About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual is the best copy we could find; it may be incomplete or contain dated information. If we find a more recent copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available. You will find any other available product information on the Agilent Test & Measurement website, www.tm.agilent.com.

HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. In other documentation, to reduce potential confusion, the only change to product numbers and names has been in the company name prefix: where a product number/name was HP XXXX the current name/number is now Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

User's Guide

Publication number E2465-97003
December 1998

For Safety information, Warranties, and Regulatory information, see the pages behind the index.

Solutions for the PowerPC 604

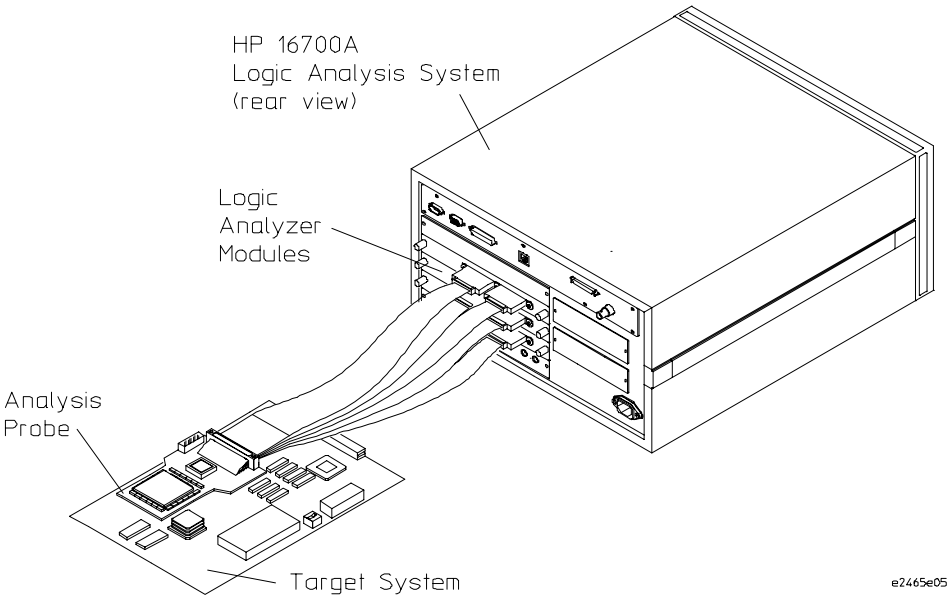
HP Solutions for the PowerPC 604—At a Glance

This manual describes several ways to connect an HP logic analysis system to your target system. These connections use an analysis probe, plus an emulation module (for an emulation solution).

Analysis Probe

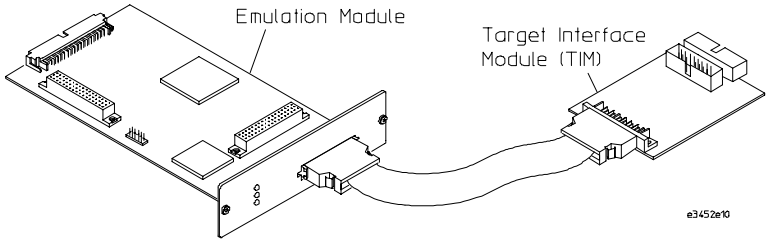
The analysis probe connects your logic analyzer to your target system for state and timing analysis. The analysis probe can be used with an HP 16600A/700A-series logic analysis system or with other HP logic analyzers.

The analysis probe can be purchased alone, or as part of an emulation solution.



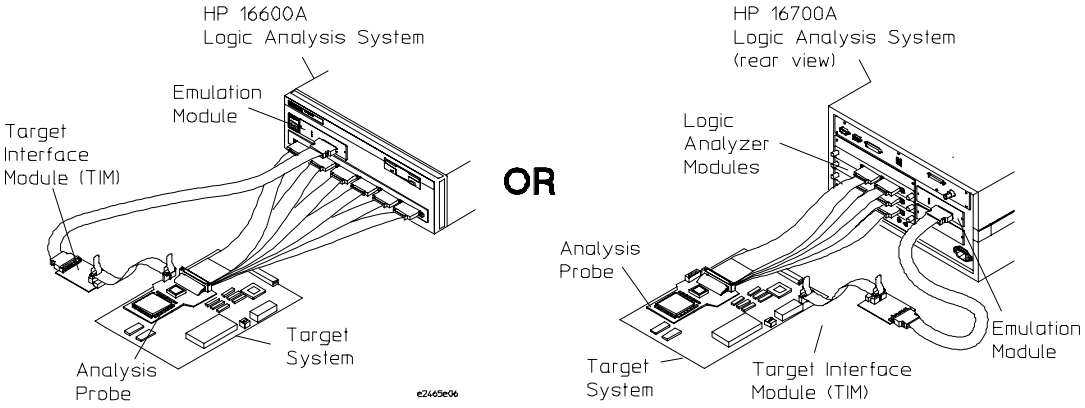
Emulation Module and Target Interface Module

The emulation module plugs into your HP 16600A/700A-series logic analysis system frame. The emulation module lets you use the target processor's built-in background debugging features, including run control and access to registers and memory. A high-level source debugger can use the emulation module to debug code running on the target system. You can use the target interface module (TIM) to connect the emulation module to the analysis probe or to a debug port on the target system.



Emulation Solution

The emulation solution includes an analysis probe, an emulation module, cables and adapters, and the HP B4620B Source Correlation Tool Set (for analyzing high-level source code). This solution is designed to be used with an HP 16600A/700A-series logic analysis system.



In This Book

This book documents the following products:

Analysis Probe

Processors supported	Product ordered	Includes
PPC604, PPC604e Up to 66 MHz 287-pin PGA package	HP E9588A Option #002	HP E2465A PGA analysis probe and inverse assembler

Emulation Solution

Processors supported	Product ordered	Includes
PPC604A, PPC604e Up to 66 MHz 287-pin PGA package	HP E9488A Option #002	HP E2465A PGA analysis probe, inverse assembler , HP 16610A emulation module, target interface module (TIM), HP B4620B Source Correlation Tool Set

Aliases

PowerPC 604e aliases: 604ev, 604ev1, 604ev2, 604e2
PowerPC 604e3 aliases: 604R

Contents

1 Overview 17

Setup Checklist 19

Setup Assistant 21

Analysis Probe 22

- Equipment supplied 22
- Minimum equipment required 24
- Additional equipment supported 24
  - Emulation module 24
  - HP B4620B Source Correlation Tool Set 24
- Logic analyzers supported 25
- Logic analyzer software version requirements 26

Emulation Module 27

- Equipment supplied 27
- Minimum equipment required 28

Emulation Solution 29

Additional Information Sources 30

2 Installing Software 31

- To install the software from CD-ROM (HP 16600A/700A) 34
- To list software packages which are installed (HP 16600A/700A) 35
- To install software on other logic analyzers 35

3 Connecting and Configuring the Analysis Probe 37

Target System Requirements 39

Analysis probe — circuit board dimensions 40

Power-on/Power-off Sequence 41

To power on HP 16600A and HP 16700A-series logic analysis systems 41
To power on all other logic analyzers 41
To power off 41

Connecting the Analysis Probe to the Target System 42

To connect to a PGA target system 44

Connecting the Analysis Probe to the Logic Analyzer 46

Analysis probe pod locations 47
To connect to the HP 16600A logic analyzer 48
To connect to the HP 16601A logic analyzer 49
To connect to the HP 16550A analyzer (two-card) 50
To connect to the HP 16554/55A/56 (two-card) 51
To connect to the HP 16554/55/56 (three-card) 52
To connect to the HP 1660A/AS/C/CS/CP logic analyzers 53
To connect to the HP 1670A/D logic analyzer 54

Configuring the Logic Analysis System 55

To load configuration and inverse assembler files—HP 16600/700 logic analysis systems 56
To load a configuration from the floppy disk (HP 16600A/700A) 57
To load configuration files—other logic analyzers 58

Connecting to the JTAG signals 60

4 Analyzing the PPC604 with a Logic Analyzer 61

Modes of Operation 63

State-per-ack mode 63
State-per-clock mode 63
Timing mode 63

Logic Analyzer Configuration 64

Format menu 64
Status Encoding 66
Predefined Logic Analyzer Symbols 68
Trigger menu 70
Qualifying Stored Data 70
Capturing an address 71
Excluding tenure from other bus masters 71
Capturing Isolated Addresses 72
Configuring for State-per-clock mode 72

Using the Inverse Assembler 73

To disable the instruction cache on the PPC604 73
To disable the cache with the emulation module: 73
To disable the cache with code: 74
To display captured state data 75
Displaying Data with the HP B4620B Source Correlation Tool Set 76
Inverse assembler output format 77
Interpreting Data 77
Branch Instructions 78
Overfetch Marking 78
Little-Endian Mode 79

To use the Invasm menu (HP 16600A and HP 16700A logic analysis systems) 80
Load 80
Display Filtering 80
Preferences 82
Options 85
To use the Invasm key (HP 1660, HP 1670, and HP 16500B/C mainframes) 86
Display Filtering 86
Code Synchronization 86
Extended Mnemonics 87

5 Symbols and Source Code in the Analyzer 91

User-Defined Symbols 93
Predefined PPC604 Symbols 93

Object File Symbols 94
To use object file symbols in the HP 16600A/700A 95
Compilers for PPC604 97
Diab Data Compiler Options 98
Green Hills Compiler Options 98
Microtec Research Inc. Compiler Options 99

Source Code 100
Requirements for source correlation 102
Inverse Assembler Generated PC (Software Address) Label 103
Access to Source Code Files 104
Source File Search Path 104
Network Access to Source Files 104
Source File Version Control 104

Triggering on Symbols and Source Code 105

Using Trigger Alignment 105
Using the Address Offset 106
Using Storage Qualification 107

6 Connecting and Configuring the Emulation Module 109

Connecting and Configuring the Emulation Module 110

Using the Emulation Control Interface 111

To start the Emulation Control Interface from the main System window 112
To start the Emulation Control Interface from the Workspace window 113
To start the Emulation Control Interface from the Workspace window for an emulation probe 113

Designing and Operating a Target System for use with the Emulation Module 114

Design requirements for PowerPC 604 114
TDO, TDI, TCK, TMS and $\overline{\text{TRST}}$ signals 114
Reset signals 114
Operating considerations for the PowerPC 604 115
JTAG bug 115
Unsupported modes 115
TLBs 115
Operating considerations for the PowerPC 604e and PowerPC 604e3 116
PowerPC 604e Rev 2 register corruption bug 116
DABR and MMCR1 registers 116
Memory model 116
Decoding of invalid instructions 116
False underflow with PowerPC 604e Rev 2 116

Unsupported modes 118
TLBs 118
PowerPC JTAG interface connections and resistors 119

Installing the Emulation Module 121

To install the emulation module in an HP 16700A-series logic analysis system or an HP 16701A expansion frame 122
To install the emulation module in an HP 16600A-series logic analysis system 124
To test the emulation module 125

Connecting the Emulation Module to the Target System 126

To connect to a target system using a JTAG port 128
To connect to a target system using an analysis probe 129

To Update Firmware 130

To display current firmware version information 131
To verify communication between the emulator and target system 131

Configuring the Emulation Module 132

To configure using the Emulation Control Interface 134
To configure using the built-in commands 135
To configure using a debugger 137
To configure restriction to real-time runs 137
To configure the JTAG clock speed (communication speed) 138
To configure reset operation 139
To set memory read delays 139
To set memory write delays 140
To generate parity bits on memory operations 140
To configure the memory read operation 141
To configure data memory write operations 142
To configure instruction memory write operations 143

Testing the emulator and target system 145

To test memory accesses 145
To test with a running program 145

7 Using the Emulation Module with a Debugger 147

Setting up Debugger Software 151

To connect the logic analysis system to the LAN 152
To change the port number of an emulation module 153
To verify communication with the emulation module 153
To export the logic analysis system's display to a workstation 154
To export the logic analysis system's display to a PC 155
To enable or disable processor caches 156

Using the Microtec Research debugger 157

Using the Software Development Systems debugger 163

8 Using the Analysis Probe and Emulation Module Together 169

Triggering the Emulation Module from the Analyzer 173

To stop the processor when the logic analyzer triggers on a line of source code (Source Viewer window) 173
To stop the processor when the logic analyzer triggers (Intermodule window) 174
To minimize the "skid" effect 175
To stop the analyzer and view a measurement 175

Tracing until the processor halts 176
To capture a trace before the processor halts 176

Triggering the Logic Analyzer from the Emulation Module 177
To trigger the analyzer when the processor halts 181
To trigger the analyzer when the processor reaches a breakpoint 182

9 Hardware Reference 185

Analysis probe—operating characteristics 187
Signal line loading 188
Theory of operation 189
Target Mastership Of Data/Address Bus 189
Normal Mode Vs. Fast-l2 Mode Of Operation 189
Direct Store Mode 189
Analysis probe—signal-to-connector mapping 190
Emulation module—operating characteristics 197
Emulation module—electrical characteristics 198

10 General-Purpose ASCII (GPA) Symbol File Format 199

GPA Record Format Summary 202
SECTIONS 204
FUNCTIONS 205
VARIABLES 206
SOURCE LINES 207
START ADDRESS 208
Comments 208

11 Troubleshooting the Analysis Probe 209

Logic Analyzer Problems 211

Intermittent data errors 211
Unwanted triggers 211
No activity on activity indicators 212
No trace list display 212
Analyzer won't power up 212

Analysis Probe Problems 213

Target system will not boot up 213
Erratic trace measurements 214
Capacitive loading 214

Inverse Assembler Problems 215

No inverse assembly or incorrect inverse assembly 215
Inverse assembler will not load or run 216

Intermodule Measurement Problems 217

An event wasn't captured by one of the modules 217

Analysis Probe Messages 218

". . . Inverse Assembler Not Found" 218
"Measurement Initialization Error" 219
"No Configuration File Loaded" 220
"Selected File is Incompatible" 220
"Slow or Missing Clock" 220
"Time from Arm Greater Than 41.93 ms" 221
"Waiting for Trigger" 221

Returning Parts to Hewlett-Packard for Service 222
To return a part to Hewlett-Packard 222
To obtain replacement parts 223

Cleaning the Instrument 223

12 Troubleshooting the Emulation Module 225

Emulation Module Troubleshooting Guide 227

Emulation Module Status Lights 228

Emulation Module Built-in Commands 229
To telnet to the emulation module 229
To use the built-in commands 230

Problems with the Target System 232
What to check first 232
To check the debug port connector signals 234
To interpret the initial prompt 235
If you see memory-related problems 239
If running from reset causes problems 241
If you see the "!ASYNC_STAT 173!" error message 241
To test the target system 242

Problems with the LAN Interface 243
If LAN communication does not work 243
If it takes a long time to connect to the network 244

Problems with the Emulation Module 245

To run the built-in performance verification test using the logic analysis system 245
To run complete performance verification tests using a telnet connection 246
If a performance verification test fails 248
TEST 3: Boundary Scan Master Test
TEST 4: I2C Test 248

Returning Parts to Hewlett-Packard for Service 249

To return a part to Hewlett-Packard 249
To obtain replacement parts 250

Cleaning the Instrument 251

Glossary 253

Index 257

1

Overview

Overview

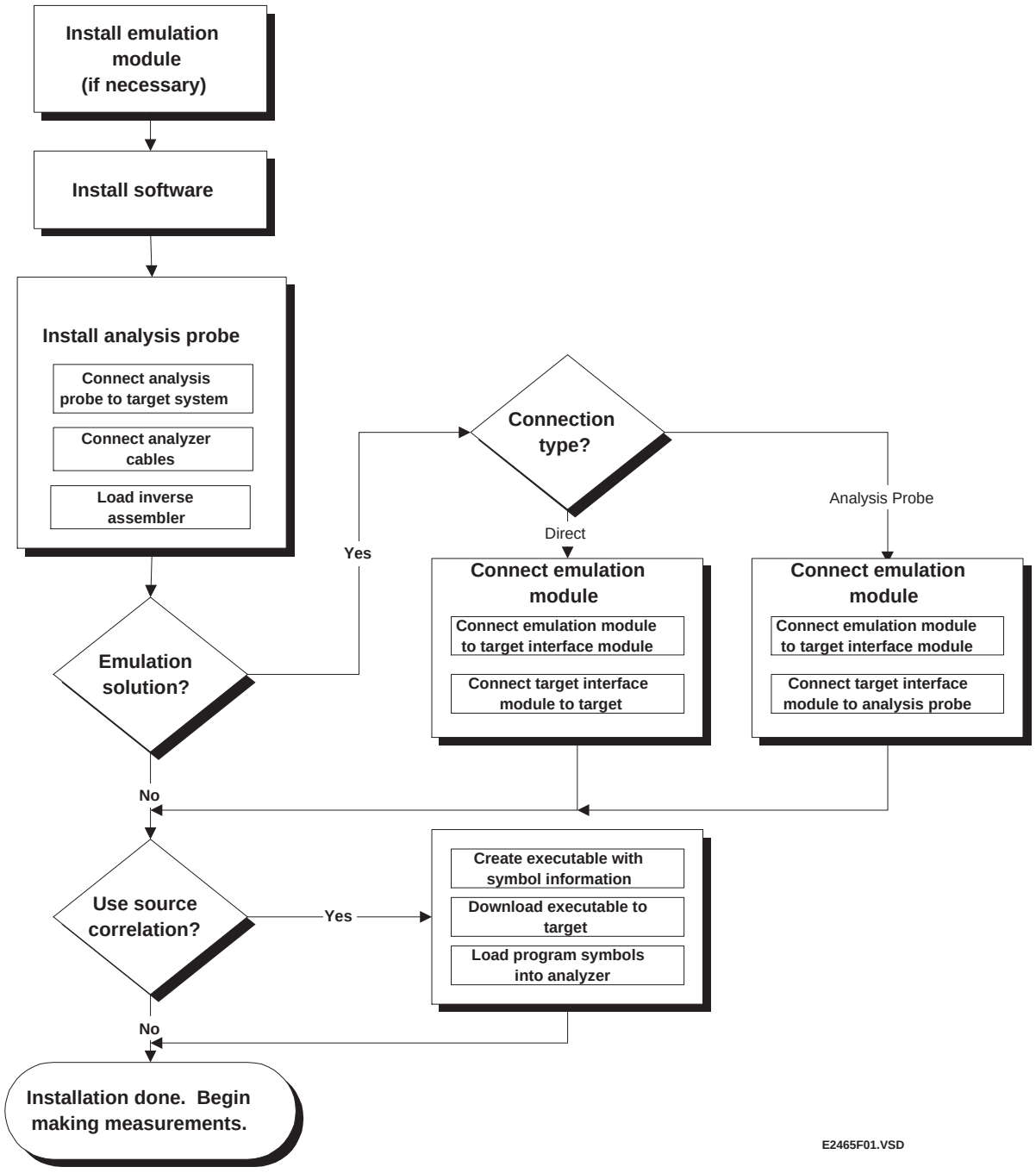
This chapter describes:

- Setup Checklist
- Setup Assistant
- Equipment used with the analysis probe (including a list of logic analyzers supported)
- Equipment used with the emulation module
- System configurations
- Additional information sources

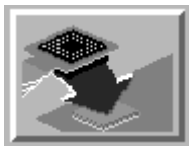
Setup Checklist

Follow these steps to connect your equipment:

- Check that you received all of the necessary equipment. See pages 22 and 27.
- If you need to install an emulation module in an HP 16600A/700A series logic analysis system, see page 110.
- Install the software. See page 31.
- If you have an HP 16600A/700A-series logic analysis system, use the Setup Assistant to help you connect and configure the analysis probe and emulation module. See page 21.
- If you do not have an HP 16600A/700A-series logic analysis system, install the analysis probe (see page 37).


Install emulation module (if necessary)
Install software
Install analysis probe
Connect analysis probe to target system
Connect analyzer cables
Load inverse assembler
Emulation solution?
Yes
No
Connection type?
Direct
Analysis Probe
Connect emulation module
Connect emulation module to target interface module
Connect target interface module to target
Connect emulation module
Connect emulation module to target interface module
Connect target interface module to analysis probe
Use source correlation?
Yes
No
Create executable with symbol information
Download executable to target
Load program symbols into analyzer
Installation done. Begin making measurements.
E2465F01.VSD

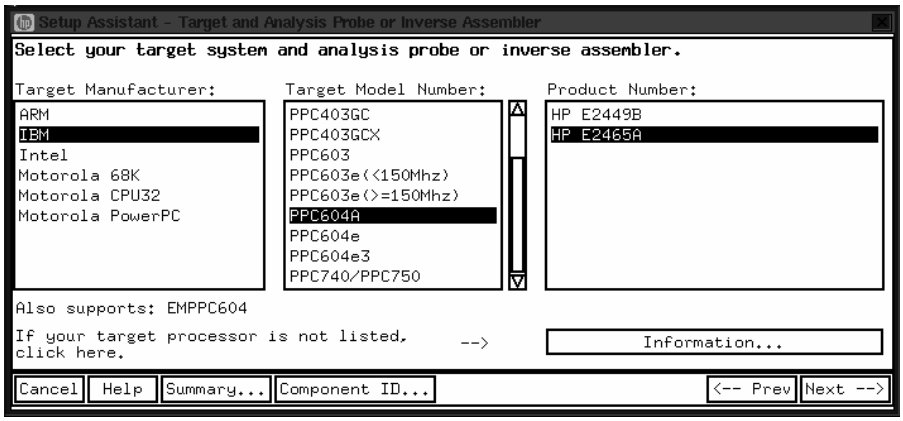
Setup Assistant



The Setup Assistant is an online tool for connecting and configuring your logic analysis system for microprocessor and bus analysis. The Setup Assistant is available on the HP 16600A and HP 16700A-series logic analysis systems. You can use the Setup Assistant in place of the connection and configuration procedures provided in this manual.

This menu-driven tool will guide you through the connection procedures for connecting the logic analyzer to an analysis probe, an emulation module, or other supported equipment. It will also guide you through connecting an analysis probe to the target system.

Start the Setup Assistant by clicking its icon in the system window.



If you ordered this analysis probe or emulation solution with your HP 16600A/700A-series logic analysis system, the logic analysis system has the latest software installed, including support for this product. If you received this product after you received your logic analysis system, see the "Installing Software" chapter (page 31).

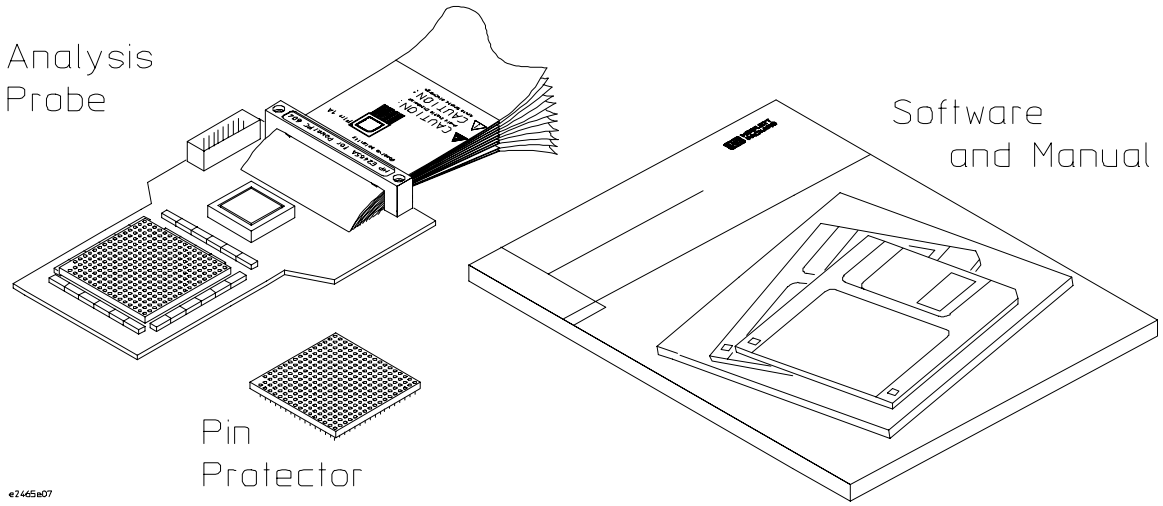
Analysis Probe

This section lists equipment supplied with the analysis probe and equipment requirements for using the analysis probe.

Equipment supplied

The equipment supplied with the analysis probe is shown in the illustration on the next page. It is listed below:

- The HP E2465A analysis probe circuit board.
- A 287-pin PGA plastic pin protector/extender socket.
- Logic analyzer configuration files and the inverse assembler on a CD-ROM (for HP 16600A/700A series logic analysis systems).
- Logic analyzer configuration files and the inverse assembler on 3.5-inch disks (for other HP logic analyzers).
- The inverse assembler and configuration files on a 3.5-inch disk (for the HP 16505A prototype analyzer).
- This User's Guide.



Equipment Supplied with the HP E2465A Analysis Probe

Minimum equipment required

For state and timing analysis of a PowerPC 604 target system, you need all of the following items.

- The HP E2465A analysis probe.
- One of the logic analyzers listed on page 25. The logic analyzer software version requirements are listed on page 26.

Additional equipment supported

Emulation module

The HP E2465A has a built-in connector for an HP 16610A emulation module.

HP B4620B Source Correlation Tool Set

The analysis probe and inverse assembler may be used with the HP B4620B Source Correlation Tool Set.

Logic analyzers supported

The table below lists the logic analyzers supported by the HP E2465A analysis probe. Logic analyzer software version requirements are shown on the following page.

The HP E2465A requires eight logic analyzer pods (136 channels) for inverse assembly. The analysis probe contains three additional pods that you can monitor.

Logic Analyzers Supported

Logic Analyzer	Channel Count	State Speed	Timing Speed	Memory Depth
16600A	204	100 MHz	125 MHz	64 k states
16601A	136	100 MHz	125 MHz	64 k states
16550A (two cards)	102/card	100 MHz	250 MHz	4 k states
16554A (two or three cards)	68/card	70 MHz	125 MHz	512 k states
16555A (two or three cards)	68/card	110 MHz	250 MHz	1 M states
16555D (two or three cards)	68/card	110 MHz	250 MHz	2 M states
16556A (two or three cards)	68/card	100 MHz	200 MHz	1 M states
16556D (two or three cards)	68/card	100 MHz	200 MHz	2 M states
1660A/AS/C/CS/CP	136	100 MHz	250 MHz	4 k states
1670A	136	70 MHz	125 MHz	64 k or .5 M states
1670D	136	100 MHz	125 MHz	64 k or 1 M states

Logic analyzer software version requirements

The logic analyzers must have software with a version number greater than or equal to those listed below to make a measurement with the HP E2465A. You can obtain the latest software at the following web site:

http://www.hp.com/go/logicanalyzer

If your software version is older than those listed, load new system software with the higher version numbers before loading the HP E2465A software.

Logic Analyzer Software Version Requirements

Logic Analyzer	Minimum Logic Analyzer Software Version for use with HP E2465A
HP 16600 Series	The latest HP 16600 logic analyzer software version is on the CD-ROM shipped with this product.
HP 1660A/AS Series	A.03.01
HP 1660C/CS/CP Series	A.02.01
HP 1670A/D Series	A.02.01
Mainframes*	
HP 16700 Series	The latest HP 16700 logic analyzer software version is on the CD-ROM shipped with this product.
HP 16500C Mainframe	A.01.06
HP 16500B Mainframe	A.03.14

* The mainframes are used with the HP 16550 and HP 16554/55/56 logic analyzers.

The HP 16505A provides a windowed user interface for the HP 16500B/C Logic Analysis System. Refer to the *HP 16505A Prototype Analyzer Installation Guide* for information on connecting the HP 16505A to the HP 16500B/C Logic Analysis System.

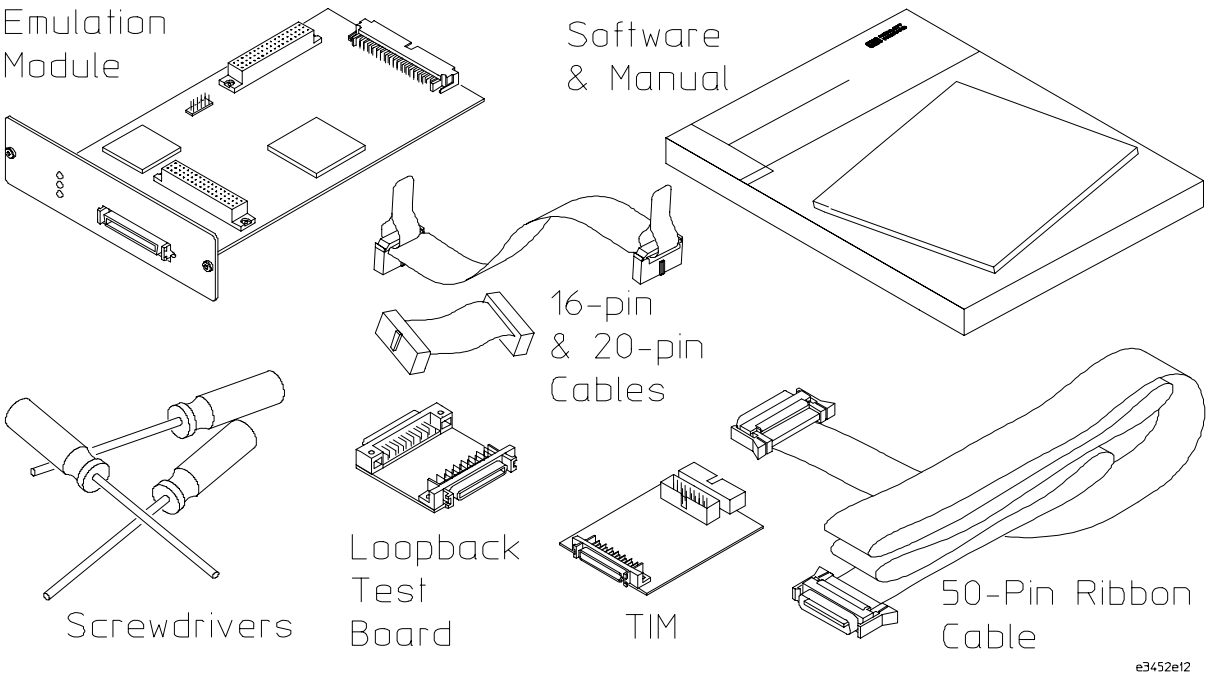
Emulation Module

This section lists equipment supplied with the emulation module and lists the minimum equipment required to use the emulation module.

Equipment supplied

The equipment supplied with your emulation module includes:

- An HP 16610A emulation module. If you ordered an emulation module as part of your HP 16600A or HP 16700A-series logic analysis system, it is already installed in the frame.
- A target interface module (TIM) circuit board.
- An emulation module loopback test board (HP part number E3496-66502).
- Firmware for the emulation module and/or updated software for the Emulation Control Interface on a CD-ROM.
- A 50-pin ribbon cable for connecting the emulation module to the target interface module.
- A 16-pin ribbon cable for connecting the target interface module to the target system or the HP E2465A Analysis Probe.
- A 20-pin ribbon cable (for use with certain third-party products).
- One Torx T-8, one Torx T-10, and one Torx T-15 screwdriver.
- This User's Guide.



Equipment Supplied with the Emulation Module

Minimum equipment required

The following equipment is required to use the emulation module:

- A method for connecting to the target system. The HP E2465A analysis probe provides a debug port connector. You can also design a debug port connector on the target system. Chapter 6 provides information on designing a debug port on the target system.
- An HP 16600A or HP 16700A logic analysis system.
- A user interface, such as a high-level source debugger or the logic analysis system's Emulation Control Interface.

Emulation Solution

An emulation solution uses the equipment and software already described in this chapter.

The combination of an analysis probe, an emulation module, and an HP 16600A or HP 16700A logic analysis system lets you both trace and control microprocessor activity on the target system.

The analysis probe supplies signals from the target microprocessor to the logic analyzer. A configuration file sets up the logic analyzer to properly interpret these signals.

You can use a debugger or the logic analysis system's Emulation Control Interface to configure and control the target processor and to download program code.

Additional Information Sources

Additional or updated information can be found in the following places:

Newer editions of this manual may be available. Contact your local HP representative.

Application notes may be available from your local HP representative or on the World Wide Web at:

http://www.hp.com/go/logicanalyzer

If you have an HP 16600A or HP 16700A logic analysis system, the **online help** for the Emulation Control Interface has additional information on using the emulation module.

The **measurement examples** include valuable tips for making emulation and analysis measurements. You can find the measurement examples in the online help in your HP 16600A/700A logic analysis system.

If you cannot easily find the information you need, send email to documentation@col.hp.com. Your comments will help HP improve future manuals. (This address is for comments only; contact your local HP representative if you need technical support.)

2

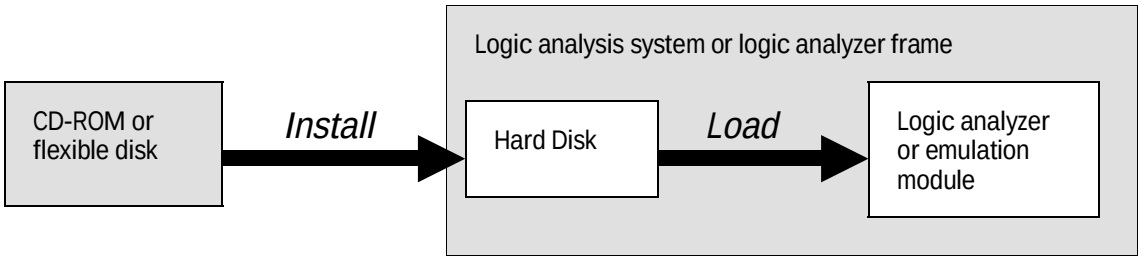
Installing Software

Installing Software

This chapter explains how to install the software you will need for your analysis probe or emulation solution.

Installing and loading

Installing the software will copy the files to the hard disk of your logic anlysis system. Later, you will need to **load** some of the files into the appropriate measurement module.



What needs to be installed

HP 16600A/700A-series logic analysis systems

If you ordered an analysis probe or emulation solution with your logic analysis system, the software was installed at the factory.

The following files are installed when you install a processor support package from the CD-ROM:

- Logic analysis system configuration files
- Inverse assembler (automatically loaded with the configuration files)
- Personality files for the Setup Assistant
- Emulation module firmware (for emulation solutions)
- Emulation Control Interface (for emulation solutions)

The HP B4620B Source Correlation Tool Set is installed with the logic analysis system's operating system. A password may be required to enable the tool set. Follow the instructions on the entitlement certificate.

Other HP logic analyzers

The following files can be installed from a floppy disk:

- Logic analyzer configuration files, which automatically load the inverse assembler

To install the software from CD-ROM (HP 16600A/700A)

Installing a processor support package from a CD-ROM will take just a few minutes. If the processor support package requires an update to the HP 16600A/700A operating system, installation may take approximately 15 minutes.

1. Insert the CD-ROM in the drive.
2. Click the **System Admin** icon.
3. Click **Install...** .

   Change the media type to "CD-ROM" if necessary.
4. Click **Apply**.
5. From the list of types of packages, select "PROC-SUPPORT."

   A list of the processor support packages on the CD-ROM will be displayed.
6. Click on the "POWERPC6XX" package.

   If you are unsure if this is the correct package, click Details for information on what the package contains.
7. Click **Install...** .

   The dialog box will display "Progress: completed successfully" when the installation is complete.
8. Click **Close**.

The configuration files are stored in /hplogic/configs/hp/ppc6xx/ppc604.

The inverse assemblers are stored in /hplogic/ia.

See Also

The instructions printed on the CD-ROM package for a summary of the installation instructions.

The online help for more information on installing, licensing, and removing software.

To list software packages which are installed (HP 16600A/700A)

- In the System Administration Tools window, click **List...** .

To install software on other logic analyzers

Consult the documentation for your logic analyzer for details.

3

Connecting and Configuring the Analysis Probe

Connecting and Configuring the Analysis Probe

This chapter shows you how to connect the logic analyzer to the target system through the analysis probe.

If you are connecting to an HP 16600A-series or HP 16700A series logic analyzer, use the Setup Assistant to connect and configure your system (see page 21). Use this manual for additional information, if desired.

If you are not using the Setup Assistant, follow the instructions given in this chapter. This chapter covers the following tasks; the order shown here is the recommended order for performing these tasks:

- Check that the target system meets the necessary requirements
- Read the power on/power off sequence
- Connect the analysis probe to the target system
- Connect the analysis probe to the logic analyzer
- Configure the logic analyzer

Target System Requirements

The analysis probe connects to the PGA socket on the target system. There are no additional target system requirements.

Clearance above the target board

See the diagram on the next page for the dimensions of the analysis probe.

