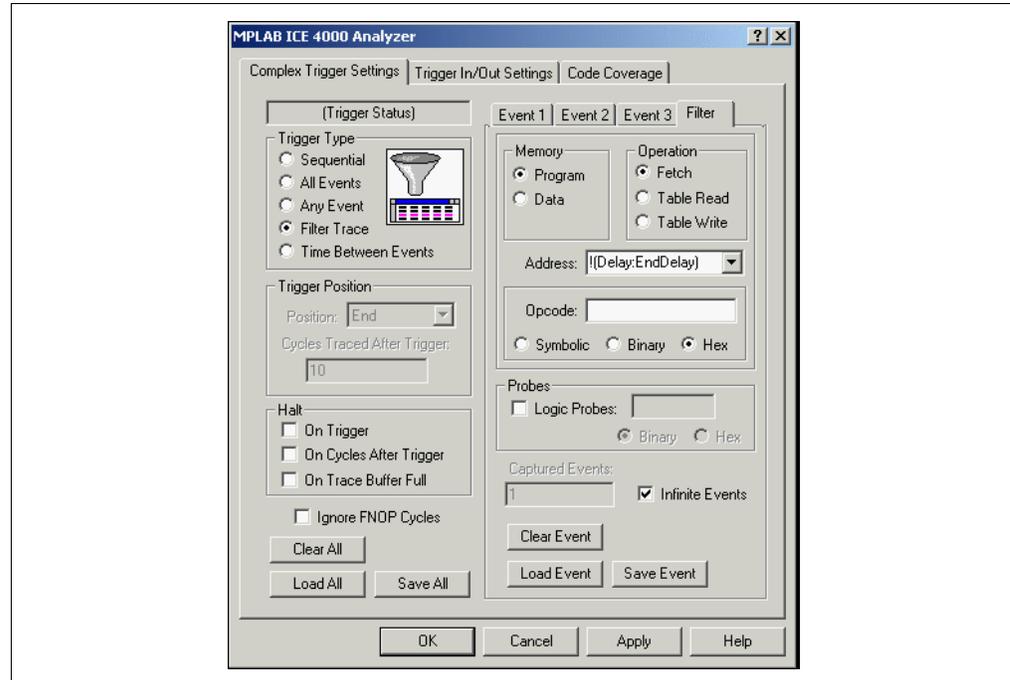
6.8.5 Filter Trace Example – Data Memory

How do I filter on data memory?

Filtering on PICmicro file register reads and writes will result in the collection of the cycles immediately following the R/W. While this is not exactly the information you are looking for, it does provide two data points:

1. The program memory after the R/W occurred, so you can track down places in your code that affect a particular variable or SFR.
2. The number of times the R/W occurred, providing you with a count of the collected cycles.

There are two ways to capture the actual data values with a filter. The first is to filter on the routine that does the R/W. This will generate some extra cycles, but will collect the proper information. The second is to make code that will generate two R/W cycles to the same location, one of which does not affect the actual data. For instance, the register can be ANDed with a value of 0xFF to generate the extra cycle like this:

```
test
    movlw 0xFF
    movwf LSDigit
test1
    decfsz LSDigit
    andwf LSDigit; generate extra read/w to LSDigit
    goto test1
```

MPLAB® ICE 4000 User's Guide

6.9   INTERNAL TRIGGERS

dsPIC processor modules have additional circuitry so that internal triggers may be set in addition to external ones (complex triggers.) dsPIC internal triggers may be set up using the Internal Triggers tab on the MPLAB ICE 4000 Analyzer dialog.

Define Event A, B, C and/or D under "Breakpoint Event Criteria". Then set up event triggering by clicking on either the interactive diagram or the check boxes.

FIGURE 6-15:   INTERNAL TRIGGERS TAB
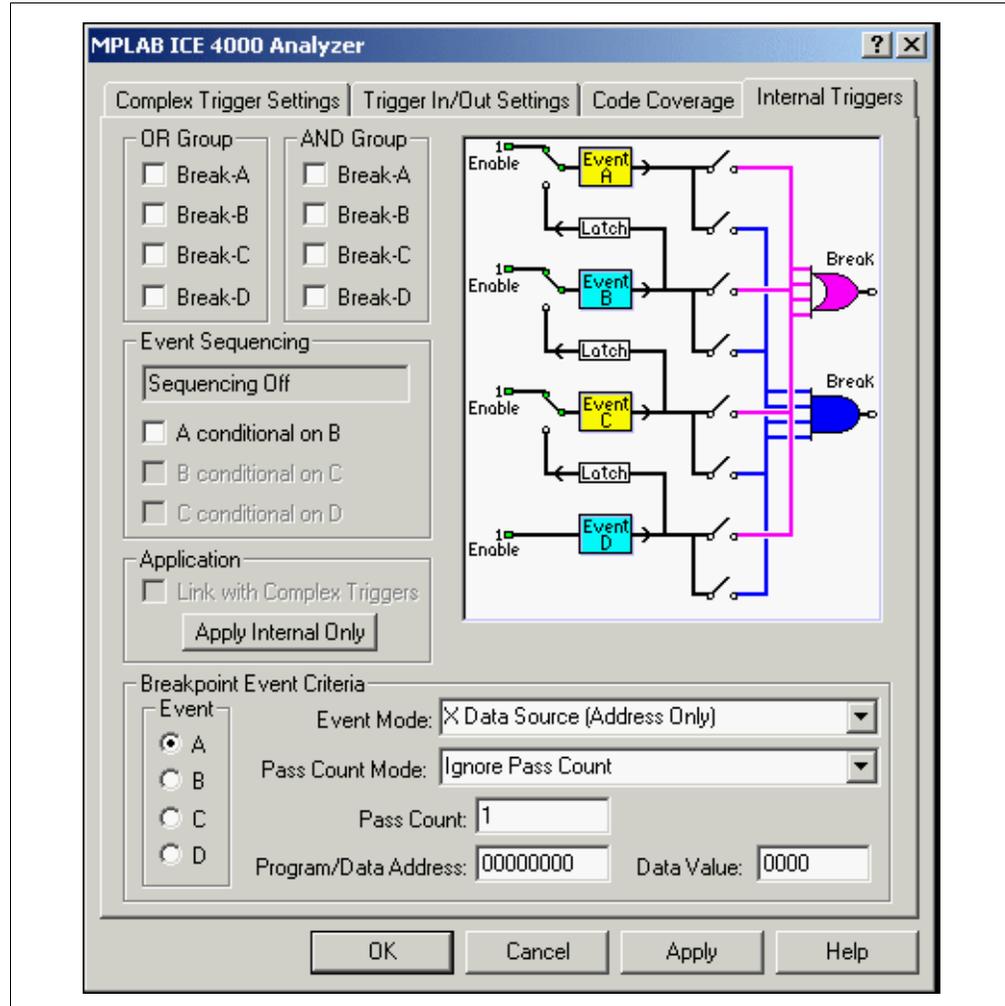
TABLE 6-3:   INTERNAL TRIGGERS TAB

Interactive Diagram	Click the switches in the diagram to set/clear OR group, AND group and Event Sequencing functions.
OR Group	Enable/disable break on event A or B or C or D.
AND Group	Enable/disable break on event A and B and C and D.
Event Sequencing	Enable/disable conditional breaks for events A through D.
Application	Link with Complex Triggers: **Currently unimplemented feature** - Select/unselect to link internal triggers with complex triggers. Apply Internal Only: - Applies internal trigger settings only. To apply all tab settings, click Apply on the bottom of the dialog.
Breakpoint Event Criteria	
Event	Select an event to set up (events A though D.)
Event Mode	Select the event mode, either: - X Data Source (Address Only) - X Data Destination (Address Only) - Y Data Source (Address Only) - X Data Source (Address and Word Data) - X Data Destination (Address and Word Data) - Y Data Source (Address and Word Data) - X Data Source (Address and Byte Data) - X Data Destination (Address and Byte Data) - Program Address (Address Only) - Table Read (Address Only) - Table Write (Address Only) - PSV (X) Data Source (Address Only) - PSV (X) Data Destination (Address Only) See the device data sheet for more information on memory organization (i.e., program and data address space.)
Pass Count Mode	**Currently unimplemented feature** Select a Pass Count mode, either: - Ignore Pass Count - Observe Pass Count - Bus Cycle Delay Count After Event
Pass Count	Enter a pass count value.
Program/Data Address	Enter a program or data address
Data Value	Enter a data value, if needed.

NOTES:

Chapter 7. Code Coverage, Trace Memory, Real-Time Reads

7.1 INTRODUCTION

For MPLAB ICE 4000, a mechanism to determine what portions of the code are being accessed (fetched, written or read) is available. Code coverage may be set using the Code Coverage tab on the MPLAB ICE 4000 Analyzer dialog.

Using trace, executed code may be captured into a trace buffer and displayed in the trace memory window (*View>ICE Trace*.)

Using real-time reads, data may be captured real-time and displayed in several windows.

7.2 HIGHLIGHTS

This chapter covers:

• Code Coverage
• Trace Memory
• Real-Time Reads

7.3 CODE COVERAGE

The code coverage feature provides visibility as to what portions of the code are being accessed (fetched, written or read). This code is checkmarked in the Program Memory window.

Code coverage differs from traced code in the following way:

• Trace displays code that has been executed and tells when it was executed.
• Code coverage marks code that has been prefetched, but does not display when it was prefetched.

> **Note:** Code coverage tags prefetched code, regardless of whether or not it actually gets executed. Therefore, not all checkmarked code may have been executed. See **Section 7.3.2 "Tracking Code"**.

This feature works by latching addresses as they appear on the bus. Thus, instructions that follow two-cycle instructions that modify the program counter may not have been actually executed.

With code coverage enabled, the next halt encountered by the emulator (software breakpoint or halt command) will cause ROM locations that have been fetched to be checkmarked in the Program Memory window.

Addresses transferred as a result of a table read (TBLRD) or table write (TBLWR) will be traced using this method.

To enable code coverage, select *Debugger>Complex Triggers and Code Coverage*, **Code Coverage** tab.

