User's Guide

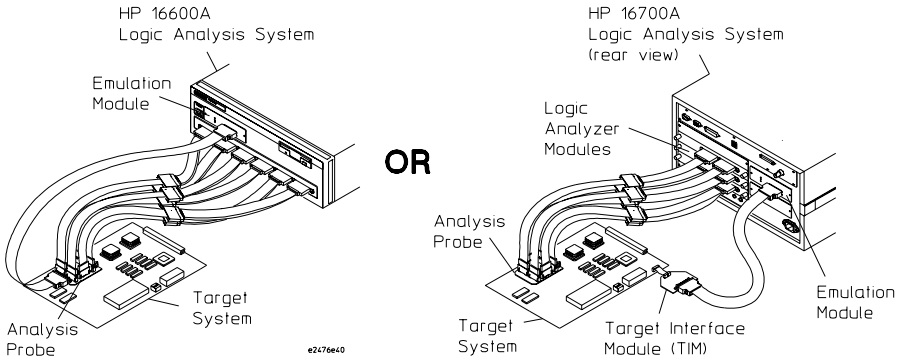
Publication number E2490-97003
May 1999

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

Solutions for the PowerPC MPC505/509

The HP E9485A emulation solution lets you use the HP 16600/16700A-series logic analysis system to debug and characterize Motorola Embedded PowerPC MPC505/509 target systems.

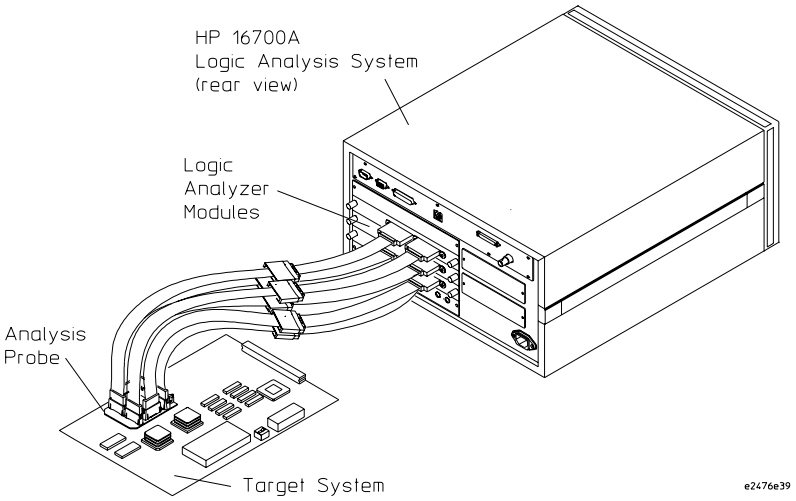
The emulation solution is a bundled product consisting of an analysis probe, an emulation module (and its cables and adapters), and the HP B4620B source correlation tool set.



Analysis Probe (and Trace Reconstruction Tool)

The analysis probe (HP E9585A when ordered separately) probes the MPC505/509 microprocessor so that you can capture and display the processor's signal values with a logic analyzer.

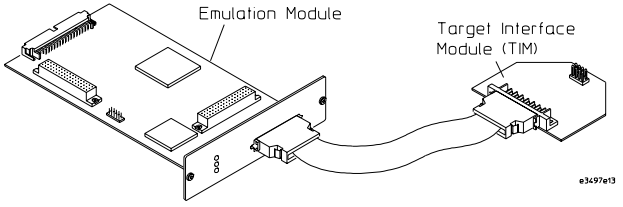
The inverse assembler includes a trace reconstruction tool that uses captured show cycles, captured VF/VFLS signals, and program image files to decode captured MPC505/509 execution into complete program traces. With the trace reconstruction tool, you get an accurate description of instruction execution, even when the instructions execute out of internal memory.



Emulation Module (and Target Interface Module)

The emulation module lets you use a microprocessor's built-in debugging features (like starting/stopping program execution, setting breakpoints, and modifying the contents of processor registers and target system memory).

The target interface module (TIM) adapts the emulation module to the MPC505/509 microprocessor's debug port.



Source Correlation Tool Set

The HP B4620B source correlation tool set lets you set up logic analyzer triggers based on source code, and it lets you view the source code associated with signal values captured by the logic analyzer.

In This Book

This book describes the following products:

Product	Supports	Includes
HP E9585A analysis probe	MPC505/509	HP E2490A analysis probe and inverse assembler
HP E9485A emulation solution	MPC505/509	HP E2490A analysis probe and inverse assembler, HP 16610A emulation module, target interface module (TIM), and HP B4620B source correlation tool set

Before you use this book, you should have already set up the HP 16600A/16700A-series logic analysis system, installed logic analyzer modules, and learned how to use the logic analysis system (see the logic analysis system's *Installation Guide*).

This book has five parts:

- Part1, "Installation Guide," describes supplied and required equipment, target system design considerations, setting up the logic analysis system, and probing the target system.
- Part 2, "Using the Logic Analyzer," describes analysis probe and logic analyzer configuration, how to set up triggers, how to interpret the captured data when it's displayed, and to troubleshoot analysis probe problems.
- Part 3, "Using the Emulation Module," describes how to use the Emulation Control Interface, how to configure the emulation module, how to use debuggers, how to coordinate emulation control and logic analysis, and how to troubleshoot emulation module problems.
- Part 4, "Reference," describes specifications and characteristics, the general-purpose ASCII symbol file format, and how to use the analysis probe with other logic analyzers.
- Part 5, "Service Guide," describes how to return parts for service, how to get replacement parts, and how to clean the instrument.

See Also

The HP 16600A/16700A-series logic analysis system's on-line help for more information on using the HP B4620B source correlation tool set.