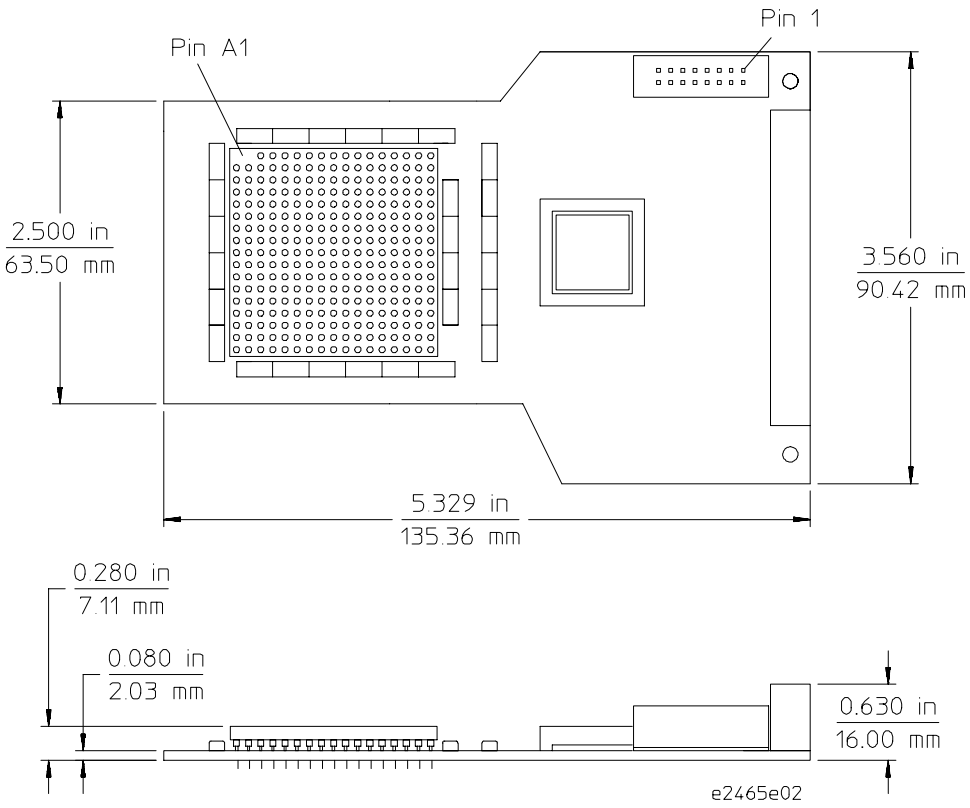
If the analysis probe interferes with components on the target system, or if a higher profile is required, additional PGA pin protectors can be used. PGA pin protectors can be ordered from Hewlett-Packard using the HP part numbers listed in the Replaceable Parts table on page 223.

See Also

The *Emulation and Analysis Solutions for Motorola/IBM PowerPC 6XX Microprocessors* data sheet, available from your HP representative, has more detailed information.

Analysis probe — circuit board dimensions

The following figure gives the dimensions for the analysis probe circuit board. The dimensions are listed in inches and millimeters.



HP E2465A Analysis Probe Circuit Board Dimension Diagram

Power-on/Power-off Sequence

Listed below are the sequences for powering on and off a fully connected system. Simply stated, your target system is always the last to be powered on, and the first to be powered off.

To power on HP 16600A and HP 16700A-series logic analysis systems

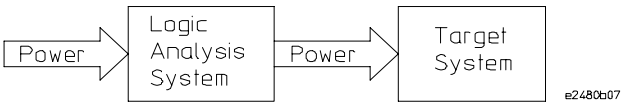
Ensure the target system is powered off.

1. Turn on the logic analyzer. The Setup Assistant will guide you through the process of connecting and configuring the analysis probe.
2. When the analysis probe is connected to the target system and logic analyzer, and everything is configured, turn on your target system.

To power on all other logic analyzers

With all components connected, power on your system in the following order:

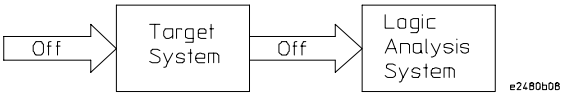
1. Logic analysis system.
2. Your target system.



To power off

Turn off power to your system in the following order:

1. Turn off your target system.
2. Turn off your logic analysis system.



Connecting the Analysis Probe to the Target System

This section explains how to connect the HP E2465A analysis probe to the target system. Connecting the analysis probe to the target system consists of the following steps, which are described on the following pages:

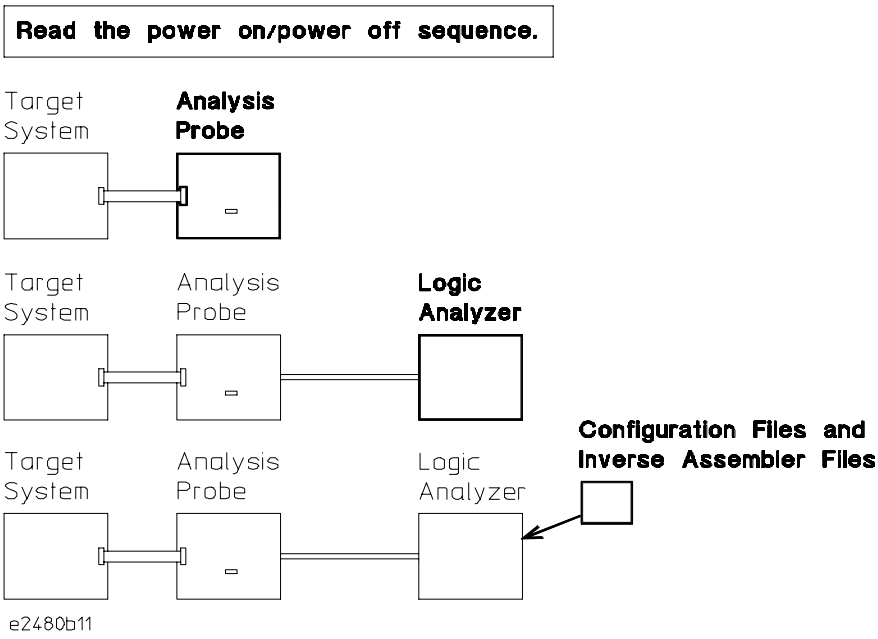
- Turn off the target system.
- Turn off the logic analyzer (unless you are using an HP 16600/16700A logic analysis system).
- Remove the PGA microprocessor from the target system.
- Insert the analysis probe into the PGA socket on the target system.
- Insert the PGA microprocessor into an extender, then insert the extender into the socket on top of the analysis probe.

The remainder of this section describes these general steps in more detail.

Protect Your Equipment

The analysis probe socket assembly pins are covered for shipment with a conductive foam wafer or conductive plastic pin protector. This is done to protect the delicate gold-plated pins from damage due to impact. When you're not using the analysis probe, protect the socket assembly pins from damage by covering them with the pin protector.

The following illustration shows the sequence for connecting the analysis probe.



Connection Sequence

To connect to a PGA target system

CAUTION

Equipment Damage. To prevent equipment damage, remove power from the target system and make sure no logic analyzer cables are connected to the analysis probe.

1. Turn off the target system and disconnect all logic analyzer cables from the analysis probe.
2. Remove the PowerPC 604 microprocessor from its socket on the target system and store it in a protected environment.
3. Prior to inserting the analysis probe connector in the socket, note the position of pin A1 on the connector and the target system socket (refer to the figure on next page).
4. Carefully align the analysis probe connector with the socket on the target system so that all pins are making contact.

CAUTION

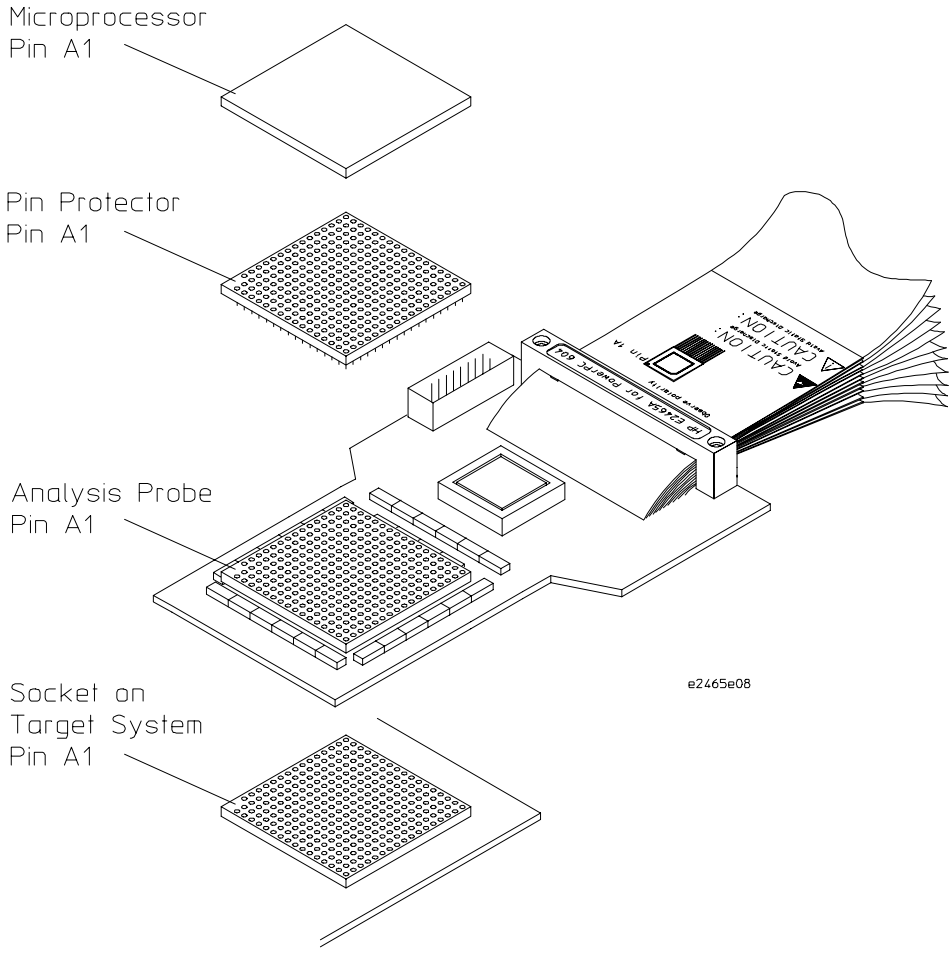
Serious damage to the target system or analysis probe can result from incorrect connection. Note the position of pin A1 on the target system, analysis probe, and microprocessor prior to making any connection. Also, take care to align the analysis probe connector pins with the target system PGA socket so that all pins are making contact.

5. Plug the analysis probe connector into the microprocessor socket on the target system.

> If the analysis probe interferes with components on the target system, or if a higher profile is required, additional PGA pin protectors can be used. PGA pin protectors can be ordered from Hewlett-Packard using the HP part numbers listed in the Replaceable Parts table on page 223.

6. Plug the PowerPC 604 microprocessor into a pin protector, then plug the pin protector into the socket on the analysis probe board.

> The PowerPC 604 microprocessor can be difficult to remove if you connect it directly to the analysis probe. Always use a pin protector between the microprocessor and the PGA socket on the analysis probe.



Connecting the Analysis Probe to the Target System

Connecting the Analysis Probe to the Logic Analyzer

This section shows the connections between the logic analyzer pod cables and the cables on the analysis probe. Use the appropriate page for your logic analyzer. The configuration file names for each logic analyzer are included with the connection diagrams.

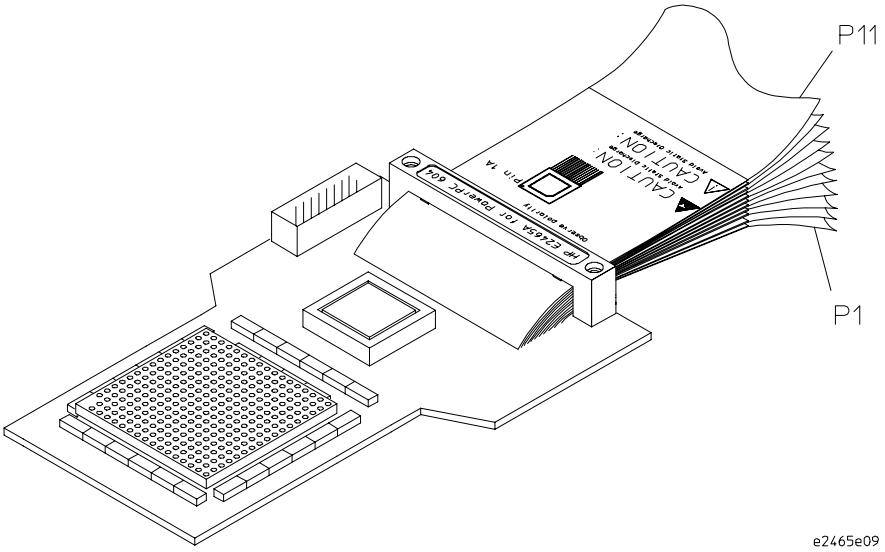
The illustration on the following page shows the analysis probe pod locations. P1 - P8 are required for inverse assembly. P9, P10, and P11 contain additional signals you might want to monitor.

This section shows diagrams for connecting the analysis probe to the logic analyzers listed below:

- HP 16600A logic analysis system (page 48)
- HP 16601A logic analysis system (page 49)
- HP 16550A logic analyzers (two cards) (page 50)
- HP 16554/55/56 logic analyzers (two or three cards) (pages 51 and 52)
- HP 1660A/AS/C/CS/CP logic analyzers (page 53)
- HP 1670A/D logic analyzers (page 54)

Analysis probe pod locations

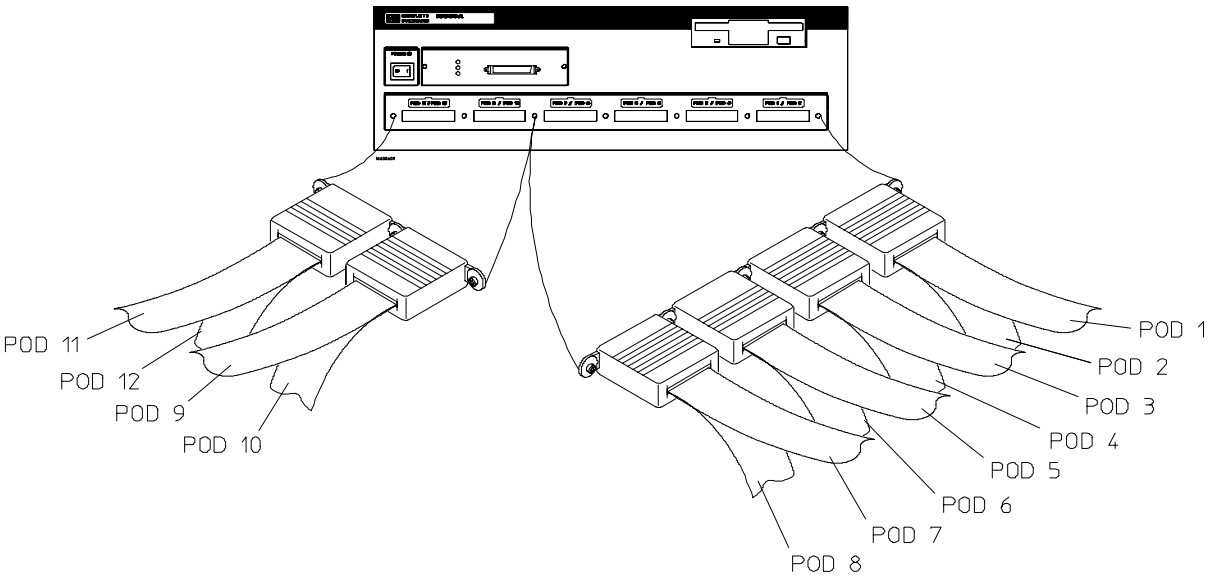
The illustration below shows the pod locations on the analysis probe.



HP E2465A Analysis Probe Pod Locations

To connect to the HP 16600A logic analyzer

Use the figure and table below to connect the analysis probe to the HP 16600A logic analysis system.



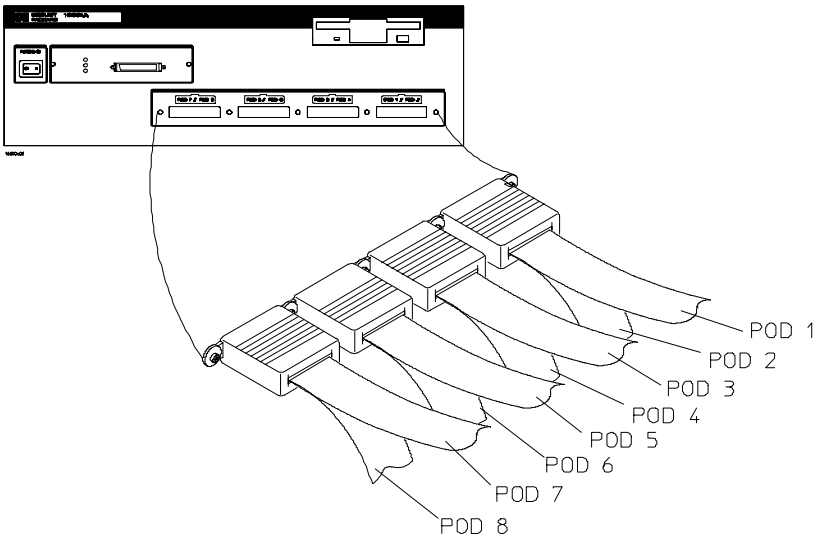
HP 16600A Pods 12 - 7	Pod 12	Pod 11	Pod 10	Pod 9	Pod 8	Pod 7
HP E2465A Connector	P6 DATA	P5 DATA	P7 STAT	P8 STAT clk ↑	P2 ADDR	P1 ADDR

HP 16600A Pods 6 - 1	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P9 Parity	P10 JTAG	P11 PLL	--	P4 DATA_B	P3 DATA_B

Configuration File
Use configuration file C604F for the HP 16600A logic analyzer.

To connect to the HP 16601A logic analyzer

Use the figure and table below to connect the analysis probe to the HP 16601A logic analyzer.



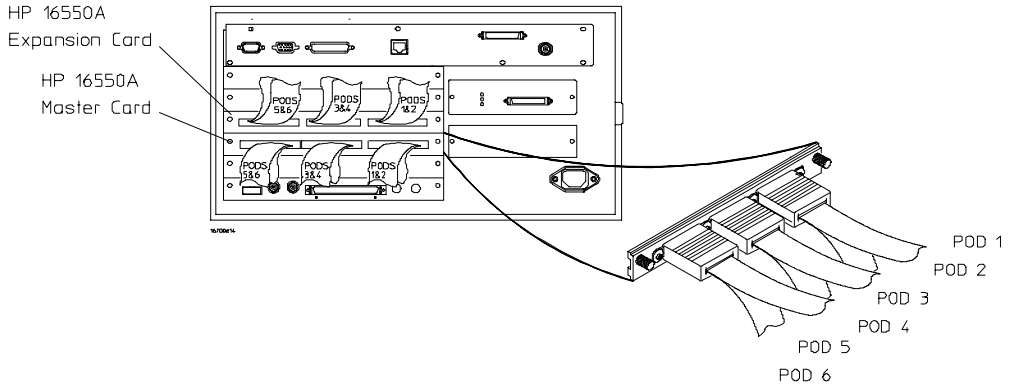
HP 16601A Pods 8 - 5	Pod 8	Pod 7	Pod 6	Pod 5
HP E2465A Connector	P2 ADDR	P1 ADDR	P7 STAT	P8 STAT clk ↑

HP 16601A Pods 4 - 1	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P4 DATA_B	P3 DATA_B	P6 DATA	P5 DATA

Configuration File
Use configuration file C604F for the HP 16601A logic analyzer.

To connect to the HP 16550A analyzer (two-card)

Use the figure and table below to connect the analysis probe to the two-card HP 16550A logic analyzer.



HP 16550A Expansion Card Pod	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B	P2 ADDR	P1 ADDR

HP 16550A Master Card Pod	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P7 STAT	P8 STAT clk ↑	P9 Parity	P10 JTAG	P11 PLL	--

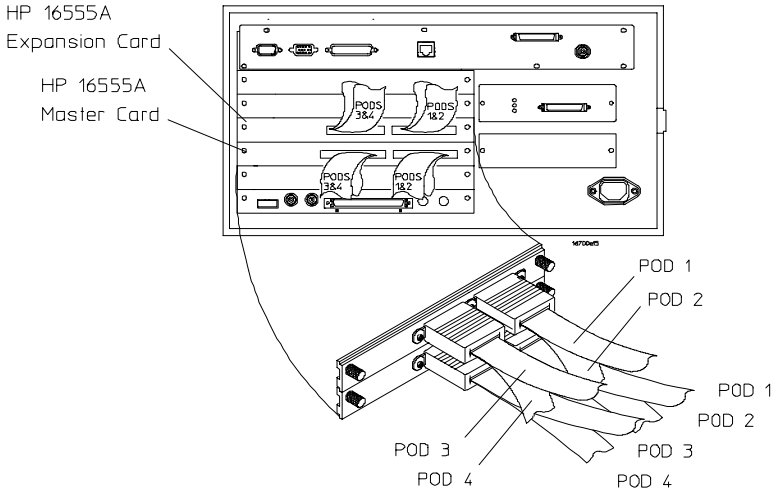
Configuration File
Use configuration file C604F for the HP 16550A logic analyzer.

Second Analyzer Machine
The configuration file C604F can support a second analyzer machine. To configure a second machine, use the logic analyzer Configuration menu to reassign Master Card pods 1 through 4 to Analyzer 2.

To connect to the HP 16554/55A/56 (two-card)

Use the figure and table below to connect the analysis probe to the two-card HP 16554A/55A/56A and HP 16555D/56D logic analyzers.

Two-card HP 16554/55/56A logic analyzer connections



HP 16554A/55A/56A Expansion Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B

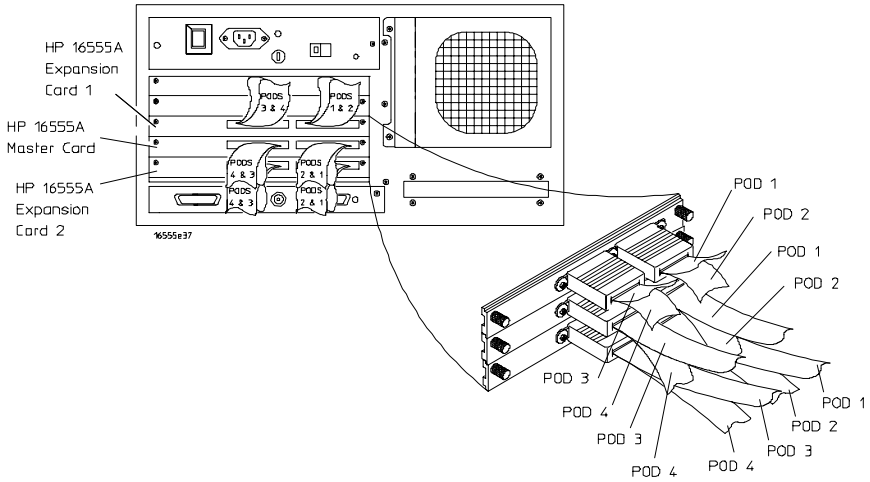
HP 16554A/55A/56A Master Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P2 ADDR	P1 ADDR	P7 STAT	P8 STAT clk ↑

Configuration File
Use configuration file C604M for the two-card HP 16554/55/56.

To connect to the HP 16554/55/56 (three-card)

Use the figure and table below to connect the analysis probe to the three-card HP 16554A/55A/56A and HP 16555D/56D logic analyzers.

Three-card HP 16554/55/56A logic analyzer connections

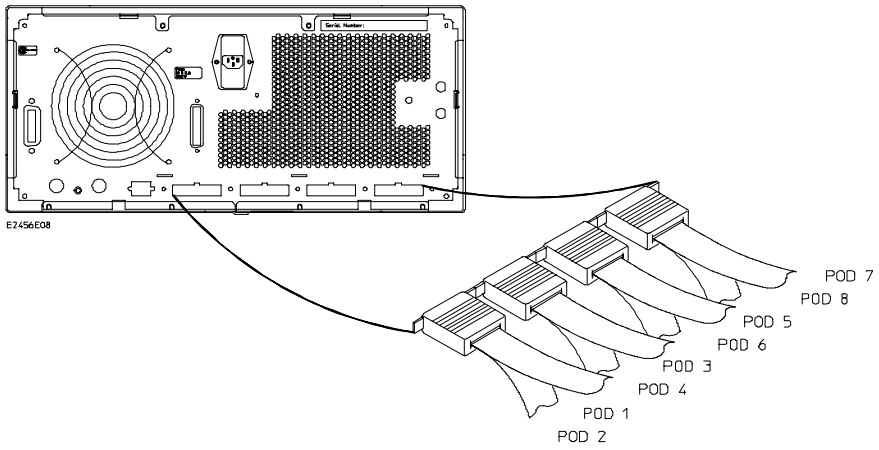


Exp. Card 1 Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B
Master Card Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	--	P11 PLL	P7 STAT	P8 STAT clk ↑
Exp. Card 2 Pod	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P9 PARITY	P10 JTAG	P2 ADDR	P1 ADDR

Configuration File
Use configuration file C604M3 for the three-card HP 16554/55/56.
Second Analyzer Machine
Configuration file C604M3 supports a second analyzer machine. To configure a second machine, use the logic analyzer Configuration menu to reassign Master Card pods 3 and 4 and Expansion Card 2 pods 3 and 4 to Analyzer 2.

To connect to the HP 1660A/AS/C/CS/CP logic analyzers

Use the figure and table below to connect the analysis probe to the HP 1660A/AS/C/CS/CP logic analyzers.

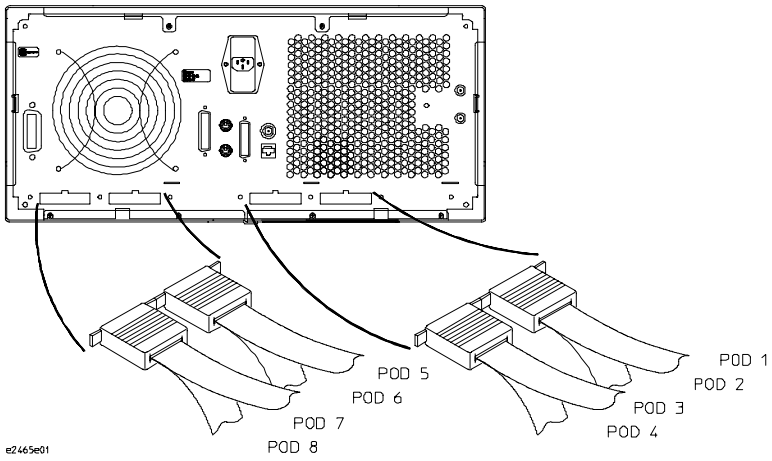


HP 1660 Pod	Pod 1	Pod 2	Pod 3	Pod 4	Pod 5	Pod 6	Pod 7	Pod 8
HP E2465A Connector	P8 STAT clk ↑	P7 STAT	P1 ADDR	P2 ADDR	P3 DATA_B	P4 DATA_B	P5 DATA	P6 DATA

Configuration File
Use configuration file C604J for the HP 1660A/AS/C/CS/CP logic analyzers.

To connect to the HP 1670A/D logic analyzer

Use the figure and table below to connect the analysis probe to the HP 1670A/D logic analyzers.



HP 1670A/D Pod	Pod 8	Pod 7	Pod 6	Pod 5	Pod 4	Pod 3	Pod 2	Pod 1
HP E2465A Connector	P6 DATA	P5 DATA	P4 DATA_B	P3 DATA_B	P2 ADDR	P1 ADDR	P7 STAT	P8 STAT clk ↑

Configuration File
Use configuration file C604M for the HP 1670A/D logic analyzers.

Configuring the Logic Analysis System

You configure the logic analyzer by loading a configuration file. The information in the configuration file includes:

- Label names and channel assignments for the logic analyzer
- Inverse assembler file name

The configuration file you use is determined by the logic analyzer you are using. The configuration file names are listed with the logic analyzer connection tables, and in a table at the end of this section.

The procedures for loading a configuration file depend on the type of logic analyzer you are using. There is one procedure for the HP 16600/700 series logic analysis systems, and another procedure for the HP 1660-series, HP 1670-series, and logic analyzer modules in an HP 16500B/C mainframe. Use the appropriate procedures for your analyzer.

To load configuration and inverse assembler files—HP 16600/700 logic analysis systems

If you did not use Setup Assistant, you can load the configuration and inverse assembler files from the logic analysis system hard disk.

1 Click on the File Manager icon. Use File Manager to ensure that the subdirectory /hplogic/configs/hp/ppc6xx/ppc604 exists.

> If the above directory does not exist, you need to install the POWERPC6XX Processor Support Package. Close File Manager, then use the procedure on the CD-ROM jacket to install the POWERPC6XX Processor Support Package before you continue.

2 Using File Manager, select the configuration file you want to load in the /hplogic/configs/hp/ppc6xx/ppc604 directory, then click Load. If you have more than one logic analyzer installed in your logic analysis system, use the Target field to select the machine you want to load.

The logic analyzer is configured for PPC604 analysis by loading the appropriate PPC604 configuration file. Loading the indicated file also automatically loads the inverse assembler. The configuration file names are located at the bottom of the table showing the connections for your particular logic analyzer. They are also shown in the following table.

3 Close File Manager.

To load a configuration from the floppy disk (HP 16600A/700A)

The preferred method is to install this functionality from the CD-ROM onto the hard disk and load from the hard disk, also described in this manual.

To install a configuration and inverse assembler file from the floppy disk that was shipped with your HP analysis probe:

1. Install the floppy disk in the floppy drive on the HP 16600A/16700A-series logic analysis system mainframe.
2. In the Logic Analysis System window, click the **File Manager** icon.
3. In the File Manager window:
   - Set Current Disk to Flexible Disk.
   - Set Target to the analyzer you wish to configure.
   - Click the name of the desired configuration file in the Contents frame. The Contents frame lists the configuration files and inverse assembler files available on the floppy disk. These may be either DOS or LIF format files. Either format can be loaded directly into the appropriate logic analyzers.

   Note that the logic analyzers read both DOS and LIF formats. However, only DOS formatted floppy disks can be used to store configurations and data. LIF format floppy disks are read-only.
4. Click **Load**.

   The configuration file you choose will set up the logic analyzer and associated tools. You may see Information, Error, and Warning dialogs that say your configuration has been loaded, and advise you about making proper connections.
5. Click the **Workspace window** icon to see the arrangement of analysis tools in your configuration.
6. Right-click the logic analyzer icon in your configuration and choose its **Setup** button to see the way your configuration file defined the Config, Format, and Trigger options.

   Under the Format tab, buses are labeled, and bits included in each bus are identified by an asterisk "*".

   This procedure restores the configuration that was in effect when the configuration file was saved. Because the file was not saved using your system, you may receive error messages about loading the enhanced inverse assembler or about pods that were truncated. Click the Config, Format, and

Trigger tabs and modify the configuration to satisfy your measurement desires. Then you can save your customized configuration to DOS format using the File→Save Configuration selection in any of your tool windows, or clicking the Save tab in the File Manager. For details about how to save configuration files, open the Help window.

To load configuration files—other logic analyzers

If you have an HP 1660-series, HP 1670-series, or logic analyzer modules in an HP 16500B/C mainframe, use these procedures to load the configuration file and inverse assembler.

The first time you set up the logic analyzer, make a duplicate copy of the flexible disk. For information on duplicating disks, refer to the reference manual for your logic analyzer.

For logic analyzers that have a hard disk, you might want to create a directory such as PPC604 on the hard drive and copy the contents of the floppy onto the hard drive. You can then use the hard drive for loading files.

1. Insert the HP E2465A disk in the front disk drive of the logic analyzer.
2. Go to the Flexible Disk menu.
3. Configure the menu to "Load" the analyzer configuration from disk.
4. Select the appropriate module (such as "100/500 MHz LA" or "Analyzer") for the load.
5. Use the knob to select the appropriate configuration file.

   Your configuration file choice depends on which analyzer you are using. See table on following page.
6. Execute the load operation to load the file into the logic analyzer.

   The logic analyzer is configured for PPC604 analysis by loading the appropriate PPC604 configuration file. Loading the indicated file also automatically loads the inverse assembler. The configuration file names are located at the bottom of the table showing the connections for your particular logic analyzer. They are also shown in the following table.

7 If you are using the HP 16505A Prototype Analyzer, insert the "16505 Prototype Analyzer" flexible disk into the disk drive of the prototype analyzer, and update the HP 16505A from the Session Manager. You must close your workspace to run the update.

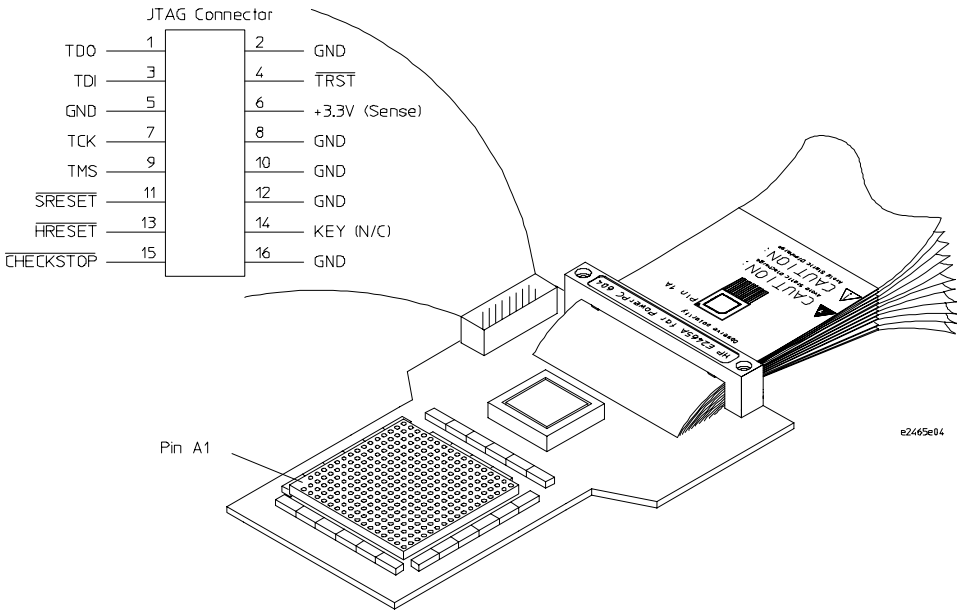
The HP 16505A Prototype Analyzer requires software version A.01.30 or higher to work with the HP E2465A.

Logic Analyzer Configuration Files

Analyzer Model	Analyzer Module Description	604 Configuration File
16600A	na	C604F
16601A	na	C604F
16550A (two card)	100 MHz STATE 500 MHz TIMING	C604F
16554A (two card)	0.5M SAMPLE 70/250 MHz LA	C604M
16555A/D (two card)	1.0M SAMPLE 110/500 MHz LA	C604M
16556A/D (two card)	1.0M SAMPLE 100/400 MHz LA	C604M
16554A (three card)	0.5M SAMPLE 70/250 MHz LA	C604M3
16555A/D (three card)	1.0M SAMPLE 110/500 MHz LA	C604M3
16556A/D (three card)	1.0M SAMPLE 100/400 MHz LA	C604M3
1660A/AS/C/CS	na	C604J
1670A/D	na	C604M

Connecting to the JTAG signals

The JTAG signals are routed to the JTAG connector. The figure below shows the location of the connector and signals.



JTAG Connector and Signals

Analyzing the PPC604 with a Logic Analyzer

Analyzing the PPC604 with a Logic Analyzer

This chapter describes modes of operation for the HP E2465A analysis probe. It also describes data, symbol encodings, and information about the inverse assembler.

The information in this chapter is presented in the following sections:

- Modes of operation
- Logic analyzer configuration
- Using the inverse assembler

Modes of Operation

The HP E2465A analysis probe can be used in three different analysis modes: State-per-ack, State-per-clock, or Timing. The following sections describe these modes and how to configure the logic analyzer for each mode.

State-per-ack mode

In State-per-ack mode, the logic analyzer uses trigger sequencer store qualification to capture only address and data-acknowledge cycles. This is the default mode set up by the configuration files.

State-per-ack mode provides the greatest information density in the logic analyzer acquisition memory.

State-per-clock mode

In State-per-clock mode, every clock cycle is captured by the logic analyzer, including idle and wait states between and during tenures. To configure the logic analyzer for State-per-clock mode, use the Trigger menu to change the trigger store qualification to "anystate". For additional information, refer to the "Trigger menu" section.

Timing mode

In Timing mode, the logic analyzer samples the microprocessor pins asynchronously, typically with 4-ns resolution. To configure the logic analyzer for timing analysis, select the Configuration menu of the logic analyzer, select the Type field for Analyzer 1, and select Timing.

Logic Analyzer Configuration

The following sections describe the logic analyzer configuration as set up by the configuration files.

Format menu

This section describes the organization of PPC604 signals in the logic analyzer's Format menu.

The configuration files contain predefined format specifications. These format specifications include all labels for monitoring the microprocessor. The tables on the following pages show the signals used in the STAT label and the predefined symbols set up by the configuration files.