FIGURE 7-1:   MPLAB ICE 4000 ANALYZER – CODE COVERAGE TAB
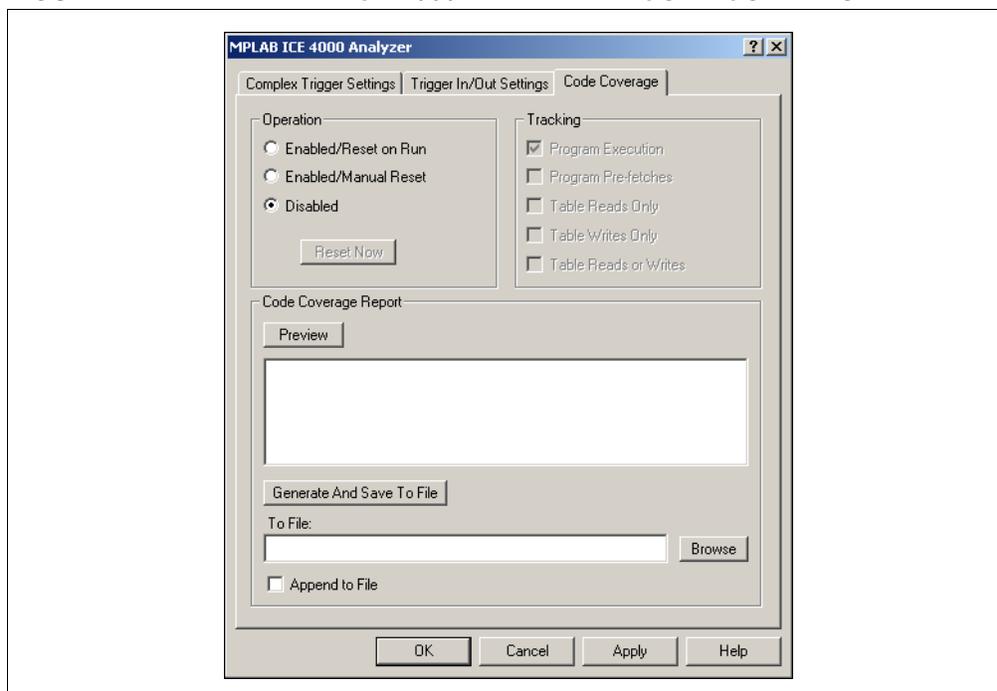
7.3.1   Enabling/Disabling Code Coverage

If Enabled/Reset on Run is selected, code coverage is reset every time emulation is started. If Enabled/Manual Reset is selected, code coverage is reset only when **Reset Now** is clicked.

When a reset has occurred, and emulation has run and then halted, all accessed program memory locations appear checkmarked in the Program Memory window. They will remain checkmarked until the next reset. Single stepping does not affect code coverage.

To disable code coverage, select *Debugger>Complex Triggers and Code Coverage*, **Code Coverage** tab, and click on Disabled.

7.3.2   Tracking Code

Determine the type of code that is covered by using the check boxes in the "Tracking" section of the dialog.

• Program Execution – include program code that has executed
• Program Prefetches – include program code that has been prefetched
• Table Reads Only – include only table reads, not writes
• Table Writes Only – include only table writes, not reads
• Table Reads or Writes – include both table reads and writes

7.3.3   Creating a Code Coverage Report

Code coverage information may be saved into a file using the controls in the "Code Coverage Report" section of the dialog tab.

1. Preview the code coverage information by clicking Preview to determine if it is what you want to save.
2. Specify a file to save the information to under "To File:".
   - For a new file, type in the file name and path or browse to a file location. When you have reached the location at which you want the new file, type in the file name and click **Save**.
   - For an existing file, type in the file name and path or browse to a file and location. To append this data to old data in the file, instead of overwriting it, check the "Append to File" check box.
3. Click **Generate and Save to File**.

7.4   TRACE MEMORY

The trace memory window in MPLAB ICE 4000 contains information that is uploaded from MPLAB ICE 4000's trace buffer. The trace buffer consists of memory hardware that can log the electrical state of the various device data, address and control signals at an instruction cycle in real time as the device executes instructions. By default, all instruction cycles are captured by the trace buffer. The Complex Trigger tab can be used to control which instruction cycles are captured (**Chapter 6. "Complex and Internal Triggers"**).

In MPLAB ICE 4000, the trace buffer captures data on a 216-bit wide analyzer that is connected to the emulator chip and other devices in the emulator.

FIGURE 7-2:   TRACE BUFFER INPUT – MPLAB ICE 4000
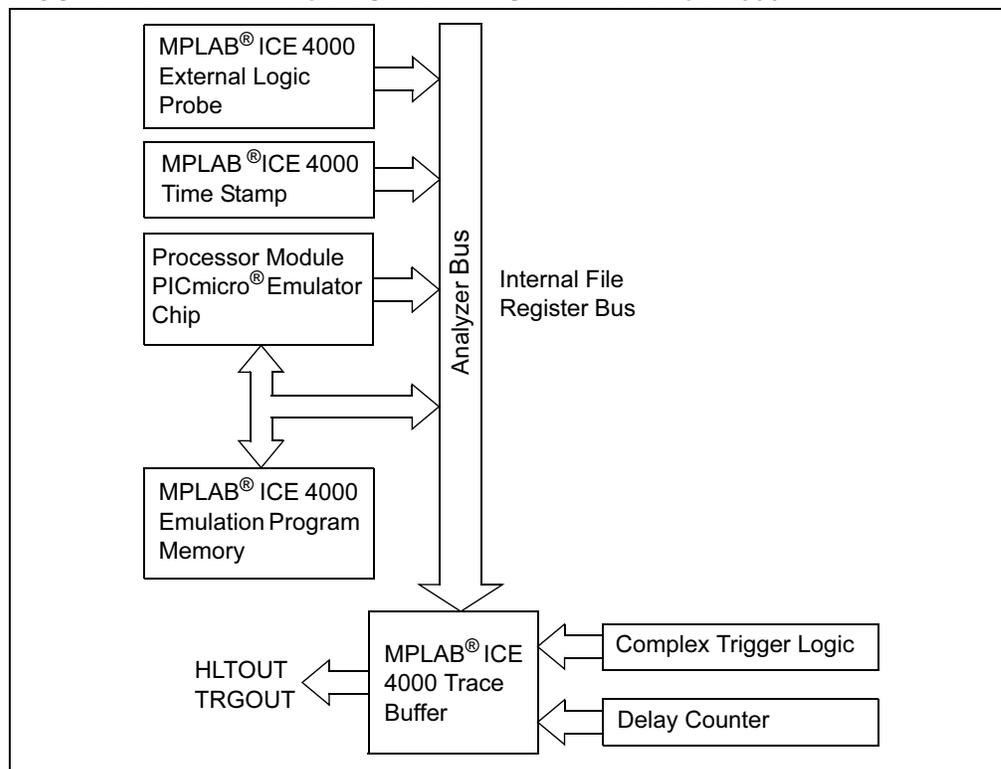
7.4.1 The Trace Memory Window

This trace buffer can be viewed by selecting *View>ICE Trace*.

FIGURE 7-3:   **TRACE MEMORY WINDOW – MPLAB ICE 4000, PIC18F452**

Line	Addr	Op	Label	Instruction	SA	SD	DA	DD	Cycles	Probe 7	Probe 6	Probe 5	Probe 4	Probe 3	Probe 2	Probe 1	Probe 0
-2	0038	2E01	D1	DECFSZ 0x1, 0x	----	--	----	--	000000D1D003	1	1	0	1	1	1	1	1
-1	003A	EF1C		GOTO 0x38	0001	4B	0001	4A	000000D1D004	1	1	0	1	1	1	1	1
0	003C	F000		(Forced NOP)	----	--	----	--	000000D1D005	1	1	0	1	1	1	1	1
1	0038	2E01	D1	DECFSZ 0x1, 0x	----	--	----	--	000000D1D006	1	1	0	1	1	1	1	1
2	003A	EF1C		GOTO 0x38	0001	4A	0001	49	000000D1D007	1	1	0	1	1	1	1	1
3	003C	F000		(Forced NOP)	----	--	----	--	000000D1D008	1	1	0	1	1	1	1	1
4	0038	2E01	D1	DECFSZ 0x1, 0x	----	--	----	--	000000D1D009	1	1	0	1	1	1	1	1
5	003A	EF1C		GOTO 0x38	0001	49	0001	48	000000D1D00A	1	1	0	1	1	1	1	1

> **Note:** Due to the timing of processor signals, the data information will be skewed by one cycle. Destination data values are actually skewed by two cycles, but for display purposes, the trace buffer display compensates for one of the delayed cycles.

MPLAB ICE 4000 offers a trace memory window that monitors much of the processor operation. Up to 64K-1 (65535) instruction cycles can be displayed. The trace memory window contains the following information for each execution cycle:

- **Line** – Cycle's position relative to the trigger or halting point.
- Address (**Addr**) – Address of the instruction being fetched from program memory.
- Opcode (**Op**) – Instruction being fetched.
- Label (**Label**) – Label (if any) associated with the program memory address.
- **Instruction** – Disassembled instruction.
- Source Data Address (**SA**) – Address or symbol of the source data, if applicable.
- Source Data Value (**SD**) – Value of the source data, if applicable.
- Destination Data Address (**DA**) – Address or symbol of the destination data, if applicable.
- Destination Data Value (**DD**) – Value of the destination data, if applicable.
- **Cycles** – Time stamp value.
- **Probe** *n* – Value of the external input from logic probe *n*, for *n* = 0 through 7.

The approximate trigger cycle will be highlighted and will be numbered as line 0. All other lines will be numbered based on this cycle. Cycles that occurred before the trigger point will have a negative line number and cycles that occurred after the trigger point will have a positive line number. If the trace buffer was displayed after a user halt, a software breakpoint, or a complex trigger that halted the processor, the trigger point will be the last traced cycle. If a complex trigger filled the trace buffer without halting the processor, the trigger point will show the approximate cycle when the complex trigger fired.