Contents

HP Solutions for the MPC505/509—At a Glance

Analysis Probe (and Trace Reconstruction Tool) 2
Emulation Module (and Target Interface Module) 3
Source Correlation Tool Set 3

In This Book

Part 1 Installation Guide

Overview of Installation and Setup 16

1 Equipment and Requirements

Equipment and Software Supplied 20
Analysis Probe 20
Emulation Module 21
Source Correlation Tool Set 22

Additional Equipment and Software Required 23
Analysis Probe 23
Emulation Module 25
Source Correlation Tool Set 25

Other Optional Equipment and Software 26

Unsupported Microprocessor Modes 27

2 Preparing the Target System

Preparing for the Analysis Probe 30
Clearance 30
Keep-Out Area 30
Boot Code for Inverse Assembly 30

Preparing for the Emulation Module 32
Target System Requirements 32
Debug Port Connection (Optional) 32

3 Setting Up the Logic Analysis System

Power-ON/Power-OFF Sequence 36
To power-ON the HP 16600A/16700A-series logic analysis systems 36
To power-OFF 36

Installing Logic Analyzer Modules 37

Installing the Emulation Module 38
To install in a HP 16700A-series logic analysis system 38
To install in a HP 16600A-series logic analysis system 41
To test the emulation module 43

Installing Software 44
To install software from CD-ROM 45

4 Probing the Target System

Using the Setup Assistant 48

Connecting the Analysis Probe to the Target System 49
To connect the probe adapter to the target system 49
To connect the analysis probe to the flexible cable 53

Contents

Connecting the Analysis Probe to a Logic Analyzer 54
To connect the high-density termination cables to the analysis probe 55
To connect to the HP 16550A logic analyzer (one-card) 56
To connect to the HP 16550A logic analyzer (two-card) 59
To connect to the HP 16554/55/56/57 logic analyzer (two or more cards) 61
To connect to the HP 16600A logic analyzer 64
To connect to the HP 16601A logic analyzer 67
To connect to the HP 16602A logic analyzer 70
To connect to the HP 16710/11/12A logic analyzer (one-card) 73
To connect to the HP 16710/11/12A logic analyzer (two-card) 76

Connecting the Emulation Module to the Target System 79
To connect to the analysis probe 79
To connect to a target system debug port 81
To update emulation module firmware 83
To display the emulation module firmware version information 84
To verify communication with the target system 84

Part 2 Using the Logic Analyzer

5 Configuring the Logic Analyzer

Configuring the Analysis Probe 88
To set the analysis probe switches 88
To program for address reconstruction 89

Loading Configuration Files 91
To load configuration files (and the inverse assembler) 91
To set the inverse assembler preferences 94

Loading Symbol Information 95
To view predefined symbols for the MPC505/509 95
To load object file symbols 97

Changing the Analysis Mode 100
To change to state-per-clock analysis 100
To change to state-per-ack analysis 101
To change to timing analysis 102

6 Capturing MPC505/509 Execution

Setting Up Logic Analyzer Triggers 105
To set up logic analyzer triggers 105
To compensate for relocated code 107
To capture valid write cycles in state-per-clock mode 108
To trigger on an address range access in state-per-clock mode 108

Triggering on Source Code 111
To set up triggers based on source code 111
To avoid capturing library code execution 112

7 Displaying Captured MPC505/509 Execution

Displaying Data from the Logic Analyzer 114
To display the captured state data 115
To display captured timing analysis mode data 117

Displaying Data from the Trace Reconstruction Tool 118
To add the trace reconstruction tool to the workspace 119
To load program image files into the trace reconstruction tool 120
To use multiple program image files 121
If undefined opcodes appear in the output 121
If the trace contains all wait states 121

Using the Listing and Source Viewer Displays 122
To display symbols 122
To interpret the inverse assembled data 122
To use the inverse assembler filters 125
To view the source code associated with captured data 128

8 Troubleshooting the Analysis Probe

Solving Logic Analyzer Problems 135
Intermittent data errors 135
Unwanted triggers 135
No activity on activity indicators 136
No trace list display 136
Analyzer won't power up 136

Solving Analysis Probe Problems 137
Target system will not boot up 137
Erratic trace measurements 138
Capacitive loading 138

Solving Inverse Assembler Problems 140
No inverse assembly or incorrect inverse assembly 140
Inverse assembler will not load or run 141

Solving Intermodule Measurement Problems 142
An event wasn't captured by one of the modules 142

Analysis Probe Messages 143
". . . Inverse Assembler Not Found" 143
"Measurement Initialization Error" 143
"No Configuration File Loaded" 145
"Selected File is Incompatible" 145
"Slow or Missing Clock" 145
"Time from Arm Greater Than 41.93 ms" 146
"Waiting for Trigger" 146

Contents

Part 3 Using the Emulation Module

9 Using the Emulation Control Interface

To start from the main System window 151
To start from the Workspace window 152

10 Configuring the Emulation Module

Entering Emulation Module Commands 155
To use the Emulation Control Interface 155
To use the built-in command interface 157
To use a debugger interface 158

Setting the MPC5xx Configuration Options 159
To configure the processor clock speed 160
To configure the “Break In” type 161
To configure the processor type 161
To configure restriction to real-time runs 162

Testing the Emulation Module and Target System 163
To test memory accesses 163
To test by running a program 163

11 Using Debuggers (with the Emulation Module)

Setting Up Debugger Software 168
To connect the logic analysis system to the LAN 169
To change the port number of an emulation module 170
To verify LAN communication with the emulation module 170
To view logic analysis system windows next to the debugger 171

Using the Green Hills Debugger 173
To get started 173
To configure using an initialization script 176
To perform common debugger tasks 177
To send commands to the emulation module 178
To view commands sent by MULTI to the emulation module 178
To reinitialize the system 179
To disconnect from the emulation module 179
Error conditions 179