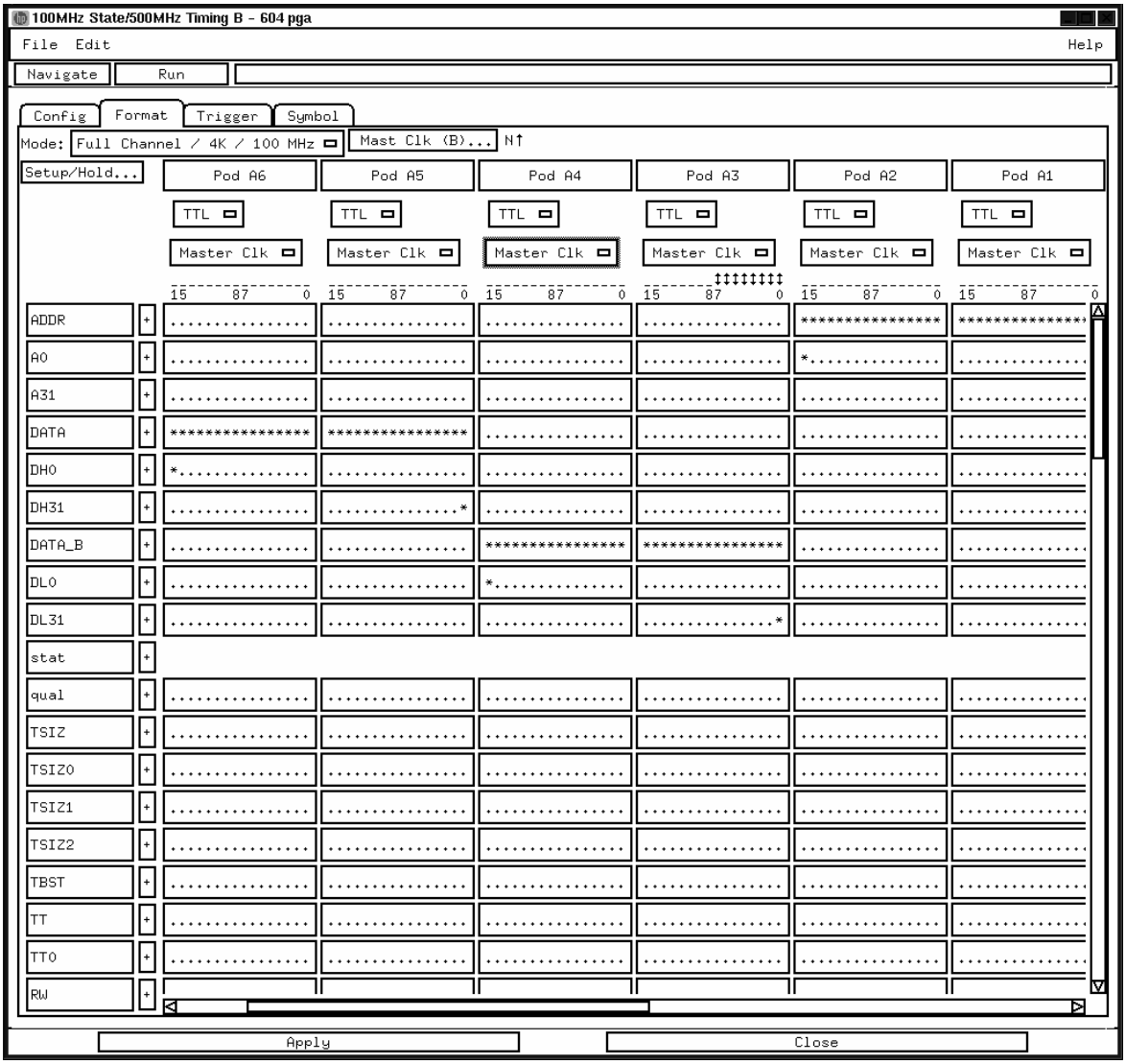
The HP logic analyzers and the PowerPC use opposite conventions to designate individual signals on a bus. In PowerPC nomenclature, bit 0 is the most significant; in the logic analyzers, bit 0 is the least significant. In PowerPC, A0 is the most significant bit of the address bus; on the analyzer, this bit is called ADDR31.

Most Significant	Least Significant	
A0	A31	PowerPC
ADDR31	ADDR0	Analyzer

This may cause confusion in the waveform menus when using Channel Mode Sequential or Individual.

Do not modify the ADDR, DATA, or STAT labels in the format specification if you want inverse assembly. Changes to these labels may cause incorrect or incomplete inverse assembly.

The configuration software sets up the analyzer format menu to display either eight or eleven pods of data, depending on the analyzer. The following figure shows the Format menu for the PowerPC 604.



Format Menu

Status Encoding

Each of the bits of the STAT label is described in the table below.

The inverse assembler uses STAT bits TC0, TSIZ0...2, TT0...3 , TBST, TA, ARTRY, DRTRY, ABB, DBB, and AACK. The signal-to-connector tables in the "Hardware Reference" chapter list all the PPC604 signals probed and their corresponding analyzer channels.

Status Bit Description

Status Bit	Description
BR	The PowerPC 604 asserts Bus Request to indicate that it has business to conduct on the address bus.
BG	The memory system asserts Bus Grant to allow the 604 onto the address bus.
ABB	Address Bus Busy indicates that the address bus is in use.
TS	The PowerPC 604 asserts TS for one cycle to commence a transaction. It also serves as the data bus request signal if the TT signals indicate a data transfer.
XATS	XATS commences a "programmed i/o" (PIO) sequence in the extended address transfer protocol.
DBG	The memory system asserts Data Bus Grant to allow the 604 onto the data bus.
DBWO	The memory system may assert Data Bus Write Only to allow the 604 to envelope a data write (snoop push, typically) between the address and data phases of a data read.
DBB	Indicates Data Bus Busy.
AACK	The memory system asserts AACK for one cycle to acknowledge an address.
ARTRY	The memory system may assert ARTRY to cause the 604 to back off the bus and retry the transaction.
TA	The memory system asserts TA to acknowledge a data transaction.
DRTRY	The memory system may assert DRTRY to cancel the effect of a TA in the previous cycle.
TEA	The memory system may assert TEA to indicate a transfer error, e.g. an unmapped part of the address space.
TT 0:4	The Transfer Type signals indicate the direction and purpose of a bus transaction.

Status Bit	Description
R/W (TT1)	TT1 is high for a read, low for a write.
TC 0:2	The Transfer Code outputs provide further information about the current transfer. For a read, they indicate whether instructions or operands are being fetched.
TBST	When asserted, TBST indicates a four-beat burst transfer of eight words.
TSIZ 0:2	Indicates the size for the data transfer, in conjunction with TBST.
WT	Write Through indicates a write-through transaction that should be pushed through local caches to shared memory.
CI	Cache Inhibit indicates that the 604 will not cache a read.
GBL	Global indicates the transaction is global, i.e., should be snooped by all other caching devices on the bus.
SRESET	Input asserted causes the 604 to undergo a soft reset.
HRESET	Input asserted causes the 604 to undergo a hard reset.
CKSTP_IN	Input asserted causes the 604 to undergo machine check processing.
INT	Input indicates an external interrupt is pending.
CKSTP_OUT	Output indicates that the 604 has entered the machine check state, i.e., stopped.

Predefined Logic Analyzer Symbols

The configuration software sets up symbol tables on the logic analyzer. The tables define a number of symbols which make several of the STAT fields easier to interpret. The following table lists the symbol descriptions.

Symbol Description

Label	Symbol	Encoding
acks	idle	1111
	ARTRY	xxx0
	DRTRY	0xxx
	TA AACK	x00x
	AACK	xx0x
	TA	x0xx
TT	Clean Block	00000
	Flush Block	00100
	SYNC	01000
	Kill Block	01100
	EIEIO	10000
	ECOWX write	10100
	TLBIE	11000
	ECIWX rd	11100
	LWARX hit	00001
	reserved	00101
	TLBSYNC	01001
	?ICBI	01101
	?reserved	1xx01
	Wr/Flush	00010
	Wr/Kill	00110
	Read	01010
	Rd/Modify	01110
	STWCX.	10010
	?reserved	10110
	LWARX cach inhib	11010
	LWARX miss	11110
	?reserved	00x11
	?Rd/No cache	01011
	?reserved	01111
	?reserved	1xx11

Label	Symbol	Encoding
R/W	rd	1
	wr	0
TSIZ	burst	xxx0
	8 byte	0001
	1 byte	0011
	2 byte	0101
	3 byte	0111
	4 byte	1001
	5?byte	1011
	6?byte	1101
	7?byte	1111
STAT	inst fetch	1xxx xxxx xxxx xxxx xxxx x1xx x1xx 0xxx

The least significant bit of the TSIZ label is the TBST signal.

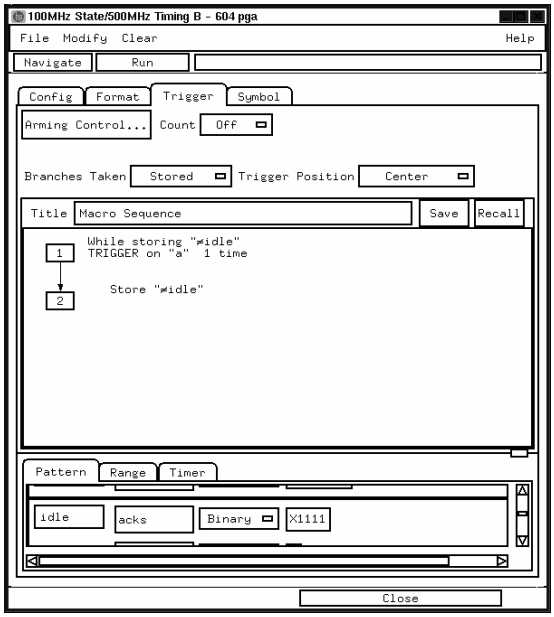
An instruction fetch is indicated by AACK asserted (address & qualifiers valid), R/W (TT1) asserted for read, and TC0 and oAdd asserted.

Trigger menu

This section describes some PowerPC 604-specific considerations in triggering the analyzer. You can use the Trigger menu to change the triggering and storage qualification to include or exclude specified cycles. The trigger specification set up by the software stores all states. If you modify the trigger specification to store only selected bus cycles, incorrect or incomplete disassembly may be displayed.

Qualifying Stored Data

The trigger menu determines what will be acquired by the analyzer and when it will be acquired. The E2465A software preconfigures a storage qualification term to exclude wait and idle states from the analyzer's memory. The following figure shows the trigger menu.



The configuration software renames pattern term "g" to "idle" and assigns it a pattern with AACK, ARTRY, TA, and DRTRY, all high (de-asserted). The sequencer is programmed to store only states ≠ idle. That is, only states where one or more of these signals is asserted will be stored.

Capturing an address

To accurately trigger on a specific address, enter the address in the ADDR field of a trigger term and also enter 0 in the AACK field of the term. This will ensure against false triggering on a floating address bus.

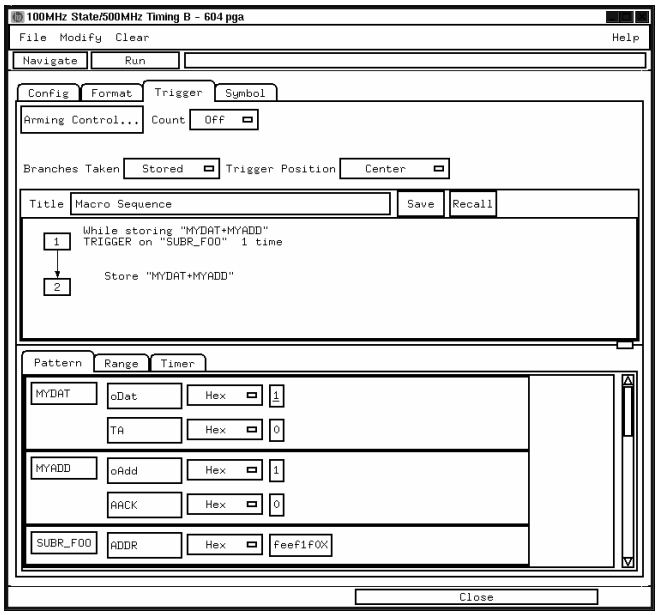
The instruction addresses presented on the PowerPC 604 address bus always end in hex 0 or hex 8. When the instruction cache is enabled, the 604 will burst four data beats per address and will not update the address as it bursts. To accurately trigger on the fetch of a particular address when bursting, the least significant five bits of the address should be "don't cares. " Change the base of the ADDR label to Binary to enter the 5 X's.

Excluding tenure from other bus masters

In a system in which devices other than the PowerPC 604 assume bus mastership, you can exclude tenures by other masters by defining new terms and changing store qualification. Define the terms MYDAT and MYADD, where:

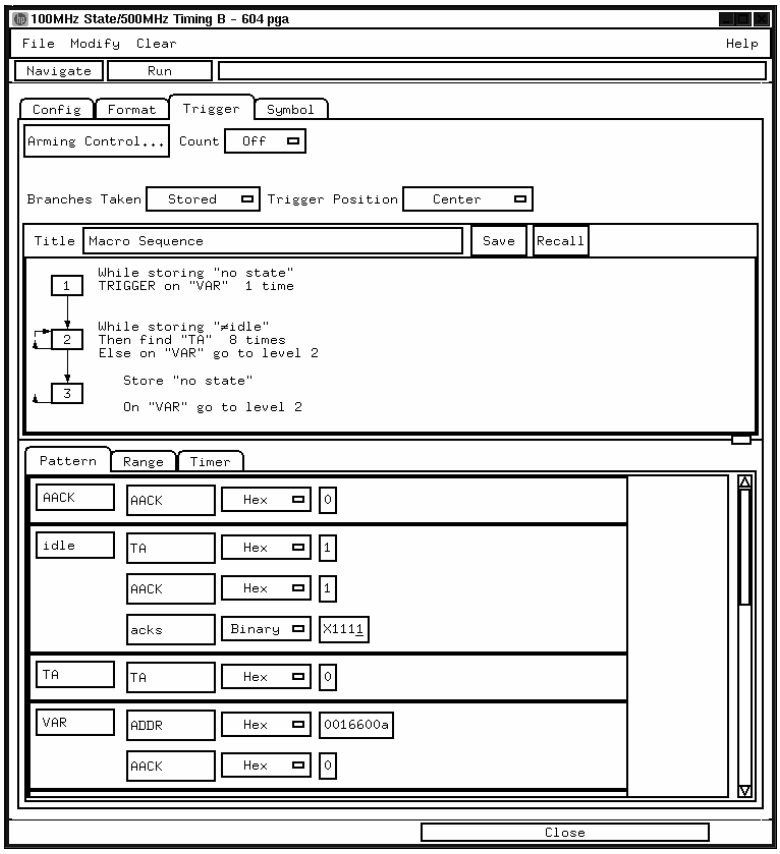
MYDAT = (oDat = 1)• (TA = 0)
MYADD = (oAdd = 1)• (AACK = 0)

Change the store qualification to the ORed combination MYDAT + MYADD, as shown in the following figure.



Capturing Isolated Addresses

The loose coupling of the address and data buses on the PowerPC 604 makes it more difficult to trace only activities associated with a given address, such as writes to a variable. Depending on how deep the pipeline is, an address of interest may be followed by up to eight data beats before the data associated with the address appears on the bus. One technique to trace writes to a variable is shown below. (The data cache is off or in write-through mode.)



Configuring for State-per-clock mode

To configure the analyzer to store wait and idle states, change the storage qualification from "≠ idle" to "anystate". Doing so will capture all states (state-per-clock).

Using the Inverse Assembler

This section discusses the general output format of the inverse assembler and processor-specific information.

Before the inverse assembler will correctly disassemble information captured on the PPC604 address bus, you must make sure the target system's cache has been disabled.

To disable the instruction cache on the PPC604

When the instruction cache is enabled, many PowerPC instructions are executed from the cache and do not appear on the external bus. To get an execution trace on the bus, the instruction cache can be disabled. This must be done in supervisor mode.

To disable the cache with the emulation module:

Use your debugger or the Emulation Control Interface to configure the HID0 register.

Register values for controlling the cache

Value	Meaning
0000 8000	Enable Instruction Cache
0000 4000	Enable Data Cache
0000 0800	Invalidate Instruction Cache
0000 0400	Invalidate Data Cache

To disable the cache with code:

- Disable the cache with the following code:

```
mfspr   r3  hid0
rlwinm  r3  r3  0  17  15   # clear bit 16 (ICE)
mtspr   hid0  r3
isync
```

- To also disable the data cache use:

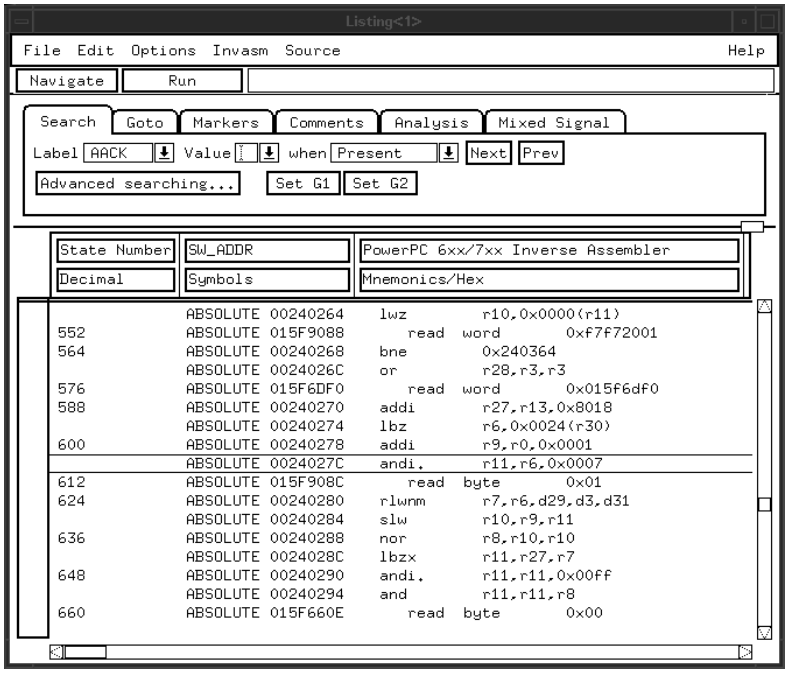
```
mfspr   r3  hid0
rlwinm  r3  r3  0  18  15   # clear ICE and DCE
mtspr   hid0  r3
isync
```

- To invalidate and disable the caches use:

```
mfspr   r3  hid0
ori     r3  0C00            # set ICFI and DCFI
mtspr   hid0  r3
rlwinm  r3  r3  0  22  19   # clear ICFI and DCFI
mtspr   hid0  r3
rlwinm  r3  r3  0  18  15   # clear ICE and DCE
mtspr   hid0  r3
isync
```

To display captured state data

The logic analyzer displays captured state data in the Listing menu. The inverse assembler display is obtained by setting the base for the DATA label to Invasm. The following figure shows a typical Listing menu.

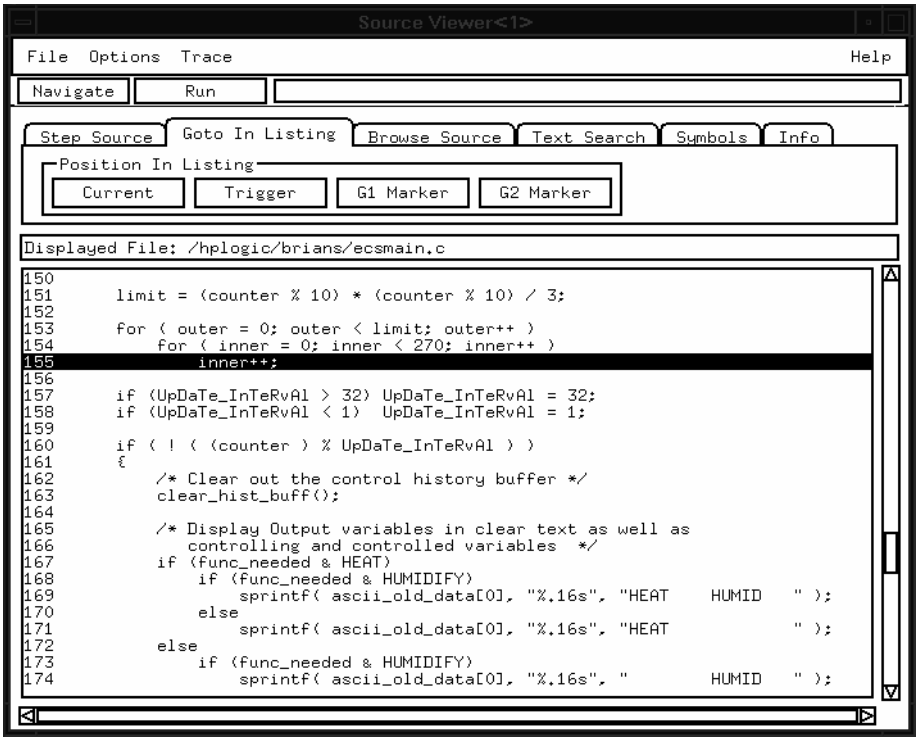


Listing Menu

This paragraph applies only when using the HP 16500C system. The columns on the left of the inverse assembly data display are the least significant hexadecimal digits of an instruction or burst address. These may be useful for matching an execution trace to an assembly listing. Because the PowerPC 604 presents one address and then reads two or eight instructions for each address, the less-significant bits are synthesized by the disassembler. On the HP 16600A/16700A-series logic analysis systems, the entire synthesized address appears under the label "SW_ADDR". The actual address bits presented by the PowerPC 604 may be observed under the ADDR label.

Displaying Data with the HP B4620B Source Correlation Tool Set

Source correlation correlates the addresses from cache with the high-level code execution. The figure below shows execution of data that is correlated to the data shown on the previous page.



Source Correlation Tool Set Data

Display Software Address Enabling this dialog allows correlation of instruction fetches external to the cache with the source code. When this dialog is enabled, you must specify the number of wait states (clocks) before the address is valid. The number of wait states is typically dependent on the type of memory hardware being accessed.

Inverse assembler output format

The following paragraphs explain the operation of the inverse assembler and the results you can expect under certain conditions.

Interpreting Data

General purpose registers are displayed as r0, r1, ..., r31. Floating point registers are displayed as f0, f1, ..., f31. Condition registers are displayed as cr0, cr1, ..., cr7. Special purpose registers are displayed using their mnemonic.

Most numerical data is displayed in hexadecimal, for example, "lwz r28, 0x0044(r1)."

Bit numbers and shift counts are displayed in decimal with a "d" prefix, for example, "cror d31,d31,d31."

A few instructions display their operands in binary with a "b" prefix, for example, "mtfsfi 4,b0101."

The inverse assembler decodes the full 32-bit mode PowerPC instruction set architecture. Instructions that are 64-bit mode or optional instructions not implemented on the PPC604 are decoded as "illegal". Any instruction that does not decode to a valid opcode is shown as "unknown".

When these unimplemented opcodes are encountered, the instruction mnemonic has a "?" prefix. If a reserved bit is set in an instruction opcode field, a "?" is appended most often to the mnemonic, but in some cases to an operand.

An instruction word of 00000000 is decoded as "illegal." Otherwise, if an opcode is invalid, it is shown as "Undefined Opcode."

Branch Instructions

If the address of a branch relative instruction is known, its target is presented as an absolute hex address (or as a symbol if it matches an ADDR pattern or range symbol). If the address of a branch relative instruction is not known, its target is displayed as a hexadecimal offset such as +00000C30 or -00000048.

Overfetch Marking

Overfetch refers to instructions which are fetched but not executed by the processor. They may arise from the following sources:

- When the 604 executes a branch instruction, the instructions between the branch and the branch target are not executed. These instructions are indicated with a asterisk "*", or if the bus trace is ambiguous, with an interrogation point "?". If the instruction cache is enabled, the branch target may already be in the cache and will not be fetched over the bus. The remaining cache line containing the branch will be marked as overfetch.

For conditional branches whose target addresses are not known, or are known but not seen in the bus traffic, the inverse assembler cannot always determine if the branch was taken and will not mark ensuing states as overfetch.

Little-Endian Mode

The inverse assembler is designed to support the native big-endian mode of operation on the PowerPC 604. When operating in little-endian mode, the 604 uses a technique known as "address munging" to convert internal little-endian addresses into external big-endian addresses. Internal and external addresses may differ from one another in the three least significant bits.

In little-endian operation, in a given data beat, the instruction word from DL0..31 (DATA_B label; external address xxx4) will be dispatched before the instruction word from DH0..31 (DATA label; external address xxx0). You can compensate for this by exchanging the DATA and DATA_B labels in the Format menu. However, while this will correctly order 32-bit word reads on the 64-bit data bus, it will cause byte- and half- word reads and writes to appear on the opposite side of the bus, and swap the halves of double-word reads and writes.

SW_ADDR Label

When an HP 16600A/700A-series logic analysis system is being used, the inverse assembler generates a "SW_ADDR" field. The SW_ADDR is displayed as another column in the listing tool. This field is the Software Address generated by the inverse assembler.

The SW_ADDR label cannot be used exactly like other labels. For example, when loading symbols, you will notice that the SW_ADDR label is not in the list of labels that the symbols can be loaded into. Symbols should still be loaded into the ADDR label. The SW_ADDR label is really only used for correlation of the listing with the Source Correlation Tool Set.

To use the Invasm menu (HP 16600A and HP 16700A logic analysis systems)

The Invasm menu provides four choices: Load, Filter, Preferences, and Options. These dialogs assist in analyzing and displaying data. Access the Invasm menu in the listing window.

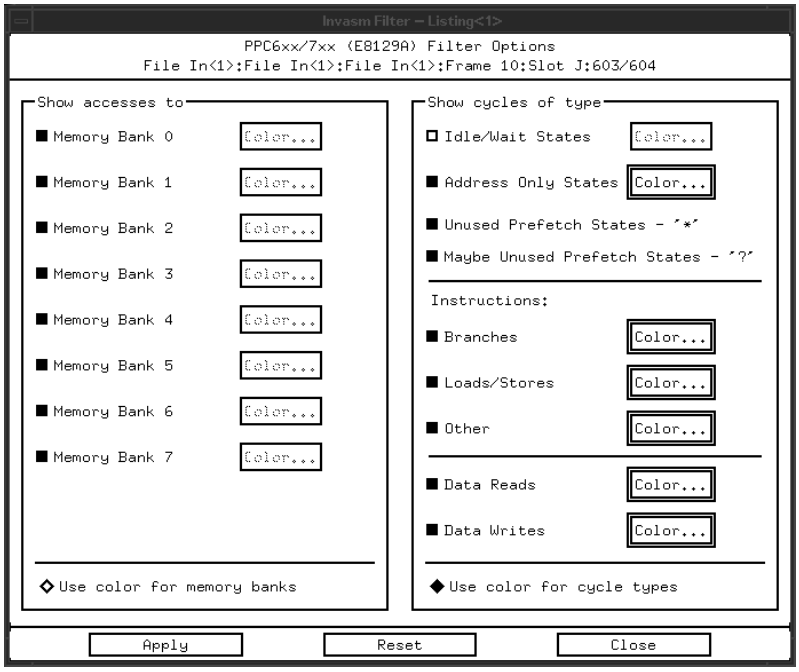
The following sections describe these dialogs.

Load

The Load dialog lets you load a different inverse assembler and apply it to the data in the Listing menu. In some cases you may have acquired raw data; you can use the Load dialog to apply an inverse assembler to that data.

Display Filtering

The inverse assembler lets you Show or Suppress and Colorize several types of states using a tool called display filtering. States can be filtered by their cycle type, or according to which memory bank was accessed for the cycle. The following figure shows the inverse assembler Filter menu.



The show/suppress settings do not affect the data that is stored by the logic analyzer; they only affect whether that data is displayed or not. You can examine the same data with different settings, for different analysis requirements.

This dialog allows faster analysis in two ways. First, you can filter unneeded information out of the display. For example, suppressing wait/idle states will show only states in which a transaction was completed. Second, you can isolate particular operations by suppressing all other operations. For example, you can obtain a quick analysis of memory writes by showing Data Reads and Data Writes, with all other states suppressed.

Suppressing wait/idle states (Idle/Wait States not selected) is useful for obtaining a state-per-cycle display of acquired data. Suppressing overfetched instructions (unused prefetches) may assist in following program execution. Suppressing instructions (by not selecting them) may be useful if your primary interest is data operand reads and writes.

"Unused Prefetch States - '*'" will include an asterisk "*" with states appearing in the inverse assembler listing that were found to be unused prefetches (also called overfetch).

"Maybe Unused Prefetch States - '?'" will include a question mark "?" with states appearing in the inverse assembler listing that were suspected to be unused prefetches. Look at the states marked with "?" to see if they were actually executed later.

When instructions are suppressed (Branches, Loads/Stores, Other not selected), the "*", "-" and "?" indicators are not shown. These prefetch indicators only apply to instructions.

If the X or O pattern markers are turned on, and the designated pattern is found in a state that has been Suppressed with display filtering, the following message will appear on the logic analyzer display:

"X (or O) pattern found, but state is suppressed."

Preferences

The inverse assembler lets you enter target-specific information that will aid in decoding the activity captured in memory. The Preferences menu lets you:

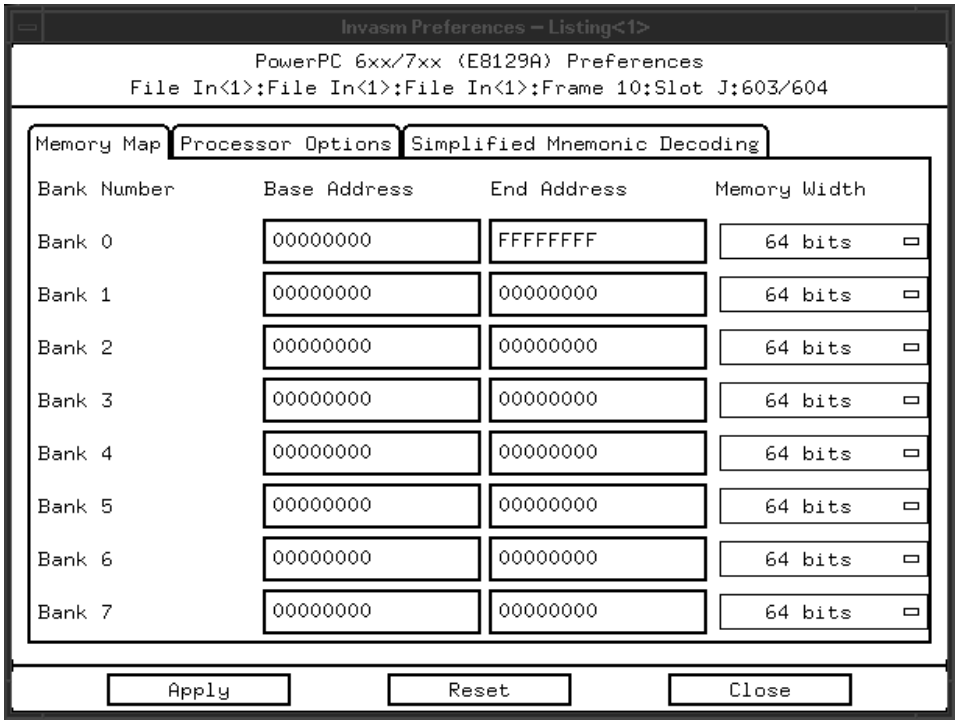
- Set up a decoding memory map of the target system (Memory Map tab).
- Specify a particular PowerPC microprocessor for bus decoding (Processor Options tab).
- Enable or disable simplified mnemonic decoding and a set of specific options (Simplified Mnemonic Decoding tab).

Memory Map For targets that contain either a mix of 32-bit and 64-bit memories or a 32-bit data bus, the inverse assembler provides a memory map to delineate where the different memory is located. For target systems that have a 32-bit bus, only the first memory bank needs to be changed. Set the memory width of Bank 0 to 32 bits. For target systems that contain a combination of 32-bit and 64-bit memory, fill out the memory map according to the address ranges having different memory widths. The memory map is set to default to a 64-bit data bus system.

For up to eight memory banks in your target system, indicate the base address, the end address, and the data bus width for each particular memory device.

The inverse assembler assumes all memory banks are valid so assign lower-numbered banks before the higher-numbered banks. Bank 0 has the highest priority, and Bank 7 has the lowest priority.

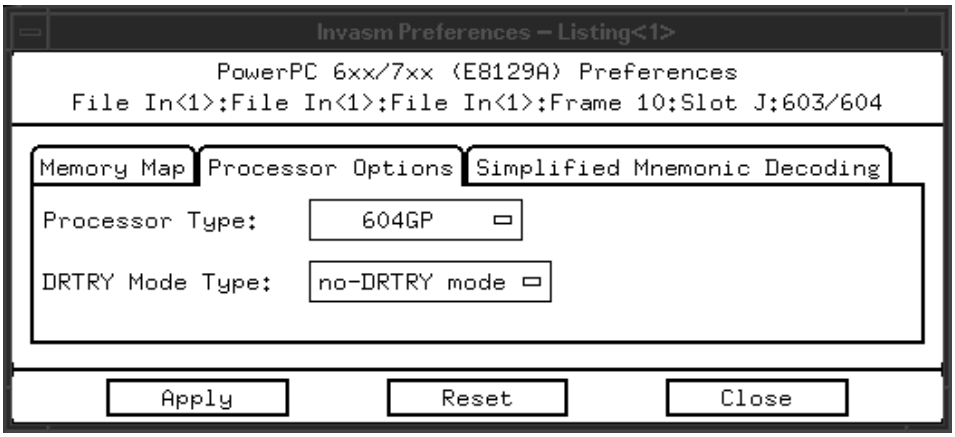
If the Inverse Assembler returns "IA Error: Address not in memory map", the address did not meet the specifications for any of the memory banks.



Inverse Assembler Memory Map

Processor Options This inverse assembler is designed to work with the PPC603, PPC604, and PPC740/750 PowerPC microprocessors. Because of this wide range of support, you must specify which processor type is currently being used. Use the following table to determine how to specify the processor type.

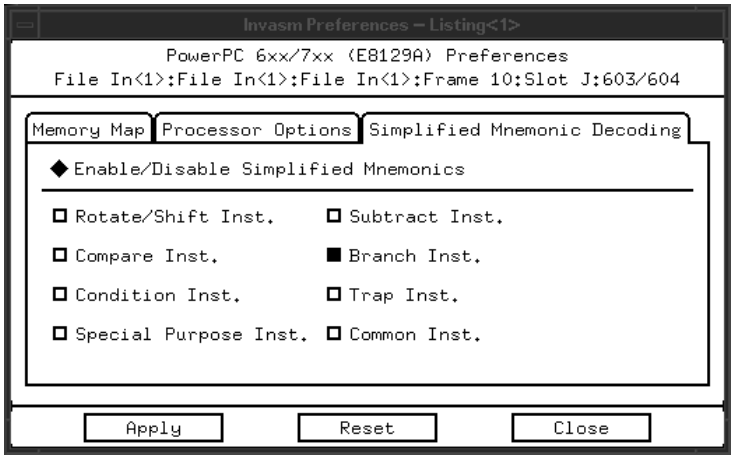
Processor Used	Set Processor Type:
All PowerPC 603 processors	603
All PowerPC 604 processors with no Analysis Probe	604GP
All PowerPC 604 processors with an Analysis Probe	604 + AP
PowerPC 740/750	740/750



Inverse Assembler Processor Options

The inverse assembler also allows you to specify inclusion of the DRTRY signal in its decoding. There are certain versions of the PowerPC microprocessors that have a no-DRTRY mode. If your processor is currently running in this mode, be sure to select the no-DRTRY mode.

Simplified Mnemonic Decoding PowerPC assemblers support a number of simplified mnemonics for some popular assembly language instructions, as described in the PowerPC Programming Environments Guide (Appendix E). The E8129A inverse assembler will show those extensions if you wish to see them. By enabling the Simplified Mnemonic Decoding, you can select which types of simplified mnemonics will be shown. Click the options for the simplified mnemonics you desire. These selections are consistent with the headings in the PowerPC Programming Environments Guide (Appendix E).



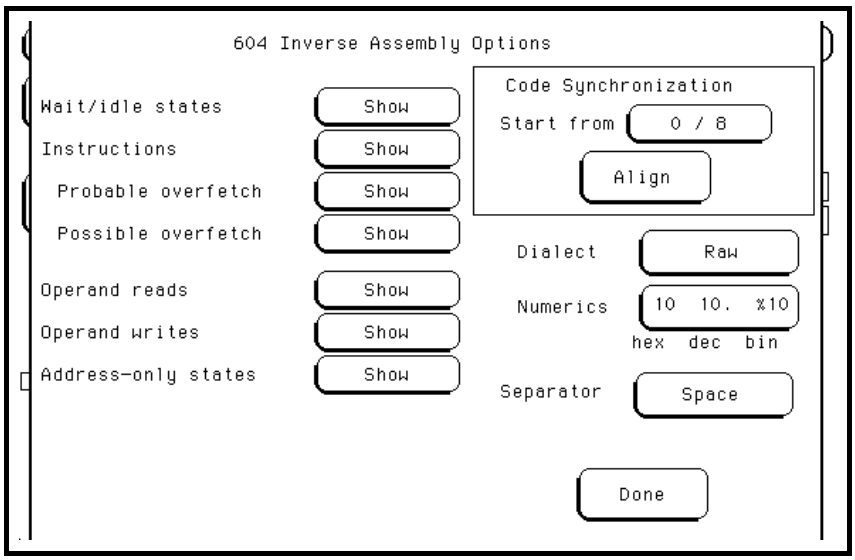
Inverse Assembler Simplified Mnemonic Decoding

Options

The Options menu lets you change the width of the display.

To use the Invasm key (HP 1660, HP 1670, and HP 16500B/C mainframes)

The HP 1660, HP 1670, and HP 16500B/C mainframes provide an Inverse Assembly Options menu, accessed by pressing the Invasm key. The Inverse Assembly Options menu contains three functions: display filtering with Show/Suppress selections, Code Synchronization, and Display Options. The figure below shows the Inverse Assembly Options menu.



Display Filtering

Display filtering is similar to the description on page 80.

Code Synchronization

Code Synchronization allows you to correct the display when the inverse assembler mispredicts a conditional branch as taken and incorrectly marks subsequent states as overfetch.