7.4.2   The Trace Memory Window Menu

Right-click in the Trace menu to show a menu with the following functions:

- Close – Close the trace buffer window.
- Find – Open the Find Text dialog.
- Find Next – Find next occurrence of text in Find Text dialog.
- Go To – Move to the line corresponding to the cycle entered here.
- Jump to Trigger – Move to the line where the trigger occurred.
- Jump to Top – Move to the first line of the trace buffer.
- Jump to Bottom – Move to the last line of the trace buffer.
- Reload – Reload the trace information from the emulator.
- Reset Time Stamp – Specify when to reset time stamp, either on processor reset, on run or manually. Or, you may reset the time stamp immediately.
- Display Time – Specify how to display time, as cycle count, in seconds elapsed or in engineering format.
- Symbolic Disassembly – View the Instructions as symbolic disassembly.
- Output to File – Output the contents of the trace buffer to an ASCII text file.
- Print – Print the contents of the trace buffer to a file.
- Refresh – Reload the contents of the viewable portion of the window from the emulator.
- Properties – Set up window properties, such as columns, fonts and colors.

7.4.3   Customizing the Trace Memory Window

The column widths can be adjusted manually by dragging the right column header border with the mouse. In addition, right-clicking on a column header pops up a list which allows you to select which columns are visible in the window.

To further customize the trace display, open the Trace properties dialog.

- Right-click in the trace window and select Properties.
- Right-click on a trace window header and select More.

The Column Settings tab allows you to set which window columns are visible and in what order they will appear in the window. The General tab allows you to set the font and colors used in the window.

7.4.4 Reading the Trace Memory Window

Reading the information in the trace memory window requires knowledge of device architecture. PICmicro MCU instructions fetch one instruction cycle while decoding and executing the previous cycle, (i.e., there is a one cycle pipeline).

In the example below:

1. At line 2, the W register (address = 0xfe8) is cleared.
2. The next cycle, the value gets written into the W register, i.e., the W register value of 0xFF is changed to 0x00.

FIGURE 7-4:   TRACE MEMORY WINDOW EXAMPLE, PIC18F452

7.5   REAL-TIME READS

This feature consists of a real-time read buffer that records the data of all processor file register write operations. The address and data are obtained from the file register address and data bus. The write operation to the real-time read buffer occurs after data buffers have latched the information from the file register address and data bus.

The real-time display (either Watch, File Register or SFR window) gives a dynamic display of data, which is being continuously captured in real time. The values displayed are the values that are collected from the last read of the real-time read buffer. The time interval between real-time reads is set up under the *Debugger>Settings*, **View** tab.

> **Note:** A longer time interval than the default value will update the values at a much slower pace, making it easier to monitor values.

Chapter 8. Emulator Function Summary

8.1   INTRODUCTION

When you select MPLAB ICE 4000 from the Debugger menu, emulator functions will be added to MPLAB IDE. The functions made available are summarized here.

8.2   HIGHLIGHTS

MPLAB ICE 4000 functions are added to MPLAB IDE as follows:

Menus

- Debugger Menu
- View Menu
- Right Mouse Button Menu

Desktop

- Toolbars
- Status Bar

Additional Commands Dialog

- Additional Commands Dialog, Data Fill Tab
- Additional Commands Dialog, Force Opcode Tab

Settings Dialog, Standard Tabs

- Settings Dialog, Port Tab
- Settings Dialog, Info Tab
- Settings Dialog, Limitations Tab
- Settings Dialog, View Tab

Settings Dialog, Device-Dependant Tabs

- Settings Dialog, Clock Tab
- Settings Dialog, Power Tab
- Settings Dialog, Break Options Tab
- Settings Dialog, Memory Tab
- Settings Dialog, Pins/Pins and Usage Tab
- Settings Dialog, Peripheral Tab

Other Dialogs/Windows

- Other Dialogs/Windows

8.3   DEBUGGER MENU

The following items are added to the Debugger menu.

- Run

  Take the processor out of the halt state and put the processor into execution until a breakpoint is encountered or until Halt is pressed.

  Execution starts at the current program counter (as displayed in the status bar). The current program counter location is also represented as a pointer in the Program Memory window. While the processor is running, the Step and Run buttons are disabled.

- Animate

  Continually Step Into instructions. Halt using *Debugger>Halt*.

- Halt

  Force the processor into the halt state. When you click Halt, the processor is forced into a Halt state (Program Counter is stopped) and the processor status information is updated.

- Step Into

  Single step the processor. This command executes one processor instruction (single or multiple cycle instructions) and then puts the processor back into halt state. After execution of one instruction, all the windows are updated with the current state of the processor. While the processor runs in real time, MPLAB IDE ignores the Step button.

> **Note:** Be careful about the instructions you are stepping into. If you step into a Sleep instruction, you will need to select *Debugger>Reinitialize ICE Hardware* in order to wake up the processor module.

- Step Over

  Execute the instruction at the current program counter location. At a call instruction, Step Over executes the called subroutine and halts at the address following the call.

- Step Out

  Step out of current location in a function and return to main program.

- Reset

  Issue a reset sequence to the target processor, either Brown-Out Reset, Power-On Reset, or Processor Reset. These reset the Program Counter to the reset vector. If the processor is running it will continue running from the reset vector address.

> **Note:** Reset will not currently wake the processor if it is in Sleep mode. To wake the processor, you must use *Debugger>Reinitialize ICE Hardware*.

- Breakpoints

  Open the Breakpoint dialog (**Section 4.5 "Using Software Breakpoints"**). Set multiple software breakpoints in this dialog.

> **Note:** You may also right-click on a line of code to set a breakpoint.

- Stop Trace

  Halt the trace buffer (**Chapter 7. "Code Coverage, Trace Memory, Real-Time Reads"**) from capturing data but allow the processor to continue running.

- Complex Triggers and Code Coverage

  Opens the MPLAB ICE 4000 Analyzer Properties dialog. Here you may set up Complex Trigger Settings (**Chapter 6. "Complex and Internal Triggers"**), Trigger In/Out Settings (**Section 4.7 "Using Trigger In/Out Settings"**) and Code Coverage (**Section 7.3 "Code Coverage"**).
- Stopwatch

  Opens the complex stopwatch dialog (**Section 4.9 "Using the Stopwatch"**).
- Additional Commands

  Fill data memory as specified or force the execution of an opcode (**Section 8.8 "Additional Commands Dialog, Data Fill Tab"**).
- Upload Program Memory from ICE

  Used for monitoring the effects of self-modifying code. Transfer program memory in the MPLAB ICE 4000 to the MPLAB IDE Program Memory window.
- Reinitialize ICE Hardware

  Performs a system reset of the MPLAB ICE 4000.
- Settings

  Opens the MPLAB ICE 4000 Settings dialog (**Section 8.10 "Settings Dialog, Port Tab"**). Set up the port, clock, power, break options and pins. Also, find out information about the current system configuration and device limitations.

8.4   VIEW MENU

The following items are added to the View menu.

- ICE Trace

  Display the window containing the current trace of your program's execution (**Chapter 7. "Code Coverage, Trace Memory, Real-Time Reads"**).

8.5   RIGHT MOUSE BUTTON MENU

Some or all of following will appear on the right mouse menus in code displays, such as program memory and source code files.

- Set/Remove Breakpoint

  Set or remove a breakpoint at the currently selected line.
- Enable/Disable Breakpoint

  Enable or disable a breakpoint at the currently selected line.
- Breakpoints

  Remove, enable or disable all breakpoints.

For more on breakpoints, see **Section 4.5 "Using Software Breakpoints"**.

- Run To Cursor

  Run the program to the current cursor location. Formerly Run to Here.
- Set PC at Cursor

  Set the program counter (PC) to the cursor location.
- Center Debug Location

  Center the current PC line in the window.
- Cursor Tracks Debug Location

  Cursor (arrow) will track the current debug location.

8.6 TOOLBARS

The standard debug toolbar is added. Hover the mouse pointer over a toolbar button to see a pop-up of the function. See **Section 8.3 "Debugger Menu"** for definitions of functions.

Run	Halt	Animate	Step Into	Step Over	Step Out	Reset

An additional toolbar is available to the right of the standard toolbar with more emulator functions. Hover the mouse pointer over a toolbar button to see a pop-up of the function. Right-click on the toolbar to customize it.

8.7 STATUS BAR

The selected debug tool (MPLAB ICE 4000), as well as other development information, is displayed in the status bar on the bottom of the MPLAB ICE desktop. Refer to the MPLAB IDE on-line help for information on the contents of the status bar.

8.8 ADDITIONAL COMMANDS DIALOG, DATA FILL TAB

This tab allows you to fill data memory as specified below.

TABLE 8-1: FILL FILE REGISTER (DATA) MEMORY

Randomize all locations	Write pseudo-random numbers to all data memory locations.
Fill all locations with value	Write the value specified in the text box to all data memory locations.
Use values from file	Write the values found in the file specified below to all data memory locations. To save the values specified to a file, click Save As to save to a .por file.
Fill Data	Execute the write to data memory.

8.9 ADDITIONAL COMMANDS DIALOG, FORCE OPCODE TAB

This tab allows you to force an opcode to be executed at the current position of halted program execution.

TABLE 8-2: FORCE OPCODE TAB

Instruction	Enter or select instruction.
Entry Is	Specify whether instruction is mnemonic or a hex value.
View	Click Check to view the resulting opcode in hex.
Run	Click Execute to execute the opcode.
Status	Status will show the result of an execution (run).

Note: Two word instructions are not supported.

8.10 SETTINGS DIALOG, PORT TAB

Select either a USB connection between the MPLAB ICE 4000 and the PC. For more information on communications, see **Section 3.4 "Configuring the Communications Port"**.

TABLE 8-3:   USB PORT SELECTION

ICEUSB-*n*	Select USB port ICEUSB-*n*, where *n* = 0, 1, 2, or 3, for MPLAB ICE 4000 to PC communications. (Default: ICEUSB-0)
Other ICEUSB-	Select a USB port other than those specified above.
Allow Auto Connection	Allow MPLAB IDE to automatically connect to the MPLAB ICE 4000.