Using the Microtec Research Debugger 181
To get started 181
To configure the emulation module using an INCLUDE file 184
To perform common debugger tasks 184
To send commands to the emulation module 185
To view commands sent by XRAY 185
To disconnect from the emulation module and target 186
Error conditions 186

Using the Software Development Systems Debugger 188
To get started 189
To send commands to the emulation module 191
Download performance 192
On-chip breakpoints and debugging ROM code 193
Error conditions 193

12 Coordinating Logic Analysis with Processor Execution

Stopping Processor Execution on a Logic Analyzer Trigger 198
To stop on a source line trigger (Source Viewer window) 198
To stop on any trigger (Intermodule window) 200
To minimize the “skid” effect 201
To stop the analyzer and view a measurement 201

Tracing Until the Processor Halts 203
To capture a trace before the processor halts 203

Triggering the Logic Analyzer when Processor Execution Stops 205
To trigger the analyzer when the processor halts 208
To trigger the analyzer when the processor reaches a breakpoint 209

13 Troubleshooting the Emulation Module

Troubleshooting Guide 213

Status Lights 214

Built-In Commands 215
To telnet to the emulation module 215
To use the built-in commands 216

Solving Target System Problems 218
What to check first 218
To interpret the initial prompt 219
If interrupts are non-recoverable 220
If hardware breakpoints have no effect 221
If the target resets itself 221
If running from reset causes problems 222
If you see the "!ASYNC_STAT 173!" error message 222
If there are problems with the debug port signals 223
To test the target system 223

Solving LAN Communication Problems 224
If LAN communication does not work 224
If it takes a long time to connect to the network 224

Contents

Solving Emulation Module Problems 226
To run the performance verification tests using the logic analysis system 226
To run complete performance verification tests using a telnet connection 227
If a performance verification test fails 228

Part 4 Reference

14 Specifications and Characteristics

Analysis Probe Operating Characteristics 234
Theory of Operation 235
Configuration File Label Descriptions 235
Analysis Probe Signal-to-Connector Mapping 238
Emulation Module Operating Characteristics 249
Emulation Module Electrical Characteristics 249

15 General-Purpose ASCII (GPA) Symbol File Format

GPA Record Format Summary 253
SECTIONS 254
FUNCTIONS 255
VARIABLES 256
SOURCE LINES 257
START ADDRESS 257
Comments 258

16 Using the Analysis Probe with Other Logic Analyzers

Logic Analyzer Software Version Requirements 260

Power-ON/Power-OFF Sequence 262
To power-ON other HP logic analyzers 262
To power-OFF 262

Contents

Installing Software 263
To install software on other logic analyzers 263

Connecting the Analysis Probe to the Logic Analyzer 264
To connect to the HP 1660 logic analyzers 265
To connect to the HP 1661 logic analyzers 268
To connect to the HP 1670 logic analyzers 271
To connect to the HP 1671 logic analyzers 274

Configuring the Logic Analyzer 277
To load configuration files 277
To view predefined symbols for the MPC5xx 279

Part 5 Service Guide

To return a part to Hewlett-Packard for service 282
To get replacement parts 282
To clean the instrument 284