Extended Mnemonics

PowerPC assemblers support a number of extended mnemonics for some popular assembly language instructions as described in the PPC604 User's Manual. The HP E8129A and HP E2465A inverse assemblers support the following extensions:

- Conditional traps and branches decode the condition mnemonically when possible. For some conditions which have no conventional mnemonics (for example, "signed less than or unsigned greater than"), the condition field is displayed in binary.
- The L bit is omitted as a compare operand. Instead, compares are decoded as "cmpw" (or "?cmpd").
- "Add immediate" instructions with a negative immediate operand are decoded as subtract immediate ("subi").
- "Subtract from" instructions subf and subfc are decoded as subtract instructions sub and subc with the operands exchanged so that "sub r3 r4 r5" is mnemonically interpreted as "r3 = r4 - r5."
- ori r0 r0 0000 is decoded as "nop".
- Add immediate and add immediate shifted instructions, addi and addis, with a null source register are decoded as load immediate and load immediate shifted, li and lis.
- or instructions with identical source registers are decoded as move register, mr.
- nor instructions with identical source registers are decoded as not register, not.
- xor and eqv instructions with identical source and destination registers are decoded as clear and set, clr and set, respectively.
- The cror, crnor, crxor, and creqv instructions map analogously to crmv, crnot, crclr, and crset.
- When the mtcrf instruction field mask specifies the entire cr, it is decoded as mtcr.

The Extended dialect adds several extended opcodes for the rotate instructions. For example, the function of the rlwinm instruction

```
rlwinm  r30  r30  16. 16. 31.
```

is to shift right word immediate, eg

```
srwi    r30  r30  16.
```

The PowerPC rotate-left instructions have extended mnemonics. The following listing shows the extended mnemonics for the integer rotate instructions.

Mnemonic	Decoded As	
rlwimi (rotate left word immediate then mask insert)	inslwi	insert from left immediate
	insrwi	insert from right immediate
rlwinm (rotate left word immediate then AND with mask)	rotlwi	rotate left immediate
	rotrwi	rotate right immediate
	slwi	shift left immediate
	srwi	shift right immediate
	extlwi	extract and left justify immediate
	extrwi	extract and right justify immediate
	clrlwi	clear left immediate
	clrrwi	clear right immediate
	clrlslwi	clear left and shift left immediate
rlwnm (rotate left word then AND with mask)	rotlw	rotate left

The HP E8129A and HP E2465A inverse assemblers support the following extensions of dialect-sensitive instructions.

Instruction Types	raw	extended
branches	bc %00100,2,FFF00230	bne cr0,FFF00230
trap	tw %10000,r5,r6	tw lt,r5,r6
compare	cmp cr1,0,r0,r16	cmpw cr1,r0,r16
	ori r0,r0,0000	nop
subtract	addi r6,r6,FCFC	subi r6,r6,0304
	subf r7,r19,r16	sub r7,r16,r19
common	addi r3,0,7000	li r3,7000
	addis r3,0,7000	lis r3,7000
	or r4,r5,r5	mr r4,r5
	nor r4,r5,r5	not r4,r5
	xor r7,r7,r7	clr r7
	eqv r8,r8,r8	set r8
special purpose	mtcrf %11111111,r5	mtcr r5
condition	creqv 7,7,7	crset 7
	crxor 8,8,8	crclr 8
	cror 7,8,8	crmv 7,8
	crnor 8,9,9	crnot 8,9
rotates and shifts	rlwnm r8,r7,r6,0,31.	rotlw r8,r7,r6
	rlwimi r3,r3,24.,8,23.	inslwi r3,r3,16.,8
	rlwimi r8,r3,17,8,23.	insrwi r8,r3,7,8
	rlwinm r6,r4,8,0,14	extlwi r6,r4,15,8
	rlwinm r6,r4,16,24,31	extrwi r6,r4,8,8
	rlwinm. r6,r4,4,0,31	rotlwi. r6,r4,4
	rlwinm r6,r4,28,0,31	rotrwi r6,r4,4
	rlwinm r6,r4,1,0,30	slwi r6,r4,1
	rlwinm r6,r4,31,1,31	srwi r6,r4,1
	rlwinm r6,r4,0,1,31	clrlwi r6,r4,1
	rlwinm r6,r4,0,0,7	clrrwi r6,r4,14
	rlwinm r6,r4,6,6,25	clrlslwi r6,r4,12,6

5

Symbols and Source Code in the Analyzer

Symbols and Source Code in the Analyzer

Symbols are more easily recognized than hexadecimal address values in logic analyzer trace displays, and they are easier to remember when setting up triggers.

HP logic analyzers let you assign user-defined symbol names to particular label values.

Also, you can download symbols from certain object file formats into HP logic analyzers.

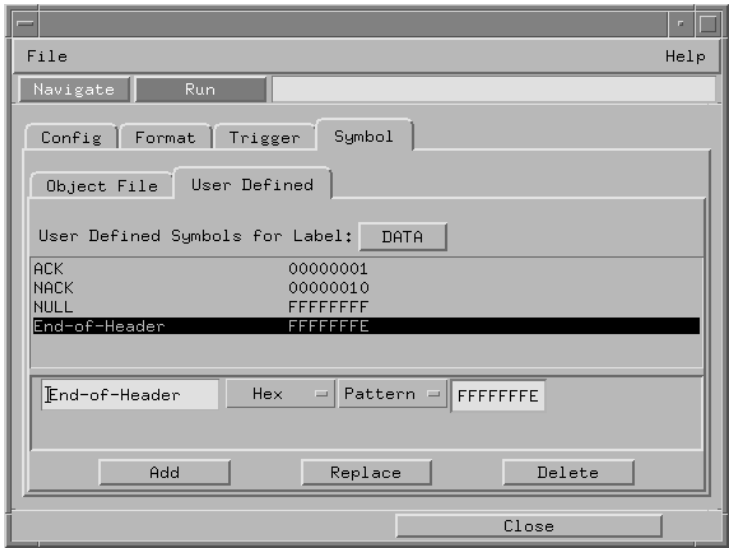
When source file line number symbols are downloaded to the logic analyzer, you can set up triggers on source lines. The HP B4620B Source Correlation Tool Set also lets you display the high-level source code associated with captured data.

After describing user-defined symbols, the rest of this chapter describes the requirements and considerations for displaying object file symbols and source code for PPC604 address values captured by a logic analyzer.

User-Defined Symbols

User-defined symbols are symbols you create from within the logic analyzer user interface by assigning symbol names to label values. Typically, you assign symbol names to address label values, but you can define symbols for data, status, or other label values as well.

User-defined symbols are saved with the logic analyzer configuration.



Predefined PPC604 Symbols

If you're using an analysis probe for the PPC604 microprocessor, the logic analyzer configuration files include predefined symbols.

These symbols appear along with the other user-defined symbols in the logic analyzer.

The predefined PPC604 symbols are listed on page 68.

Object File Symbols

The most common way to load program symbols into the logic analyzer is from an object file that is created when the program is compiled.

Requirements

In order for object file symbols and source code to be accurately assigned to address values captured by the logic analyzer, you need:

An accurate bus trace

An HP E2465A analysis probe captures PPC604 microprocessor data.

Direct address translation

The Memory Management Unit must perform direct address translation.

An inverse assembler

The inverse assembler software is included with HP analysis probes. The PPC604 inverse assembler decodes captured data into program counter (PC) addresses (also known as software addresses) and assembly language mnemonics. Refer to the previous chapter on PPC604 inverse assembly.

A symbol file

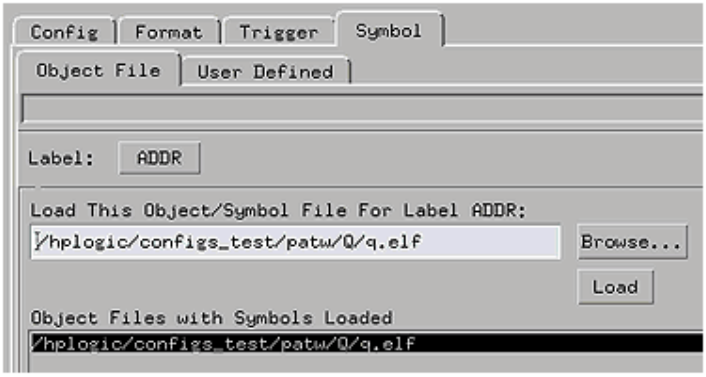
You need an object file containing symbolic debug information in a format the logic analyzer understands. Alternatively, you can use a General Purpose ASCII (GPA) symbol file (see page 199).

See Also

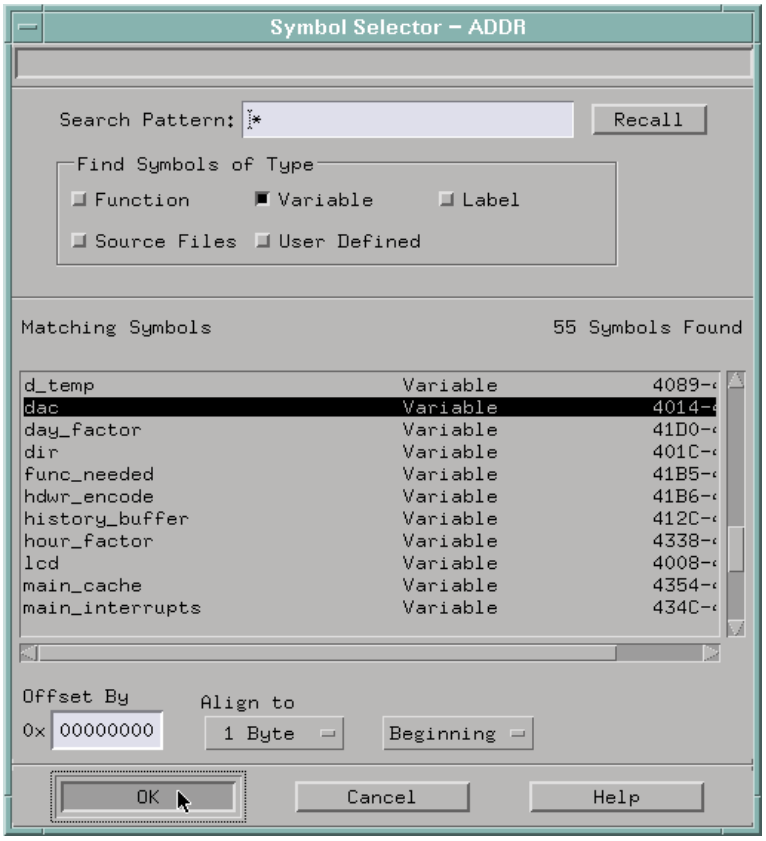
This chapter does not give you task-based instructions for loading object file symbols into a logic analyzer. Refer to your logic analyzer documentation or online help for these instructions.

To use object file symbols in the HP 16600A/700A

To load symbols in the HP 16600A/16700A-series logic analysis system, open the logic analyzer module's Setup window and select the Symbol tab; then, select the Object File tab. Make sure the label is ADDR. From this dialog you can select object files and load their symbol information.



When you load object file symbols into a logic analyzer, a database of symbol/line number to address assignments is generated from the object file. The Symbol Selector dialog allows you to use a symbol in place of a hexadecimal value when when defining trigger patterns, trigger ranges, and so on.



If your language tool is not one of those listed on page 97, you can create a symbol file in the General-Purpose ASCII (GPA) file format (refer to the "General-Purpose ASCII (GPA) File Format" chapter).

See Also

Refer to your logic analyzer documentation or online help for information on how to load symbol files.

Compilers for PPC604

The following PPC604 compilers and their ELF/DWARF format object files can be used with HP logic analyzers and the HP B4620B Source Correlation Tool Set:

Object File Formats

Language System & Version	Format
Diab Data version 4.1a	ELF/DWARF
Green Hills version 1.8.8	ELF/DWARF
Microtec Research, Inc. version 1.4	ELF/DWARF

In order to use symbols in the logic analyzer, file name and line number information must be present in the object file. Your compiler may have options that include or exclude this information.

Limitations: - For C++ files, symbols are not demangled. Mangled names are available for use and the trace listing will still correctly correlate to the appropriate source file lines.

When compiling code, if possible, specify that code and data be put in different memory 'blocks'. A 'block' is 32 Kbytes. 32 Kbytes is the smallest area of memory that can be distinguished by each memory block.

It is also useful to put the stack in the data block.

By separating the code and data in this way, the inverse assembler can be configured to properly decode both code and data.

See Also

Contact your Hewlett-Packard sales engineer to find out if there are other compilers for the PPC604 microprocessor that can be used with HP logic analyzers.

Diab Data Compiler Options

The following options should be used:

-g	Specifies to generate symbolic debugger information (same as -g2).
-WDDOBJECT=E	Specifies the ELF/DWARF file format.
-WDDENVIRON=cross	Specifies the cross development environment.
-WDDTARGET=PPC604	Specifies the type of processor.
-Xdebug-mode=0xff	Turns off Diab Data extensions to the file format.

Diab Data provides a utility that you can use to generate the compiler options you need. Enter "dctrl -t" and follow the instructions. When it is finished, it will present you with a string that you can use for the compiler options.

Please refer to the language tool supplier's documentation for more information about the options available.

More information is available on the World Wide Web at:
http://www.diabdata.com

Green Hills Compiler Options

The following options should be used:

-dwarf	Generates DWARF debugging information.
-G	Generates extended debugging information.
-cpu=ppc604 (or -cpu=ppc604e)	Specifies code generation for the PPC604 (or PPC604e) processor.

If you are using the Green Hills MULTI builder interface, use the following selections:

Options→Advanced, enable "Output DWARF on ELF targets"	Generates DWARF debugging information.
Options→File Options, select "Debugging Level MULTI"	Generates extended debugging information.
Options→CPU, select processor	Specifies code generation for the PPC604 (or PPC604e) processor.

Please refer to the language tool supplier's documentation for more information about the options available. More information is available on the World Wide Web at:

http://www.ghs.com

Microtec Research Inc. Compiler Options

The following options should be used:

-g	Specifies to generate debugging information.
-p604	Specifies code generation for the PPC604 processor.

Please refer to the language tool supplier's documentation for more information about the options available.

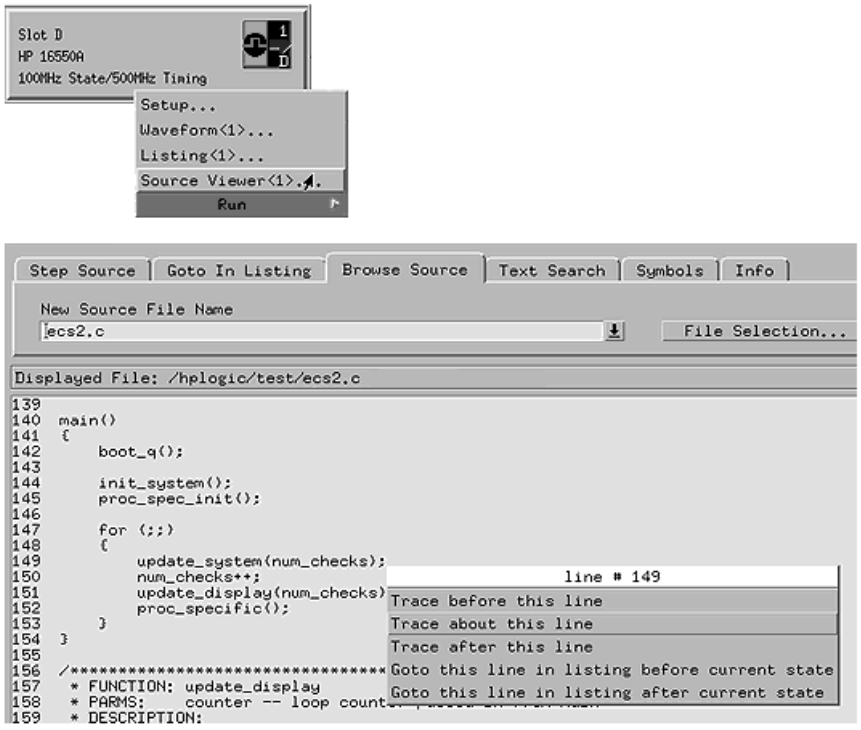
More information is available on the World Wide Web at:
http://www.mentorg.com/microtec

Source Code

The HP B4620B Source Correlation Tool Set lets you:

- View the high-level source code associated with captured data.
- Set up triggers based on source code.

The source correlation tool set correlates the logic analyzer's address label with a line of high-level source code whose address, symbol name, file name, and line numbers are described in a symbol file downloaded to the logic analyzer.



If you purchased a solution, the HP B4620B Source Correlation Tool Set was included. Otherwise, the source correlation tool set is available as an add-on product for the HP 16600A/16700A-series logic analysis system and must be licensed before you can use it (see the System Admin dialogs for information on licensing).

See Also

More information on configuring and using the source correlation tool set can be found in the online help for your logic analysis system.

Requirements for source correlation

The source correlation tool set works with many microprocessors and their embedded software development environments.

However, the overall effectiveness of the source correlation tool set will vary to some degree depending on the specific development environment it is being used in. The following areas affect the performance of the source correlation tool set for different development environments:

- Analysis probe and inverse assembler.

  All the information needed to reconstruct the complete address bus of the target system must be acquired by the logic analyzer. The HP E2465A analysis probe meets this requirement.

  The logic analyzer's inverse assembler may need to reconstruct any incomplete address bus information and/or filter out any unexecuted instructions. Also, the Memory Management Units must perform direct address translation.

  When displaying the next or previous instances of a source line, the Source Viewer display uses the PC or SW_ADDR (Software Address) label generated by the inverse assembler.

- Object file symbols.

  The source correlation tool set requires that symbols be loaded into the logic analyzer (refer to the "Object File Symbols" section earlier in this chapter).

  The compiler needs to produce an object file format that is readable by the logic analyzer; otherwise; a general-purpose ASCII (GPA) format file needs to be generated.

- Access to source code files.

  The source correlation tool set requires that you give the logic analysis system access to your program's high-level source files (either by NFS mounting the file system that contains the source files or by copying source files to the logic analysis system disk).

Inverse Assembler Generated PC (Software Address) Label

In the HP 16600A/16700A-series logic analysis system, the PPC604 inverse assembler generates a "PC" label. The PC label is displayed as another column in the Listing tool. This label is also known as the Software Address generated by the inverse assembler.

The "Goto this line in listing" commands in the HP 16600A/16700A-series logic analysis system perform a pattern search on the PC label in the Listing display (when an inverse assembler is loaded). Because the inverse assembler is called for each line that is searched, the search can be slow, especially with a deep memory logic analyzer.

Also, a single source code line will generate many assembly instructions. The "Goto this line in listing" commands will not find a given source code line unless the first assembly instruction generated from the source line has been acquired by the logic analyzer.

For example, if the compiler unrolls a loop in the source code, the trace could begin after the first assembly instruction of the loop has been executed. A "Goto this line in listing" command would not find the source line.

Access to Source Code Files

The source correlation tool set must be able to access the high-level source code files referenced by the symbol information so that these source files can be displayed next to and correlated with the logic analyzer's execution trace acquisition. This requires you to be aware of a number of issues.

Source File Search Path

Verify that the correct file search paths for the source code have been entered into the source correlation tool set. The HP B4620B Source Correlation Tool Set can often read and access the correct source code file from information contained in the symbol file, if the source code files have not been moved since they were compiled.

Network Access to Source Files

If source code files are being referenced across a network, the HP logic analyzer networking must be compatible with the user's network environment. HP logic analyzers currently support Ethernet networks running a TCP/IP protocol and support ftp, telnet, NFS client/server and X-Window client/server applications. Some PC networks may require extensions to the normal LAN protocols to support the TCP/IP protocol and/or these networking applications. Users should contact their LAN system administrators to help setup the logic analyzer on their network.

Source File Version Control

If the source code files are under a source code or version control utility, check the file names and paths carefully. These utilities can change source code file paths and file names. If these names are changed from the information contained in the symbol file, the source correlation tool set will not be able to find the proper source code file. These version control utilities usually provide an "export" command that creates a set of source code files with unmodified names. The source correlation tool set can then be given the correct path to these files.

Triggering on Symbols and Source Code

When setting up trigger specifications to capture PPC604 execution:

- Use the logic analyzer trigger alignment to avoid missed triggers.
- Use the logic analyzer address offset to compensate for relocated code.
- Use the logic analyzer storage qualification to capture the software execution you're interested in and filter out library code execution (whose source file lookups can take a long time if the library source code is not available).

Using Trigger Alignment

You should use an 8-bit alignment to avoid missed triggers. The PPC604 has a 64-bit data bus. Instructions for the PPC604 are 32 bits long and must be located on even address boundaries. This means that an instruction will often be fetched as the lower 32 bits of one 64-bit memory cycle. When this happens, the address of the instruction in the lower 32 bits of the fetch will not be seen on the address bus. If a trigger was set to occur on this instruction's address, then the trigger will not occur. For example:

	Bus Activity		**High Level**	
Address	Data	Mnemonic	Line	C-Source
00000080	39600000	li r11,0	#13	i = 0;
00000084	39400001	li r10,1	#14	j = 1;
00000088	7D4A5A14	add r10,r10,r11	#15	k = j+i;

In this case, instruction fetches will occur at addresses 80 and 88, so that a trigger set on line #14 (address 84) will not occur. The instruction at address 84 was actually fetched with the 64-bit memory fetch at address 80, so you needed to trigger on address 80 to catch the fetch of 84.

To help avoid these missed triggers, the trigger dialogs for symbol addresses allow you to "Align" the address to a 1-, 2-, 4-, or 8-byte boundary. Alignment affects the least significant address bits of the trigger specification, either setting them to a "don't care" or "zero" value, depending on the logic analyzer.

Set the alignment for program fetches to the width of the program memory in bytes. For the PPC604 with 64-bit (8-byte) wide program memory, use 8-byte alignment. Eight-byte alignment will change the least significant three bits ($2^3 = 8$) of the trigger. Address 84 with 8-byte alignment results in a trigger address range of 80 through 87 for some logic analyzers(3 bits of don't care), or an address of 80 on the other analyzers (three bits of 0). Note that either of these triggers would catch line #14 in the example above.

Using the Address Offset

You need to adjust the source correlation tool set to compensate for relocatable code segments or memory management units that produce fixed code offsets. The offset field in the trigger menu allows you to offset the symbol address. Entering the appropriate address offset will cause the source correlation tool set to reference the correct symbol information for the relocatable or offset code.

To adjust for prefetches, use a trigger offset of 8 (prefetch queue depth) to avoid triggering on prefetched instructions. Note that this is not a foolproof scheme, since this may result in a missed trigger if a branch takes place between the base address and the offset address. For the PPC604, an offset of 8 is large enough to overcome the prefetch queue.

Be aware of prefetches and adjust your triggering to compensate for them. Note that the PPC604 has a good Branch Prediction Unit (PBU), and often, when executing loops the processor will NOT fetch instructions beyond the end of the loop. This means that on most loops an offset may not be required to avoid a false trigger.

Using Storage Qualification

You should configure the logic analyzer's storage qualification capabilities to store only those cycles that correspond to software execution (non-idle, etc.). To set up the store qualification to store only non-idle states, first verify that the Store Qualify field is set to "Term Selection", (as opposed to "Anystate"). Next, replace the STAT label with the ACKs label. While the cursor is over the STAT label, hold down the right mouse button, select the "Replace label..." option, select the ACKs label from the list. Now change the format of the ACKs label to Binary Not Pattern. Finally, edit the pattern field to become "X1111". Only states that are not Idle will be stored.

The source correlation tool set can exhibit long responses to requests for the next source line if the current trace listing corresponds to code from a library that is not in the source code search path. Logic analyzer storage qualification can be used to avoid capturing library code routines.

6

Connecting and Configuring the Emulation Module

Connecting and Configuring the Emulation Module

This chapter shows you how to connect the emulation module to the target system and how to configure the emulation module and target processor.

Overview

Here is a summary of the steps for connecting and configuring the emulation module:

1. Make sure the target system is designed to work properly with the emulation module. (Page 114.)
2. Install the emulation module in your logic analysis system, if necessary. (Page 121.)

   Use the Setup Assistant to guide you through steps 3-6 (see page 21). Use this manual for additional information, if desired.
3. Connect the emulation module to your target system or an analysis probe using the 50-pin cable and the TIM. (Page 126.)
4. Update the firmware of the emulation module, if necessary. (Page 130.)
5. Verify communication between the emulation module and the target. (Page 131.)
6. Configure the emulation module. (Page 132.)
7. Test the connection between the emulation module and the target. (Page 145.)
8. Connect a debugger to the emulation module, if applicable. (Page 147.)

See Also

"Using the Emulation Module with a Debugger" beginning on page 147 for information on configuring the emulation module with a debugger, and for information on configuring LAN port numbers.

Using the Emulation Control Interface

The Emulation Control Interface in your HP 16600A/700A-series logic analysis system allows you to control an emulator (an emulation module or an emulation probe).

As you set up the emulation module, you will use the Emulation Control Interface to:

- Update firmware (which reloads or changes the processor-specific personality of the emulator).
- Change the LAN port assignment (rarely necessary).
- Run performance verification tests on the emulator.

The Emulation Control Interface allows you to:

- Run, break, reset, and step the target processor.
- Set and clear breakpoints.
- Read and write registers.
- Read and write memory.
- Read and write I/O memory.
- View memory in mnemonic form.
- Read and write the emulator configuration.
- Download programs (in Motorola S-Record or Intel Hex format) to the target system RAM or ROM.
- View emulator status and errors.
- Write and play back emulator command script files .

If you have an emulation probe, this interface also allows you to configure the LAN address of the emulation probe.

Using the logic analysis system's intermodule bus does not require the Emulation Control Interface to be running. If the emulation module icon is in the Intermodule window, then it will be able to send and receive signals. Therefore if you are using a debugger, you can use an analyzer to cause a break.

Using a debugger with the Emulation Control Interface is not recommended because:

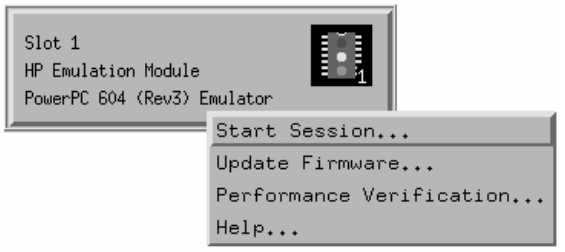
- The interfaces can get out of synchronization when commands are issued from both interfaces. This causes windows to be out-of-date and can cause confusion.
- Most debuggers cannot tolerate another interface issuing commands and may not start properly if another interface is running.

See Also

All of the Emulation Control Interface windows provide online help with a **Help** button or a **Help→On this window** menu selection. Refer to the online help for complete details about how to use a particular window.

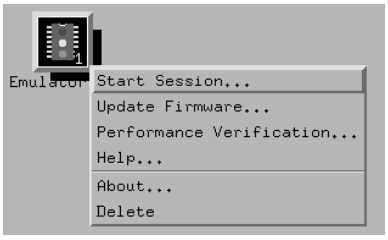
To start the Emulation Control Interface from the main System window

1 In the System window, click the emulation module icon.
2 Select **Start Session...**.



To start the Emulation Control Interface from the Workspace window

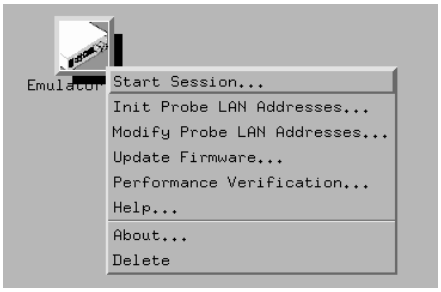
1 Open the Workspace window.
2 Drag the Emulator icon onto the workspace.
3 Right-click on the Emulator icon, then select **Start Session...**.



To start the Emulation Control Interface from the Workspace window for an emulation probe

If you have a stand-alone emulation probe connected to the logic analysis system via LAN, use the Emulation Probe icon instead of the Emulation Module icon.

1 Open the Workspace window.
2 Drag the Emulation Probe icon onto the workspace.
3 Right-click on the Emulation Probe icon, then select **Start Session...**.



4 In the Session window, enter the IP address or LAN name of the emulation probe, then click **Start Session**.

Designing and Operating a Target System for use with the Emulation Module

The following sections describe design and operating considerations for your target system to perform properly with the emulation module. Note that there are different requirements for PPC604, PPC604e, and PPC604e3 target systems.

Design requirements for PowerPC 604

TDO, TDI, TCK, TMS and $\overline{\text{TRST}}$ signals

TDO, TDI, TCK, TMS and $\overline{\text{TRST}}$ signal traces between the JTAG connector and the PPC604 must be less than 3 inches long. If these signals are connected to other nodes, the other nodes must be daisy chained between the JTAG connector at one end and the PowerPC microprocessor at the other end. These signals are sensitive to crosstalk and must not be routed along active signals such as clock lines on the target board.

The TDI, TCK, TMS and $\overline{\text{TRST}}$ signals must not be actively driven by the target system when the JTAG port is being used.

Reset signals

The $\overline{\text{HRESET}}$, $\overline{\text{SRESET}}$ and $\overline{\text{TRST}}$ signals from the JTAG connector must be logically ORed with the $\overline{\text{HRESET}}$, $\overline{\text{SRESET}}$ and $\overline{\text{TRST}}$ signals that connect to the processor on the target system. They cannot be "dotted" or "wire-ORed" on the board. The ORed signals should only reset the processor and no other devices on the target system.

The HP emulation module adds capacitance to all target system signals routed to the JTAG connector. This added capacitance may reduce the rise time of the $\overline{\text{SRESET}}$ or the $\overline{\text{HRESET}}$ signal beyond the processor specifications. If so, the target may need to increase the pull-up current on these signal lines.

Operating considerations for the PowerPC 604

JTAG bug

The June 1997 rev of the PPC604 processor has a bug that renders debug via the JTAG port cumbersome.

What triggers the bug When an external interrupt or a decrementer underflow exception is pending, and the emulation module requests the processor to stop (enter monitor mode), the processor does not report status back correctly to the emulation module.

What happens The emulation module freezes the processor clocks. Once in this state, the emulation module cannot step or run the processor. It can only read/write registers and memory. A processor reset is required to clear this state.

Workaround A possible workaround may be implemented in the user's code by setting the MSR[EE] bit to 1 and coding a minimum of an 'rfi' instruction at the decrementer and external interrupt vector locations. As such, there can never be a pending external interrupt or decrementer underflow exception. The emulation module modifies the decrementer whenever it can to prolong the time it takes for the decrementer to underflow.

Unsupported modes

Target systems which use any of the following modes of operation are not currently supported:

- MMU when it is used for address translation. Only physical memory addresses are accessible. MMU may be used for memory protection as long as no address translation is being performed.
- Caches. Caches must be disabled before using the emulation module. Debug of a PPC604 with either of its caches enabled is not possible. The coherency model is not yet functional.
- Little-endian byte ordering. Memory display/modify is always in big-endian mode. Byte swapping may be handled by the host software.

TLBs

The emulation module cannot access the TLBs (translation look-aside buffers).

Operating considerations for the PowerPC 604e and PowerPC 604e3

PowerPC 604e Rev 2 register corruption bug

A hardware bug in the PowerPC 604e Rev 2 processor causes certain registers to be corrupted when the JTAG port is scanned. The DEC, TBL and TBU registers will always be read as "0xdeadbeef".

DABR and MMCR1 registers

The emulation module firmware cannot read the DABR and MMCR1 registers correctly.

Memory model

Writing to data memory using the "memory model" configuration does not work correctly. Use "cf dmwrop=thru". Do not use "cf dmwrop=mm".

Decoding of invalid instructions

The PowerPC 604e instruction cache is encoded. The emulation module decodes valid instructions before they are displayed. Any invalid instructions will be displayed as-is, in their encoded form, and thus might not match the contents of memory.

False underflow with PowerPC 604e Rev 2

Because the DEC register is corrupted when the JTAG port is scanned, the processor may detect that an underflow has occurred. This can result in unexpected interrupts when using the emulation module.

There are two workarounds for this bug:

- Suppress the interrupts by keeping the EE bit of the MSR cleared, or
- Use Rev 3 of the processor.

MMU Support

Full MMU support is provided for PowerPC 604e and PowerPC 604e3.

When the MMU is enabled in the PowerPC hardware, and the HP Emulator is configured for effective addresses, all memory addresses given to the emulator are assumed to be effective addresses (logical addresses). The emulator uses the MMU block address translation (BAT) registers, segment registers, hash tables, and other special-purpose MMU registers to compute

each corresponding physical address. The requested memory operation is then performed using the physical address.

Operational notes:

- The emulator attempts to perform address translation only if the MSR[IR] and/or the MSR[DR] bits are set (=1) AND the emulator is configured to do translation (cf address=effective). The emulator configuration may be changed using the cf command:
  - cf address=effective (power up default value)
  - cf address=physical
- If both the MSR[IR] and MSR[DR] are set, the emulator will perform address translations by first searching the IBATs and then the DBATS, if no match is found in the IBATs. Note that the PowerPC silicon allows the IBAT and DBAT registers to specify overlapping effective address ranges. Avoid defining overlapping ranges. These make debugging more difficult because the emulator can use the IBATs to translate addresses intended for the DBATs.
- If an effective address is not found in the MMU translation tables, the emulator will return an error and will not perform the requested operation.
- Cache coherency is maintained during emulator MMU translations.
- Be sure the translation enable/disable condition is the same when you set and clear breakpoints. If a breakpoint is set while translation is enabled and then cleared while translation is disabled, the result will be erroneous and unpredictable. This is also true if a breakpoint is set while translation is disabled and then cleared while translation is enabled.
- The emulator ignores read-only restrictions defined in the MMU. (i.e. The emulator may attempt to write to memory that has been defined by the MMU as read-only.)
- MMU translation is automatic and transparent to debuggers connected to the emulator.
- Note the following when using either Rev2 or Rev3 of PowerPC 604e:
  - Writing DBAT0U also writes SR0 and vise versa.
  - Writing DBAT0L also writes SR1 and vise versa.
  - Writing DBAT1U also writes SR2 and vise versa.
  - Writing DBAT1L also writes SR3 and vise versa.

- Writing DBAT2U also writes SR4 and vise versa.
- Writing DBAT2L also writes SR5 and vise versa.
- Writing DBAT3U also writes SR6 and vise versa.
- Writing DBAT3L also writes SR7 and vise versa.

Unsupported modes

Target systems which use any of the following modes of operation are not currently supported:

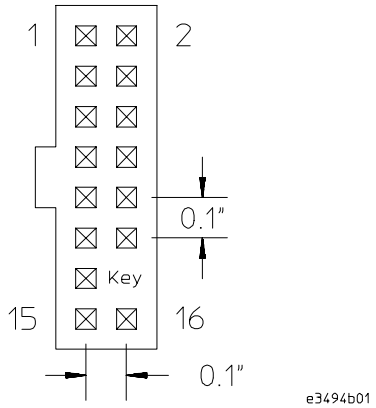
- Little-endian byte ordering. Memory display/modify is always in big-endian mode. Byte swapping may be handled by the host software.
- Caches. For PowerPC 604e3, the data cache must be disabled before using the emulation module.

TLBs

The emulation module cannot access the TLBs (translation look-aside buffers).

PowerPC JTAG interface connections and resistors

The target system must have a 16-pin male 2x8 header connector with the following dimensions:



JTAG Header Connector (top view)

Position 14 of the connector on the target system must not contain a pin. The cable supplied with the emulation module can only be installed if pin 14 has been removed from the header.

The connections for these pins are shown in a table on the next page.

The connector should be placed as close as possible to the processor to ensure signal integrity.

PowerPC 6xx Connections

Header Pin Number	Signal Name	I/O	Board Resistor
1	TDO	Out	
2	Not connected		
3	TDI	In	1KΩ pulldown
4	$\overline{\text{TRST}}$	In	10KΩ pullup
5	Not connected		
6	+POWER[1]		1KΩ series[2]
7	TCK	In	10KΩ pullup
8	Not connected		
9	TMS	In	10KΩ pullup
10	Not connected		
11	$\overline{\text{SRESET}}$	In	10KΩ pullup
12	Not connected		
13	$\overline{\text{HRESET}}$	In	10KΩ pullup
14	KEY		
15	$\overline{\text{CHECKSTOP}}$[3]	Out	1KΩ pullup
16	GND		
	$\overline{\text{QACK}}$ [4]	In	1KΩ pulldown
	L2_TEST_CLK	In	10KΩ pullup
	L1_TEST_CLK	In	10KΩ pullup
	$\overline{\text{LSSD\_MODE}}$	In	10KΩ pullup
	ARRAY_WR	In	10KΩ pullup

[1] The +POWER signal is sourced from the target system and is used as a reference signal. It should be the power signal being supplied to the processor (either +3.3V or +5V). It does not supply power to the HP emulation module.

[2] This 1KΩ series resistor provides short circuit current limiting protection only. If the resistor is present, it should be 1KΩ or less.

[3] For the PowerPC 604 processors, this line is called $\overline{\text{CKSTP\_OUT}}$.

[4] If the target system does not use this signal, the board must have a 1KΩ pulldown resistor connected to this pin. This signal allows the HP emulation module to force the processor into soft stop mode. If the target system does use this signal, it should provide logic so that $\overline{\text{QACK}}$ goes low in response to a $\overline{\text{QREQ}}$.

Installing the Emulation Module

Your emulation module may already be installed in your logic analysis system. If you need to install an emulation module yourself, follow the instructions on the pages which follow.