8.11 SETTINGS DIALOG, INFO TAB

View information about the emulator system.

TABLE 8-4:   INFO TAB

ICE Object	In-circuit emulator selected in MPLAB IDE.
PMF Object	Processor module in use.
Base Unit (Pod)	Part number of pod being used.
Processor Module	Part number of processor module being used.
Device Selected	Device selected in MPLAB IDE.
Device Adapter	Part number of device adapter being used, if any.
Other	Other information about the emulator system, such as the current emulator DLL being used in MPLAB IDE.

8.12 SETTINGS DIALOG, LIMITATIONS TAB

View emulator limitations for the device selected.

TABLE 8-5:   LIMITATIONS TAB

Device Selected	Device selected in MPLAB IDE.
MPLAB ICE 4000 Limitations	
Text Box	A brief list of emulator limitations for the device selected.
Details	Detailed emulator limitations for the device selected.

8.13 SETTINGS DIALOG, VIEW TAB

Set viewing properties.

TABLE 8-6:   VIEW TAB

Analyzer	
Complex Trigger (Analyzer) Property Sheet Always On Top When Opened	When the Complex Trigger Settings tab of the MPLAB ICE 4000 Analyzer dialog is open, it will always appear on top of other MPLAB IDE windows or dialogs when this is checked. If not checked, the dialog can disappear behind other windows or dialogs. **Note:** The MPLAB ICE 4000 Analyzer dialog must be closed and then reopened for selections here to take effect.
Periodic (Real-Time) Reads	
Enabled	If checked, real-time memory reads will occur at intervals specified in the Read Period for registers specified in Apply To.
Read Period (mS)	Specify a the period for memory reads.
Apply To	Specify if all memory or only memory for those registers displayed in Watches is read.
Other	Click **Show State and Operations Gauge** to open an emulator monitor window (**Section 4.10 "Monitoring Emulator States and Operations"**).

8.14 SETTINGS DIALOG, CLOCK TAB

Set up the emulator clock or select the target board clock. Other options may be available, depending on device selected and PMF emulator chip.

For more information, see **Section 3.6 "Setting Up the Processor Clock"**.

8.14.1   PIC18X Devices (PMFs with PIC18C01/C02)

TABLE 8-7:   CLOCK TAB – PIC18X DEVICES (PMFS WITH PIC18C01/C02)

Emulator	
Desired Frequency	The frequency at which you would like emulation to run. Consult your device data sheet for appropriate frequencies for your device.
Actual Frequency (and Percent Error)	The emulator-generated clock frequency and deviation from the desired frequency.
Dialog	
Use Target Board Clock	Use the target board clock, and not the emulator clock, for timing.
Optimize for Fast Step	Check to speed up stepping. This assumes that you will not change the frequency while executing/stepping your code, i.e., frequency checking is suspended. **Note:** Do not use with INT RC or Timer 1 Osc.

8.14.2   PIC18X Devices (PMFs with PIC18C03)

TABLE 8-8:   CLOCK TAB – PIC18X DEVICES (PMFS WITH PIC18C03)

Emulator	
Desired Frequency	The frequency at which you would like emulation to run. Consult your device data sheet for appropriate frequencies for your device.
Actual Frequency (and Percent Error)	The emulator-generated clock frequency and deviation from the desired frequency.
Target	
Use Target Board Clock	Use the target board clock, and not the emulator clock, for timing.
Fast access while halted	
State	Fast access is enabled/disabled.
Enable Fast Oscillator when halted	When enabled, stepping will be faster. For more on this operation, see "Single stepping extremely slow" under Troubleshooting.
Enable Fast Oscillator even when OSC2 is clock out	Use fast oscillator even though OSC2 is used as a clock output. Care should be taken here as the OSC2 frequency will switch to a fast frequency when halted.
OSC2 pin	
Freeze CLKOUT While Halted	Freeze the clock output function when halted. Useful with "Enable Fast Oscillator even when OSC2 is clock out".

8.14.3   dsPIC30F Devices

TABLE 8-9:   CLOCK TAB – dsPIC30F DEVICES

Emulator	
Desired Frequency	The frequency at which you would like emulation to run. Consult your device data sheet for appropriate frequencies for your device.
Actual Frequency	The emulator-generated clock frequency.
Dialog	
Use Target Board Clock	Use the target board clock, and not the emulator clock, for timing.
Diagram	A diagram of available oscillator configurations for the device is displayed. The Oscillator Source chosen in the configuration bits will be highlighted in this diagram.

8.15   SETTINGS DIALOG, POWER TAB

Select the power source for the emulation system. Other options may be available, depending on device selected. For more information, see **Section 3.5 "Selecting Processor Power"**.

TABLE 8-10:   POWER TAB

Processor Power	*From Emulator* Use the emulator power supply to power the emulator system (pod and device adapter). This is the default. *From Target Board* Use power from the target board to power the emulator system.
Analog Voltage Source	**dsPIC Devices Only** *From Emulator* Use the emulator power supply as the analog voltage source. This is the default. *From Target Board* Use power from the target board as the analog voltage source.
Analog Voltage Reference	**dsPIC Devices Only** *From Emulator* Use the emulator power supply to provide the reference for analog voltage. This is the default. *From Target Board* Use power from the target board to provide the reference for analog voltage.
Low Voltage Usage	**PIC18X Devices Only** Display whether or not Low Voltage mode is being used for emulation.

8.16   SETTINGS DIALOG, BREAK OPTIONS TAB

Set up options for halting/resetting the processor (device). Other options may be available, depending on device selected.

TABLE 8-11:   BREAK OPTIONS TAB

General	
Global Hardware Break Enable	When selected, enables all hardware breakpoints. When not selected, all hardware breakpoints are disabled.
Freeze Peripherals on Halt	This option freezes peripherals (timers, ports, PWM, etc.) on a halt operation. By default, this option is on so that you will get a true picture of the status. When Freeze Peripherals on Halt is on, you can write to a port after a halt, but you cannot read from the port until you perform a single-step. You may wish to turn this option off if a peripheral must continue to run after a halt. This option facilitates troubleshooting of timers in situations when timers are continuing to get set/reset and cleared after a halt. **Note:** If Freeze Peripherals On Halt is selected, the I/O port bits in the SFR or the watch windows will not update when single stepping.
Stack Full/Underflow Operation When Enabled	
Stack Reset	Display whether or not stack reset is enabled in the configuration bits.
Disable Stack Full/Underflow Warning	When selected, the stack full/underflow warning message is disabled. However, the halt or break on this condition is not disabled.
Reset On Full/Underflow	When selected, causes the processor to reset as soon as a stack full or underflow occurs. If you do not wish to reset the processor specifically for a stack full or underflow, clear this switch.
Break On Full/Underflow	When selected, causes the processor to halt as soon as a stack full or underflow occurs. If you do not wish to stop processing specifically for a stack full or underflow, clear this switch.
Watchdog Timer Operation When Enabled	
WDT	Display whether or not watchdog timer is enabled in the configuration bits.
Reset On Overflow	When selected, resets the processor when the watchdog timer times out.
Break On Overflow	When selected, halts the processor when the watchdog timer times out.
Enable Warning on WDT Expiration Break	Check to enable the display of a Warning message when the WDT times-out and breaks your code execution. Uncheck to not display this message.

8.17   SETTINGS DIALOG, MEMORY TAB

Set up external (off-chip) program memory for devices which have this feature available. For more information, see **Chapter 5. "External Memory Usage"**.

TABLE 8-12:   MEMORY TAB

External (Off-Chip) Program Memory	
Mode	Display the Memory mode selected in the configuration bits.
Supplied by Emulator	If the mode selected supports external memory, selecting this option causes external memory to be supplied by the emulator. You may wish to use this option because of speed issues (no external connection delays) while developing your code. Also select any Memory Mapped Peripheral Range and its associated Target 16-bit Access mode.
On Target Board	If the mode selected supports external memory, selecting this option causes external memory to be supplied by the target. Also select the Target 16-bit Access mode.
Memory Mapped Peripheral Range	For external memory supplied by the emulator, you may map a range of target memory if you want access to its associated peripheral connections. - Click to select/deselect Enable Banked Access mode. - Specify the Start and End Address values, A28-A8. A7-A0 are already specified (00h and FFh). - Select the Target 16-bit Access mode.
Target 16-Bit Access Mode	For external memory supplied by the target or any memory mapped range, you must set up the external memory interface. Choose the mode of 16-bit access to the target, either Word Write, Byte Select, or Byte Write, and any delay (Wait = 0, 1, 2 or 3 TCY.)
CSEL	For PIC18C601/801 devices, specify values for the CSEL2 and CSELIO registers.
ICE Memory Mapping	
Emulator Supplied	Displays range of program memory supplied by the emulator.
Target Supplied	Displays range of program memory supplied by the target.

8.18   SETTINGS DIALOG, PINS/PINS AND USAGE TAB

Set up the operation of special pins for your selected device. Other options may be available, depending on device selected.

8.18.1   PIC18X Devices Only

TABLE 8-13:   PINS AND USAGE TAB PIC18X DEVICES ONLY

User Master Clear and Master Clear Pull-up	
$\overline{\text{MCLR}}$ Pin Usage	Display how the $\overline{\text{MCLR}}$ pin will be used, either as master clear or an input port pin.
$\overline{\text{MCLR}}$ Pull-up Resistor Supplied by ICE	Connect/disconnect the resistor on the MPLAB ICE 4000 header used to pull up the $\overline{\text{MCLR}}$ pin. **Warning:** Disconnecting may lock up the emulator hardware.
Enable $\overline{\text{MCLR}}$ Low Warning (if pull-up supplied by ICE	Display a Warning message if $\overline{\text{MCLR}}$ voltage level is low. This would indicate a problem, as the pull-up should be keeping the level high.
Emulator Operation	
Preserve user's external bus access method when halted	The purpose of this check box is to keep the emulator from changing external bus access methods during the "halted" state of the emulator when using Extended Microcontroller mode. Generally, the emulator would change external bus access methods to the most efficient method for the emulator during Halt. However, this process tri-states the external bus, causing the bus and control lines to float. This may cause problems for the target system. By selecting "Preserve user's external bus access method when halted", the bus stays active during "halted" states. This may cause the emulator's response to be slightly slower and there will be bus activity generated by the emulator system.