Glossary

Index

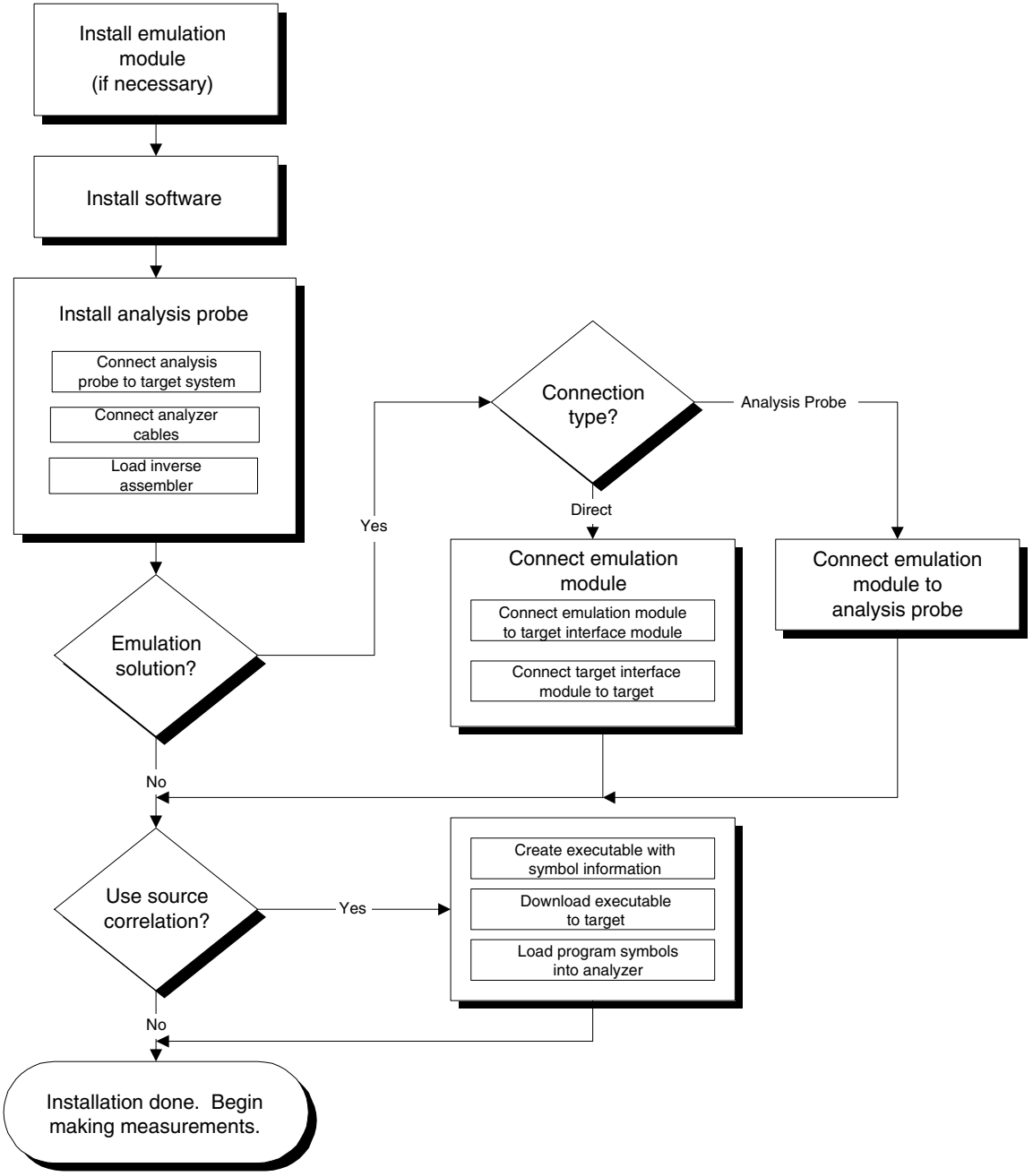
Part 1

Installation Guide

Overview of Installation and Setup

Follow these steps to connect your equipment:

1. Check that you received all of the necessary equipment. See the "Equipment and Requirements" chapter on page 19.
2. If you need to install an emulation module in an HP 16600A/16700A-series logic analysis system, see "Installing the Emulation Module" on page 38.
3. Install the software. See "Installing Software" on page 44.
4. If you have an HP 16600A/16700A-series logic analysis system, use the Setup Assistant to help you connect and configure the analysis probe and emulation module. See "Using the Setup Assistant" on page 48.
5. If you do not have an HP 16600A/16700A-series logic analysis system, see the "Using the Analysis Probe with Other Logic Analyzers" chapter on page 259.


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?
Analysis Probe
Direct
Connect emulation module
Connect emulation module to target interface module
Connect target interface module to target
Connect emulation 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.

1

Equipment and Requirements

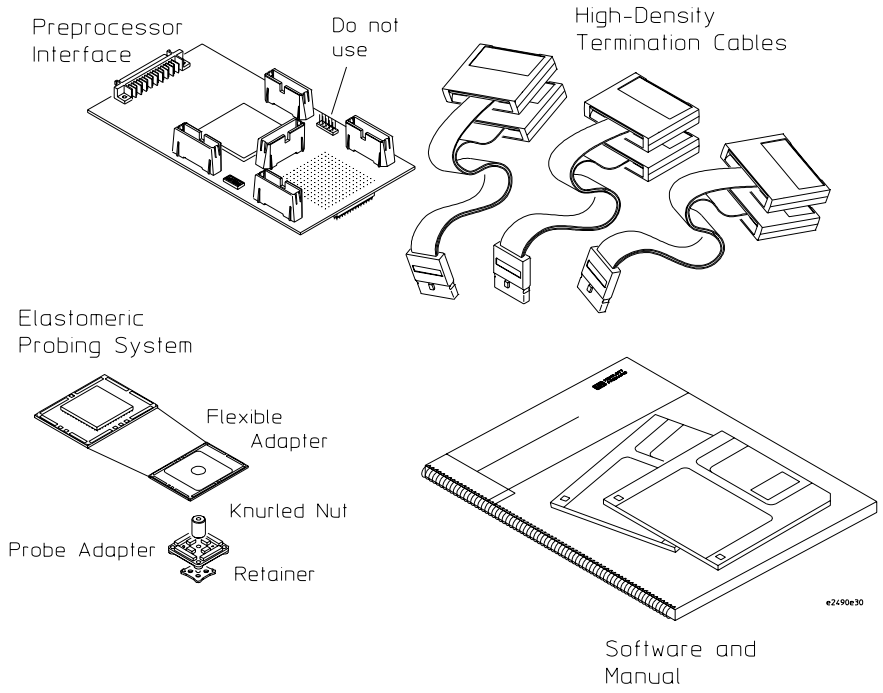
Equipment and Software Supplied

Listed below is the equipment and software supplied with the HP E9485A MPC5xx emulation solution (which includes an analysis probe, emulation module, and source correlation tool set).

Analysis Probe

The analysis probe includes:

- The HP E2490A analysis probe, which includes
  - A pre-installed male-to-male header.
  - Logic analyzer configuration files and the inverse assembler software on a CD-ROM (for HP 16600A/16700A-series logic analysis systems).
  - Logic analyzer configuration files and the inverse assembler software on one 3.5-inch disk (for other HP logic analyzers).
  - Logic analyzer configuration files and the inverse assembler software on one 3.5-inch disk (for the HP 16505A prototype analyzer).
  - This *User's Guide*.
- The HP E5373A elastomeric probing system, which includes:
  - The HP E5348A probe adapter (and its *Installation Guide*).
  - The HP E5350A flex adapter.
- Three HP E5346A high-density termination cables.



Emulation Module

The emulation module includes:

- An HP 16610A emulation module.

  If you ordered an emulation module as part of your HP 16600A/16700A-series logic analysis system, it is already installed in the frame.

- Firmware for the emulation module and/or updated software for the Emulation Control Interface on a CD-ROM.