CAUTION

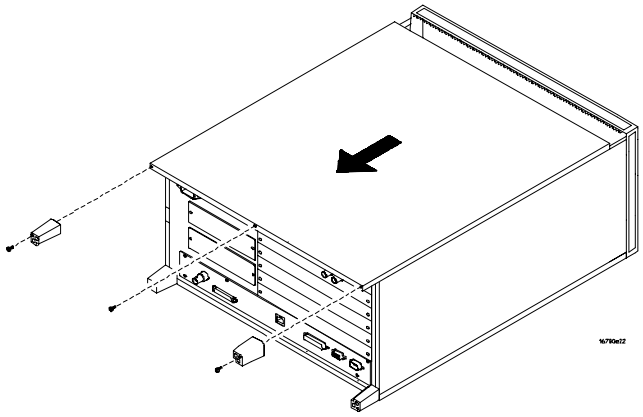
These instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

Electrostatic discharge can damage electronic components. Use grounded wrist straps and mats when you handle modules.

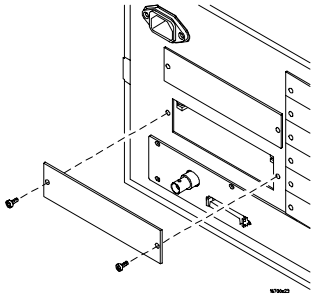
To install the emulation module in an HP 16700A-series logic analysis system or an HP 16701A expansion frame

You will need T-10 and T-15 Torx screwdrivers (supplied with the emulation module).

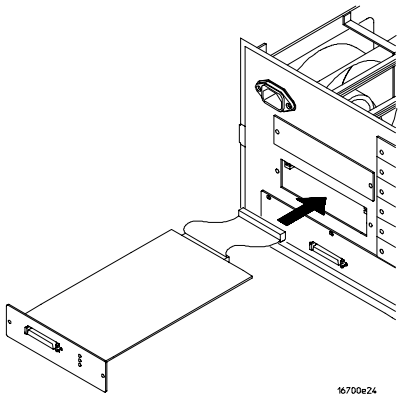
1. Turn off the logic analysis system and REMOVE THE POWER CORD.
   Remove any other cables (including mouse or video monitor cables).
2. Turn the logic analysis system frame upside-down.
3. Remove the bottom cover.



4. Remove the slot cover.
   You may use either slot.

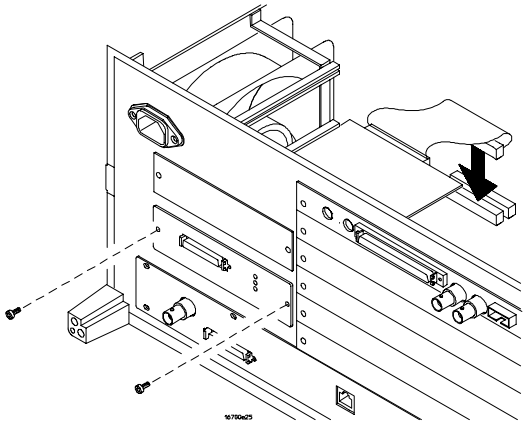


5 Install the emulation module.



6 Connect the cable and re-install the screws.

You may connect the cable to either of the two connectors. If you have two emulation modules, note that many debuggers will work only with the "first" module: the one toward the top of the frame ("Slot 1"), plugged into the connector nearest the back of the frame.



7 Reinstall the bottom cover, then turn the frame right-side-up.

8 Plug in the power cord, reconnect the other cables, and turn on the logic analysis system.

The new emulation module will be shown in the system window.

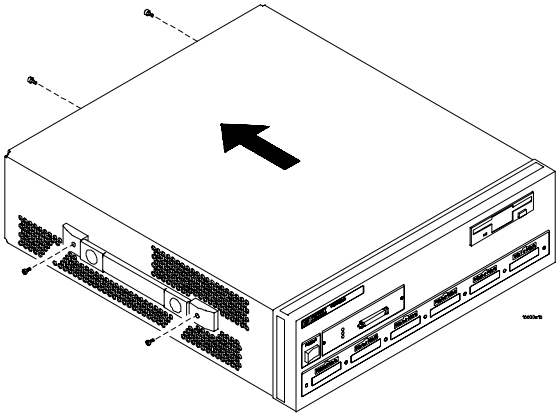
See Also

See page 130 for information on giving the emulation module a "personality" for your target processor.

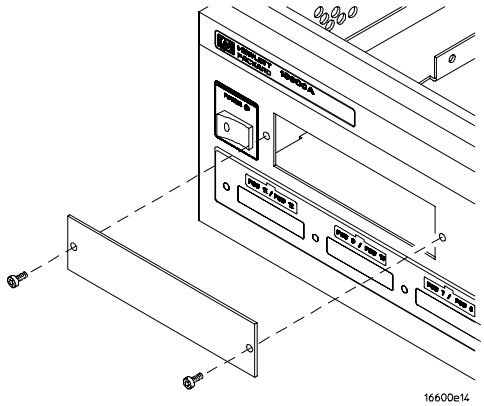
To install the emulation module in an HP 16600A-series logic analysis system

You will need T-8, T-10, and T-15 Torx screwdrivers (supplied with the emulation module).

1. Turn off the logic analysis system and REMOVE THE POWER CORD.
   Remove any other cables (such as probes, mouse, or video monitor).
2. Slide the cover back.

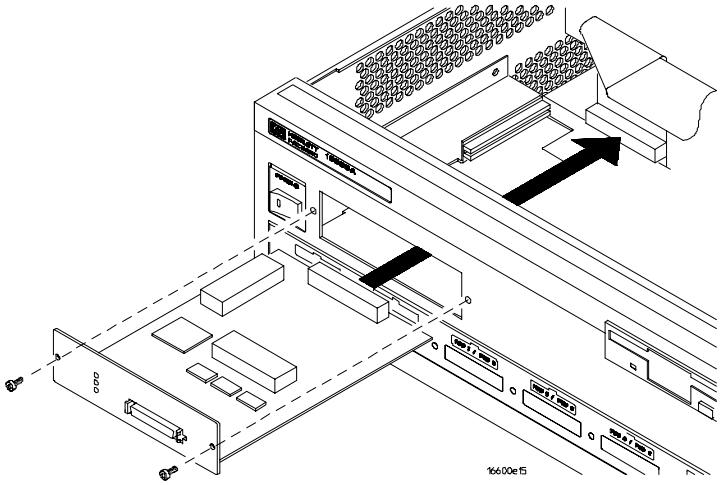


3. Remove the slot cover.



4 Install the emulation module.

5 Connect the cable and re-install the screws.



6 Reinstall the cover.

Tighten the screws snugly (2 N•m or 18 inch-pounds).

7 Plug in the power cord, reconnect the other cables, and turn on the logic analysis system.

The new emulation module will be shown in the system window.

See Also

See page 130 for information on giving the emulation module a "personality" for your target processor.

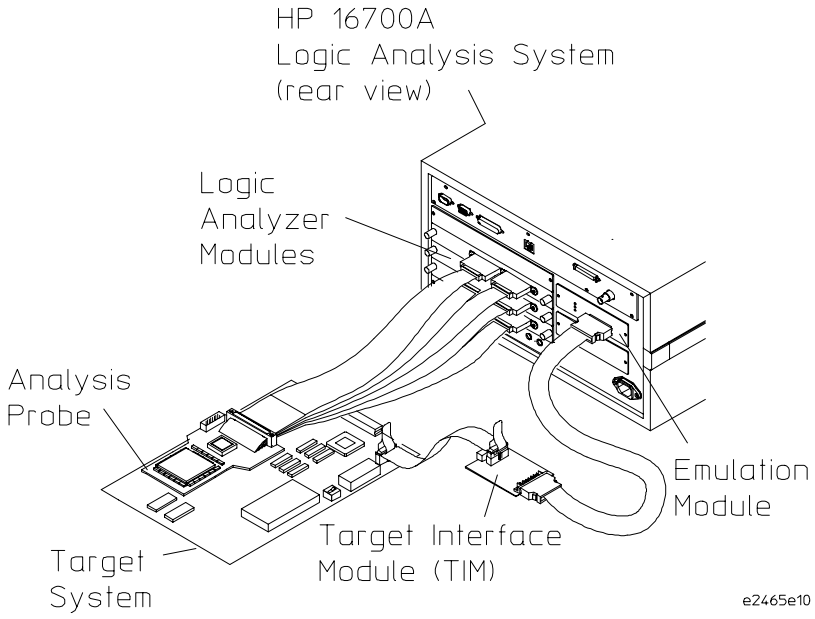
To test the emulation module

If this is the first time that you have used the emulation module, you should run the built-in performance verification test before you connect to a target system. Refer to page 245 for information on performance verification.

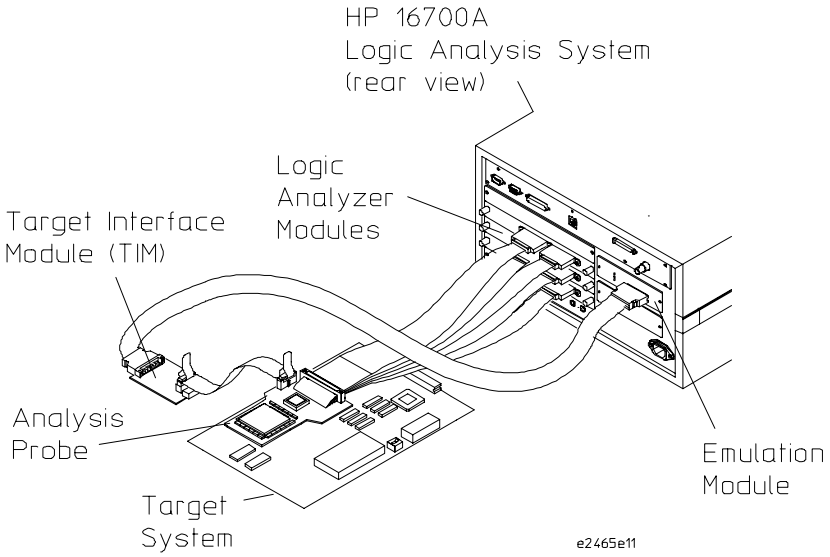
Connecting the Emulation Module to the Target System

Choose one of the following methods for connecting the emulation module and TIM to a target system.

- Directly through a JTAG connector on the target board.



- Through an HP E2465A analysis probe, which provides a direct connection to the JTAG port pins.



After you have connected the emulation module to your target system, you may need to update the firmware in the emulation module.

See Also

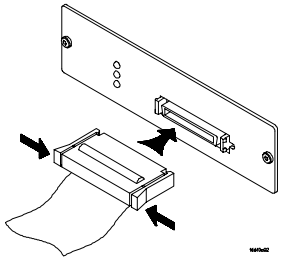
For information on designing a JTAG port on your target board, see page 114.

For a list of the parts supplied with the emulation module, see page 27.

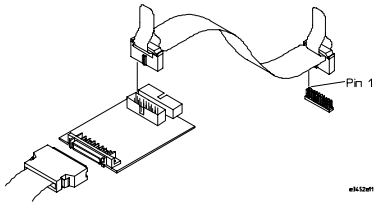
To connect to a target system using a JTAG port

The emulation module can be connected to a target system through a 16-pin JTAG port connector. The emulation module should be connected to a 16-pin male 2x8 header connector on the target system using the 16-conductor cable assembly provided.

1. Turn off the target system and disconnect it from all power sources.
2. Plug one end of the 50-pin cable into the emulation module.



3. Plug the other end of the 50-pin cable into the target interface module.
4. Plug one end of the 16-pin cable into the target interface module.
5. Plug the other end of the 16-pin cable into the JTAG port on the target system.



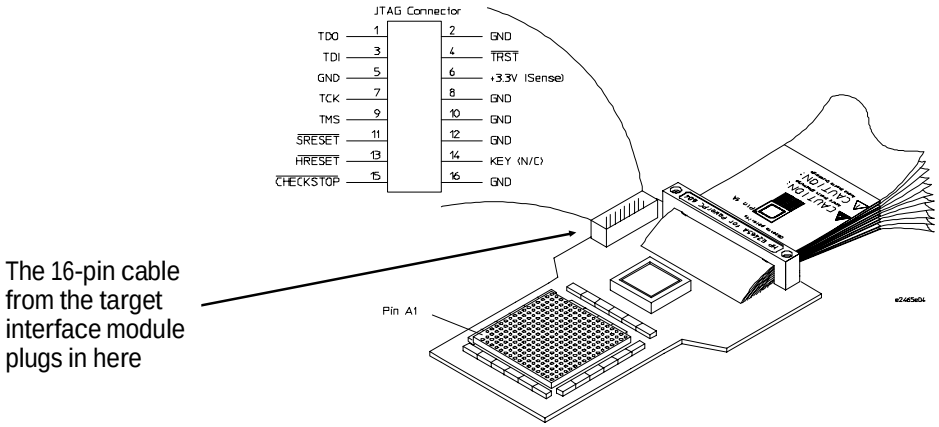
6. Turn on the power to the logic analysis system and then the target system.

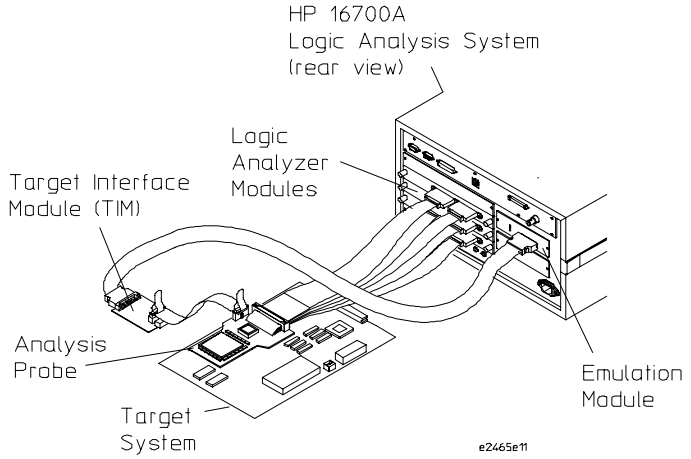
See Also

"Designing a Target System" (page 114) for information on designing a target system for use with the emulation module.

To connect to a target system using an analysis probe

1. Remove power from the target system.
2. Plug one end of the 50-pin cable into the emulation module.
3. Plug the other end of the 50-pin cable into the target interface module.
4. Plug one end of the 16-pin cable into the target interface module.
5. Plug the other end of the 16-pin cable into the JTAG port on the analysis probe.





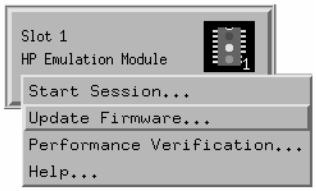
To Update Firmware

After you have connected the emulation module to your target system, you may need to update the firmware to give it the right "personality" for your processor. You must update the firmware if:

- The emulation module is being connected to a new analysis probe or TIM, or
- The emulation module was not shipped already installed in the logic analysis system, or
- You have an updated version of the firmware from HP.

To update the firmware:

1. End any run control sessions which may be running.
2. In the Workspace window, remove any Emulator icons from the workspace.
3. Install the firmware onto the logic analysis system's hard disk, if necessary.
4. In the system window, click the emulation module and select **Update Firmware**.



5. In the Update Firmware window, select the firmware version to load into the emulation module.
6. Click **Update Firmware**.

   In about 20 seconds, the firmware will be installed and the screen will update to show the current firmware version.

See Also

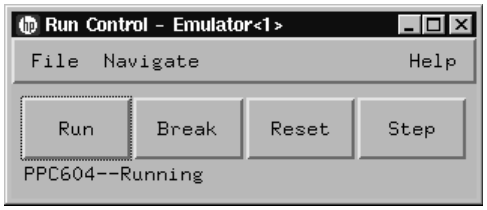
"Installing Software" beginning on page 31 for instructions on how to install the firmware files on the hard disk.

To display current firmware version information

- In the Update Firmware window, click **Display Current Version**.
  There are usually two firmware version numbers: one for "Generics" and one for the personality of your processor.

To verify communication between the emulator and target system

1. Turn on the target system.
2. Start the Emulation Control Interface.
   If the electrical connections are correct, and if the emulator firmware and analysis probe or TIM match your target processor, the Run Control window should be displayed:



Configuring the Emulation Module

The emulation module has several user-configurable options. These options may be customized for specific target systems and saved in configuration files for future use.

The easiest way to configure the emulation module is through the Emulation Control Interface in an HP 16600A or HP 16700A logic analysis system.

If you use the Emulation Control Interface, please refer to the online help in the Configuration window for information on each of the configuration options.

Other ways to configure the emulation module are by using:

- the emulation module's built-in terminal interface
- your debugger, if it provides an "emulator configuration" window which can be used with this HP emulation module

What can be configured

There are two categories of configuration items: general configuration and cache configuration.

The default powerup configuration will generally work with many target systems if the cache is turned off.

If the instruction and data caches are both turned off, the cache configuration items are meaningless and can be ignored.

The following option can be configured using built-in commands:

- Restriction to real-time runs.

The built-in "help cf" command also lists the following options, which are provided only for compatibility with standalone emulation probes:

- BNC break in behavior.
- BNC trigger out behavior.

General Configuration

- JTAG clock speed
- Reset operation
- Memory read delays
- Memory write delays
- Parity bit information

Cache Configuration

- Memory read operation
- Data memory write operations
- Instruction memory write operations

To configure using the Emulation Control Interface

The easiest way to configure the emulation module is to use the Emulation Control Interface.

1. Start an Emulation Control Interface session.

   In the system window, click the Emulation Control Interface icon, and then select "Start Session...".

2. Open a Configuration window.

   Select "Configuration..." from the Emulation Control Interface icon or from the Navigate menu in any Emulation Control Interface window.

3. Set the configuration options, as needed.

   The configuration selections will take effect when you close the configuration window or when you move the mouse pointer outside the window.

4. Save the configuration settings.

   To save the configuration settings, open the File Manager window and click **Save...**.

See Also

Help →Help on this window in the Configuration window for information on each of the configuration options.

Help in the Emulation Control Interface menu for help on starting an Emulation Control session.

To configure using the built-in commands

If you are unable to configure the emulation module with the Emulation Control Interface or a debugger interface, you can configure the emulation module using the built-in "terminal interface" commands.

1. Connect a telnet session to the emulation module over the LAN.

   For example, on a UNIX system, for an emulation module in Slot 1 enter:

   `telnet` *`LAN_address`* `6472`

2. Enter `cf` to see the current configuration settings.
3. Use the `cf` command to change the configuration settings.

See Also

Enter `help cf` for help on the configuration commands.

For information on connecting using telnet, and for information on other built-in commands, see page 229.

Example

To see a complete list of configuration items, type "help cf". This command displays:

```
cf - display or set emulation configuration

  cf                - display current settings for all config items
  cf <item>         - display current setting for specified <item>
  cf <item>=<value> - set new <value> for specified <item>
  cf <item> <item>=<value> <item> - set and display can be combined

help cf <item>    - display long help for specified <item>

--- VALID CONFIGURATION <item> NAMES ---
  rrt       - Restrict to real-time runs
  reset     - Configure reset actions
  speed     - Set JTAG clock
  mrdop     - Configure mem read operation
  dmwrop    - Configure D mem write operation
  imwrop    - Configure I mem write operation
  mrddel    - Set memory read delay
  mwrdel    - Set memory write delay
  breakin   - Select BNC break input option
  trigout   - Select BNC trigger output option
  parity    - Enable/disable data parity
  drtry     - Select DRTRY mode
```

```
    32bitmode - Enable/disable 32 bit mode
M>
```

To see a more detailed description of any configuration item, use the command "help cf <item>". For example:

```
M>help cf rrt

  Restrict to real-time runs

    cf rrt=yes
    cf rrt=no

  If yes (and while the processor is running the user program), any
  command that requires the processor to be stopped will be rejected.
  For example 'reg' and 'm'.

  If no, commands that require the processor to be stopped will
  actually stop the processor, execute then resume running the
  processor.

M>
```

To see a list of the current configuration settings, use "cf":

```
M>cf
  cf rrt=yes
  cf reset=runrom
  cf speed=1
  cf mrdop=mm
  cf dmwrop=mm
  cf imwrop=upd_dcu
  cf mrddel=0
  cf mwrdel=0
  cf breakin=off
  cf trigout=fixhigh
  cf parity=off
  cf drtry=off
  cf 32bitmode=off
M>
```

To configure using a debugger

Because the HP emulation module can be used with several third-party debuggers, specific details for sending the configuration commands from the debugger to the emulation module cannot be given here. However, all debuggers should provide a way of directly entering terminal mode commands to the emulation module. Ideally, you would create a file that contains the modified configuration entries to be sent to the emulation module at the beginning of each debugger session.

See Also

Information about specific debuggers in the "Using the Emulation Module with a Debugger" chapter (page 147).

Your debugger manual.

To configure restriction to real-time runs

Real-time runs configuration

Value	Emulation module configured for	Built-in command
no	Allows commands which break to the monitor. Examples include commands which display memory or registers. These commands break to the monitor to access the target processor, then resume the user program.	`cf rrt=no`
yes	No commands are allowed which break to the monitor, except "break," "reset," "run," or "step." The processor must be explicitly stopped before these commands can be performed. (Default)	`cf rrt=yes`

If your debugger allows displaying or modifying memory or registers while the processor is running, you must set rrt=no in order to use this feature.

To configure the JTAG clock speed (communication speed)

The HP emulation module needs to be configured to communicate at a rate which is compatible with your target processor. The JTAG Clock speed is independent of processor clock speed. In general, speed=1 can always be used and provides the best performance. With some target systems that have additional loads on the JTAG lines or with target systems that do not quite meet the requirements described in the "Designing a Target System" chapter (page 114), setting speed to a slower setting may enable the module to work.

Processor clock speed configuration

Value	Processor clock (TCK) is at least	Built-in command
1	10 MHz (default)	`cf speed=1`
2	5 MHz	`cf speed=2`
3	2.5 MHz	`cf speed=3`
4	1.25 MHz	`cf speed=4`
5	625 kHz	`cf speed=5`
6	312 kHz	`cf speed=6`
7	156 kHz	`cf speed=7`

To configure reset operation

The reset configuration item controls what kind of reset is performed and what state the processor will be in after the reset.

Reset configuration

Value	Effect of a reset from the emulation module	Built-in command
runrom	Reset the processor and cause it to start running user code at address FFF00100H.(Default)	`cf reset=runrom`
rom	Reset the processor and cause it to stop at address 0FFF00100H.	`cf reset=rom`
runram	Reset the processor and cause it to start running user code at address 00000100H.	`cf reset=runram`
ram	Reset the processor and cause it to stop at address 00000100H.	`cf reset=ram`
jtag	Just reset the JTAG interface on the processor. The processor itself will not be reset. This may help in some cases where communications are lost, however all the other reset settings reset the JTAG interface as part of the reset sequence so this setting will only rarely be useful.	`cf reset=jtag`

To set memory read delays

The memory read delay setting delays the number of microseconds specified during memory reads. It is provided for accessing slow devices like memory mapped IO.

- To set the memory read delay using the built-in terminal interface, use the `cf mrddel=<`*delay in usec*`>` command.

  The <*delay in usec*> must be in the range 0-10000000. This should be set to the smallest number possible for best performance since it delays all reads by the number of microseconds specified.

  Default: cf mrddel=0

To set memory write delays

The memory write delay setting delays memory writes by the number of microseconds specified. It is provided for accessing slow devices like memory mapped IO.

- To set the memory write delay using the built-in terminal interface, use the `cf mwrdel=<`*delay in usec*`>` command.

  The <*delay in usec*> must be in the range 0-10000000. This should be set to the smallest number possible for best performance.

  Default: cf mwrdel=0

To generate parity bits on memory operations

The PowerPC processor generates parity bits on both address and data lines when running user code. When used in debug mode these bits must be generated separately slowing down memory operations. Since memory operations on the PowerPC are slow as it is and many target systems do not check parity, parity is only generated if requested.

Parity configuration

Value	Emulation module configured for	Built-in command
off	Do not generate the parity bits for memory operations from the emulation module. This provides better performance, but will not work correctly when accessing devices that check the parity bits.(Default)	`cf parity=off`
on	Generate the parity bits for memory operations. Currently, only parity bits for the memory data lines are generated. Parity bits on the address lines are not. This may change in future firmware versions.	`cf parity=on`

To configure the memory read operation

The memory read operation configuration entry defines how the memory and cache interact during a memory read operation. If both instruction and data caches are turned off (bits ICE and DCE in the register HID0 are zero), this configuration setting has no effect and a memory read will always return the contents of physical memory.

Memory read configuration

Value	emulation module configured for	Built-in command
mm	A memory read from an address that is valid in either the data or instruction cache will return the contents of the cache. Memory reads from addresses not valid in either cache will return the contents of the physical memory.(Default)	`cf mrdop=mm`
phys	A memory read will always return the contents of physical memory.	`cf mrdop=phys`

Using the mrdop=phys setting with the cache enabled may show data that is no longer valid. Use this setting only for solving cache problems where you really need to see the contents of physical memory. For general operation, the "mm" setting should always be used.

The instruction cache in PPC604e and PPC604e3 is encoded. The emulator will decode the content of the instruction cache before displaying it. However, the emulator will only decode valid instructions. Invalid instructions in the cache will be displayed in coded form, which might not match the content of memory.

To configure data memory write operations

Although the PowerPC processor has one contiguous physical memory address space that can hold both data and instructions, it has separate caches for instructions and data. These separate caches must be considered in order to keep the caches and memory coherent during memory write operations. These settings are only used for memory write operations. Code download always writes to physical memory and disables any cache entries containing addresses written for improved performance. Some host interfaces use the code download mode for all memory write operations so this setting may or may not have any effect on your debugger.

Only the memory write command allows specifying instruction or data memory operations. This may not be provided by your debugger interface. If not specified, memory write operations are always instruction memory.

If the data cache is disabled, a data memory write will always write to physical memory and this configuration setting is ignored.

Memory write configuration

Value	emulation module configured for	Built-in command
mm	A data writes to addresses that are valid in the data cache will write the value only to the cache and mark the cache line modified as "dirty" which will indicate to the cpu that the cache line must be written to memory. A data write that is not valid in the data cache will only be written to physical memory.(Default)	`cf dmwrop=mm`
thru	A data memory write to an address that is valid in the data cache will write to both cache and physical memory. If the address is not valid in the cache, only physical memory will be modified.	`cf dmwrop=thru`
bypass	A data memory write will only be written to physical memory ignoring the cache.	`cf dmwrop=bypass`

The **cf dmwrop=bypass** setting should be used with extreme caution because dirty cache entries may be written by the processor over the new data value written to memory by the emulation module.

To configure instruction memory write operations

Although the PowerPC processor has one contiguous physical memory address space that can hold both data and instructions, it has separate caches for instructions and data. These separate caches must be considered in order to keep the caches and memory coherent during memory write operations. Code download always writes to physical memory and disables any cache entries containing addresses written for improved performance. Some host interfaces use the code download mode for all memory write operations so this setting may or may not have any effect on your debugger.

Only the memory write command allows specifying instruction or data memory operations. Access to this may not be provided by your debugger interface. If not specified, memory write operations are always instruction memory.

If the instruction and data caches are both disabled, an instruction memory write will always write to physical memory and this configuration setting is ignored. If the instruction cache is disabled, instruction memory writes will always write to physical memory and the data cache will be either updated or bypassed depending on this configuration setting.

This configuration setting controls the behavior of both caches when doing instruction memory writes so that instruction memory writes can be used for all memory operations if desired.

Instruction memory write configuration

Value	Emulation module configured for	Built-in command
upd_dcb	This stands for instruction cache update, data cache bypass. An instruction memory write to an address that is valid in the instruction cache will write the value to both the instruction cache and memory. The data cache will be bypassed even if the address is valid in the data cache.	`cf imwrop=upd_dcb`
upd_dcu	This stands for update instruction cache and update data cache. An instruction memory write to an address that is valid in both caches will write the value to both caches and physical memory. (Default)	`cf imwrop=upd_dcu`

Value	Emulation module configured for	Built-in command
inv_dcb	This stands for instruction cache invalidate and data cache bypass. An instruction memory write will invalidate the instruction cache if valid and write only to physical memory. The data cache is not modified even if valid.	`imwrop=inv_dcb`
inv_dcu	This stands for instruction cache invalidate and data cache update. An instruction memory write will invalidate the instruction cache if valid and write to physical memory. The data cache will also be updated if the address is valid in the data cache	`imwrop=inv_dcu`

Setting imwrop to upd_dcb or inv_dcb should be used with caution since dirty cache entrys in the data cache may overwrite the memory just modified by the HP emulation module.

Testing the emulator and target system

After you have connected and configured the emulator, you should perform some simple tests to verify that everything is working.

See Also "Troubleshooting the Emulation Module" on page 225 for information on testing the emulator hardware.

To test memory accesses

1. Start the Emulation Control Interface and configure the emulator, if necessary.
2. Open the Memory window.
3. Write individual locations or fill blocks of memory with patterns of your choosing.

   The access size is the size of memory access that will be used to write or read the memory values.
4. Use the Memory I/O window to stimulate I/O locations by reading and writing individual memory locations.

To test with a running program

To more fully test your target, you can load simple programs and execute them.

1. Compile or assemble a small program and store it in a Motorola S-Record or Intel Hex file.
2. Use the Load Executable window to download the program into RAM or flash memory.

3. Use the Breakpoints window to set breakpoints. Use the Registers window to initialize register values.

   The new register or breakpoint values are sent to the processor when you press the Enter key or when you move the cursor out of the selected register field.
4. In the Run Control window, click **Run**.
5. Use the Memory Mnemonic window to view the program and use the Memory window to view any output which has been written to memory.

7

Using the Emulation Module with a Debugger

Using the Emulation Module with a Debugger

Several prominent companies design and sell state-of-the-art source debuggers which work with the HP emulation module and emulation probe.

Benefits of using a debugger

The debugger will enable you to control the execution of your processor from the familiar environment of your debugger. Using a debugger lets you step through your code at the source-code level.

With a debugger connection, you can set breakpoints, single-step through source code, examine variables, and modify source code variables from the debugger interface. The debugger can also be used to download executable code to your target system.

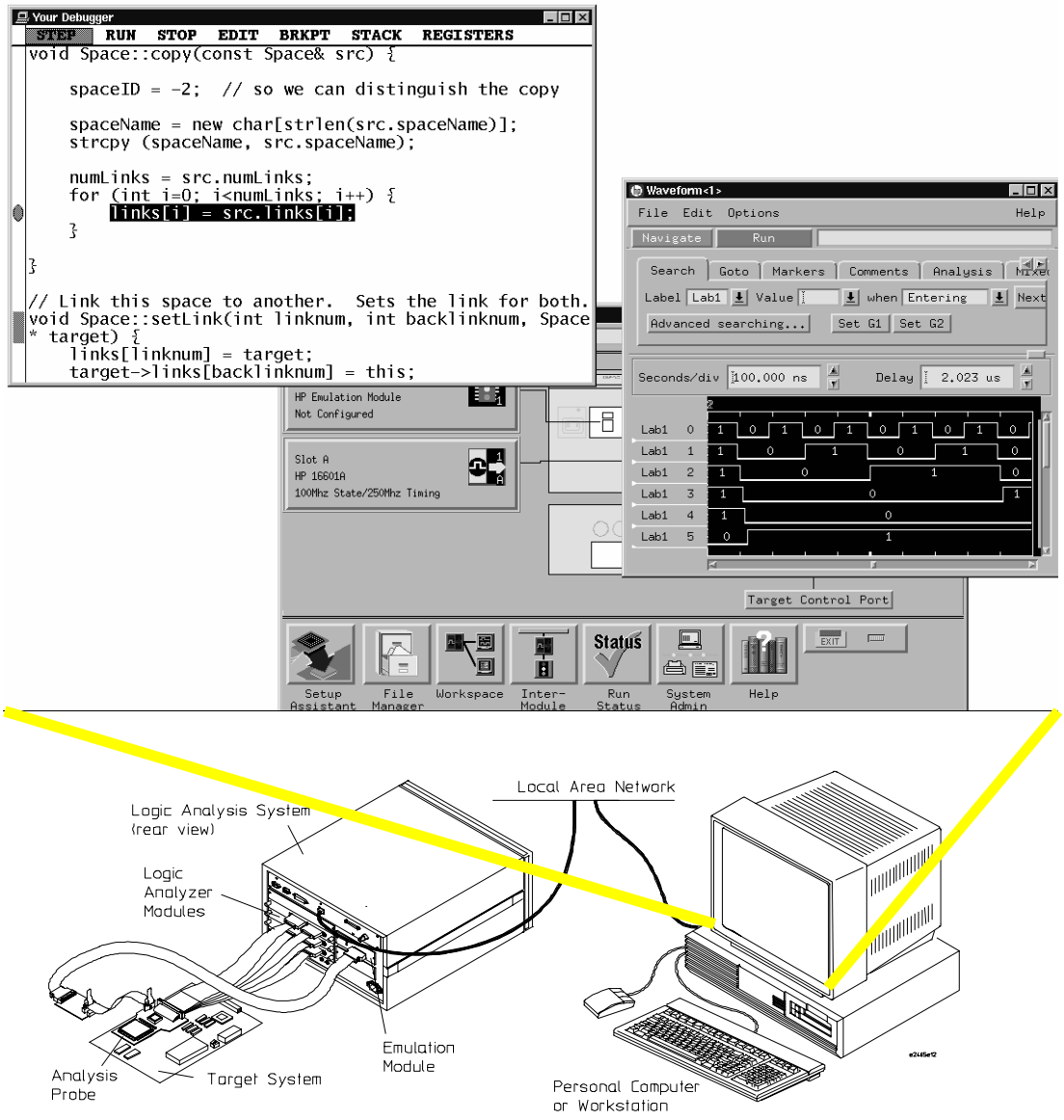
Using a debugger to connect the emulator allows the entire design team to have a consistent interface from software development to hardware/software integration.

Debugger interfaces must be ordered directly from the debugger vendor.

Compatibility with other logic analysis system tools

You can use your logic analysis system to collect and analyze trace data while you use your debugger. If you are using an X windows workstation or a PC with an X terminal emulator, you can display the logic analyzer windows right next to your debugger.

Here is an example of what the display on your PC or workstation might look like:



Minimum requirements

To use a debugger with the emulation module, you will need:

- A debugger which is compatible with the emulation module
- A LAN connection between the PC or workstation that is running the debugger, and the HP 16600A or HP 16700A logic analysis system
- X windows or an X terminal emulator, such as Reflection X on a PC. This is required only if you wish to have the logic analysis system user interface displayed on your PC or workstation screen, along with the debugger.

Is your debugger compatible with the emulation module?

Ask your debugger vendor whether the debugger can be used with an HP emulation module or HP emulation probe (also known as a "processor probe" or "software probe").

LAN connection

You will use a LAN connection to allow the debugger to communicate with the emulation module.

Compatibility with the Emulation Control Interface

Do not use the logic analysis system's Emulation Control Interface and your debugger at the same time.

Setting up Debugger Software

The instructions in this manual assume that your PC or workstation is already connected to the LAN, and that you have already installed the debugger software according to the debugger vendor's documentation.

To use your debugger with the emulation module, follow these general steps:

- Connect the emulation module to your target system (page 110).
- Connect the logic analysis system to the LAN (page 152).
- Export the logic analysis system's display to your PC or workstation (page 154).
- Configure the emulation module (page 132).
- Begin using your debugger.

If you use the Emulation Control Interface to configure the emulation module, remember to end the Emulation Control Interface session before you start the debugger.

CAUTION

Do not use the Emulation Control Interface at the same time as a debugger.

The Emulation Control Interface and debuggers do not keep track of commands issued by other tools. If you use both at the same time, the tools may display incorrect information about the state of the processor, possibly resulting in lost data.

See Also

Refer to the documentation for your debugger for more information on connecting the debugger to the emulation module.

To connect the logic analysis system to the LAN

Information on setting up a LAN connection is provided in the online help or installation manual for your logic analysis system.

Your debugger will require some information about the LAN connection before it can connect to the emulation module. This information may include:

- IP address (Internet address) or LAN name of the logic analysis system.
- Gateway address of the logic analysis system.
- Port number of the emulation module.

Port numbers for emulation modules

Port number	Use for
Debugger connections	
6470	Slot 1 (First emulation module in an HP 16600A/700A-series logic analysis system)
6474	Slot 2 (Second emulation module in an HP 16700A-series system)
6478	Slot 3 (Third emulation module in an expansion frame)
6482	Slot 4 (Fourth emulation module in an expansion frame)
Telnet connections	
6472	Slot 1 (First emulation module)
6476	Slot 2 (Second emulation module)
6480	Slot 3 (Third emulation module)
6484	Slot 4 (Fourth emulation module)

Write the information here for future reference:

IP Address of Logic Analysis System _____

LAN Name of Logic Analysis System _____

Gateway Address _____

Port Number of Emulation Module _____

To change the port number of an emulation module

Some debuggers do not provide a means to specify a port number. In that case, the debugger will always connect to port 6470 (the first emulation module). If you need to connect to another module, or if the port number of the first module has been changed, you must change the port number to be 6470.

To view or change the port number:

1. Click on the emulation module icon in the system window of the logic analysis system, then select **Update Firmware**.
2. Select **Modify Lan Port...**.
3. If necessary, enter the new port number in the **Lan Port Address** field.

   The new port number must be greater than 1024 and must not already be assigned to another emulation module.

To verify communication with the emulation module

1. telnet to the IP address.

   For example, on a UNIX system, enter "telnet <*IP_address*> 6472". This connection will give you access to the emulation module's built-in terminal interface. You should see a prompt, such as "M>".
2. At the prompt, type:

   ```
   ver
   ```

   You should then see information about the emulation module and firmware version.
3. To exit from this telnet session, type <CTRL>D at the prompt.