8.18.2   dsPIC Devices Only

TABLE 8-14:   PINS AND USAGE TAB dsPIC DEVICES ONLY

Pins	
$\overline{\text{MCLR}}$ Pull-up Resistor Connected	Connect/disconnect the resistor on the MPLAB ICE 4000 header used to pull up the MCLR pin. **Warning:** Disconnecting may lock up the emulator hardware.
Emulator Operation and Usage	
Enable Clip-on Emulation on Target Board Device	**This feature not yet implemented.** For target dsPIC device in proper mode, emulate using target device.

8.19   SETTINGS DIALOG, PERIPHERAL TAB

Set up which peripherals will freeze on halt. Other options may be available, depending on device selected.

8.20   OTHER DIALOGS/WINDOWS

Other dialogs and windows added to MPLAB IDE are:

- Breakpoint Dialog

    Set multiple software breakpoints in this dialog (**Section 4.5 "Using Software Breakpoints"**).

- MPLAB ICE 4000 Analyzer Properties Dialog

    Set up Complex Trigger Settings (**Chapter 6. "Complex and Internal Triggers"**), Trigger In/Out Settings (**Section 4.7 "Using Trigger In/Out Settings"**) and Code Coverage (**Section 7.3 "Code Coverage"**).

- Stopwatch Dialog

    Set up the stopwatch (**Section 4.9 "Using the Stopwatch"**).

- ICE Trace Window

    View the contents of the Trace buffer (**Chapter 7. "Code Coverage, Trace Memory, Real-Time Reads"**).

Appendix A. Troubleshooting

A.1   INTRODUCTION

This section is designed to help you troubleshoot any problems, errors or issues you encounter while using MPLAB ICE 4000. If none of this information helps you, please see the Preface for ways to contact Microchip Technology.

A.2   HIGHLIGHTS

This chapter discusses the following:

- Common Problems/FAQ
- Error Messages
- Limitations

A.3   COMMON PROBLEMS/FAQ

The following are problems and issues that many users may encounter when using the emulator system.

- Communications cannot be established with MPLAB ICE 4000
- MPLAB ICE 4000 driver not loaded on installation
- Single stepping extremely slow when programmer enabled
- Single stepping extremely slow
- Program Memory appears correct, but the file registers and program execution do not appear to work correctly.
- Program keeps resetting
- On reset, an error message appears saying that there was an error resetting the processor and to check power.
- Power light is blinking

Communications cannot be established with MPLAB ICE 4000

(1) Check the port.

Open the Settings dialog (*Debugger>Settings*) and select the **Ports** tab.

Verify that the port specified in MPLAB IDE is the port you are using, i.e., make sure MPLAB ICE 4000 is plugged into the correct communications port.

(2) Check target clock/power.

If you are using target clock, select emulator clock (*Debugger>Settings*, **Clock** tab, uncheck "Use Target Board Clock") and see if you can establish communications. If so, there may be a problem with the target clock.

If you are using target power, select emulator power (*Debugger>Settings*, **Power** tab, select "Processor Power From Emulator") and see if you can establish communications. If so, there may be a problem with the target power.

MPLAB ICE 4000 driver not loaded on installation

If you did not install the MPLAB ICE 4000 driver according to the instructions displayed at MPLAB IDE installation and a standard Windows driver has been used instead, MPLAB ICE 4000 will not work. Follow the directions in the file `MPUsbClean.htm` found in the `Driversnn\ICE4k_USB` subdirectory of the MPLAB IDE installation directory, where *nn* is the version of Windows OS. Then go to **Section 2.4 "Driver and Software Installation"** to install the correct driver.

Cannot select MPLAB ICE 4000 from MPLAB IDE Debug menu

For some operating systems, like Windows XP, you may have to wait up to 10 seconds from the time you power your USB device until the device is recognized by the operating system and thus allows MPLAB IDE to see it.

Single stepping extremely slow when programmer enabled

When using MPLAB ICE 4000 to emulate a part with EEDATA, if you have a programmer enabled your stepping through code will be very slow. Disable to the programmer to speed up stepping time.

The slowdown occurs because the programmer must attach to the entire range of EEDATA so it can be programmed. The debugger, seeing that someone is attached to the EEDATA (whether a programmer or EEDATA Display) will read back the entire range after each step in case the data was modified while running.

Single stepping extremely slow

In general, to speed up execution when single stepping through code, close the following windows: EEDATA, Stack and SFR/GPR window. The EEDATA and Stack windows require extra time to populate with data. The SFR/GPR window requires more time simply because of the large number of registers it may contain. In this case, you may want to consider using a Watch window.

Additionally, select *Debugger>Settings*, **Clock** tab and choose to either check the check box for "Optimize for Fast Step" or change the frequency to be faster while stepping.

For some newer PMFs, if you are using a low frequency clock, you can speed stepping by selecting "Enable Fast Oscillator While Halted" on the *Debugger>Settings*, **Clock** tab. This allows the PMF/PCM to switch frequencies while stepping to a 10 MHz clock. However, you should be careful using this feature:

- If you are using an oscillator mode that uses the pin OSC2 as an output with $F_{OSC}/4$, the "Enable Fast Oscillator While Halted" check box will be grayed out. This is to avoid enabling a feature that would cause the OSC2 pin to go from its current operating frequency / 4 to 10 MHz / 4. However, you can still use the fast oscillator by selecting the checkbooks "Enable Fast Oscillator Even When OSC2 is clock out".
- If you are running with peripherals not frozen, the peripherals operation will change when you step to 10 MHz.

Program Memory appears correct, but the file registers and program execution do not appear to work correctly.

You may have a problem with clock or power setup.

Make sure that the processor clock is set to a valid speed. Select *Debugger>Settings*, **Clock** tab. Make sure the frequency is within the correct range for the emulated device at the current operating voltage.

Verify that the correct power source is selected by selecting *Debugger>Settings*, **Power** tab. Make sure the processor power option is correct. If using a target board for a power source, verify that the target board power has been applied.

Program keeps resetting

Check the configuration bits settings (*Configure>Configuration Bits*) for your selected device. Some reset functions (such as Watchdog Timer Reset) are enabled by default.

On reset, an error message appears saying that there was an error resetting the processor and to check power.

If you are doing low voltage emulation, this message is the point after doing a System Reset where the target board should be connected and voltage applied. However, MPLAB IDE and MPLAB ICE 4000 do not always synchronize on the first try. In most cases, clicking Yes to retry will work and initialization will continue. If it doesn't work after two or three times, click No and MPLAB IDE will try to continue, reporting any other errors it encounters.

Power light is blinking

On some pods, the power light will blink when there is a system fault (**Section B.6.1 "Power LED – Green"**). Turn the pod off and then back on to clear the fault. If this does not clear the fault, contact Microchip support.

A.4   ERROR MESSAGES

You may receive the following error messages when using the emulator system.

• Cannot Perform Requested Operation
• Unable to Find Emulator System
• Error Halting Processor
• No Target Clock
• Processor Frequency Too Slow
• Stack Overflow Has Occurred After Break
• Unable to Communicate with Emulator

Cannot Perform Requested Operation

Emulation may have been halted by the user while the processor was in sleep mode.

It is also possible that the power is expected to come from the target board, but power is not applied to the device. See *Debugger>Settings* and click the **Power** tab to check this setting.

Unable to Find Emulator System

Make sure the emulator is connected to the power supply and is powered on. Also, make sure the emulator is connected to the PC's communication port.

Error Halting Processor

Emulation may have been halted by the user while the processor was in sleep mode.

It is also possible that the power is expected to come from the target board, but power is not applied to the device. See *Debugger>Settings* and click the **Power** tab to check this setting.

No Target Clock

MPLAB IDE cannot detect a clock on the target board. Click *Debugger>Settings* and click the **Clock** tab. Check the target clock or switch to the emulator clock.

Processor Frequency Too Slow

The clock frequency chosen is too slow (below 32 kHz) for the MPLAB ICE 4000 processor module to emulate. Select *Debugger>Settings* and click the **Clock** tab. Select a higher frequency. Refer to the selected PICmicro MCU data sheet for operating frequencies.

Stack Overflow Has Occurred After Break

An overflow of the stack has occurred. If this has happened after a program break, check *Debugger>Settings*, **Break Options** tab and make sure "Freeze Peripherals on Halt" is checked. If it is not, peripherals will continue to run during a Halt and could cause a stack overflow.

Unable to Communicate with Emulator

The device you selected is not supported on this tool. Select *Configure>Select Device* to select a supported device. Select the appropriate configuration options from the MPLAB ICE 4000 Settings dialog tabs in order to configure the emulator processor module or probe for a specific device. For additional help, see Common Problems.

Make sure the processor module is plugged in to the emulator pod.

Make sure the emulator is powered on, connected to the PC and is functioning properly.

A.5   LIMITATIONS

General and device-specific limitations for the emulator may be found in on-line help for MPLAB ICE 4000.

Appendix B. Pod Electrical Specification

B.1   INTRODUCTION

The hardware electrical specifications of the MPLAB ICE 4000 in-circuit emulator pod are shown here.

Refer to *MPLAB ICE 4000 Processor Module and Device Adapter Specification* (DS51298) for information on a specific processor module/device adapter.

Refer to *MPLAB ICE Transition Socket Specification*s (DS51194) for information on a specific transition socket.