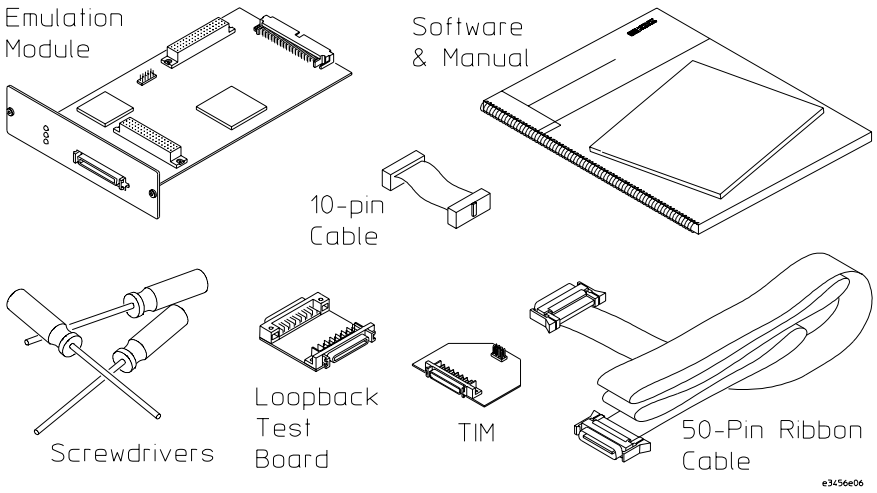
- A 50-pin ribbon cable for connecting the emulation module to the HP E2490A analysis probe or to the target interface module (TIM).

- A target interface module (TIM) circuit board.

  The target interface module (TIM) is used to connect the emulation module to a debug port in the target system.

The target interface module (TIM) is not needed when using the HP E2490A analysis probe because the target interface module's functionality is already built into the analysis probe.

- A 10-pin ribbon cable for connecting the target interface module (TIM) to a debug port connector in the target system.
- One Torx T-10 and one Torx T-15 screwdriver.
- An emulation module loopback test board (HP part number E3496-66502).



Source Correlation Tool Set

The source correlation tool set includes:

- An entitlement certificate for licensing the software.
- The HP 16600A/16700A-series logic analysis system software CD-ROM.

The HP B4620B source correlation tool set software is already installed on the HP 16600A/16700A-series logic analysis system's disk. All you need is the entitlement certificate for licensing the source correlation tool set software. The CD-ROM is included in case you need to re-install the software.

Additional Equipment and Software Required

Listed below is the additional equipment and software required by the HP E9485A emulation solution for the MPC5xx.

Analysis Probe

Emulation Module/Probe

An emulation module/probe is used to program the HP E2490A analysis probe for address reconstruction.

Logic Analyzer

The analysis probe requires a logic analyzer to provide state and timing analysis of MPC505/509 target systems.

HP 16600/16700A-Series Logic Analysis Systems. When used as part of the emulation solution, the HP E2490A analysis probe is used with logic analyzers in the HP 16600/16700A-series logic analysis systems:

- The HP 16600A-series logic analysis systems have built-in logic analyzers.
- The HP 16700A-series logic analysis systems have logic analyzer cards installed like the HP 16500A, HP 16554A, HP 16555A/D, HP 16556A/D, HP 16557D, HP 16710A, HP 16711A, or HP 16712A.

Other HP Logic Analyzers. By itself, the HP E2490A analysis probe can also be used with other HP logic analyzers like the HP 1660A/AS/C/CP/CS/E/EP/ES, HP 1661A/AS/C/CP/CS/E/EP/ES, HP 1670A/D/E, HP 1671A/D/E, or logic analyzer cards in the HP 16500B/C logic analysis system (see the “Using the Analysis Probe with Other Logic Analyzers” chapter on page 259).

Logic Analyzers Supported. The HP E2490A analysis probe has 5 connectors for high-density logic analyzer probe cables. (Each of these high-density connectors connect to two 16-channel logic analyzer pods.)

Three high-density connectors (6 logic analyzer pods) are required for inverse assembly with state analysis and for timing analysis.

One of the other high-density connectors (2 logic analyzer pods) contains additional status signals. If you want to monitor signals on the optional pods, you will need an additional high-density cable.

The following table lists the logic analyzers that meet the HP E2490A analysis probe's requirements.

Logic Analyzer	Channel Count	State Speed	Timing Speed (Full/Half Channels)	Memory Depth (Full/Half Channels)
HP 16550A (one or two cards)	102/card	100 MHz	250/500 MHz	4K/8K states
HP 16554A (two or more cards)	68/card	70 MHz	125/250 MHz	500K/1M states
HP 16555A (two or more cards)	68/card	110 MHz	250/500 MHz	1M/2M states
HP 16555D (two or more cards)	68/card	110 MHz	250/500 MHz	2M/4M states
HP 16556A (two or more cards)	68/card	100 MHz	200/400 MHz	1M/2M states
HP 16556D (two or more cards)	68/card	100 MHz	200/400 MHz	2M/4M states
HP 16557D (two or more cards)	68/card	135 MHz	250/500 MHz	2M/4M states
HP 16600A	204	100 MHz	125/250 MHz	64K/128K states
HP 16601A	136	100 MHz	125/250 MHz	64K/128K states
HP 16602A	102	100 MHz	125/250 MHz	64K/128K states
HP 16710A (one or two cards)	102/card	100 MHz	250/500 MHz	8K/16K states
HP 16711A (one or two cards)	102/card	100 MHz	250/500 MHz	32K/64K states
HP 16712A (one or two cards)	102/card	100 MHz	250/500 MHz	128K/256K 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 measurements with the HP E2490A analysis probe.

Logic Analyzer	Minimum Software Version
HP 16600A-series	The latest HP 16600A logic analyzer software version is on the CD-ROM shipped with this product.
HP 16700A-series*	The latest HP 16700A logic analyzer software version is on the CD-ROM shipped with this product.
* Used with the HP 16550, HP 16554/55/56/57, and HP 16710/11/12 logic analyzers.	

Emulation Module

The emulation module requires:

- An HP 16600A/16700A-series logic analysis system into which it can be installed.
- Interface software that gives you access to the emulation module's functionality.