See Also

The online help or manual for your logic analysis system, for information on physically connecting the system to the LAN and configuring LAN parameters.

"Troubleshooting," page 243, if you have problems verifying LAN communication.

To export the logic analysis system's display to a workstation

By exporting the logic analyzer's display, you can see and use the logic analysis system's windows on the screen of your workstation. To do this, you must have telnet software and X windows installed on your computer.

1. On the workstation, add the host name of the logic analysis system to the list of systems allowed to make connections:

   `xhost +<IP_address>`

2. Use **telnet** to connect to the logic analysis system.

   `telnet <IP_address>`

3. Log in as "hplogic".

   The logic analysis system will open a Session Manager window on your display.

4. In the Session Manager window, click **Start Session on This Display**.

Example

On a UNIX workstation, you could use the following commands to export the display of a logic analysis system named "mylogic":

```
$ xhost +mylogic
$ telnet mylogic
Trying...
Connected to mylogic.mycompany.com.
Escape character is '^]'.
Local flow control on
Telnet TERMINAL-SPEED option ON

HP Logic Analysis System

Please Log in as: hplogic [displayname:0]

login: hplogic
Connection closed by foreign host.
$
```

To export the logic analysis system's display to a PC

By exporting the logic analyzer's display, you can see and use the logic analysis system's windows on the screen of your PC . To do this, you must have telnet software and an X terminal emulator installed on your computer. The following instructions use the Reflection X emulator from WRQ, running on Windows 95, as an example.

1. On the PC, start the X terminal emulator software.

   To start Reflection X, click the Reflection X Client Startup icon.

2. Start a telnet connection to the logic analysis system.

   Log in as "hplogic".

   For Reflection X, enter the following values in the Reflection X Client Startup dialog:

   a. In the Host field, enter the LAN name or IP address of the logic analysis system.
   b. In the User Name field, enter "hplogic".
   c. Leave the Password field blank.
   d. Leave the Command field blank.
   e. Click Run to start the connection.

   The logic analysis system will open a Session Manager window on your display.

3. In the Session Manager window, click **Start Session on This Display**.

To enable or disable processor caches

The Power PC 6xx processors have instruction and data caches. Debugging using an third party debugger will have the greatest performance if the caches are disabled during debugging. There are three ways to disable the caches prior to a debug session:

- Set bits 16 and 17 of register HID0 to zero (bit 0 being the MSB). This will turn off I and D caches.

  Ensure that your startup code does not reset the HID0 register as this could re-enable the caches.

- Issue the following probe commands:

  "cf reset=rom"
  "rst" ("rst" will turn off all caches)

  Ensure that your startup code does not reset the HID0 register after the "rst" command as this could re-enable the caches.

- Keep the caches enabled but tell the HP emulation module to bypass them. To do this, issue the probe commands:

  "cf mrdop=phys" (so only physical memory is read)
  "cf dmwrop=bypass" (to bypass the updating of the data cache)
  reference all addresses with the @dmem modifier.

  Example:

  ```
  M> cf mrdop=phys
  M> cf dmwrop=bypass
  M> m -d4 -a4 0..                    (this will read physical memory only)
  M> m -d4 -a4 0@dmem=12345678 (this will write physical memory only)
  ```

  When caches are bypassed, all memory accesses occur out of physical memory and the cache information is ignored. **This means that cache coherency is not maintained.**

If cache handling is not modified using one of the above three methods, execution with the third party debugger may be slower due to the HP emulation module making sure the cache information stays coherent with physical memory.

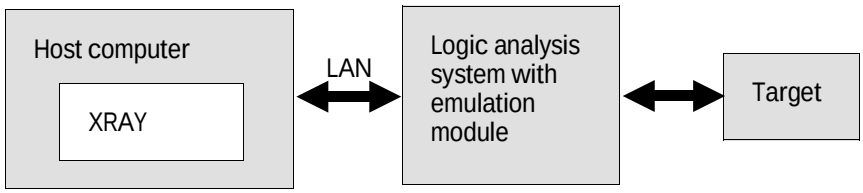
Using the Microtec Research debugger

Compatibility

Version 4.1 of the XRAY HP Probe debugger from Microtec Research, a Mentor Graphics Company, is one of several debuggers which connect to the HP emulation module.

This section provides information that is specific to using XRAY with the HP emulation module. It is intended to be used in conjunction with the XRAY documentation provided by Microtec Research.

Overview



Getting started

1 Check that your Emulation Module is programmed with firmware for a PPC604 processor.

Go to the system window of the logic analyzer interface and verify that the Emulation Module icon is described as a "PowerPC 604 Emulator". If it is not, follow the instructions on page 130 to update the firmware.

2 Edit the gtw.brd file.

The file gtw.brd includes example hostnames, port numbers and initialization information for HP emulation modules that might be on the network for XRAY to connect to. The gtw.brd file is in the "etc" directory under the Microtec tools directory.

a Modify gtw.brd to identify the emulation probe.

Modify the file to include the emulation probe that you want XRAY to communicate with.

See page 152 for information on which port number to use for your emulation module.

b Add commands to initialize the target system.

The target system must have various memory locations initialized before it can access RAM and before XRAY can download an application. Normally, code in the target's boot ROM performs this initialization. However, when XRAY resets the target, it immediately places the processor in debug mode. Therefore, any initialization code which may exist on the target board will not have been executed.

XRAY provides a way for target initialization to occur through the gtw.brd file. The initialization sequences (contained in "{}" pairs) included in the gtw.brd commands specify the commands that will be sent to the HP emulation module to initialize it and prepare it for code download.

The example gtw.brd file provided by Microtec Research contains initialization sequences which can be referenced. If the configuration for your target board is very involved, you can use the "gtwinit" definition in gtw.brd to merely reset the processor and break and use an include file to do the many configuration steps. Please refer to the section "Using an INCLUDE file to configure the emulation module and target" on page 160 for more information on using an include file.

If you are unsure of the configuration needed for your emulation module, you can use the Configuraton window in the logic analysis system's Emulation Control Interface to explore the configuration options. If you use this interface to actually configure your emulation module while connected to XRAY, exit the Emulation Control Interface after the configuration is complete and before you start debugging with the debugger.

NOTE: You must start up XRAY from scratch after gtw.brd is modified for the changes you have made in gtw.brd to be recognized by XRAY.

3 Start XRAY.

After modifying gtw.brd, bring up the XRAY debugger. When XRAY comes up, the Managers dialog will be highlighted. (If the dialog is not present, the Managers dialog can be brought up from the Output Logging Window by selecting **Managers→Connection Manager**).

Using the Managers dialog, set up the connection to your HP emulation module by selecting the Connections tab, clicking on your emulation module name in the lower Available Connections table and click on the **connect** button. You should see your emulation module name appear in the Active Connections table in the top half of the dialog. At this point, you are connected to the emulation module and the initialization commands specified in the gtw.brd file have been sent to your emulation module. If you look in the Output Logging Window, you can verify that the connection and initialization did in fact take place.

4 Download the application code.

> You can significantly increase download performance by disabling the caches. Disable caches by writing the appropriate bits to the HID0 register (see page73).

In the Managers dialog, select the Debug tab, then select **execution→Load File to Target** or **control→Load File to Target**. This will open the 'Load File To Target' dialog. (Alternatively, you may select the Files tab and select **load→Load File to Target**.)

Use the Load File To Target dialog to choose the file you would like to download. When the file you want is listed in the center window, you may double click on it to start the load.

When the load is complete, you will see the file you loaded appear in the Active Files window of the File tab and in the Active Processes window of the Debug tab. You are now ready to debug your application code.

To configure the emulation module and target using an INCLUDE file

You can use an include file to configure the emulation module and set up your target system after bringin up the XRAY debugger. If a complex configuration is needed for your emulation module and target (such as multi-commands sent to the emulation module) this will save time and reduce errors.

1. Save the configuration commands in a text file, one command per line. Microtec Research provides an example include file in its tools\etc\xhippchp\ directory in the file "mo8xxads.inc" (for the MPC8xx family).
2. To run the include file, select "Include Commands from File" under the Debug menu in the Code window and double click on the include filename you want to execute.

To perform common debugger tasks

- To display registers, select Register under the Windows menu in the Code window.
- To set a breakpoint, double click on the source code line where the breakpoint is to be located.
- To clear a breakpoint, double click on the line where the breakpoint is set.
- To step through code, select one of the step icons at the top of the Code window.
- To run from current PC, click on the first icon in the Code window.
- To toggle the display between source code and source code interlaced with assembly code, click on the Dsm button at the bottom of the code display window.
- To load program symbols, reset the PC, reset the stack pointer, and run from start, click restart.

gtwinit and gtwreset command sequences

The gtwinit command sequence defined in the gtw.brd file is sent to the HP emulation module when XRAY is establishing connection with the module.

The gtwreset command sequence is sent to the emulation module when the XRAY "Reset" command is invoked.

To send commands to the emulation module

"Terminal interface" commands may be sent directly to the emulation module from XRAY. There are two ways to do this:

- Using an include file (as explained in the "Using an INCLUDE file to configure the emulation module and target" section)

OR

- Using the XRAY "cf" command.

  This command takes a string as a parameter and sends it to the emulation module. For example, if you want to send the emulation module command cf rst=soft, you can type

  ```
  cf "rst=soft"
  ```

  in the XRAY Debugger command line.

  Note that the command must be surrounded by double quotes.

To view commands sent by XRAY

XRAY communicates with the emulation module using the emulation module's "terminal interface" commands. XRAY automatically generates and sends the commands required for normal operation. The communication between XRAY and the emulation module can be logged to a file after a connection has been established between XRAY and the emulation module and viewed later. To enable logging, enter the command:

```
PROBEMESSAGE ON,msgfile
```

This will create the "msgfile" and log a summary of the messages that occur between XRAY and the emulation module to it. The logging can be turned off with the following command:

```
PROBEMESSAGE OFF
```

To disconnect from the emulation module and target

In the Managers winodw, select the Connect tab. Click on the emulation module name that you want to disconnect. Under the Control menu, select "Disconnect from Board" (or you can "Reconnect to Board" if you have lost connection to the emulation module).

Error conditions

"!ERROR 800! Invalid command: bcast" usually means that there is not a target interface module (TIM) connected to the emulation module or the emulation module does not have firmware for the PPC604 family. Verify that the emulation module is connected to the target through a TIM. Next, go to the system window of the logic analyzer interface and verify that the Emulation Module icon (stop-light) is described as a PowerPC 604 Emulator. If it is not, follow the steps on page 130 to update the firmware in the Emulation module for PPC604 processors.

"command socket connection failed: WSAECONNREFUSED: connection refused" usually means the emulation module is not at port #6470 on the Logic Analysis System.

See Also

The Microtec Research web site: **http://www.mentorg.com/microtec**

The *XRAY Debugger Reference Manual* by Microtec Research.

The configuration section beginning on page 132 for more information on configuration options and the "cf" command.

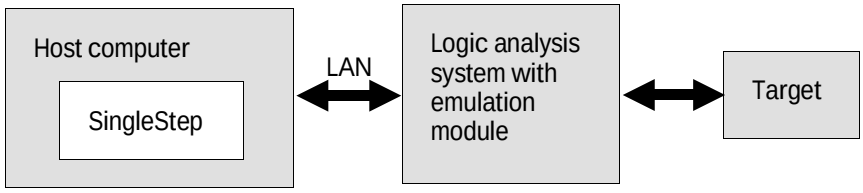
Using the Software Development Systems debugger

Compatibility

Version 7.2 of the SingleStep debugger from Software Development Systems, Inc. is one of several debuggers which connect to the HP emulation module.

This section provides information that is specific to using SingleStep with the HP emulation module. It is intended to be used in conjunction with the SingleStep documentation provided by SDS.

Overview



Startup behavior

The following actions are performed at the start of a session and when you select **File→Debug:**

- If the target reset option is selected, the target is reset and programmed with the register values in the configuration file (<*filename*>.cfg).
- Hardware breakpoints are disabled.
- Software breakpoints are enabled.
- All breakpoints are cleared.
- main() _exit breakpoints are set, if that option is selected.

Getting started

1 Check that your Emulation Module is programmed with firmware for a PPC604 processor:

Go to the system window of the logic analyzer interface and verify that the Emulation Module icon is described as a "PowerPC 604 Emulator". If it is not, follow the instructions on page 130 to update the firmware.

2 Connect to the emulation module:

a Start SingleStep running on your PC or workstation.

b When the small Debug dialog box appears in the middle of the screen, click the Connection tab and then enter the IP address of the HP logic analysis system which contains the emulation module.

If the Debug dialog box is not visible, select **File→Debug.**

Note: SingleStep is hard-coded to connect to the emulation module at port 6470 of the logic analysis system frame. See page 153 for more information on port numbers.

3 Initialize the target system.

The target system must have various registers and memory locations initialized before it can access RAM and before SingleStep can download an application. Normally, code in the target's boot ROM performs this initialization. However, when SingleStep resets the target, it immediately places the processor in debug mode. Any initialization code which may exist on the target board has not been run.

SingleStep provides a way for target initialization to occur without running application code through the use of the "_config" alias. _config is used to define a list of commands that will be used to initialize the target after a reset. The _config alias should be defined in the sstep.ini file (in the "cmd" directory); it can list the actual commands used to initialize the executable, or it may point to a file of type .cfg which contains the actual initialization commands. The config aliases contained in sstrp.ini are provided by SDS for some common targets. Refer to the SDS SingleStep User's Guide for information about additional initialization techniques.

An alternate way of creating the _config alias is to use the Target Configuration tab in the "Debug" dialog box. The "Debug" dialog method and the sstep.ini method are mutually exclusive. Use one or the other, but not both.

Initialization of the target (that is, execution of the _config alias) will not actually occur until the "Debug" dialog is successfully exited.

4 Set up the download and execution options in the Options tab of the Debug dialog.
5 Download the application and run:

> You can significantly increase download performance by disabling the caches. Disable caches by writing the appropriate bits to the HID0 register (see page73).

Select the File tab and enter the application file name. Exit the "Debug" dialog box by clicking OK.

Emulation module initialization and target initialization occur every time the "Debug" dialog is terminated via the OK button. A summary of the actions taken by SingleStep is given here:

- Initialize the emulation module with the communication speed specified in the "Debug" dialog.
- If "reset target" was selected then execute the commands specified by the _reset alias. The _reset alias should be used to specify commands that are specific to initializing the processor. It is executed each time the processor is reset. The value of the _reset alias can be viewed by issuing a "alias _reset" from the command window.
- Execute the commands specified by the _config alias. The _config alias should be used to specify commands that are specific to initializing (configuring) the target system. It is executed each time the processor is reset and each time the debug dialog is exited. The value of the _config alias can be viewed by issuing an "alias _config" from the command window.
- If "load image" was selected then download the application and set the PC based on object module file contents.
- If "execute until main" was selected then set a breakpoint at main() and run.

To send commands to the emulation module

To view commands sent by SingleStep

SingleStep communicates to the emulation module using the emulation module's "terminal interface" commands. SingleStep automatically generates and sends the commands required for normal operation. This communication between SingleStep and the emulation module can be observed by entering the following command in the SingleStep command window:

```
control -ms
```

To send commands

"Terminal interface" commands may be sent directly to the emulation module from the SingleStep command window or included in SingleStep's .cfg or .dbg command files.

Commands should be enclosed in double quotes and given the prefix: control -c.

Examples

To see see what is defined to happen at reset you would issue the following command in the SingleStep command window:

```
control -c "cf reset"
```

To change the reset definition you would issue the following command in the command window:

```
control -c "cf reset=runrom"
```

For more information about "terminal interface" commands see page 132.

Error conditions

"!ERROR 800! Invalid command: bcast" usually means that there is not a target interface module (TIM) connected to the emulation module or the emulation module does not have firmware for the PPC604. Verify that the emulation module is connected to the target through a TIM. Next, go to the system window of the logic analyzer interface and verify that the Emulation Module icon (stop-light) is described as a PowerPC 604 Emulator. If it is not, follow the steps on page 130 to update the firmware in the Emulation module for PPC604 processors.

"command socket connection failed: WSAECONNREFUSED: connection refused" usually means the emulation module is not at port #6470 on the Logic Analysis System. See step 2 of the getting started section above.

"unrecognized hostname" usually means that the debugger is unable to establish communication with the emulator. Verify communication to the emulation module by doing a ping to the logic analyzer. If you are unable to ping the logic analyzer refer to page 243 for more information.

See Also

The SDS web site: **http://www.sdsi.com**

The *SDS SingleStep Users Guide*.

The configuration section beginning on page 132 for more information on configuration options and the "cf" command.

8

Using the Analysis Probe and Emulation Module Together

Using the Analysis Probe and Emulation Module Together

This chapter describes how to use an analysis probe, an emulation module, and other features of your HP 16600A or HP 16700A logic analysis system to gain insight into your target system.

What are some of the tools I can use?

You can use a combination of all of the following tools to control and measure the behavior of your target system:

- Your analysis probe, to acquire data from the processor bus while it is running full-speed.
- Your emulation module, to control the execution of your target processor and to examine the state of the processor and of the target system.
- The Emulation Control Interface, to control and configure the emulation module, and to display or change target registers and memory.
- Display tools including the Listing tool, Chart tool, and System Performance Analyzer tool to make sense of the data collected using the analysis probe.
- Your debugger, to control your target system using the emulation module. Do not use the debugger at the same time as the Emulation Control Interface.
- The HP B4620B Source Correlation Tool Set, to relate the analysis trace to your high-level source code.

Which assembly-level listing should I use?

Several windows display assembly language instructions. Be careful to use to the correct window for your purposes:

- The Listing tool shows processor states that were captured during a "Run" of the logic analyzer. Those states are disassembled and displayed in the Listing window.
- The Emulation Control Interface shows the disassembled contents of a section of memory in the Memory Disassembly window.
- Your debugger shows your program as it was actually assembled, and (if it supports the emulation module) shows which line of assembly code corresponds to the value of the program counter on your target system.

Which source-level listing should I use?

Different tools display source code for different uses:

- The Source Viewer window allows you to follow how the processor executed code as the analyzer captured a trace. Use the Source Viewer to set analyzer triggers. The Source Viewer window is available only if you have licensed the HP B4620B Source Correlation Tool Set.
- Your debugger shows which line of code corresponds to the current value of the program counter on your target system. Use your debugger to set breakpoints.

Where can I find practical examples of measurements?

The Measurement Examples section in the online help contains examples of measurements which will save you time throughout the phases of system development: hardware turn-on, firmware development, software development, and system integration.

A few of the many things you can learn from the measurement examples are:

- How to find glitches.
- How to find NULL pointer de-references.
- How to profile system performance.

To find the measurement examples, click on the Help icon in the logic analysis system window, then click on "Measurement Examples."

Triggering the Emulation Module from the Analyzer

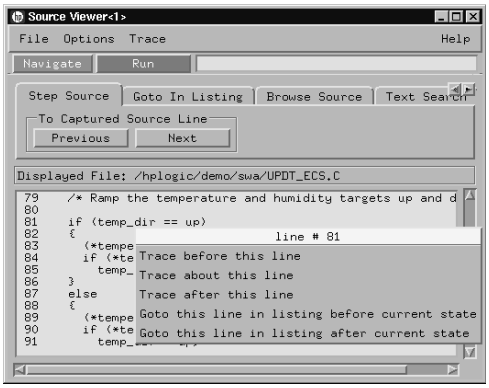
You can trigger the emulation module from the logic analyzer using either the Source Viewer window or the Intermodule window. If you are using the HP B4620B Source Correlation Tool Set, using the Source Viewer window is the easiest method.

To stop the processor when the logic analyzer triggers on a line of source code (Source Viewer window)

If you have the HP B4620B Source Correlation Tool Set, you can easily stop the processor when a particular line of code is reached.

1 In the Source window, click on the line of source code where you want to set the trigger, then select **Trace about this line**.

The logic analyzer trigger is now set.



2 Select **Trace→Enable - Break Emulator On Trigger.**

The emulation module is now set to halt the processor after receiving a trigger from the logic analyzer.

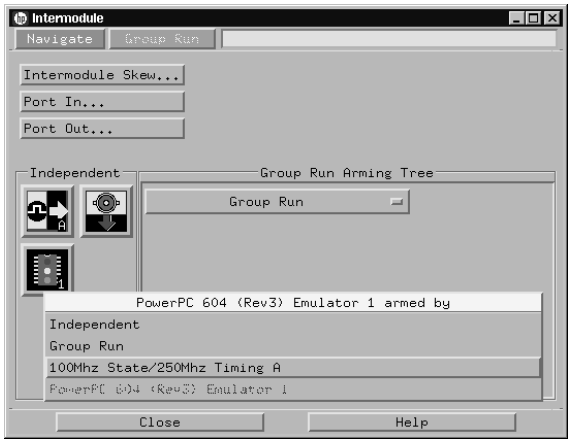
To disable the processor stop on trigger, select **Trace→Disable - Break Emulator On Trigger.**

3 Click **Group Run** in the Source window (or other logic analyzer window).
4 If your target system is not already running, click **Group Run** in the emulation Run Control window to start your target.

To stop the processor when the logic analyzer triggers (Intermodule window)

Use the Intermodule window if you do not have the HP B4620B Source Correlation Tool Set or if you need to use a more sophisticated trigger than is possible in the Source Viewer window.

1 Create a logic analyzer trigger.
2 In the Intermodule window, click the emulation module icon, then select the analyzer which is intended to trigger it.



The emulation module is now set to stop the processor when the logic analyzer triggers.

3 Click **Run** in the Source window (or other logic analyzer window).
4 If your target system is not already running, click **Group Run** in the emulation Run Control window to start your target.

See Also

See the online help for your logic analysis system for more information on setting triggers.

To minimize the "skid" effect

There is a finite amount of time between when the logic analyzer triggers, and when the processor actually stops. During this time, the processor will continue to execute instructions. This latency is referred to as the skid effect.

To minimize the skid effect:

1. In the Emulation Control Interface, open the Configuration window.
2. Set processor clock speed to the maximum value which your target can support.

   The amount of skid will depend on the processor's execution speed and whether code is executing from the cache. See page 138 for information on how to configure the clock speed.

To stop the analyzer and view a measurement

- To view an analysis measurement you may have to click **Stop** after the trigger occurs.

  When the target processor stops it may cause the analyzer qualified clock to stop. Therefore most intermodule measurements will have to be stopped to see the measurement.

Example

An intermodule measurement has been set up where the analyzer is triggering the emulation module. The following sequence could occur:

1. The analyzer triggers.
2. The trigger ("Break In") is sent to the emulation module.
3. The emulation module stops the user program which is running on the target processor. The processor enters a background debug monitor.
4. Because the processor has stopped, the analyzer stops receiving a qualified clock signal.
5. If the trigger position is "End", the measurement will be completed.

   If the trigger position is not "End", the analyzer may continue waiting for more states.
6. The user clicks **Stop** in a logic analyzer window, which tells the logic analyzer to stop waiting, and to display the trace.

Tracing until the processor halts

If you are using a state analyzer, you can begin a trace, run the processor, then manually end the trace when the processor has halted.

To halt the processor, you can set a breakpoint using the Emulation Control Interface or a debugger.

Some possible uses for this measurement are:

- To store and display processor bus activity leading up to a system crash.
- To capture processor activity before a breakpoint.
- To determine why a function is being called. To do this, you could set a breakpoint at the start of the function then use this measurement to see how the function is getting called.

This kind of measurement is easier than setting up an intermodule measurement trigger.

To capture a trace before the processor halts

1. Set the logic analyzer to trigger on **nostate**.
2. Set the trigger point (position) to **End**.
3. In a logic analyzer window, click **Run**.
4. In the Emulation Control Interface or debugger click **Run**.
5. When the emulation module halts click **Stop** in the logic analyzer window to complete the measurement.

   This is the recommended method to do state analysis of the processor bus when the processor halts.

   If you need to capture the interaction of another bus when the processor halts or you need to make a timing or oscilloscope measurement you will need to trigger the logic analyzer from the emulation module (described in the next section).

Triggering the Logic Analyzer from the Emulation Module

You can create an intermodule measurement which will allow the emulation module to trigger another module such as a timing analyzer or oscilloscope.

If you are only using a state analyzer to capture the processor bus then it will be much simpler to use "Tracing until processor halts" as described on page 176.

Before you trigger a logic analyzer (or another module) from the emulation module, you should understand a few things about the emulation module trigger:

The emulation module trigger signal

The trigger signal coming from the emulation module is an "In Background Debug Monitor" ("In Monitor") signal. This may cause confusion because a variety of conditions could cause this signal and falsely trigger your analyzer.

The "In Monitor" trigger signal can be caused by:

- The most common method to generate the signal is to click **Run** and then click **Break** in the Emulation Control Interface. Going from "Run" (Running User Program) to "Break" ("In Monitor") generates the trigger signal.
- Another method to generate the "In Monitor" signal is to click **Reset** and then click **Break**. Going from the reset state of the processor to the "In Monitor" state will generate the signal.
- In addition, an "In Monitor" signal is generated any time a debugger or other user interface reads a register, reads memory, sets breakpoints or steps. Care must be taken to not falsely trigger the logic analyzers listening to the "In Monitor" signal.

Group Run

The intermodule bus signals can still be active even without a Group Run.

The following setups can operate independently of Group Run:

- Port In connected to an emulation module
- Emulation modules connected in series
- Emulation module connected to Port Out

Here are some examples:

- If "Group Run" is armed from "Port In" and an emulation module is connected to Group Run, then any "Port In" signal will cause the emulation module to go into monitor. The Group Run button does not have to be pressed for this to operate.
- If two emulation modules are connected together so that one triggers another, then the first one going into monitor will cause the second one to go into monitor.
- If an emulation module is connected to Port Out, then the state of the emulation module will be sent out the Port Out without regard to "Group Run".

The current emulation module state (Running or In Monitor) should be monitored closely when they are part of a Group Run measurement so that valid measurements are obtained.

Group Run into an emulation module does not mean that the Group Run will Run the emulation module.

The emulation module Run, Break, Step, and Reset are independent of the Group Run of the Analyzers.

For example, suppose you have the following IMB measurement set up:

Clicking the **Group Run** button (at the very top of the Intermodule window or a logic analyzer window) will start the analyzer running. The analyzer will then wait for an arm signal. Now when the emulation module transitions into "Monitor" from "Running" (or from "Reset"), it will send the arm signal to the analyzer. If the emulation module is "In Monitor" when you click **Group Run**, you will then have to go to the emulation module or your debugger interface and manually start it running.

Debuggers can cause triggers

Emulation module user interfaces may introduce additional states into your analysis measurement and in some cases falsely trigger your analysis measurement.

When a debugger causes your target to break into monitor it will typically read memory around the program stack and around the current program counter. This will generate additional states which appear in the listing.

You can often distinguish these additional states because the time tags will be in the μs and ms range. You can use the time tag information

to determine when the processor went into monitor. Typically the time between states will be in the nanoseconds while the processor is running and will be in the μs and ms range when the debugger has halted the processor and is reading memory.

Note also that some debugger commands may cause the processor to break temporarily to read registers and memory. These states that the debugger introduces will also show up in you trace listing.

If you define a trigger on some state and the debugger happens to read the same state, then you may falsely trigger your analyzer measurement. In summary, when you are making an analysis measurement be aware that the debugger could be impacting your measurement.

To trigger the analyzer when the processor halts

Remember: if you are only using a state analyzer to capture the processor bus then it will be much simpler to use "Tracing until processor halts" as described on page 176.

1. Set the logic analyzer to trigger on **anystate**.
2. Set the trigger point to **center** or **end**.
3. In the Intermodule window, click on the logic analyzer you want to trigger and select the emulation module.

   The logic analyzer is now set to trigger on a processor halt.
4. Click **Group Run** to start the analyzer(s).
5. Click **Run** in the Emulation Control Interface or use your debugger to start the target processor running.

   Clicking **Group Run** will *not* start the emulation module. The emulation module run, break, step, reset are independent of the Group Run of the analyzers.
6. Wait for the Run Control window in the Emulation Control Interface or the status display in your debugger to show that the processor has stopped.

   The logic analyzer will store states up until the processor stops, but may continue running.

   You may or may not see a "slow clock" error message. In fact, if you are using a state analyzer on the processor bus the status may never change upon receiving the emulation module trigger (analysis arm). This occurs because the qualified processor clock needed to switch the state analyzer to the next state is stopped. For example, the state analyzer before the arm event may have a status of "Occurrences Remaining in Level 1: 1" and after the arm event it may have the same status of "Occurrences Remaining in Level 1: 1"
7. If necessary, in the logic analyzer window, click **Stop** to complete the measurement.

   If you are using a timing analyzer or oscilloscope the measurement should complete automatically when the processor halts. If you are using a state logic analyzer, click **Stop** if needed to complete the measurement.

To trigger the analyzer when the processor reaches a breakpoint

This measurement is exactly like the one on the previous page, but with the one additional complexity of setting breakpoints. Be aware that setting breakpoints may cause a false trigger and that the breakpoints set may not be valid after a reset.

Remember: if you are only using a state analyzer to capture the processor bus then it will be much simpler to use "Tracing until processor halts" as described on page 176.

1. Set the logic analyzer to trigger on **anystate**.
2. Set the trigger point to **center** or **end**.
3. In the Intermodule window, click on the logic analyzer you want to trigger and select the emulation module.

   The logic analyzer is now set to trigger on a processor halt.
4. Set the breakpoint.

   If you are going to run the emulation module from Reset you must do a **Reset** followed by **Break** to properly set the breakpoints. The Reset will clear all on-chip hardware breakpoint registers. The Break command will then reinitialize the breakpoint registers. If you are using software breakpoints which insert an illegal instruction into your program at the breakpoint location you will not need to do the Reset, Break sequence. Instead you must take care to properly insert your software breakpoint in your RAM program location.
5. Click **Group Run** to start the analyzer(s).
6. Click **Run** in the Emulation Control Interface or use your debugger to start the target processor running.

   Clicking **Group Run** will *not* start the emulation module. The emulation module run, break, step, reset are independent of the Group Run of the analyzers.
7. Wait for the Run Control window in the Emulation Control Interface or the status display in your debugger to show that the processor has stopped.

   The logic analyzer will store states up until the processor stops, but may continue running.

You may or may not see a "slow clock" error message. In fact, if you are using a state analyzer on the processor bus the status may never change upon receiving the emulation module trigger (analysis arm). This occurs because the qualified processor clock needed to switch the state analyzer to the next state is stopped. For example, the state analyzer before the arm event may have a status of "Occurrences Remaining in Level 1: 1" and after the arm event it may have the same status of "Occurrences Remaining in Level 1: 1"

8 If necessary, in the logic analyzer window, click **Stop** to complete the measurement.

If you are using a timing analyzer or oscilloscope the measurement should complete automatically when the processor halts. If you are using a state logic analyzer, click **Stop** if needed to complete the measurement.

Hardware Reference

Hardware Reference

This chapter contains additional reference information including the specifications and characteristics for the analysis probe and the emulation probe, as well as signal mapping for the HP E2465A analysis probe. It consists of the following information:

- Analysis probe reference
- Emulation module reference

Analysis probe—operating characteristics

The following operating characteristics are not specifications, but are typical operating characteristics for the HP E2465A PPC604 analysis probe.

Operating Characteristics

Microprocessor Compatibility	PowerPC 604
Package Supported	287-pin PGA
Microprocessor Clock Speed	66 MHz maximum
Logic Analyzers Supported	HP 16600A, HP 16601A, HP 1660A/AS/C/CS/CP, HP 1670A/D, HP 16550A (two cards), HP 16554A/55A/56A (two or three cards), HP 16555D/56D (two or three cards)
Accessories Required	None
Optional Accessories	An emulation module can be connected to the analysis probe.
Pods Required	Eight 16-channel pods are required for disassembly. Eleven 16-channel pods are available.

Electrical Characteristics

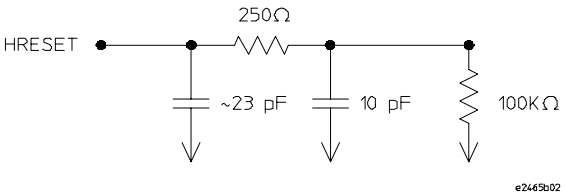
Power Requirements	120 mA @ 5V, supplied by the logic analyzer. CAT I, Pollution degree 2.
Signal Line Loading	See diagrams on next page.
Setup/Hold Requirement	For address and data alignment, the PLD requires that the AACK line has a 10.9 ns setup/0 s hold. For all signals, the logic analyzers require a minimum combined setup/hold window. For the HP 16600-series logic analysis system, the combined setup/hold must be at least 4.5 ns (such as 0/4.5, 1.0/3.5. etc). For all other logic analyzers, the combined window must be at least 3.5 ns.

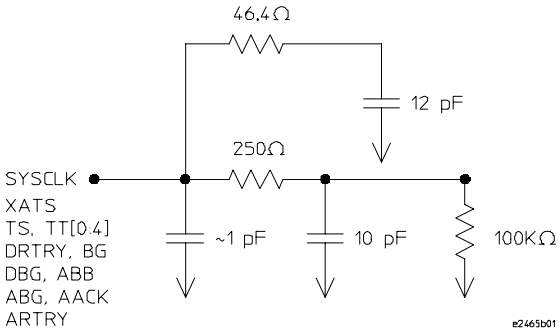
Environmental Characteristics

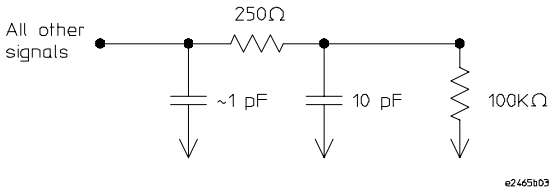
Temperature	Operating	0 to + 50 degrees C
Altitude	Operating	4,600 m
Humidity	Up to 75% noncondensing. Avoid sudden, extreme temperature changes which could cause condensation on the circuit board. For indoor use only.	

Signal line loading

The following diagrams show the loading for all signals.







Theory of operation

The HP E2465A analysis probe contains a programmable logic device (PLD) which tracks bus ownership, and tags address and data tenures owned by the PowerPC 604 with three-bit counts. These counts are used by the disassembler to correlate address and data tenures, and appear on the logic analyzer as labels "Addc" and "Datc". The bus mastership signals are true high (asserted with a logic level 1), and are labeled oAdd and oDat. A block diagram of the PLD is shown on the next page.

Target Mastership Of Data/Address Bus

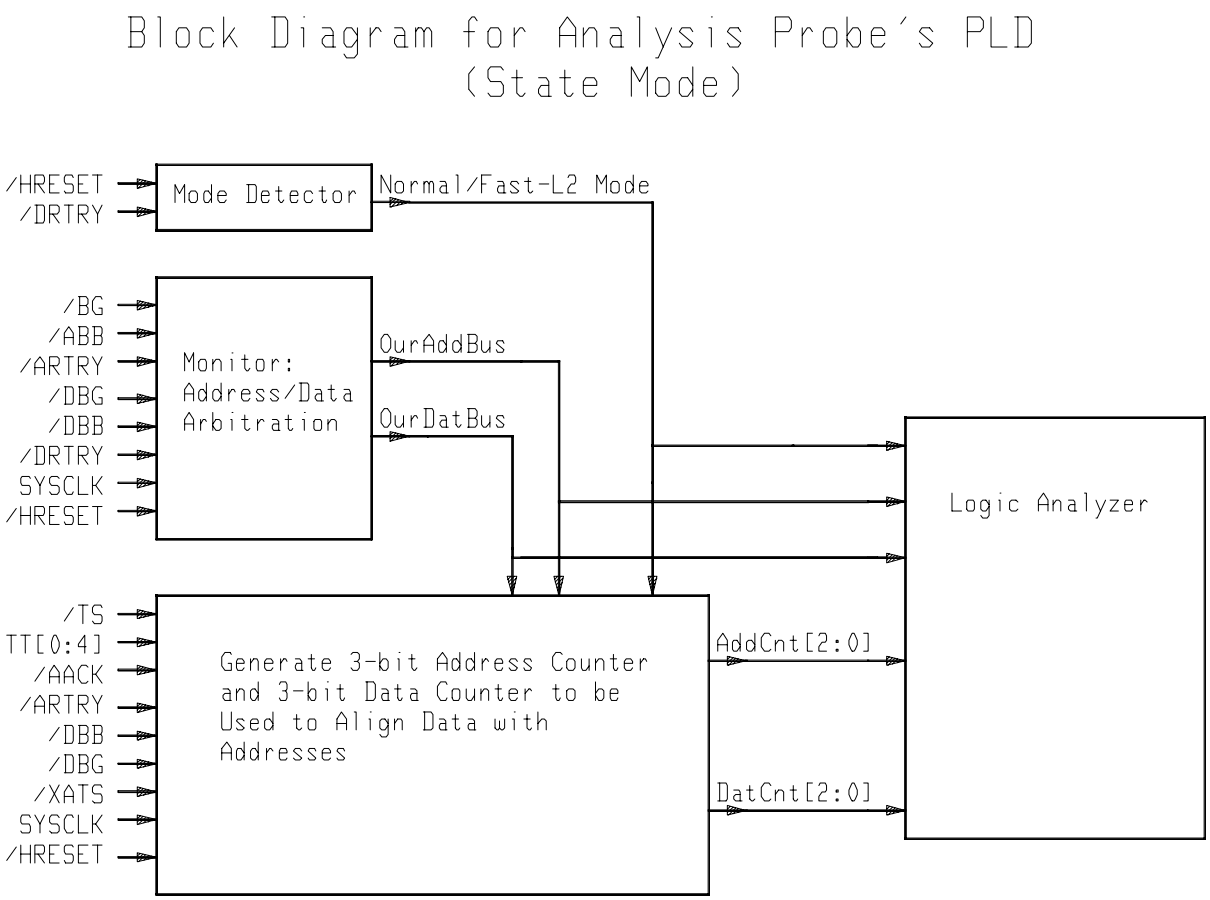
Mastership of the data bus is indicated after the target receives a qualified data bus grant. Mastership of the address bus is indicated by sampling the ABB signal one cycle after BG is asserted. If ABB is asserted, then the target processor has the bus. This will be indicated to the logic analyzer one cycle later (two cycles after BG is asserted). If ABB is NOT asserted the cycle following BG assertion, then the target chose not to take over the address bus and mastership of the address bus will not be indicated to the logic analyzer.