B.2   HIGHLIGHTS

This appendix addresses the following electrical specifications of the emulator pod:

• Declaration of Conformity
• Power
• USB Port
• Indicator Lights
• Logic Probes

B.3   DECLARATION OF CONFORMITY

We

Microchip Technology, Inc.
2355 W. Chandler Blvd.
Chandler, Arizona 85224-6199
USA

hereby declare that the product:

MPLAB ICE 4000

complies with the following standards, provided that the restrictions stated in the operating manual are observed:

Standards: EN61010-1   Laboratory Equipment
Microchip Technology, Inc.
Monday, August 09, 2004

Important Information Concerning the Use of the MPLAB ICE 4000

Due to the special nature of the MPLAB ICE 4000 development system, the user is advised that it can generate higher than normal levels of electromagnetic radiation which can interfere with the operation of all kinds of radio and other equipment.

To comply with the European Approval Regulations therefore, the following restrictions must be observed:

1.   The development system must be used only in an industrial (or comparable) area.
2.   The system must not be operated within 20 meters of any equipment which may be affected by such emissions (radio receivers, TV´s etc.).

B.4 POWER

Power to the MPLAB ICE 4000 system is supplied by an external power supply included with the system. The input is located on the back of the pod, as indicated in Figure B-1. The power on/off switch is also located on the back of the pod.

FIGURE B-1:   MPLAB ICE 4000 REAR VIEW
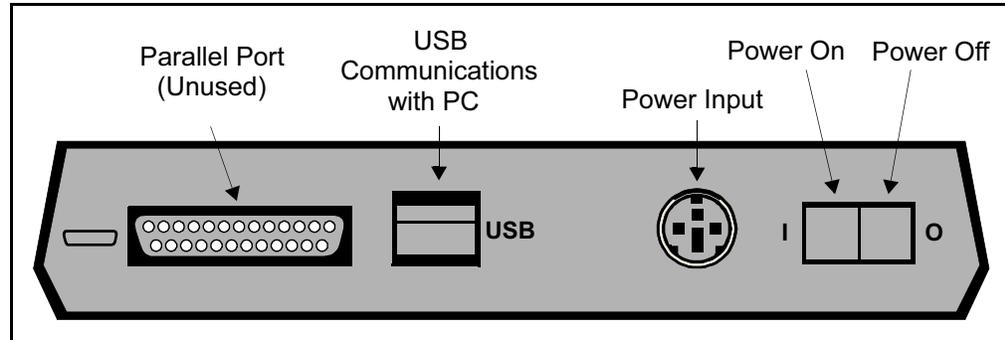
Power Supply Requirements:

+5V, ±5%, 0.75 A, and +3.3V, ±5%, 5.0 A

Power supplied by emulator: emulator loads system at 10 mA typical

Power supplied by target: emulator loads system at 80 mA typical

B.5 USB PORT

MPLAB ICE 4000 may be connected to the host PC via a universal serial bus (USB) port, version 1.1 compliant. The USB connector is located on the rear panel of the pod (Figure B-1). A USB port on the host PC is required.

Cable Length – The PC to MPLAB ICE 4000 cable length for proper operation has been tested to be 6 feet. This length cable is shipped with MPLAB ICE 4000.

B.6 INDICATOR LIGHTS

Four indicator lights are located on the front of the emulator (Figure B-2):

• Power LED – Green
• Status LED – Green
• Halt LED – Red
• Run LED – Green

Information about the indicator lights is detailed in the following sections.

FIGURE B-2:   MPLAB ICE 4000 FRONT VIEW
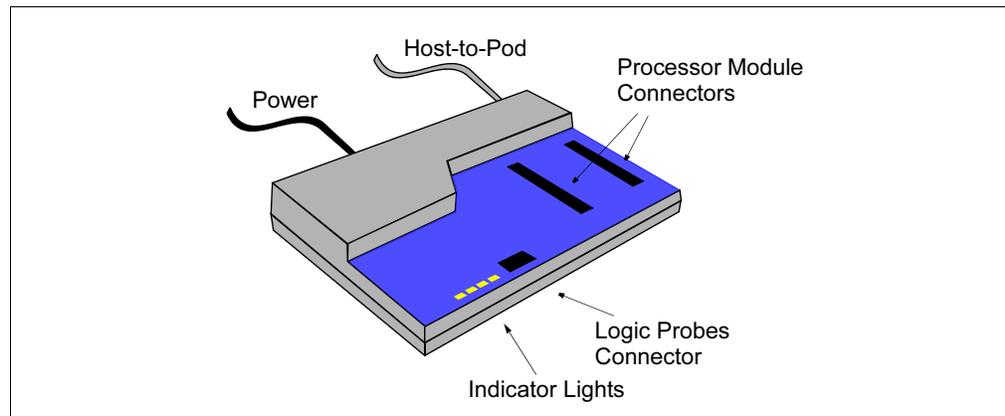
B.6.1 Power LED – Green

LED	Condition
On	System is sufficiently powered.
Off	No or low power condition.
Blinking	Fault detected.

This LED is located on the front panel of the pod and is lit when the system is sufficiently powered.

The LED will blink slowly if the internal system voltage falls below a 4.65 (±5%) volts, indicating a low power condition. If this occurs, power should be removed until the cause of the low voltage is determined.

On pods with an electronic circuit breaker, the LED will blink if there is a system fault. The rate of blinking determines the type of fault.

Blink Rate	Meaning
5 per second	Overcurrent condition (more than 0.75 A for 5V supply, 5 A for 3.3V supply)
2 per second	Undervoltage condition (equal to or less than 4.65V for 5V supply, 2.7V for 3.3V supply.)
3 rapid, pause, 3 rapid	Undervoltage condition on processor module.

The system may be reset from a fault by turning off the pod and then turning it back on. If a condition persists, please contact Microchip support.

B.6.2 Status LED – Green

LED	Condition
On	Processor module inserted correctly.
Blink	Processor module not inserted completely or correctly.

This LED will turn on when the processor module is not inserted completely or correctly. If this LED is on, turn off the power and remove and reinsert the processor module.

B.6.3 Halt LED – Red

LED	Condition
On	Processor is halted.
Off	Processor is running.

This LED will be on when the processor is halted. This LED and the Power LED should be on after MPLAB ICE 4000 has been chosen as the development mode from the MPLAB IDE.

The function of this LED is the compliment of the Run LED.

B.6.4 Run LED – Green

| LED	Condition
On	Processor is running.
Off	Processor is halted.
Blink	Processor executes single step.

This LED will be on when the processor is actually running and will blink momentarily when a single step is executed.

The function of this LED is the compliment of the Halt LED.

B.7   LOGIC PROBES

The 14-pin connector (Figure B-3) on the front panel of the MPLAB ICE 4000 emulator (Figure B-2) provides power, ground, an external break input, a trigger input, a trigger output and up to eight trace/trigger inputs.

FIGURE B-3:   LOGIC PROBE PINOUT

Logic Probe Pinout on Emulator

```
13  ┌ □ □ □ □ □ □ □ ┐ 1
14  └ □ □ □ □ □ □ □ ┘ 2
```
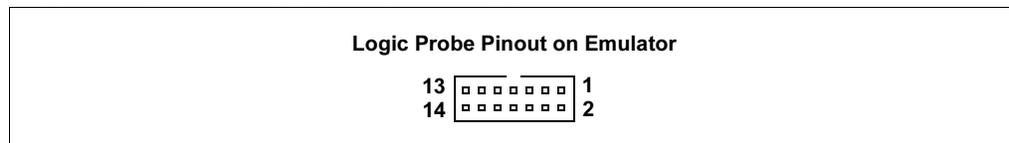
Logic probes may be attached to this connector to give the functionality described in Table B-1. The probes are color-coded for easy identification.

TABLE B-1:   LOGIC PROBE PINOUT DESCRIPTION

Pin	I/O	Name	Function	Color
1	O	VDD	System Power. Will supply +5V ±5%, up to 250 mA	Red
2	O	HLTOUT	Processor Halted Signal. Indicates whether the processor is halted (high) or running (low).	Gray
3	O	TRGOUT	Trigger/Break Out	Gray
4	I	TRGIN	Trigger In. Active-high input will freeze the trace buffer without halting the processor or edge triggered input will halt the processor.	Gray
5	I	EXT7	External input bit 7 of the trace/trigger inputs.	White
6	I	EXT6	External input bit 6 of the trace/trigger inputs.	White
7	I	EXT5	External input bit 5 of the trace/trigger inputs.	White
8	I	EXT4	External input bit 4 of the trace/trigger inputs.	White
9	I	EXT3	External input bit 3 of the trace/trigger inputs.	White
10	I	EXT2	External input bit 2 of the trace/trigger inputs.	White
11	I	EXT1	External input bit 1 of the trace/trigger inputs.	White
12	I	EXT0	External input bit 0 of the trace/trigger inputs.	White
13	Gnd	GND	System Ground	Black
14	Gnd	GND	System Ground	Black

The electrical specifications for logic probes are listed in Table B-2.

TABLE B-2:   LOGIC PROBE ELECTRICAL SPECIFICATIONS

Logic Inputs	VIH = 3.2V min
	VIL = 1.8V max
Logic Outputs	VOH = 2.4V min
	VOL = 0.4V max
TRGIN	Minimum input pulse width = 15 nsec
TRGOUT	The output pulse width of the filter trace is one bus cycle, which is ¼ the clock speed (PICmicro MCU devices use four clocks for each instruction).

Glossary

Absolute Section

A section with a fixed (absolute) address that cannot be changed by the linker.

Access Memory (PIC18 Only)

Special registers on PIC18XXXXX devices that allow access regardless of the setting of the bank select register (BSR).

Address

Value that identifies a location in memory.

Alphabetic Character

Alphabetic characters are those characters that are letters of the arabic alphabet (a, b, …, z, A, B, …, Z).

Alphanumeric

Alphanumeric characters are comprised of alphabetic characters and decimal digits (0,1, …, 9).

ANSI

American National Standards Institute is an organization responsible for formulating and approving standards in the United States.

Application

A set of software and hardware that may be controlled by a PICmicro microcontroller.

Archive

A collection of relocatable object modules. It is created by assembling multiple source files to object files, and then using the archiver to combine the object files into one library file. A library can be linked with object modules and other libraries to create executable code.