  You can use the HP 16600A/16700A-series logic analysis system' s Emulation Control Interface. Or, you can use a third-party high-level source debugger to access and control the emulation module.

Source Correlation Tool Set

The source correlation tool set requires the HP 16600A/16700A-series logic analysis system.

Other Optional Equipment and Software

The emulation module works with several debuggers offered by other vendors.

Refer to the *Emulation and Analysis Solutions for Motorola MPC 505 and 509 Microprocessors* product overview data sheet (HP literature number 5966-4829E) for a list of supported debuggers. You can find this data sheet on the logic analysis and emulation world-wide web site at:

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

Unsupported Microprocessor Modes

The HP E2490A analysis probe does not support the following MPC505/509 configurations:

LAST Burst Mode. LAST Burst Mode is not supported. BDIP mode is supported.

GPIO Pins. The primary signals on the 50X are needed for address reconstruction, data display, and status decoding. Using the general purpose ports will produce incorrect results in the listing display. If a Port Replacement Unit is installed on the target system, the external memory-mapped ports can still be used, as long as the primary signals are still sent to the analysis probe.

Endian Mode. The inverse assembler supports big-endian mode. It does not support little-endian mode.

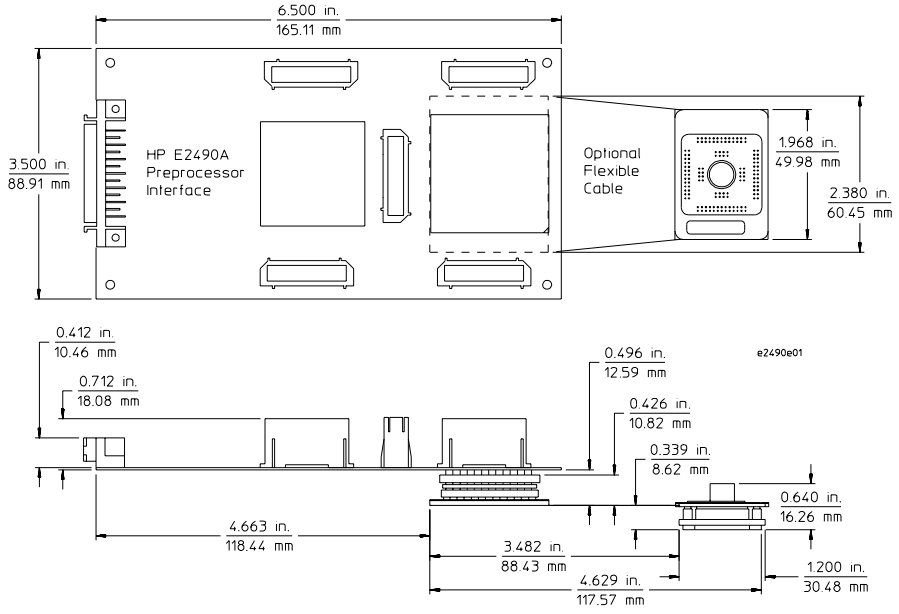
2

Preparing the Target System

Preparing for the Analysis Probe

Clearance

The dimensions of the analysis probe are shown below.



Keep-Out Area

A target system with sufficient clearance around the microprocessor (keep out area) for the probe adapter.

The required clearances are shown in the probe adapter *Installation Guide*.

Boot Code for Inverse Assembly

For the inverse assembler to work properly, the CR/DS must be

configured for DS mode. Boot code to configure for DS mode follows.

If the CR/DS pin must be configured in CR mode for your target system, show cycle information captured in the trace is still valid.

Address	Data	Instruction		Comments
FFF00100 FFF00104 FFF00108 FFF0010C	7C908AA6 3CA00200 7C832B78 7C708BA6	mfspr lis or mtspr	r4 ic_cst r5 0200 r3 r4 r5 ic_cst r3	Enable the instruction cache. This will allow show cycles to appear on the bus, which will synchronize the 50X analysis probe. Only a single show cycle is required. Subsequent instructions can turn off show cycles and/or the instruction cache.
FFF00110 FFF00114 FFF00118 FFF0011C FFF00120 FFF00124	38E0FC64 3CA0FFFF 7CE42878 64868007 38E000DF 98E60000	li lis andc oris li stb	r7 FFFFFC64 r5 FFFF r4 r7 r5 r6 r4 8007 r7 000000DF r7 0000(r6)	Configure CR/DS as the Data Strobe.

Preparing for the Emulation Module

When using the MPC5xx emulation module, you need to be aware of the requirements it makes of target systems, and you need to consider how and when the emulation module connects to the target system.

Target System Requirements

The DSDI and DSCK signals must not be actively driven by the target system when the debug port is being used.

The $\overline{\text{RESET}}$ signal from the debug connector must be ORed with the $\overline{\text{RESET}}$ signal that connects to the processor on the target system. They can be logically ORed or "wire-ORed" on the board. The emulation module drives $\overline{\text{RESET}}$ through a 100 ohm resistor with an open-collector driver. There is also 1.79 Kohm pullup to 3.3 volts on the $\overline{\text{RESET}}$ line.

The HP emulation module adds about 40 pF to all target system signals routed to the debug connector. This added capacitance may reduce the rise time of the $\overline{\text{RESET}}$ signal beyond the processor specifications. If so, the target may need to increase the pull-up current on these signal lines.

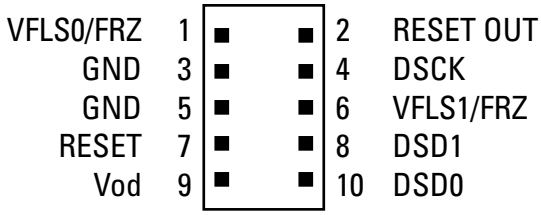
Additional target requirements may be specified in the release notes in the "readme" file on the provided floppy disk.