Normal Mode Vs. Fast-l2 Mode Of Operation

On power-up, the PLD detects whether the 604 is running in the Normal mode or Fast-L2 mode of operation by sampling DRTRY on the rising edge of HRESET. If DRTRY is asserted, then the processor will run in Fast-L2 mode instead of Normal mode. There are no data retries in Fast-L2 mode. In Normal mode, the clock to the data counter is a rising DBB. In Fast-L2 mode, the data counter clock is also a rising DBB. As a result of using DBB as the data counter clock when in Fast-L2 mode, there must always be a DBB pulse between burst operations in Fast-L2 mode. By using DBB, the PLD does not need to look at DRTRY or TBST. Using DBB also eliminates the need for pipelining the TBST bit.

Direct Store Mode

Direct store operations occur in either Normal mode or Fast-L2 mode. Direct store operations occur when XATS is strobed low. The address and data counters are not affected when direct store transactions are in progress.



HP E2465A Block Diagram

Analysis probe—signal-to-connector mapping

The following tables show the electrical signal-to-connector mapping implemented by the HP E2465A PPC604 Analysis Probe.

PowerPC 604 Signal List

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P6	3	CLK1	D7	NAPRUN		NAPRUN
P6	7	15	P13	DH0	DATA	
P6	9	14	N12	DH1	DATA	
P6	11	13	T15	DH2	DATA	
P6	13	12	U15	DH3	DATA	
P6	15	11	R13	DH4	DATA	
P6	17	10	U14	DH5	DATA	
P6	19	9	N10	DH6	DATA	
P6	21	8	P11	DH7	DATA	
P6	23	7	T11	DH8	DATA	
P6	25	6	U11	DH9	DATA	
P6	27	5	R10	DH10	DATA	
P6	29	4	U10	DH11	DATA	
P6	31	3	U9	DH12	DATA	
P6	33	2	T9	DH13	DATA	
P6	35	1	N9	DH14	DATA	
P6	37	0	P9	DH15	DATA	
P5	3	CLK1	D15	MCP		MCP
P5	7	15	R9	DH16	DATA	
P5	9	14	U8	DH17	DATA	
P5	11	13	R8	DH18	DATA	
P5	13	12	U7	DH19	DATA	
P5	15	11	N8	DH20	DATA	
P5	17	10	P7	DH21	DATA	
P5	19	9	T7	DH22	DATA	
P5	21	8	U6	DH23	DATA	
P5	23	7	R7	DH24	DATA	
P5	25	6	R6	DH25	DATA	
P5	27	5	N7	DH26	DATA	
P5	29	4	U5	DH27	DATA	
P5	31	3	T5	DH28	DATA	
P5	33	2	U4	DH29	DATA	
P5	35	1	R5	DH30	DATA	
P5	37	0	U3	DH31	DATA	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P4	3	CLK1	E14	INT	INT	
P4	7	15	L16	DL0	DATA_B	
P4	9	14	K15	DL1	DATA_B	
P4	11	13	M17	DL2	DATA_B	
P4	13	12	L14	DL3	DATA_B	
P4	15	11	N17	DL4	DATA_B	
P4	17	10	M15	DL5	DATA_B	
P4	19	9	N16	DL6	DATA_B	
P4	21	8	L13	DL7	DATA_B	
P4	23	7	M13	DL8	DATA_B	
P4	25	6	N15	DL9	DATA_B	
P4	27	5	P17	DL10	DATA_B	
P4	29	4	R17	DL11	DATA_B	
P4	31	3	N14	DL12	DATA_B	
P4	33	2	P15	DL13	DATA_B	
P4	35	1	R16	DL14	DATA_B	
P4	37	0	U16	DL15	DATA_B	
P3	3	CLK1	C9	CKSTP_IN		CHKIN
P3	7	15	R14	DL16	DATA_B	
P3	9	14	N11	DL17	DATA_B	
P3	11	13	T13	DL18	DATA_B	
P3	13	12	R12	DL19	DATA_B	
P3	15	11	U13	DL20	DATA_B	
P3	17	10	R11	DL21	DATA_B	
P3	19	9	U12	DL22	DATA_B	
P3	21	8	N3	DL23	DATA_B	
P3	23	7	P3	DL24	DATA_B	
P3	25	6	N4	DL25	DATA_B	
P3	27	5	R2	DL26	DATA_B	
P3	29	4	T1	DL27	DATA_B	
P3	31	3	T3	DL28	DATA_B	
P3	33	2	R4	DL29	DATA_B	
P3	35	1	P5	DL30	DATA_B	
P3	37	0	N6	DL31	DATA_B	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P2	3	CLK1	E9	CKSTP_OUT		CHKOUT
P2	7	15	F13	A0	ADDR	
P2	9	14	E1	A1	ADDR	
P2	11	13	D17	A2	ADDR	
P2	13	12	F3	A3	ADDR	
					ADDR	
P2	15	11	E16	A4	ADDR	
P2	17	10	F1	A5	ADDR	
P2	19	9	E17	A6	ADDR	
P2	21	8	G5	A7		
P2	23	7	F15	A8	ADDR	
P2	25	6	G4	A9	ADDR	
P2	27	5	G13	A10	ADDR	
P2	29	4	G3	A11	ADDR	
P2	31	3	F17	A12	ADDR	
P2	33	2	G2	A13	ADDR	
P2	35	1	G14	A14	ADDR	
P2	37	0	G1	A15	ADDR	
P1	3	CLK1	L15	TS	TS	
P1	7	15	G15	A16	ADDR	
P1	9	14	H1	A17	ADDR	
P1	11	13	G16	A18	ADDR	
P1	13	12	H3	A19	ADDR	
P1	15	11	G17	A20	ADDR	
P1	17	10	J1	A21	ADDR	
P1	19	9	H13	A22	ADDR	
P1	21	8	H5	A23	ADDR	
P1	23	7	H15	A24	ADDR	
P1	25	6	J2	A25	ADDR	
P1	27	5	H17	A26	ADDR	
P1	29	4	J3	A27	ADDR	
P1	31	3	J13	A28	ADDR	
P1	33	2	L1	A29	ADDR	
P1	35	1	K13	A30	ADDR	
P1	37	0	M1	A31	ADDR	

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels	
P7	3	CLK1	K17	XATS		XATS	
P7	7	15	*	OurAddBus	STAT	oAdd	
P7	9	14	*	AddCnt2	STAT	Addc	
P7	11	13	*	AddCnt1	STAT	Addc	
P7	13	12	*	AddCnt0	STAT	Addc	
P7	15	11	*	OurDatBus	STAT	oDat	
P7	17	10	*	DatCnt2	STAT	Datc	
P7	19	9	*	DatCnt1	STAT	Datc	
P7	21	8	*	DatCnt0	STAT	Datc	
P7	23	7	B9	HRESET	STAT	HRSET	
P7	25	6	D13	SRESET	STAT	SRSET	
P7	27	5	E8	BR	STAT	BR	busarb
P7	29	4	J5	BG	STAT	BG	busarb
P7	31	3	L3	ABB	STAT	ABB	busarb
P7	33	2	K1	DBG	STAT	DBG	busarb
P7	35	1	L17	DBB	STAT	DBB	busarb
P7	37	0	J14	TEA	STAT	TEA	
* These signals are generated by the analysis probe.							
P8	3	CLK1	C10	SYSCLK		SYSCLK	
P8	7	15	E11	TSIZ0	STAT	TSIZ	
P8	9	14	A15	TSIZ1	STAT	TSIZ	
P8	11	13	B15	TSIZ2	STAT	TSIZ	
P8	13	12	E12	TBST	STAT	TSIZ	TBST
P8	15	11	A16	TT0	STAT	TT	TT0
P8	17	10	C14	TT1	STAT	TT	R/W
P8	19	9	C16	TT2	STAT	TT	TT2
P8	21	8	C17	TT3	STAT	TT	TT3
P8	23	7	E15	TT4	STAT	TT	TT4
P8	25	6	C6	TC0	STAT	TC	TC0
P8	27	5	A5	TC1	STAT	TC	TC1
P8	29	4	C7	TC2	STAT	TC	TC2
P8	31	3	K3	AACK	STAT	ACKs	AACK
P8	33	2	K5	ARTRY	STAT	ACKs	ARTRY
P8	35	1	J16	TA	STAT	ACKs	TA
P8	37	0	J17	DRTRY	STAT	ACKs	DRTRY

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P9	3	CLK1	--			
P9	7	15	L4	DP0	DP	
P9	9	14	N1	DP1	DP	
P9	11	13	M3	DP2	DP	
P9	13	12	N2	DP3	DP	
P9	15	11	P1	DP4	DP	
P9	17	10	L5	DP5	DP	
P9	19	9	R1	DP6	DP	
P9	21	8	M5	DP7	DP	
P9	23	7	E6	AP0	AP	
P9	25	6	C4	AP1	AP	
P9	27	5	C5	AP2	AP	
P9	29	4	A4	AP3	AP	
P9	31	3	B7	DPE	DPE	
P9	33	2	C8	APE	APE	
P9	35	1	E7	CSE0	CSE	CSE0
P9	37	0	B5	CSE1	CSE	CSE1
P10		CLK1	C12	TCK	TCK	
P10		15	A12	TRST	TRST	
P10		14	C13	TMS	TMS	
P10		13	B13	TDI	TDI	
P10		12	A14	TDO	TDO	
P10		11	J15	DBDIS	DBDIS	
P10		10	J4	DBWO	DBWO	
P10		9	L2	SHD	SHD	
P10		8	F5	GBL	GBL	
P10		7	D1	WT	WT	
P10		6	E2	CI	CI	
P10		5	D5	RSRV	RSRV	
P10		4	E4	TBEN	TBEN	
P10		3	D9	HALTED	HALT	
P10		2	B17	SMI	SMI	
P10		1	A7	L2_INT	L2_INT	
P10		0	*	FastL2	FastL2	

* This signal is generated by the analysis probe.

Pod	2x20 pin	Analyzer Bit	PGA Pin #	Signal Name	Analyzer Labels	Analyzer Labels
P11	3	CLK1				
P11	7	15				
P11	9	14				
P11	11	13				
P11	13	12				
P11	15	11				
P11	17	10				
P11	19	9				
P11	21	8				
P11	23	7				
P11	25	6				
P11	27	5				
P11	29	4				
P11	31	3	A9	PLLCFG0	PLLCFG	PLLCFG0
P11	33	2	A10	PLLCFG1	PLLCFG	PLLCFG1
P11	35	1	A13	PLLCFG2	PLLCFG	PLLCFG2
P11	37	0	C11	PLLCFG3	PLLCFG	PLLCFG3

Emulation module—operating characteristics

The following operating characteristics are not specifications, but are typical operating characteristics for the HP 16610A emulation module and PPC604 target interface module.

Operating Characteristics

Microprocessor Compatibility	PPC604A and PPC604e microprocessors.
Environmental Characteristics (Temperature, Altitude, Humidity)	The HP 16610A emulation module meets the environmental characteristics of the logic analysis system in which it is installed. For indoor use only.

Emulation module—electrical characteristics

Maximum Ratings

Characteristics for the PowerPC 604 emulation module	Symbol	Min	Max
TDO, $\overline{\text{CHECKSTOP}}$	V_{ih}	2.0 V	5.5 V
	V_{il}		0.8 V
	I_i		±1 µA
	C_{in}		15 pF
TDI, TCK, TMS, $\overline{\text{TRST}}$ [1]	V_{oh} @ I_{oh} = -32 mA	2.0 V	2.8 V
	V_{ol} @ I_{ol} = 64 mA; V_{CC}=4.5V		0.55 V
TDI, TMS, $\overline{\text{TRST}}$	C_o		25 pF
TCK	C_o		45 pF
+3.3V Power Sense [2]	V_{ih}	2.0 V	5.3 V
	V_{il}	-0.3 V	0.8 V
SRESET, $\overline{\text{HRESET}}$ [3]	V_{ol} @ I_{ol} = 12 ma		0.5 V
	C_o		25 pF
TS0 - TS6, SYSCLK	C_{in}		10 pF
	V_{ih}	2.0 V	5.5 V
	V_{il}		0.8 V
	I_i		±1 µA

1 These signals must not be actively driven by the target system when the debug port is being used.

2 Power Sense is used only to determine target powered status. The processor probe does not draw power from this source.

3 Open collector outputs, pulled up to a generated voltage equivalent to the Power Sense voltage with a 2.61 K pullup resistor

10

General-Purpose ASCII (GPA) Symbol File Format

General-Purpose ASCII (GPA) Symbol File Format

General-purpose ASCII (GPA) format files are loaded into a logic analyzer just like other object files, but they are usually created differently.

If your compiler is not one of those listed on page 97, if your compiler does not include symbol information in the output, or if you want to define a symbol not in the object file, you can create an ASCII format symbol file.

Typically, ASCII format symbol files are created using text processing tools to convert compiler or linker map file output that has symbolic information into the proper format.

You can typically get symbol table information from a linker map file to create a General-Purpose ASCII (GPA) symbol file.

Various kinds of symbols are defined in different records in the GPA file. Record headers are enclosed in square brackets; for example, [VARIABLES]. For a summary of GPA file records and associated symbol definition syntax, refer to the "GPA Record Format Summary" that follows.

Each entry in the symbol file must consist of a symbol name followed by an address or address range.

While symbol names can be very long, the logic analyzer only uses the first 16 characters.

The address or address range corresponding to a given symbol appears as a hexadecimal number. The address or address range must immediately follow the symbol name, appear on the same line, and be separated from the symbol name by one or more blank spaces or tabs. Ensure that address ranges are in the following format:

```
beginning address..ending address
```

Example

```
main     00001000..00001009
test     00001010..0000101F
var1     00001E22     #this is a variable
```

This example defines two symbols that correspond to address ranges and one point symbol that corresponds to a single address.

For more detailed descriptions of GPA file records and associated symbol definition syntax, refer to these topics that follow:

- SECTIONS
- FUNCTIONS
- VARIABLES
- SOURCE LINES
- START ADDRESS
- Comments

GPA Record Format Summary

Format

```
[SECTIONS]
section_name  start..end  attribute

[FUNCTIONS]
func_name  start..end

[VARIABLES]
var_name   start [size]
var_name   start..end

[SOURCE LINES]
File: file_name
line#  address

[START ADDRESS]
address

#Comments
```

If no record header is specified, [VARIABLES] is assumed. Lines without a preceding header are assumed to be symbol definitions in one of the VARIABLES formats.

Example

This is an example GPA file that contains several different kinds of records:

```
[SECTIONS]
prog     00001000..0000101F
data     40002000..40009FFF
common   FFFF0000..FFFF1000

[FUNCTIONS]
main     00001000..00001009
test     00001010..0000101F

[VARIABLES]
total    40002000  4
value    40008000  4
```

```
[SOURCE LINES]
File: main.c
10       00001000
11       00001002
14       0000100A
22       0000101E

File: test.c
 5       00001010
 7       00001012
11       0000101A
```

SECTIONS

Format

```
[SECTIONS]
section_name  start..end  attribute
```

Use SECTIONS to define symbols for regions of memory, such as sections, segments, or classes.

`section_name` A symbol representing the name of the section.

`start` The first address of the section, in hexadecimal.

`end` The last address of the section, in hexadecimal.

`attribute` This is optional, and may be one of the following:

NORMAL (default)—The section is a normal, relocatable section, such as code or data.

NONRELOC—The section contains variables or code that cannot be relocated; this is an absolute segment.

> Enable Section Relocation
>
> To enable section relocation, section definitions must appear before any other definitions in the file.

Example

```
[SECTIONS]
prog           00001000..00001FFF
data           00002000..00003FFF
display_io     00008000..0000801F  NONRELOC
```

If you use section definitions in a GPA symbol file, any subsequent function or variable definitions must be within the address ranges of one of the defined sections. Functions and variables that are not within the range are ignored.

FUNCTIONS

Format

```
[FUNCTIONS]
func_name  start..end
```

Use FUNCTIONS to define symbols for program functions, procedures, or subroutines.

`func_name` A symbol representing the function name.

`start` The first address of the function, in hexadecimal.

`end` The last address of the function, in hexadecimal.

Example

```
[FUNCTIONS]
main     00001000..00001009
test     00001010..0000101F
```

VARIABLES

Format

```
[VARIABLES]
var_name   start [size]
var_name   start..end
```

You can specify symbols for variables either by using the address of the variable, the address and the size of the variable, or a range of addresses occupied by the variable. If you specify only the address of a variable, the size is assumed to be one byte.

`var_name` A symbol representing the variable name.

`start` The first address of the variable, in hexadecimal.

`end` The last address of the variable, in hexadecimal.

`size` This is optional, and indicates the size of the variable, in bytes, in decimal.

Example

```
[VARIABLES]
subtotal     40002000   4
total        40002004   4
data_array   40003000..4000302F
status_char  40002345
```

SOURCE LINES

Format

```
[SOURCE LINES]
File: file_name
line#  address
```

Use SOURCE LINES to associate addresses with lines in your source files.

`file_name` The name of a file.

`line#` The number of a line in the file, in decimal.

`address` The address of the source line, in hexadecimal.

Example

```
[SOURCE LINES]
File: main.c
10       00001000
11       00001002
14       0000100A
22       0000101E
```

START ADDRESS

Format

```
[START ADDRESS]
address
```

`address` The address of the program entry point, in hexadecimal.

Example

```
[START ADDRESS]
00001000
```

Comments

Format

```
#comment text
```

Use the # character to include comments in a file. Any text following the # character is ignored. You can put comments on a line alone or on the same line following a symbol entry.

Example

```
#This is a comment.
```

11

Troubleshooting the Analysis Probe

Troubleshooting the Analysis Probe

If you encounter difficulties while making measurements, use this chapter to guide you through some possible solutions. Each heading lists a problem you may encounter, along with some possible solutions.

If you still have difficulty using the analyzer after trying the suggestions in this chapter, please contact your local Hewlett-Packard service center.

CAUTION

When you are working with the analyzer, be sure to power down both the analyzer and the target system before disconnecting or connecting cables, probes, and analysis probes. Otherwise, you may damage circuitry in the analyzer, analysis probe, or target system.

Logic Analyzer Problems

This section lists general problems that you might encounter while using the logic analyzer.

Intermittent data errors

This problem is usually caused by poor connections, incorrect signal levels, or marginal timing.

- ☐ **Remove and reseat all cables and probes, ensuring that there are no bent pins on the analysis probe or poor probe connections.**
- ☐ **Adjust the threshold level of the data pod to match the logic levels in the system under test.**
- ☐ **Use an oscilloscope to check the signal integrity of the data lines.**

  Clock signals for the state analyzer must meet particular pulse shape and timing requirements. Data inputs for the analyzer must meet pulse shape and setup and hold time requirements.

See Also See "Capacitive Loading" in this chapter for information on other sources of intermittent data errors.

Unwanted triggers

Unwanted triggers can be caused by instructions that were fetched but not executed.

- ☐ **Add the prefetch queue or pipeline depth to the trigger address to avoid this problem.**

  The logic analyzer captures prefetches, even if they are not executed. When you are specifying a trigger condition or a storage qualification that follows an instruction that may cause branching, an unused prefetch may generate an unwanted trigger.

No activity on activity indicators

- ☐ Check for loose cables, board connections, and analysis probe connections.
- ☐ Check for bent or damaged pins on the analysis probe.

No trace list display

If there is no trace list display, it may be that your trigger specification is not correct for the data you want to capture, or that the trace memory is only partially filled.

- ☐ Check your trigger sequencer specification to ensure that it will capture the events of interest.
- ☐ Try stopping the analyzer; if the trace list is partially filled, this should display the contents of trace memory.

Analyzer won't power up

If logic analyzer power is cycled when the logic analyzer is connected to a target system or emulation probe that remains powered up, the logic analyzer may not be able to power up. Some logic analyzers are inhibited from powering up when they are connected to a target system or emulation probe that is already powered up.

- ☐ Remove power from the target system, then disconnect all logic analyzer cabling from the analysis probe. This will allow the logic analyzer to power up. Reconnect logic analyzer cabling after power up.

Analysis Probe Problems

This section lists problems that you might encounter when using an analysis probe. If the solutions suggested here do not correct the problem, you may have a damaged analysis probe. Contact your local Hewlett-Packard Sales Office if you need further assistance.

Target system will not boot up

If the target system will not boot up after connecting the analysis probe, the microprocessor (if socketed) or the analysis probe may not be installed properly, or they may not be making electrical contact.

- ☐ **Ensure that you are following the correct power-on sequence for the analysis probe and target system.**

  1 Power up the analyzer and analysis probe.

  2 Power up the target system.

  If you power up the target system before you power up the analysis probe, interface circuitry in the analysis probe may latch up and prevent proper target system operation.

- ☐ **Verify that the microprocessor and the analysis probe are properly rotated and aligned, so that the index pin on the microprocessor (pin A1) matches the index pin on the analysis probe.**

- ☐ **Verify that the microprocessor and the analysis probe are securely inserted into their respective sockets.**

- ☐ **Verify that the logic analyzer cables are in the proper sockets of the analysis probe and are firmly inserted.**

Erratic trace measurements

- **Do a full reset of the target system before beginning the measurement.**

  Some analysis probe designs require a full reset to ensure correct configuration.

- **Ensure that your target system meets the timing requirements of the processor with the analysis probe installed.**

  See “Capacitive loading” in this chapter. While analysis probe loading is slight, pin protectors, extenders, and adapters may increase it to unacceptable levels. If the target system design has close timing margins, such loading may cause incorrect processor functioning and give erratic trace results.

- **Ensure that you have sufficient cooling for the microprocessor.**

  Ensure that you have ambient temperature conditions and airflow that meet or exceed the requirements of the microprocessor manufacturer.

Capacitive loading

Excessive capacitive loading can degrade signals, resulting in incorrect capture by the analysis probe, or system lockup in the microprocessor. All analysis probes add additional capacitive loading, as can custom probe fixtures you design for your application.

Careful layout of your target system can minimize loading problems and result in better margins for your design. This is especially important for systems that are running at frequencies greater than 50 MHz.

- **Remove as many pin protectors, extenders, and adapters as possible.**

- **If multiple analysis probe solutions are available, use one with lower capacitive loading.**

Inverse Assembler Problems

This section lists problems that you might encounter while using the inverse assembler.

When you obtain incorrect inverse assembly results, it may be unclear whether the problem is in the analysis probe or in your target system. If you follow the suggestions in this section to ensure that you are using the analysis probe and inverse assembler correctly, you can proceed with confidence in debugging your target system.

No inverse assembly or incorrect inverse assembly

This problem may be due to incorrect synchronization, modified configuration, incorrect connections, or a hardware problem in the target system. A locked status line can cause incorrect or incomplete inverse assembly.

- ☐ **Ensure that each logic analyzer pod is connected to the correct analysis probe connector.**

  There is not always a one-to-one correspondence between analyzer pod numbers and analysis probe cable numbers. Analysis Probes must supply address (ADDR), data (DATA), and status (STAT) information to the analyzer in a predefined order. The cable connections for each analysis probe are often altered to support that need. Thus, one analysis probe might require that you connect cable 2 to analyzer pod 2, while another will require you to connect cable 5 to analyzer pod 2. See Chapter 3 for connection information.

- ☐ **Check the activity indicators for status lines locked in a high or low state.**

- ☐ **Verify that the STAT, DATA, and ADDR format labels have not been modified from their default values.**

  These labels must remain as they are configured by the configuration file. Do not change the names of these labels or the bit assignments within the labels.

Some analysis probes also require other data labels. See Chapter 3 for more information.

- ☐ **Verify that all microprocessor caches and memory managers have been disabled.**

  In most cases, if the microprocessor caches and memory managers remain enabled you should still get inverse assembly. It may be incorrect because a portion of the execution trace was not visible to the logic analyzer.

  To determine if a cache is on or off, examine the most significant bit of the ICCST register (for the instruction cache) or the DCCST register (for data cache). If this bit is 1, the cache is on; if the bit is 0, the cache is off.

  For instructions on how to disable the cache, see page 73.

- ☐ **Verify that storage qualification has not excluded storage of all the needed opcodes and operands.**

- ☐ **Verify that the data format is big-endian.**

  Unlike most processors, the PPC604 can run in either little-endian or big-endian mode. The inverse assembler can only decode data in big-endian format. To verify the format of the data, the MSR or Machine State Register must be examined. The least significant bit (bit 0 according to Motorola convention, or bit 31 according to IBM convention) indicates the mode of the processor. A value of 1 indicates little-endian mode. A value of 0 indicates big-endian mode.

Inverse assembler will not load or run

You need to ensure that you have the correct system software loaded on your analyzer.

- ☐ **Ensure that the inverse assembler is on the same disk as the configuration files you are loading.**

  Configuration files for the state analyzer contain a pointer to the name of the corresponding inverse assembler. If you delete the inverse assembler or rename it, the configuration process will fail to load the disassembler.

  See Chapter 2 for details.

Intermodule Measurement Problems

Some problems occur only when you are trying to make a measurement involving multiple modules.

An event wasn't captured by one of the modules

If you are trying to capture an event that occurs very shortly after the event that arms one of the measurement modules, it may be missed due to internal analyzer delays. For example, suppose you set an oscilloscope module to trigger upon receiving a trigger signal from the logic analyzer because you are trying to capture a pulse that occurs right after the analyzer's trigger state. If the pulse occurs too soon after the analyzer's trigger state, the oscilloscope will miss the pulse.

- ☐ **Adjust the skew in the Intermodule menu.**

  You may be able to specify a skew value that enables the event to be captured.

- ☐ **Change the trigger specification for modules upstream of the one with the problem.**

  If you are using a logic analyzer to trigger an oscilloscope module, try specifying a trigger state one state before the one you are using. This may be more difficult than working with the skew because the prior state may occur more often and not always be related to the event you are trying to capture with the oscilloscope.

Analysis Probe Messages

This section lists some of the messages that the analyzer displays when it encounters a problem.

". . . Inverse Assembler Not Found"

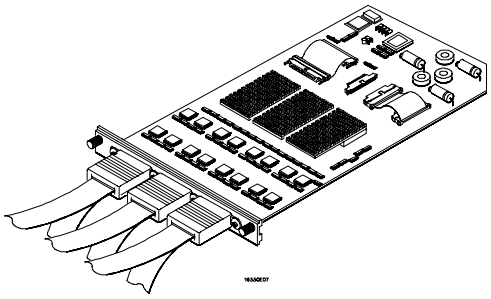
This error occurs if you rename or delete the inverse assembler file that is attached to the configuration file.

Ensure that the inverse assembler file is not renamed or deleted, and that it is located in the the correct directory:

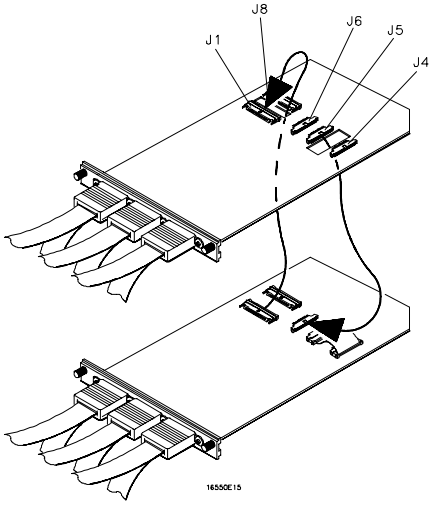
- For HP 16600A/700A-series logic analysis systems it should be in /hplogic/ia.
- For other logic analyzers it should be in the same directory as the configuration file.

"Measurement Initialization Error"

This error occurs when you have installed the cables incorrectly for one or two HP 16550A logic analysis cards. The following diagrams show the correct cable connections for one-card and two-card installations. Ensure that your cable connections match the silk screening on the card, and that they are fully seated in the connectors. Then, repeat the measurement.



Cable Connections for One-Card HP 16550A Installations



Cable Connections for Two-Card HP 16550A Installations

See Also

The *HP 16550A 100-MHz State/500-MHz Timing Logic Analyzer Service Guide*.

"No Configuration File Loaded"

This is usually caused by trying to load a configuration file for one type of module/system into a different type of module/system.

- ☐ Verify that the appropriate module has been selected from the Load {module} from File {filename} in the disk operation menu. Selecting Load {All} will cause incorrect operation when loading most analysis probe configuration files.

See Also Chapter 3 describes how to load configuration files.

"Selected File is Incompatible"

This occurs when you try to load a configuration file for the wrong module. Ensure that you are loading the appropriate configuration file for your logic analyzer.

"Slow or Missing Clock"

- ☐ This error message might occur if the logic analyzer cards are not firmly seated in the logic analysis system frame. Ensure that the cards are firmly seated.
- ☐ This error might occur if the target system is not running properly. Ensure that the target system is on and operating properly.
- ☐ If the error message persists, check that the logic analyzer pods are connected to the proper connectors on the analysis probe. See Chapter 3 to determine the proper connections.

"Time from Arm Greater Than 41.93 ms"

The HP 16550A state/timing analyzers have a counter to keep track of the time from when an analyzer is armed to when it triggers. The width and clock rate of this counter allow it to count for up to 41.93 ms before it overflows. Once the counter has overflowed, the system does not have the data it needs to calculate the time between module triggers. The system must know this time to be able to display data from multiple modules on a single screen.

"Waiting for Trigger"

If a trigger pattern is specified, this message indicates that the specified trigger pattern has not occurred. Verify that the triggering pattern is correctly set.

- ☐ When analyzing microprocessors that fetch only from word-aligned addresses, ensure that the trigger condition is set to look for an opcode fetch at an address corresponding to a word boundary.

Returning Parts to Hewlett-Packard for Service

The repair strategy for this emulation solution is board replacement.

Exchange assemblies are available when a repairable assembly is returned to Hewlett-Packard. These assemblies have been set up on the "Exchange Assembly" program. This lets you exchange a faulty assembly with one that has been repaired, calibrated, and performance verified by the factory. The cost is significantly less than that of a new assembly.

To return a part to Hewlett-Packard

1. **Follow the procedures in this chapter to make sure that the problem is caused by a hardware failure, not by configuration or cabling problems.**
2. **In the U.S., call 1-800-403-0801. Outside the U.S., call your nearest HP sales office. Ask them for the address of the nearest HP service center.**
3. **Package the part and send it to the HP service center.**

   Keep any parts which you know are working. For example, if only the target interface module is broken, keep the emulation module and cables.
4. **When the part has been replaced, it will be sent back to you.**

   The unit returned to you will have the same serial number as the unit you sent to HP.

The HP service center can also troubleshoot the hardware and replace the failed part. To do this, send your entire measurement system to the service center, including the logic analysis system, analysis probe, and cables.

In some parts of the world, on-site repair service is available. Ask the HP sales or service representative for details.

To obtain replacement parts

The following table lists some parts that may be replaced if they are damaged or lost. Contact your nearest Hewlett-Packard Sales Office for further information.

Analysis Probe Replaceable Parts

HP Part Number	Description
E2465-66501	PowerPC 604 Analysis Probe Circuit Board
1200-1975	Pin Protector
E2465-68702	Configuration and Inverse Assembler Software

Cleaning the Instrument

If the instrument requires cleaning:

1. Remove power from the instrument.
2. Clean the instrument with a mild detergent and water.
3. Make sure that the instrument is completely dry before reconnecting it to a power source.

12

Troubleshooting the Emulation Module

Troubleshooting the Emulation Module

If you have problems with the emulation module, your first task is to determine the source of the problem. Problems may originate in any of the following places:

- The connection between the emulation module and your debugger
- The emulation module itself
- The connection between the emulation module and the target interface module
- The connection between the target interface module and the target system
- The target system

You can use several means to determine the source of the problem:

- The troubleshooting guide on the next page
- The status lights on the emulation module
- The emulation module "performance verification" tests
- The emulation module's built-in "terminal interface" commands

Emulation Module Troubleshooting Guide

Common problems and what to do about them

Symptom	What to do	See also
Commands from the Emulation Control Interface have no effect	Check that you are using the correct firmware.	
Commands from debugger have no effect	Use the Emulation Control Interface to try a few built-in commands. If this works, your debugger may not be configured properly. If this does not work, continue with the steps for the next symptom....	page 229
Emulation module built-in commands do not work	**1** Check that the emulation module has been properly configured for your target system.	page 132
	2 Run the emulation module performance verification tests.	page 245
	3 If the performance verification tests pass, then there is an electrical problem with the connection to the target processor OR the target system may not have been designed according to "Designing a Target System."	page 114, page 232
"Slow or missing clock" message after a logic analyzer run	Check that the target system is running user code or is in reset. (This message can appear if the processor is in background mode.)	
"Slow clock" message in the Emulation Control Interface or "c>" prompt in the built-in terminal interface	Check that the clock rate is properly configured.	page 138
Some commands fail	Check the "restrict to real-time runs" configuration	page 137

Emulation Module Status Lights

The emulation module uses status lights to communicate various modes and error conditions.

The following table gives more information about the meaning of the power and target status lights.

❍ = LED is off
● = LED is on
✳ = Not applicable (LED is off or on)

Power/Target Status Lights

Pwr/Target LEDs	Meaning
❍ Reset ❍ Break ❍ Run	No target system power, or emulation module is not connected to the target system
● Reset ❍ Break ❍ Run	Target system is in a reset state
❍ Reset ● Break ❍ Run	The target processor is executing in background mode
❍ Reset ❍ Break ● Run	The target processor is executing user code
❍ Reset ● Break ● Run	Only boot firmware is good (other firmware has been corrupted)

Emulation Module Built-in Commands

The emulation module has some built-in "terminal interface" commands which you can use for troubleshooting.

You can access the terminal interface using:

- A telnet (LAN) connection
- The Command Line window in the Emulation Control Interface
- A "debugger command" window in your debugger

To telnet to the emulation module

You can establish a telnet connection to the emulation module if:

- A host computer and the logic analysis system are both connected to a local-area network (LAN), and
- The host computer has the telnet program (often part of the operating system or an internet software package).

To establish a telnet connection:

1. **Find out the port number of the emulation module.**

   The default port number of the first emulation module in an HP 16600A/700A series logic analysis system is 6472. The default port of a second module in an HP 16600A-series system is 6476. The default port numbers of a third and fourth module in an expansion frame are 6480 and 6484. These port numbers can be changed, but that is rarely necessary.
2. **Find out the LAN address or LAN name of the logic analysis system.**
3. **Start the telnet program.**

   If the LAN name of the logic analysis system is "test2" and you have only one emulation module installed, the command might look like this:

   ```
   telnet test2 6472
   ```
4. **If you do not see a prompt, press the <Return> key a few times.**

   To exit from this telnet session, type <CTRL>D at the prompt.

To use the built-in commands

Here are a few commonly used built-in commands:

Useful built-in commands

b	Break—go into the background monitor state
cf	Configuration—read or write configuration options
help	Help—display online help for built-in commands
init	Initialize—init -c re-initializes everything in the emulation module except for the LAN software; init -p is the equivalent of cycling power (it will break LAN connections)
lan	configure LAN address (emulation probes only)
m	Memory—read or write memory
reg	Register—read or write a register
r	Run—start running user code
rep	Repeat—repeat a command or group of commands
rst	Reset—reset the target processor (the emulation module will wait for you to press the target's RESET button)
s	Step—do a low-level single step
ver	Version—display the product number and firmware version of the emulation module

The prompt indicates the status of the emulation module:

Emulation module prompts

U	Running user program
M	Running in background monitor
c	Target is checkstopped
p	No target power
d	No target interface module connected to emulation module
?	Unknown state

Examples

To set register GPR0, then view GPR0 to verify that it was set, enter:

```
R>rst -m
M>reg GPR0=ffff
M>reg GPR0
  reg GPR0=0000ffff
```

To break execution then step a single instruction, enter:

```
M>b
M>s
  PC=xxxxxxxx
M>
```

To determine what firmware version is installed in the emulation module, enter:

```
M>ver
```

See Also

Use the `help` command for more information on these and other commands. Note that some of commands listed in the help screens are generic commands for HP emulators and may not be available for your product.

If you are writing your own debugger, contact HP for more information.

Problems with the Target System

This section describes how to determine whether your target system is causing problems with the operation of the emulation module.

What to check first

1 Try some basic built-in commands using the Command Line window or a telnet connection:

```
U>rst
R>
```

This should reset the target and display an "R>" prompt.

```
R>b
M>
```

This should stop the target and display an "M>" prompt.

```
M>reg r1
  reg r1=00000000
M>
```

This should read the value of the r1 register (the value will probably be different on your target system).

```
M>m 0..
  00000000   7c3043a6 7c2802a6 7c3143a6 4bf04111
  00000010   00000000 00000000 00000000 00000000
  00000020   00000000 00000000 00000000 00000000
  00000030   00000000 00000000 00000000 00000000
  00000040   00000000 00000000 00000000 00000000
  00000050   00000000 00000000 00000000 00000000
  00000060   00000000 00000000 00000000 00000000
  00000070   00000000 00000000 00000000 00000000
M>
```

This should display memory values starting at address 0.

```
M>s
```

This should execute one instruction at the current program counter.