Archiver

A tool that creates and manipulates libraries.

ASCII

American Standard Code for Information Interchange is a character set encoding that uses 7 binary digits to represent each character. It includes upper and lower case letters, digits, symbols and control characters.

Assembler

A language tool that translates assembly language source code into machine code.

Assembly Language

A programming language that describes binary machine code in a symbolic form.

Asynchronous Stimulus

Data generated to simulate external inputs to a simulator device.

Breakpoint, Hardware

An event whose execution will cause a halt.

Breakpoint, Software

An address where execution of the firmware will halt. Usually achieved by a special break instruction.

Build

Compile and link all the source files for an application.

C

A general-purpose programming language which features economy of expression, modern control flow and data structures, and a rich set of operators.

Calibration Memory

A special function register or registers used to hold values for calibration of a PICmicro microcontroller on-board RC oscillator or other device peripherals.

COFF

Common Object File Format. An object file of this format contains machine code, debugging and other information.

Command Line Interface

A means of communication between a program and its user based solely on textual input and output.

Compiler

A program that translates a source file written in a high-level language into machine code.

Configuration Bits

Special-purpose bits programmed to set PICmicro microcontroller modes of operation. A configuration bit may or may not be preprogrammed.

Control Directives

Directives in assembly language code that cause code to be included or omitted based on the assembly-time value of a specified expression.

Cross Reference File

A file that references a table of symbols and a list of files that references the symbol. If the symbol is defined, the first file listed is the location of the definition. The remaining files contain references to the symbol.

Data Directives

Data directives are those that control the assembler's allocation of program or data memory and provide a way to refer to data items symbolically; that is, by meaningful names.

Data Memory

On Microchip MCU and DSC devices, data memory (RAM) is comprised of general purpose registers (GPRs) and special function registers (SFRs). Some devices also have EEPROM data memory.

Device Programmer

A tool used to program electrically programmable semiconductor devices such as microcontrollers.

Directives

Statements in source code that provide control of the language tool's operation.

Download

Download is the process of sending data from a host to another device, such as an emulator, programmer or target board.

EEPROM

Electrically Erasable Programmable Read Only Memory. A special type of PROM that can be erased electrically. Data is written or erased one byte at a time. EEPROM retains its contents even when power is turned off.

Emulation

The process of executing software loaded into emulation memory as if it were firmware residing on a microcontroller device.

Emulation Memory

Program memory contained within the emulator.

Emulator

Hardware that performs emulation.

Emulator System

The MPLAB ICE 2000 and 4000 emulator systems include the pod, processor module, device adapter, cables and MPLAB IDE software.

EPROM

Erasable Programmable Read Only Memory. A programmable read-only memory that can be erased usually by exposure to ultraviolet radiation.

Event

A description of a bus cycle which may include address, data, pass count, external input, cycle type (fetch, R/W) and time stamp. Events are used to describe triggers, breakpoints and interrupts.

Export

Send data out of the MPLAB IDE in a standardized format.

Extended Microcontroller Mode

In extended microcontroller mode, on-chip program memory as well as external memory is available. Execution automatically switches to external if the program memory address is greater than the internal memory space of the PIC17CXXX or PIC18CXXX device.

External Label

A label that has external linkage.

External Linkage

A function or variable has external linkage if it can be referenced from outside the module in which it is defined.

External Symbol

A symbol for an identifier which has external linkage. This may be a reference or a definition.

External Symbol Resolution

A process performed by the linker in which external symbol definitions from all input modules are collected in an attempt to resolve all external symbol references. Any external symbol references which do not have a corresponding definition cause a linker error to be reported.

External Input Line

An external input signal logic probe line (TRIGIN) for setting an event based upon external signals.

External RAM

Off-chip Read/Write memory.

File Registers

On-chip data memory, including general purpose registers (GPRs) and special function registers (SFRs).

Flash

A type of EEPROM where data is written or erased in blocks instead of bytes.

FNOP

Forced No Operation. A forced NOP cycle is the second cycle of a two-cycle instruction. Since the PICmicro microcontroller architecture is pipelined, it prefetches the next instruction in the physical address space while it is executing the current instruction. However, if the current instruction changes the program counter, this prefetched instruction is explicitly ignored, causing a forced NOP cycle.

GPR

General Purpose Register. The portion of device data memory (RAM) available for general use.

Halt

A stop of program execution. Executing Halt is the same as stopping at a breakpoint.

HEX Code

Executable instructions stored in a hexadecimal format code. HEX code is contained in a HEX file.

HEX File

An ASCII file containing hexadecimal addresses and values (HEX code) suitable for programming a device.

High Level Language

A language for writing programs that is further removed from the processor than assembly.

ICD

In-Circuit Debugger. MPLAB ICD and MPLAB ICD 2 are Microchip's in-circuit debuggers for PIC16F87X and PIC18FXXX devices, respectively. These ICDs work with MPLAB IDE.

ICE

In-Circuit Emulator. MPLAB ICE 2000 and 4000 are Microchip's in-circuit emulators that work with MPLAB IDE.

IDE

Integrated Development Environment. MPLAB IDE is Microchip's integrated development environment.

Import

Bring data into the MPLAB IDE from an outside source, such as from a hex file.

Instruction Set

The collection of machine language instructions that a particular processor understands.

Instructions

A sequence of bits that tells a central processing unit to perform a particular operation and can contain data to be used in the operation.

Internal Linkage

A function or variable has internal linkage if it can not be accessed from outside the module in which it is defined.

International Organization for Standardization

An organization that sets standards in many businesses and technologies, including computing and communications.

Interrupt

A signal to the CPU that suspends the execution of a running application and transfers control to an Interrupt Service Routine (ISR) so that the event may be processed.

Interrupt Handler

A routine that processes special code when an interrupt occurs.

Interrupt Request

An event which causes the processor to temporarily suspend normal instruction execution and to start executing an interrupt handler routine. Some processors have several interrupt request events allowing different priority interrupts.

Interrupt Service Routine

User-generated code that is entered when an interrupt occurs. The location of the code in program memory will usually depend on the type of interrupt that has occurred.

IRQ

See Interrupt Request.

ISO

See International Organization for Standardization.

ISR

See Interrupt Service Routine.

Librarian

See Archiver.

Library

See Archive.

Linker

A language tool that combines object files and libraries to create executable code, resolving references from one module to another.

Linker Script Files

Linker script files are the command files of a linker. They define linker options and describe available memory on the target platform.

Listing Directives

Listing directives are those directives that control the assembler listing file format. They allow the specification of titles, pagination and other listing control.

Listing File

A listing file is an ASCII text file that shows the machine code generated for each C source statement, assembly instruction, assembler directive, or macro encountered in a source file.

Local Label

A local label is one that is defined inside a macro with the LOCAL directive. These labels are particular to a given instance of a macro's instantiation. In other words, the symbols and labels that are declared as local are no longer accessible after the ENDM macro is encountered.

Logic Probes

Up to 14 logic probes can be connected to some Microchip emulators. The logic probes provide external trace inputs, trigger output signal, +5V and a common ground.

Machine Code

The representation of a computer program that is actually read and interpreted by the processor. A program in binary machine code consists of a sequence of machine instructions (possibly interspersed with data). The collection of all possible instructions for a particular processor is known as its "instruction set".

Machine Language

A set of instructions for a specific central processing unit, designed to be usable by a processor without being translated.

Macro

Macroinstruction. An instruction that represents a sequence of instructions in abbreviated form.

Macro Directives

Directives that control the execution and data allocation within macro body definitions.

Make Project

A command that rebuilds an application, re-compiling only those source files that have changed since the last complete compilation.

MCU

Microcontroller Unit. An abbreviation for microcontroller. Also uC.

Message

Text displayed to alert you to potential problems in language tool operation. A message will not stop operation.

Microcontroller

A highly integrated chip that contains a CPU, RAM, program memory, I/O ports and timers.

Microcontroller Mode

One of the possible program memory configurations of the PIC17CXXX and PIC18CXXX families of microcontrollers. In microcontroller mode, only internal execution is allowed. Thus, only the on-chip program memory is available in microcontroller mode.

Microprocessor Mode

One of the possible program memory configurations of the PIC17CXXX and PIC18CXXX families of microcontrollers. In microprocessor mode, the on-chip program memory is not used. The entire program memory is mapped externally.

Mnemonics

Text instructions that can be translated directly into machine code. Also referred to as Opcodes.

MPASM Assembler

Microchip Technology's relocatable macro assembler for PICmicro microcontroller devices, KeeLoq devices and Microchip memory devices.

MPLAB ASM30

Microchip's relocatable macro assembler for dsPIC30F digital signal controller devices.

MPLAB C1X

Refers to both the MPLAB C17 and MPLAB C18 C compilers from Microchip. MPLAB C17 is the C compiler for PIC17CXXX devices and MPLAB C18 is the C compiler for PIC18CXXX and PIC18FXXXX devices.

MPLAB C30

Microchip's C compiler for dsPIC30F digital signal controller devices.

MPLAB ICD 2

Microchip's in-circuit debugger for PIC16F87X, PIC18FXXX and dsPIC30FXXXX devices. The ICD works with MPLAB IDE. The main component of each ICD is the module. A complete system consists of a module, header, demo board, cables and MPLAB IDE Software.

MPLAB ICE 2000

Microchip's in-circuit emulator for PICmicro MCU's that works with MPLAB IDE.

MPLAB ICE 4000

Microchip's in-circuit emulator for dsPIC DSC's that works with MPLAB IDE.

MPLAB IDE

Microchip's Integrated Development Environment.

MPLAB LIB30

MPLAB LIB30 archiver/librarian is an object librarian for use with COFF object modules created using either MPLAB ASM30 or MPLAB C30 C compiler.

MPLAB LINK30

MPLAB LINK30 is an object linker for the Microchip MPLAB ASM30 assembler and the Microchip MPLAB C30 C compiler.

MPLAB SIM

Microchip's simulator that works with MPLAB IDE in support of PICmicro MCU devices.