Debug Port Connection (Optional)

Because the analysis probe has a built-in connector for the emulation module, your target system doesn't need a debug port connector and you don't have to use the supplied target interface module to connect to it.

However, if you plan to connect the emulation module directly to the target system, the target system should have a debug port (BDM) connector.

The connector should be a dual row header strip (“Berg connector”), 10 pins per inch, with 25 mil pins.



Pins 1 and 6 may be connected to VFLS0 and VFLS1 respectively, or, if a single freeze line is used, to the FRZ line.

3

Setting Up the Logic Analysis System

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 the HP 16600A/16700A-series logic analysis systems

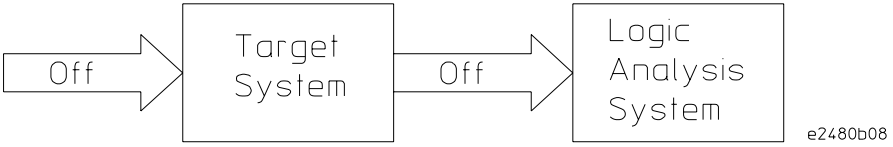
Ensure the target system is powered off.

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-OFF

Turn off power to your system in the following order:

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



Installing Logic Analyzer Modules

You should install logic analyzer, oscilloscope, or pattern generator modules in your logic analysis system before you install an emulation module and software.

Refer to the HP 16600A/16700A-series logic analysis system's *Installation Guide*.

Installing the Emulation Module

Your emulation module may already be installed in your logic analysis system. However, if you need to install an emulation module, 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 in a HP 16700A-series logic analysis system

Or, to install in an HP 16701A expansion frame:

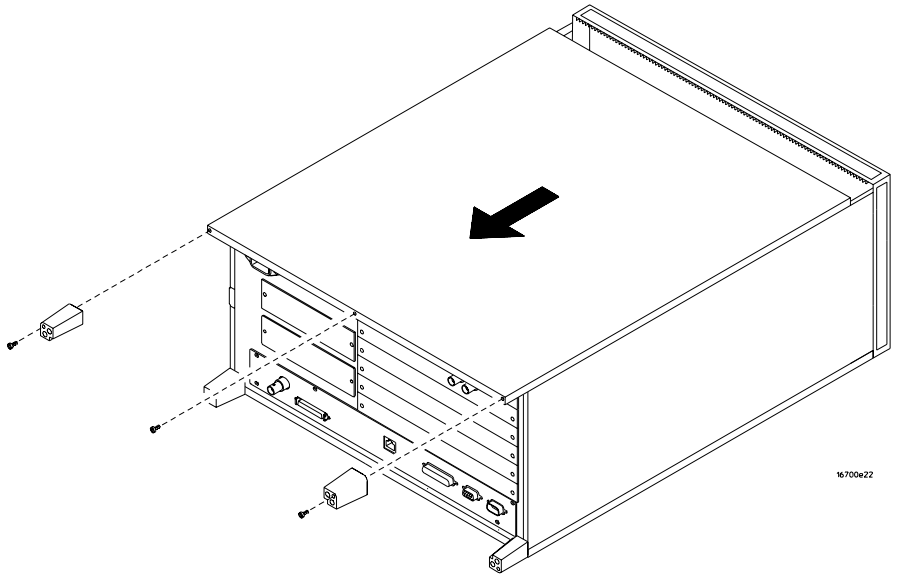
You will need T-10 and T-15 Torx screw drivers.

1. Turn off the logic analysis system and REMOVE THE POWER CORD.

   Remove any other cables (including mouse or video monitor cables).

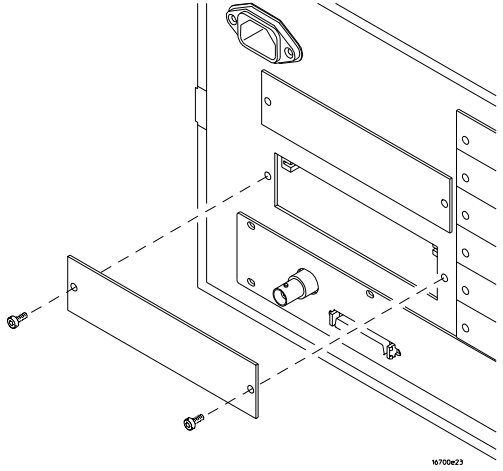
2. Turn the logic analysis system frame upside-down.

3. Remove the bottom cover.

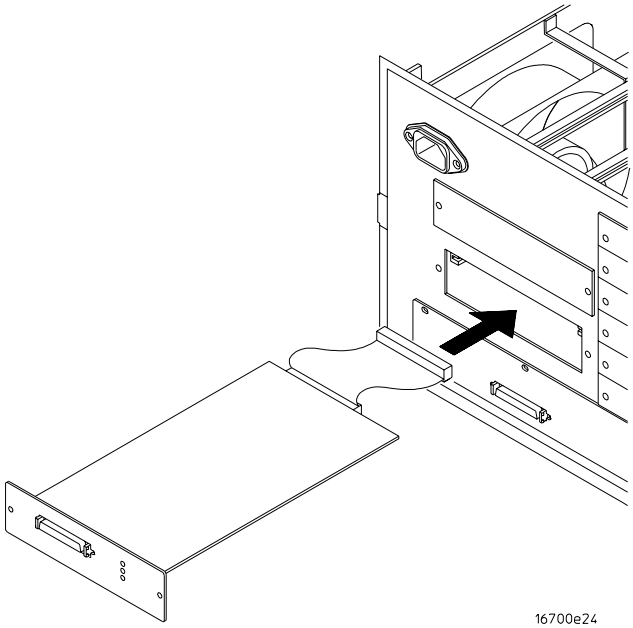


4 Remove the slot cover.

You may use either slot.