If any of these commands don't work, there may be a problem with the design of your target system, a problem with the revision of the processor you are using, or a problem with the configuration of the emulation module.

2 Check that the emulation module firmware matches your processor. To do this, enter:

```
M>ver
```

See Also Page 229 for information on entering built-in commands.

To check the debug port connector signals

- Check for the following logic levels on the target debug port. The signal names are for the PPC 6xx.

Levels with the emulation module not connected

Header Pin	Signal Name	Level
3	TDI	Low
4	$\overline{\text{TRST}}$	High
6	+POWER	V_{DD}
7	TCK	High
9	TMS	High
11	$\overline{\text{SRESET}}$	High
13	$\overline{\text{HRESET}}$	High
15	$\overline{\text{CHECKSTOP}}$	High
16	GND	Low

Levels with the emulation module connected

Header Pin	Signal Name	I/O
1	TDO	Toggle with "es" command
3	TDI	Toggle with "es" command
4	$\overline{\text{TRST}}$	Low pulse with "rst" command
6	+POWER	V_{DD}
7	TCK	10+ MHz clock (default)
9	TMS	Low, pulse with "es" command
11	$\overline{\text{SRESET}}$	High, pulse low with "rst" command
13	$\overline{\text{HRESET}}$	High, pulse low with "rst" command
15	$\overline{\text{CHECKSTOP}}$	High
16	GND	Low

For the COP on the PowerPC 604 to work reliably, the following pins must be tied high: L1_TSTCLK, L2_TSTCLK, ARRAY_WR, LSSD_MODE, DRVMOD0, and DRVMOD1.

To interpret the initial prompt

The initial prompt can be used to diagnose several common problems. To get the most information from the prompt, follow this procedure:

1 Connect the emulation module to your target system.

2 Set the default configuration settings. Enter:

```
M>init -c
```

You can enter this command at any prompt. The emulation module will respond with the same information as printed by the "ver" command.

If the response is "!ERROR 905! Driver firmware is incompatible with ID of attached device"

Make sure the target interface module is connected to the cable of the emulation module, then try the "init -c" command again.

If the initial prompt is "p>"

Check pin 6 on header, 3.3V (V_{OD}).

If the initial prompt is "M>"

The processor entered debug mode without the help of the emulation module. Is another debugger connected?

If the initial prompt is "c>"

Processor is checkstopped. Something caused a machine exception before the emulation module connected or $\overline{\text{CHECKSTOP}}$ is being pulled or held low.

If the initial prompt is "?>" with "ERROR 171!"

A bad status code (0xXX) was received from the processor. Valid status is 0x01 or 0x05. Any other status indicates a bad scan of the instruction register. Check TCK, TDO, TDI, TMS, and $\overline{\text{TRST}}$_L signals. Check the firmware revision.

If the initial prompt is "U>"

The emulation module is scanning the instruction register correctly. Now you can do some more tests:

3 Enter the reset command:

```
U>rst
U>
```

The "U>" prompt is a good response that indicates SRESET and HRESET are working. Continue with "If the prompt after rst is U>".

If the prompt after rst is "?>" with "ERROR 171!"

A bad status code (0xXX) was received from the processor. Valid status is 0x01, any other status indicates bad scan of IR or failure of the reset signals. Verify TCK, TDO, TDI,TMS, and $\overline{\text{TRST}}$ are all changing state on an $\overline{\text{HRESET}}$.

If the rst command fails

Set "cf reset=rom" (no external bus cycles used in this mode), then enter the "rst" command again:

```
*>cf reset=rom
*>rst
M>
```

You can enter these commands at any prompt, shown here as "*>".

- If the prompt is "M>" with no error messages, all scans worked. We have control as long as we don't try to run code. Continue with "If you can get to the "M>" prompt.
- If an error message is displayed, verify that $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ are being driven.
- If the prompt is "c>", there was bad scanning of the data scan chain. Check processor mask revision.
- If the prompt is "U>", the processor failed to stop soft or hard. Check reset lines, mask revision, processor type and firmware version.

If the prompt after rst is "U>"

The HRESET and SRESET lines are working. Continue with more tests:

4 Enter the break command:

```
U>b
M>
```

If the prompt after b is "M>" with error messages

If you see: "!ERROR 145! Unable to soft stop - freezing the processor clocks" the processor is hard stopped. Check the mask revision, processor type, and firmware version. If all of these look good, then the target may not be terminating cycles (pending external bus cycles). Successive run ("r") and step ("s") commands will fail. The processor may have fetched an invalid instruction.

Check the value of the PC (IAR):

```
M>reg PC
  reg PC=xxxxxxxx
M>
```

If the value is fff00100, the processor had a problem accessing the boot ROM and crashed during boot.

Processor and/or board level reset is required to recover from "freezing" processor clocks" -- register and memory commands should still work.

If the prompt after b is "M>" with no error messages

Everything is still working correctly. Continue with more tests:

If you can get to the "M>" prompt

5 At the "M>" prompt , check register and memory access:

```
M>reg GPR0
  reg GPR0=xxxxxxxx
M>reg GPR0=12345678
M>reg GPR0
   reg GPR0=12345678
M>
```

If the returned value is equal to the written value, then the dd level of the chip is probably correct.

Now enter:

```
M>m -d4 -a4 0=11111111,22222222,33333333,44444444
M>m -d4 -a4 0..
  00000000   11111111 22222222 33333333 44444444
  00000010   00000000 00000000 00000000 00000000
  00000020   00000000 00000000 00000000 00000000
  00000030   00000000 00000000 00000000 00000000
  00000040   00000000 00000000 00000000 00000000
  00000050   00000000 00000000 00000000 00000000
  00000060   00000000 00000000 00000000 00000000
  00000070   00000000 00000000 00000000 00000000
M>
```

- Returned value is equal to the written value implies that memory is working.
- Returned value is not equal to the written value implies that memory control may not be initialized. Try to initialize by:

  ```
  M>cf reset=runrom;rst;w 5
    #waiting for 5 seconds...
  U>b
  M>
  ```

  Repeat above memory test.

- If every other word is wrong, set 32 bit mode:

  ```
  M>cf 32bitmode=on
  M>
  ```

  Repeat above memory test.

If you see memory-related problems

1 Set caches and translation off:

```
M>reg HID0=0
M>reg MSR=0
M>
```

If these commands fail, just try again.

2 Now enter:

```
M>m -d4 -a4 0=11111111,22222222,33333333,44444444
M>m -d4 -a4 0..
  00000000   11111111 02222222 33333333 44444444
  00000010   00000000 00000000 00000000 00000000
  00000020   00000000 00000000 00000000 00000000
  00000030   00000000 00000000 00000000 00000000
  00000040   00000000 00000000 00000000 00000000
  00000050   00000000 00000000 00000000 00000000
  00000060   00000000 00000000 00000000 00000000
  00000070   00000000 00000000 00000000 00000000
M>
```

- If you do not see correct values written in memory, try increasing memory delay (page 139).
- If the read value is not equal to the written value, the memory controller may not be set up correctly.
- If the read value is equal to the written value, but you still suspect memory problems, the emulator firmware might not be working with cache.

3 Enter:

```
M>cf reset=rom
M>rst
M>m -d4 -a4 0..
```

- Read value not equal to the written value implies that reset is tied to memory controller. Check $\overline{\text{HRESET}}$ and $\overline{\text{SRESET}}$ for correct connections.

4 If you have memory problems running Windows NT, you may have this problem:

- System normally runs in little endian mode
- "rst" returns processor to big endian, memory controller on target still little endian, so memory access doesn't work.

5 Hand load a little program:

```
M>m -d4 -a4 100=38210001,60000000,60000000,4bfffff4
M>reg GPR1=0
M>
```

This means: Add 1, GPR1, NOP, NOP, JMP .-4

Set the PC to this program:

```
M>reg PC=100
M>
```

Step, then check the register:

```
M>s
  PC=00000104
M>reg GPR1
  reg GPR1=00000001
M>
```

This should return"reg GPR1=00000001" .

Step some more and verify that GPR1 increments after every four steps:

```
M>s 4
  PC=00000104
M>reg GPR1
  reg GPR1=00000002
M>
```

If running from reset causes problems

Running from reset may cause some problems once background is entered. To ensure proper operation, the DER register must have bits 31,30,29,28 set (0x0000000f), and the SYPCR register must have the 'Disable watchdog freeze' bit set (0x00000080).

If you see the "!ASYNC_STAT 173!" error message

If after a break, the following error arises:

```
!ASYNC_STAT  173! MSR.RI bit not set - Break may not be
recoverable
```

This indicates that the MSR.RI bit is not set, implying that a non-maskable break was needed, and the interrupt may not be recoverable. If this occurs while breaking out of regular code, then the MSR.RI bit was not set in the boot code. This can be fixed by 'ORing' in 0x00000002 into the SRR1 register and resuming the run.

To test the target system

The following program can be placed into memory.

```
start:  addi  r1,1     - 0x38210001
nop             - 0x60000000
nop             - 0x60000000
bra   start     - 0x4bffffff4
```

The opcode 0x4bffffff4 is a branch to a relative offset, so this program can be placed at any start address.

```
M>reg r1=0
M>m -a2 -d2 10000=3821,1,6000,0,6000,0,4bff,fff4
M>r 10000
U>reg r1
reg r1=00034567    # or some number
U>reg r1
reg r1=00102333    # or some number
U>
```

This program will loop forever, incrementing `r1`. This is a good test program to load once a memory system is up to make sure the microprocessor can run code out of memory.

Problems with the LAN Interface

If LAN communication does not work

If you cannot verify the connection , or if the commands are not accepted by the emulation module:

- ☐ Make sure that you wait for the power-on self test to complete before connecting.
- ☐ Make sure that the LAN cable is connected. Watch the LAN LED's on the back of the logic analysis system to see whether the system is seeing LAN activity. Refer to your LAN documentation for testing connectivity.
- ☐ Check that the host computer or debugger was configured with the correct LAN address. If the logic analysis system is on a different subnet than the host computer, check that the gateway address is correct.
- ☐ Make sure that the logic analysis system's IP address is set up correctly.

If it takes a long time to connect to the network

☐ **Check the subnet masks on the other LAN devices connected to your network. All of the devices should be configured to use the same subnet mask.**

Subnet mask error messages do not indicate a major problem. You can continue using the emulation module.

The subnet mask is set in the logic analysis system's System Admin window. If it then detects other subnet masks, it will generate error messages.

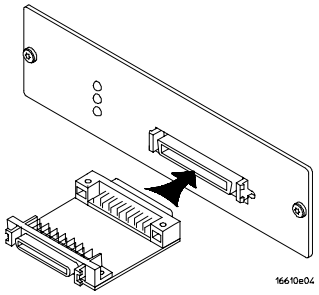
If there are many subnet masks in use on the local subnet, the logic analysis system may take a very long time to connect to the network after it is turned on.

Problems with the Emulation Module

Occasionally you may suspect a hardware problem with the emulation module or target interface module. The procedures in this section describe how to test the hardware, and if a problem is found, how to repair or replace the broken component.

To run the built-in performance verification test using the logic analysis system

1. End any Emulation Control Interface or debugger sessions.
2. Disconnect the 50-pin cable from the emulation module, and plug the loopback test board (HP part number E3496-66502) into the emulation module.



3. In the system window, click the emulation module and select **Performance Verification**.
4. Click **Start PV**.

   The results will appear onscreen.

To run complete performance verification tests using a telnet connection

1. Disconnect the 50-pin cable from the emulation module, and plug the loopback test board (HP part number E3496-66502) directly into the emulation module. Do not plug anything into the other end of the loopback test board.

   On a good system, the RESET LED will light and the BKG and USER LEDs will be out.
2. telnet to the emulation module.
3. Enter the **pv 1** command.

See Also

Options available for the "pv" command are explained in the help screen displayed by typing "help pv" or "? pv" at the prompt. Note, however, that some of the options listed may not apply to your emulation module.

Examples:

If you are using a UNIX system, to telnet to a logic analysis system named "mylogic", enter:

```
telnet mylogic 6472
```

Here are some examples of ways to use the **pv** command.

To execute both tests one time:

```
pv 1
```

To execute test 2 with maximum debug output repeatedly until a ^C is entered:

```
pv -t2 -v9 0
```

To execute tests 3, 4, and 5 only for 2 cycles:

```
pv -t3-5 2
```

The results on a good system with the loopback test board connected, are as follows:

```
M>pv 1

  Testing: HPE3499C Series Emulation System
    Test  1: Powerup PV Results                              Passed!
    Test  2: Target Probe Feedback Test                      Passed!
    Test  3: Boundary Scan Master Test                       Passed!
    Test  4: I2C Test                                        Passed!
    Test  5: Data Lines Test                                 Passed!
  PASSED  Number of tests: 1           Number of failures: 0

         Copyright (c) Hewlett-Packard Co. 1987
All Rights Reserved.  Reproduction, adaptation, or translation without prior
written permission is prohibited, except as allowed under copyright laws.

  HPE3499C Series Emulation System
    Version:   A.07.54 22Apr98
    Location:  Generics

  HPE3498A PowerPC 604 (Rev3) JTAG Emulator
    Version:   A.02.04 14Apr98
M>
```

You may get an error like "!ERROR 172! Bad status code (0xff) from the hard reset sequence" just before the prompt. This is because the selftest loopback connector is installed instead of being connected to a real PowerPC target system. You may also get a "?>" prompt for the same reason, and this is normal and expected. Any errors after the "PASSED Number of tests: 1 Number of failures: 0" line can be ignored.

If a performance verification test fails

There are some things you can do if a failure is found on one of these tests. Details of the failure can be obtained through using a -v option ("verbose" level) of 2 or more.

> If the particular failure you see is not listed below, contact HP for assistance.

TEST 3: Boundary Scan Master Test
TEST 4: I2C Test

If these tests are not executed, check that you have connected the loopback test board.

If these tests fail, return the emulation module to HP for replacement.

Returning Parts to Hewlett-Packard for Service

The repair strategy for this emulation solution is board replacement.

Exchange assemblies are available when a repairable assembly is returned to Hewlett-Packard. These assemblies have been set up on the "Exchange Assembly" program. This lets you exchange a faulty assembly with one that has been repaired, calibrated, and performance verified by the factory. The cost is significantly less than that of a new assembly.

To return a part to Hewlett-Packard

1. **Follow the procedures in this chapter to make sure that the problem is caused by a hardware failure, not by configuration or cabling problems.**
2. **In the U.S., call 1-800-403-0801. Outside the U.S., call your nearest HP sales office. Ask them for the address of the nearest HP service center.**
3. **Package the part and send it to the HP service center.**

   Keep any parts which you know are working. For example, if only the target interface module is broken, keep the emulation module and cables.
4. **When the part has been replaced, it will be sent back to you.**

   The unit returned to you will have the same serial number as the unit you sent to HP.

The HP service center can also troubleshoot the hardware and replace the failed part. To do this, send your entire measurement system to the service center, including the logic analysis system, target interface module, and cables.

In some parts of the world, on-site repair service is available. Ask an HP sales or service representative for details.

To obtain replacement parts

The following table lists some parts that may be replaced if they are damaged or lost. Contact your nearest Hewlett-Packard Sales Office for further information.

Part numbers

Exchange Assemblies

Part Number	Description
E5901A #060	Emulation module

Replacement Assemblies

Part number	Description
16700-61608	Expansion cable
E3494-61604	16-pin cable
E3496-61601	50-pin cable
E3496-66502	emulation module loopback test board
E3481-61601	20-pin cable
E3452-66502	Target interface module (PPC JTAG board)

Cleaning the Instrument

If the instrument requires cleaning:

1. Remove power from the instrument.
2. Clean the instrument with a mild detergent and water.
3. Make sure that the instrument is completely dry before reconnecting it to a power source.

Glossary

Analysis Probe A probing solution connected to the target microprocessor. It provides an interface between the signals of the target microprocessor and the inputs of the logic analyzer. Formerly called a "preprocessor."

Elastomeric Probe Adapter A connector that is fastened on top of a target microprocessor using a retainer and knurled nut. The conductive elastomer on the bottom of the probe adapter makes contact with pins of the target microprocessor and delivers their signals to connection points on top of the probe adapter.

Emulation Module An emulation module is installed within the mainframe of a logic analyzer. It provides run control within an emulation and analysis test setup. See Emulation Probe.

Emulation Migration By loading new firmware and connecting a different TIM, your emulator migrates from support of one PowerPC model to support of another PowerPC model.

Emulation Probe An emulation probe is a standalone instrument connected via LAN to the mainframe of a logic analyzer or to a host computer. It provides run control within an emulation and analysis test setup. Formerly called a "processor probe" or "software probe." See Emulation Module.

Emulator As used in this manual, the term Emulator applies equally to both the Emulation Module and the Emulation Probe.

Extender A part whose only function is to provide connections from one location to another. One or more extenders might be stacked to raise a probe above a target micoprocessor to avoid mechanical contact with other components installed close to the target microprocessor. Sometimes called a "connector board."

Flexible Adapter Two connection devices coupled with a flexible cable. Used for connecting probing hardware on the target microprocessor to the analysis probe.

General-Purpose Flexible Adapter A cable assembly that connects the signals from an elastomeric probe adapter to an analysis probe. Normally, a male-to-male header or transition board makes the connections from the general-purpose flexible adapter to the analysis probe.

High-Density Adapter Cable A cable assembly that delivers signals from an analysis probe hardware interface to the logic analyzer pod cables. A high-density adapter cable has a single Mictor connector that is installed into the analysis probe, and two cables that are connected to corresponding odd and even logic analyzer pod cables.

High-Density Termination Adapter Cable Same as a High-Density Adapter Cable, except it has a termination in the Mictor connector.

Jumper Moveable direct electrical connection between two points.

Mainframe Logic Analyzer A logic analyzer that resides on one or more board assemblies installed in an HP 16500 or HP 16600A/700A-series mainframe.

Male-to-male Header A board assembly that makes point-to-point connections between the female pins of a flexible adapter or transition board and the female pins of an analysis probe.

Preprocessor See Analysis Probe.

Preprocessor Interface See Analysis Probe.

Probe adapter See Elastomeric Probe Adapter.

Processor Probe See Emulation Probe.

Prototype Analyzer The HP 16505A prototype analyzer acts as an analysis and display processor for the HP 16500B/C logic analysis system. It provides a windowed in-

terface and powerful analysis capabilities. Replaced by HP 16600A/700A-series logic analysis systems.

Run Control Probe See Emulation Probe and Emulation Module.

Setup Assistant A software program that guides a user through the process of connecting and configuring a logic analyzer to make measurements on a specific microprocessor.

Shunt Connector. See Jumper.

Software Probe See Emulation Probe.

Solution HP's term for a set of tools for debugging your target system. A solution includes probing, inverse assembly, the HP B4620B Source Correlation Tool Set, and an emulation module.

Stand-alone Logic Analyzer A standalone logic analyzer has a predefined set of hardware components which provide a specific set of capabilities. It is designed to perform logic analysis. A standalone logic analyzer differs from a mainframe logic analyzer in that it does not offer card slots for installation of additional capabilities, and its specifications are not modified based upon selection from a set of optional hardware boards that might be installed within its frame.

Target Control Port An 8-bit, TTL port on a logic analysis system that you can use to send signals to your target system. It does not function like a pattern generator or emulation module, but more like a remote control for the target's switches.

Target Interface Module A small circuit board which connects the 50-pin cable from an emulation module or emulation probe to signals from the debug port on a target system.

TIM See Target Interface Module.

Trigger Specification A set of conditions that must be true before the instrument triggers. See the printed or online documentation for your logic analyzer for details.

Transition Board A board assembly that obtains signals connected to one side and rearranges them in a different order for delivery at the other side of the board.

1/4-Flexible Adapter An adapter that obtains one-quarter of the signals from an elastomeric probe adapter (one side of a target microprocessor) and makes them available for probing.

A
ADDR label, modifying, 64
addresses
  branch target, 78
  mask, 80
  offset, 106
  PC label, 103
  See also software addresses
  triggering on a specific address, 71
  type, 66
  wait states, 76
aliases, PowerPC, 4
analysis probe
  additional equipment supported, 24
  connecting to, 129
  connection illustration, 45
  connection overview, 43
  definition, 253
  dimensions, 40
  equipment required, 24
  equipment supplied, 22
  inverse assembly, 73
  microprocessors supported, 187
  modes of operation, 63
  operating characteristics, 187
  overview, 2
  pod locations, 47
  power on/power off sequence, 41
  processors supported, 4
  product numbers, 4
  storage qualification, 70
  target system requirements, 39–40
analysis probe problems, 213
  erratic trace measurements, 214
  target system will not boot, 213
analyzer problems, 211
  capacitive loading, 214
  intermittent data errors, 211
  unwanted triggers, 211
analyzing the processor, 61
ASCII format (GPA), 200
assistant
  See setup assistant
AT status bit, 66

B
b prefix, 77
BB status bit, 66
BDM port
  See debug port
BGA socket, 39–40, 42–45
big endian mode, 79
bits
  labels, 64
  LSB and MSB, 64
  STAT, 66
BKG light, 228
branch instructions, 78
branches, displaying, 80
breakpoints
  tracing until, 182
built-in commands
  configuration, 135
  list of commands, 229–231

C
cables
  emulator, 128
  replacing, 223, 250
caches
  disabling, 73
  enabling and disabling, 73–74, 156
  trace problems and, 216
cards
  See logic analyzers
carrier, BGA, 42–45
CD-ROM, installing software from, 34
cf commands, 135
characteristics
  analysis probe, 187
  emulation module, 198
checklist, setup, 19–20
circuit board, dimensions, 40
cleaning, 223, 251
clocks
  logic analyzer, 63
  maximum speed, 187
  qualification, 70
  qualified, and emulator, 175
  slow, 181–182, 220
colors, 80
comments, in GPA files, 208
compilers, 97
configuration
  checklist, 19–20
  logic analyzers, 55–59, 64–72
configuration file floppy disk, 57
configuration files
  installing, 31, 33–35
  loading, 55–59
  names of, 59
configuration, emulation module
  overview, 132–144
  using debugger, 137
connection
  analysis probe, 37, 39–60
  emulation module, 109–146
  host workstation, 147, 149–167
  PGA target systems, 44
  problems, LAN, 243–244
  setup checklist, 19–20
connection illustration, 45
connector
  10BASE-T, 152
  10BASE2, 152
  debug port, 119
  JTAG, 114
  JTAG, levels, 234
connector board, 254
creating GPA symbol files, 200

D
"d" number prefix, 77
DATA label, modifying, 64
debug port
  connecting to, 128
debuggers
  benefits of, 148
  configuration, 137
  MRI, 157–162
  SDS, 163–167
  setting up, 151–156
  See also under debugger name
  writing, 231
delays, configuring, 139–140
development port
  See debug port
Diab Data
  compiler, 98
dimensions of analysis probe, 40
directories
  configuration files, 56
  software installation, 34
  source code, 104
display filtering, 80
displaying on PC, 154–155
dmwrop, configuring, 142
driver firmware error, 235

E
elastomeric probe adapter
  definition, 253
Emulation Control Interface
  configuration, 134
  debugger conflict, 151
  introduction, 111–113
  when to use, 170
emulation module
  configuration, 137
  connecting, 110, 126–129
  definition, 253
  description of, 3
  HP 16600 installation, 124
  HP 16700A installation, 122
  port number, 153
  product numbers, 4
  target system design, 114–120
emulation probe
  definition, 253
emulation solution
  See solution
endian mode, 79
enhanced inverse assembler
  logic analyzer requirements, 26
environmental characteristics
  emulation module, 197
equipment required
  analysis probe, 24
  emulation module, 28

equipment supplied, 22–26
  analysis probe, 22
  emulation module, 27–28
  ordering information, 4
  overview, 4
error messages
  inverse assembler, 218–221
examples, measurement, 171
exporting a display, 154–155
extended mnemonics, 87
extender, 254

F
files
  loading vs. installing, 32
  workstation setup, 151–156
filtering, display, 80
firmware
  emulation module, 131
  updating, 130–131
  version, 131
flash ROM
  updating emulator, 130–131
flexible adapter
  definition, 254
floppy disks
  duplicating, 58
  installing software from, 57
flowchart, setup, 19–20
format menu, 64
full solution, 3
FUNCTIONS in GPA format, 205

G
General-Purpose ASCII format, 200
  address format, 200
  comments, 208
  FUNCTIONS, 205
  record format summary, 202
  record headers, 200
  SECTIONS, 204
  simple form, 200
  SOURCE LINES, 207
  START ADDRESS, 208
  VARIABLES, 206
general-purpose flexible adapter
  definition, 254
Green Hills compiler, 98

H
high-density adapter cable
  definition, 254
high-density termination adapter
  definition, 254
host computer
  connecting to, 147, 149–167
HRESET signal, 114–120

I
IEEE 802.3, 152
imwrop, configuring, 143
information sources, 30
init command, 235
installation, software, 31, 33–35
instruction cache
  See cache
instruction decodings, 87
intermodule measurement
  creating, 174
intermodule measurement problems, 217
  an event wasn't captured, 217
  analyzer doesn't stop, 175
inverse assembler
  configuration file names, 59
  loading files, 56, 58
  output format, 77
  requirements for, 73–87, 89
  requirements for enhanced, 26
inverse assembler floppy disk, 57
inverse assembler problems, 215
  failure to load or run, 216
  incorrect inverse assembly, 215
  no inverse assembly, 215
inverse assembly, 73
  displays, 171
  pods required, 25
IP address, 152

J
JTAG port
  connections, 119
jumper, definition, 254

L
LAN
  emulation module, 152
  problems, 243–244
lights
  See status lights
listing
  incorrect, 215
Listing menu, 75
listing windows, 171
little endian mode, 79
little-endian ordering, 115, 118
Load menu, 80
loading configurations
  logic analyzer, 55–59
loading configurations, vs.
  installing, 31, 33–35
logic analyzers
  clocking, 70
  configuration, 64–72
  configuring, 56, 58
  HP 16550A two-card connections, 50
  HP 16554/55/56 three-card
connections, 52
  HP 16554/55/56 two-card connections, 51
  HP 1660 series connections, 53
  HP 16600A and HP 16700A-series, 21
  HP 16600A connections, 48
  HP 16601A connections, 49
  HP 1670 series connections, 54
  loading configuration files, 55–59
  software version requirements, 26
  storage qualification, 70
  supported, 25
LSB, 64, 75

M
mainframe logic analyzer
  definition, 254
male-to-male header
  definition, 254
mask, subnet, 244
measurement examples, 171
memory
  configuring delays, 139–140
  configuring parity, 140
  configuring read, 141
  configuring write, 143

memory (continued)
confituring write, 142
testing, 239
memory banks, 80
microprocessor aliases, 4
microprocessors supported, 4
Microtec Research Inc.
compiler, 99
debugger, 157–162
MMU support, 115
mnemonics, 87
modes
operating, 63
monitor, 137
mrdop, configuring, 141
MRI debugger, 157–162
MSB, 64, 75

O
object module file symbols, 94–99
offset, address, 106
online configuration help, 21
operating characteristics
analysis probe, 187
emulation module, 197
Options menu, 80, 85
orientation, socket, 42–45

P
parity, configuring, 140
parts supplied, 22–26
path, source file, 104
PC (personal computer)
connecting to, 147, 149–167
PC label, 79, 103
performance verification test, 245
PGA target systems, 44
pin A1 position, 44
pod locations, 47
port number, emulation module, 153, 229
power on/power off sequence, 41
PowerPC aliases, 4
prefix ?, 77
preprocessor
See analysis probe
problems
analysis probe, 209, 211–223
emulation module, 225–251
processor support package, 34
processors supported, 4
program symbols, 94–99
prompts, 231
list of, 231
troubleshooting, 235
prototype analyzer
definition, 254
PV
See performance verification

Q
? prefix and suffix, 77

R
R/W status bit, 66
real-time runs, configuring, 137
record format, General-Purpose ASCII, 202
record headers, 200
references, 30
register commands, 232
registers
listing format, 77
repair
analysis probe, 222
emulation module, 249–250
requirements
target system, 39–40, 114–120
reset
configuring, 139
effects of, 241
light, 228
troubleshooting, 236
RESET signals, 114
run control tool
See emulation control interface

S
SDS debugger, 163–167
Section Format, 200
SECTIONS in GPA format, 204
service, how to obtain, 222, 249
setup
See configuration
setup assistant, 21
definition, 255
setup checklist, 19–20
signals
debug port, 119
logic analyzer, 190–196
signals, expected levels, 234
SingleStep debugger, 163–167
skid, reducing, 175
slow clock, 227
slow clock message, 181–182
socket, BGA, 39–40, 42–45
software
installing, 31, 33–35
list of installed, 35
requirements, 26
software addresses, 76, 103
software analyzer, 24
software probe
See emulation module
See emulation probe
software requirements, 26
software,emulation module firmware, 131
solution
at a glance, 2
definition, 255
description of, 2
equipment required, 29
product numbers, 4
source code, 91, 93–107
displays, 171
source correlation
data display, 76
in analyzer, 91, 93–107
using, 100–104
source file search path, 104
SOURCE LINES in GPA format, 207
specifications
See characteristics
$\overline{\text{SRESET}}$ signal, 114–120
START ADDRESS in GPA format, 208
STAT
encoding, 66
label, 66
modifying, 64
state-per-ack, mode of operation, 63
state-per-transfer, mode of operation, 63
status bits, 66
status encoding, 66
status lights, 228
storage qualification, 70

STS status bit, 66
subnet mask, 244
SW_ADDR
  See software addresses
suffix ?, 77
symbol files
  creating GPA, 200
symbols
  in analyzer, 91, 93–107
  object file, 94–99
  object module file, 94–99
  predefined, 68, 93
  program, 94–99
  user-defined, 93
synchronous mode, 63

T
TA status bit, 66
target control port, 255
target interface module (TIM)
  connecting, 128
  definition, 255
target system
  boot failure, 213
  connecting to, 110
  PGA connections, 44
  power sequence, 41
  problems with, 232–242
  requirements for analysis probe, 39–40
  requirements for emulation, 114–120
TEA status bit, 66
telnet, 153, 229
terminal interface, 153
  See also built-in commands
tests, emulation module, 245–248
timing, mode of operation, 63
TLBs, 115, 118
trace
  erratic, 214
  missing display, 212
transition board
  definition, 255
trigger
  emulation module, 173–175
  on break, 177–183
  source code, 105–107
  specification, 70
  unwanted, 211
Trigger menu, 70
troubleshooting, 227
  analysis probe, 209–223
  emulation module, 225–251
TS status bit, 66
TSIZ status bit, 66

U
Undefined Opcode, 77
USER light, 228

V
VARIABLES in GPA format, 206
versions
  emulation module firmware, 131
  logic analyzer software, 26
voltage
  emulation module, 198

W
wait states, 76
web sites
  HP logic analyzers, 30
  See also under debugger names
wizard
  See setup assistant

X
X windows, 154–155
XRAY debugger, 157–162

DECLARATION OF CONFORMITY

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company

Manufacturer's Address: Colorado Springs Division
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

declares that the product

Product Name: Logic Analyzer

Model Number(s): HP 16600A, HP 16601A, HP 16602A, HP 16603A

Product Option(s): All

conforms to the following Product Specifications:

Safety:	IEC 1010-1:1990+A1 / EN 61010-1:1993	
	UL 3111	
	CSA-C22.2 No. 1010.1:1993	
EMC:	CISPR 11:1990 / EN 55011:1991	Group 1, Class A
	IEC 555-2:1982 + A1:1985 / EN 60555-2:1987	
	IEC 555-3:1982 + A1:1990 / EN 60555-3:1987 + A1:1991	
	IEC 801-2:1991 / EN 50082-1:1992	4 kV CD, 8 kV AD
	IEC 801-3:1984 / EN 50082-1:1992	3 V/m, {1kHz 80% AM, 27-1000 MHz}
	IEC 801-4:1988 / EN 50082-1:1992	0.5 kV Sig. Lines, 1 kV Power Lines

Supplementary Information:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC and carries the CE marking accordingly.

This product was tested in a typical configuration with Hewlett-Packard test systems.

Colorado Springs, 08/18/97

John Strathman, Quality Manager

European Contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department ZQ / Standards Europe, Herrenberger Strasse 130, D-71034 Böblingen Germany (FAX: +49-7031-14-3143)

Product Regulations

Safety

IEC 1010-1:1990+A1 / EN 61010-1:1993
UL 3111
CSA-C22.2 No.1010.1:1993

EMC

This Product meets the requirement of the European Communities (EC) EMC Directive 89/336/EEC.

CE
ISM 1-A

Emissions EN55011/CISPR 11 (ISM, Group 1, Class A equipment), IEC 555-2 and IEC 555-3

N279

Immunity

EN50082-1	Code[1]	Notes[2]
IEC 801-2 (ESD) 4kV CD, 8kV AD	3	
IEC 801-3 (Rad.) 3 V/m	1	
IEC 801-4 (EFT) 0.5 kV, 1kV	3	

[1] Performance Codes:
1 PASS - Normal operation, no effect.
2 PASS - Temporary degradation, self recoverable.
3 PASS - Temporary degradation, operator intervention required.
4 FAIL - Not recoverable, component damage.

[2] Notes: (none)

Sound Pressure Level <60 dBA

DECLARATION OF CONFORMITY

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company

Manufacturer's Address: Colorado Springs Division
1900 Garden of the Gods Road
Colorado Springs, CO 80907 USA

declares that the product

Product Name: Logic Analyzer Mainframe

Model Number(s): HP 16700A

Product Option(s): All

conforms to the following Product Specifications:

Safety:	IEC 1010-1:1990+A1 / EN 61010-1:1993	
	UL 3111	
	CSA-C22.2 No. 1010.1:1993	
EMC:	CISPR 11:1990 / EN 55011:1991	Group 1 Class A
	IEC 555-2:1982 + A1:1985 / EN 60555-2:1987	
	IEC 555-3:1982 + A1:1990 / EN 60555-3:1987 + A1:1991	
	IEC 801-2:1991 / EN 50082-1:1992	4 kV CD, 8 kV AD
	IEC 801-3:1984 / EN 50082-1:1992	3 V/m, {1kHz 80% AM, 27-1000 MHz}
	IEC 801-4:1988 / EN 50082-1:1992	0.5 kV Sig. Lines, 1 kV Power Lines

Supplementary Information:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC and carries the CE marking accordingly.

This product was tested in a typical configuration with Hewlett-Packard test systems.

Colorado Springs, 09/22/97

John Strathman, Quality Manager

European Contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department ZQ / Standards Europe, Herrenberger Strasse 130, D-71034 Böblingen Germany (FAX: +49-7031-14-3143)

Product Regulations

Safety

IEC 1010-1:1990+A1 / EN 61010-1:1993
UL 3111
CSA-C22.2 No.1010.1:1993

EMC

CE
ISM 1-A

This Product meets the requirement of the European Communities (EC) EMC Directive 89/336/EEC.

Emissions EN55011/CISPR 11 (ISM, Group 1, Class A equipment), IEC 555-2 and IEC 555-3

N279

Immunity

EN50082-1	Code[1]	Notes[2]
IEC 801-2 (ESD) 4kV CD, 8kV AD	3	
IEC 801-3 (Rad.) 3 V/m	1	
IEC 801-4 (EFT) 0.5 kV, 1kV	1	

[1] Performance Codes:
1 PASS - Normal operation, no effect.
2 PASS - Temporary degradation, self recoverable.
3 PASS - Temporary degradation, operator intervention required.
4 FAIL - Not recoverable, component damage.

[2] Notes: (none)

Sound Pressure Level Less than 60 dBA

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraph (C) (1) (ii) of the Rights in Technical Data and Computer Software Clause in DFARS 252.227-7013.
Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304 U.S.A.
Rights for non-DOD U.S. Government Departments and Agencies are set forth in FAR 52.227-19(c)(1,2).

Document Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety

This apparatus has been designed and tested in accordance with IEC Publication 1010, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

Warning

- Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.
- Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock or fire hazard.
- Service instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.
- If you energize this instrument by an auto transformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.
- Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.
- Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.
- Do not install substitute parts or perform any unauthorized modification to the instrument.
- Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.
- Use caution when exposing or handling the CRT. Handling or replacing the CRT shall be done only by qualified maintenance personnel.

Safety Symbols

Instruction manual symbol: the product is marked with this symbol when it is necessary for you to refer to the instruction manual in order to protect against damage to the product.

Hazardous voltage symbol.

Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.

Hewlett-Packard
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901

Product Warranty

This Hewlett-Packard product has a warranty against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard.

For products returned to Hewlett-Packard for warranty service, the Buyer shall prepay shipping charges to Hewlett-Packard and Hewlett-Packard shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Hewlett-Packard from another country.

Hewlett-Packard warrants that its software and firmware designated by Hewlett-Packard for use with an instrument will execute its programming instructions when properly installed on that instrument. Hewlett-Packard does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. Hewlett-Packard specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are the buyer's sole and exclusive remedies. Hewlett-Packard shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales Office.

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

About this edition

This is the *Solutions for the PowerPC 604 User's Guide*.

Publication number E2465-97003, December 1998
Printed in USA.

The information in this manual previously appeared in:

E2465-97002, May 1998

E2465-97001, October 1996

E3452-97001, August 1997

New editions are complete revisions of the manual. Many product updates do not require manual changes; and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

Comments welcome!

Send your comments or suggestions regarding this manual to: documentation@col.hp.com