MPLAB SIM30

Microchip's simulator that works with MPLAB IDE in support of dsPIC DSC devices.

MPLIB Object Librarian

MPLIB librarian is an object librarian for use with COFF object modules created using either MPASM assembler (mpasm or mpasmwin v2.0) or MPLAB C1X C compilers.

MPLINK Object Linker

MPLINK linker is an object linker for the Microchip MPASM assembler and the Microchip MPLAB C17 or C18 C compilers. MPLINK linker also may be used with the Microchip MPLIB librarian. MPLINK linker is designed to be used with MPLAB IDE, though it does not have to be.

MRU

Most Recently Used. Refers to files and windows available to be selected from MPLAB IDE main pull down menus.

Nesting Depth

The maximum level to which macros can include other macros.

Node

MPLAB IDE project component.

Non Real-Time

Refers to the processor at a breakpoint or executing single step instructions or MPLAB IDE being run in simulator mode.

Non-Volatile Storage

A storage device whose contents are preserved when its power is off.

NOP

No Operation. An instruction that has no effect when executed except to advance the program counter.

Object Code

The machine code generated by an assembler or compiler.

Object File

A file containing machine code and possibly debug information. It may be immediately executable or it may be relocatable, requiring linking with other object files, e.g. libraries, to produce a complete executable program.

Object File Directives

Directives that are used only when creating an object file.

Off-Chip Memory

Off-chip memory refers to the memory selection option for the PIC17CXXX or PIC18CXXX device where memory may reside on the target board, or where all program memory may be supplied by the Emulator. The Memory tab accessed from Options > Development Mode provides the Off-Chip Memory selection dialog box.

Opcodes

Operational Codes. *See* Mnemonics.

Operators

Symbols, like the plus sign '+' and the minus sign '-', that are used when forming well-defined expressions. Each operator has an assigned precedence that is used to determine order of evaluation.

OTP

One Time Programmable. EPROM devices that are not in windowed packages. Since EPROM needs ultraviolet light to erase its memory, only windowed devices are erasable.

Pass Counter

A counter that decrements each time an event (such as the execution of an instruction at a particular address) occurs. When the pass count value reaches zero, the event is satisfied. You can assign the Pass Counter to break and trace logic, and to any sequential event in the complex trigger dialog.

PC

Personal Computer or Program Counter.

PC Host

Any IBM™ or compatible personal computer running a supported Windows operating system.

PICmicro MCUs

PICmicro microcontrollers (MCUs) refers to all Microchip microcontroller families.

PICSTART Plus

A developmental device programmer from Microchip. Programs 8-, 14-, 28- and 40-pin PICmicro microcontrollers. Must be used with MPLAB IDE Software.

Pod, Emulator

The external emulator box that contains emulation memory, trace memory, event and cycle timers and trace/breakpoint logic.

Power-on-Reset Emulation

A software randomization process that writes random values in data RAM areas to simulate uninitialized values in RAM upon initial power application.

PRO MATE II

A device programmer from Microchip. Programs all PICmicro microcontrollers and most memory and Keeloq devices. Can be used with MPLAB IDE or stand-alone.

Program Counter

The location that contains the address of the instruction that is currently executing.

Program Memory

The memory area in a device where instructions are stored. Also, the memory in the emulator or simulator containing the downloaded target application firmware.

Project

A set of source files and instructions to build the object and executable code for an application.

Prototype System

A term referring to a user's target application, or target board.

PWM Signals

Pulse Width Modulation Signals. Certain PICmicro MCU devices have a PWM peripheral.

Qualifier

An address or an address range used by the Pass Counter or as an event before another operation in a complex trigger.

Radix

The number base, HEX, or decimal, used in specifying an address.

RAM

Random Access Memory (Data Memory). Memory in which information can be accessed in any order.

Raw Data

The binary representation of code or data associated with a section.

Real-Time

When released from the halt state in the emulator or MPLAB ICD mode, the processor runs in real-time mode and behaves exactly as the normal chip would behave. In real-time mode, the real-time trace buffer of MPLAB ICE is enabled and constantly captures all selected cycles, and all break logic is enabled. In the emulator or MPLAB ICD, the processor executes in real-time until a valid breakpoint causes a halt, or until the user halts the emulator. In the simulator real-time simply means execution of the microcontroller instructions as fast as they can be simulated by the host CPU.

Recursion

The concept that a function or macro, having been defined, can call itself. Great care should be taken when writing recursive macros; it is easy to get caught in an infinite loop where there will be no exit from the recursion.

ROM

Read Only Memory (Program Memory). Memory that cannot be modified.

Run

The command that releases the emulator from halt, allowing it to run the application code and change or respond to I/O in real time.

SFR

See Special Function Registers.

Shell

The MPASM assembler shell is a prompted input interface to the macro assembler. There are two MPASM assembler shells: one for the DOS version and one for the Windows version.

Simulator

A software program that models the operation of devices.

Single Step

This command steps though code, one instruction at a time. After each instruction, MPLAB IDE updates register windows, watch variables and status displays so you can analyze and debug instruction execution. You can also single step C compiler source code, but instead of executing single instructions, MPLAB IDE will execute all assembly level instructions generated by the line of the high level C statement.

Skew

The information associated with the execution of an instruction appears on the processor bus at different times. For example, the executed Opcodes appears on the bus as a fetch during the execution of the previous instruction, the source data address and value and the destination data address appear when the Opcodes is actually executed, and the destination data value appears when the next instruction is executed. The trace buffer captures the information that is on the bus at one instance. Therefore, one trace buffer entry will contain execution information for three instructions. The number of captured cycles from one piece of information to another for a single instruction execution is referred to as the skew.

Skid

When a hardware breakpoint is used to halt the processor, one or more additional instructions may be executed before the processor halts. The number of extra instructions executed after the intended breakpoint is referred to as the skid.

Source Code

The form in which a computer program is written by the programmer. Source code is written in some formal programming language which can be translated into or machine code or executed by an interpreter.

Source File

An ASCII text file containing source code.

Special Function Registers

The portion of data memory (RAM) dedicated to registers that control I/O processor functions, I/O status, timers, or other modes or peripherals.

Stack, Hardware

Locations in PICmicro microcontroller where the return address is stored when a function call is made.

Stack, Software

Memory used by an application for storing return addresses, function parameters and local variables. This memory is typically managed by the compiler when developing code in a high-level language.

Static RAM or SRAM

Static Random Access Memory. Program memory you can Read/Write on the target board that does not need refreshing frequently.

Status Bar

The Status Bar is located on the bottom of the MPLAB IDE window and indicates such current information as cursor position, development mode and device and active tool bar.

Step Into

This command is the same as Single Step. Step Into (as opposed to Step Over) follows a CALL instruction into a subroutine.

Step Over

Step Over allows you to debug code without stepping into subroutines. When stepping over a CALL instruction, the next breakpoint will be set at the instruction after the CALL. If for some reason the subroutine gets into an endless loop or does not return properly, the next breakpoint will never be reached. The Step Over command is the same as Single Step except for its handling of CALL instructions.

Stimulus

Input to the simulator, i.e., data generated to exercise the response of simulation to external signals. Often the data is put into the form of a list of actions in a text file. Stimulus may be asynchronous, synchronous (pin), clocked and register.

Stopwatch

A counter for measuring execution cycles.

Symbol

A symbol is a general purpose mechanism for describing the various pieces which comprise a program. These pieces include function names, variable names, section names, file names, struct/enum/union tag names, etc. Symbols in MPLAB IDE refer mainly to variable names, function names and assembly labels. The value of a symbol after linking is its value in memory.

System Window Control

The system window control is located in the upper left corner of windows and some dialogs. Clicking on this control usually pops up a menu that has the items "Minimize," "Maximize" and "Close."

Target

Refers to user hardware.

Target Application

Software residing on the target board.

Target Board

The circuitry and programmable device that makes up the target application.

Target Processor

The microcontroller device on the target application board.

Template

Lines of text that you build for inserting into your files at a later time. The MPLAB Editor stores templates in template files.

Tool Bar

A row or column of icons that you can click on to execute MPLAB IDE functions.

Trace

An emulator or simulator function that logs program execution. The emulator logs program execution into its trace buffer which is uploaded to MPLAB IDE's trace window.

Trace Memory

Trace memory contained within the emulator. Trace memory is sometimes called the trace buffer.

Trigger Output

Trigger output refers to an emulator output signal that can be generated at any address or address range, and is independent of the trace and breakpoint settings. Any number of trigger output points can be set.

Uninitialized Data

Data which is defined without an initial value. In C,

```
int myVar;
```

defines a variable which will reside in an uninitialized data section.

Upload

The Upload function transfers data from a tool, such as an emulator or programmer, to the host PC or from the target board to the emulator.

Warning

An alert that is provided to warn you of a situation that would cause physical damage to a device, software file, or equipment.

Watch Variable

A variable that you may monitor during a debugging session in a watch window.

Watch Window

Watch windows contain a list of watch variables that are updated at each breakpoint.

Watchdog Timer

A timer on a PICmicro microcontroller that resets the processor after a selectable length of time. The WDT is enabled or disabled and set up using configuration bits.

WDT

See Watchdog Timer.

Index

Index

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: 480-792-7627
Web Address: www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
16200 Addison Road, Suite 255
Addison Plaza
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
25950 Acero St., Suite 200
Mission Viejo, CA 92691
Tel: 949-462-9523
Fax: 949-462-9608

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444
Fax: 650-961-0286

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Unit 32 41 Rawson Street
Epping 2121, NSW
Sydney, Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde
District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-22290061 Fax: 91-80-22290062

Japan
Yusen Shin Yokohama Building 10F
3-17-2, Shin Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or
82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4816
Fax: 886-7-536-4817

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

Taiwan
Taiwan Branch
13F-3, No. 295, Sec. 2, Kung Fu Road
Hsinchu City 300, Taiwan
Tel: 886-3-572-9526
Fax: 886-3-572-6459

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Salvatore Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
Waegenburghtplein 4
NL-5152 JR, Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

07/12/04