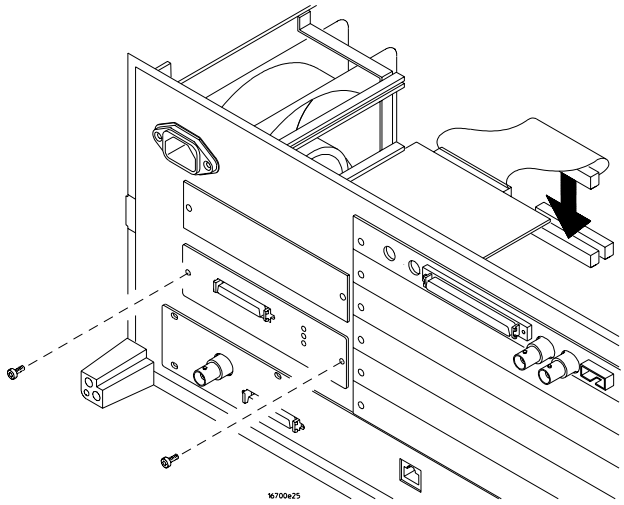


5 Install the emulation module.



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 "To update emulation module firmware" on page 83 for information on giving the emulation module a "personality" for your target processor.

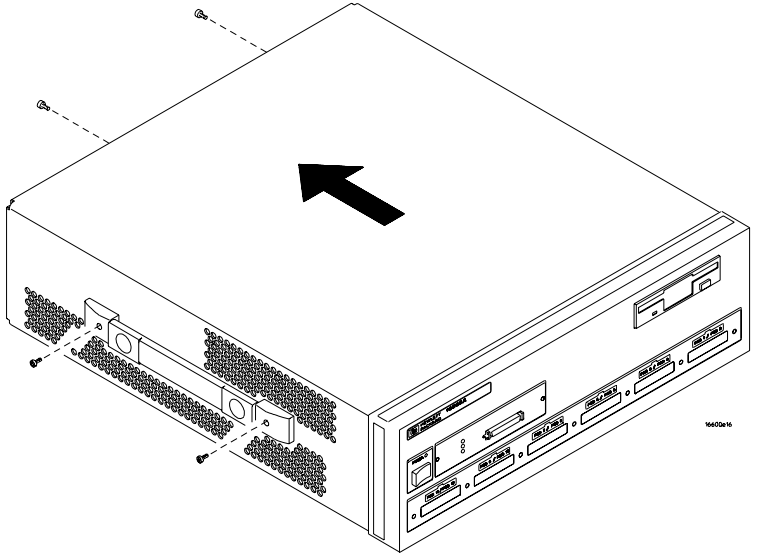
To install in a HP 16600A-series logic analysis system

You will need T-8, T-10, and T-15 Torx screw drivers.

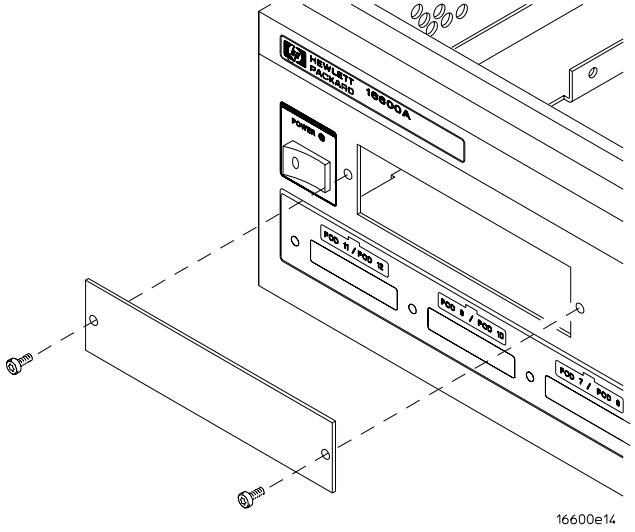
1 Turn off the logic analysis system and REMOVE THE POWER CORD.

Remove any other cables (such as probes, mouse, or video monitor).

2 Slide the cover back.

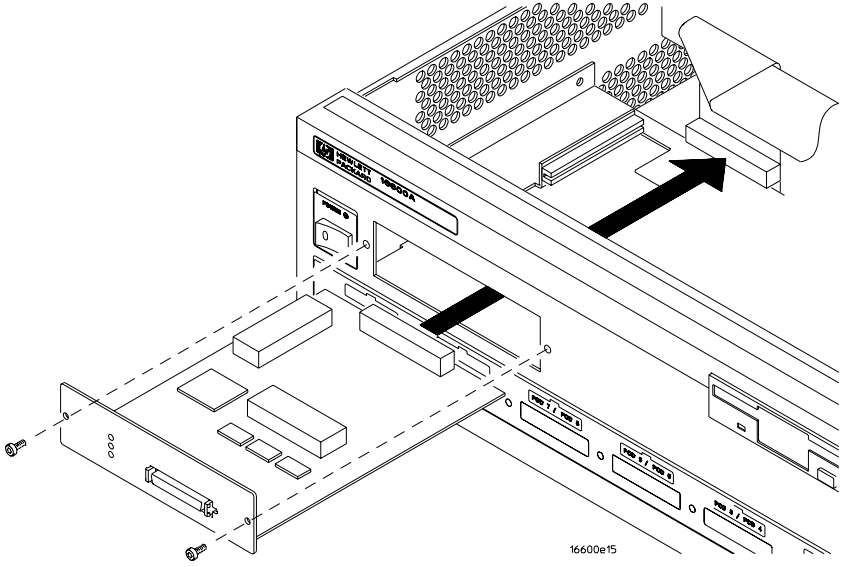


3 Remove the slot cover.



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 "To update emulation module firmware" on page 83 for information on giving the emulation module a "personality" for your target processor.

To test the emulation module

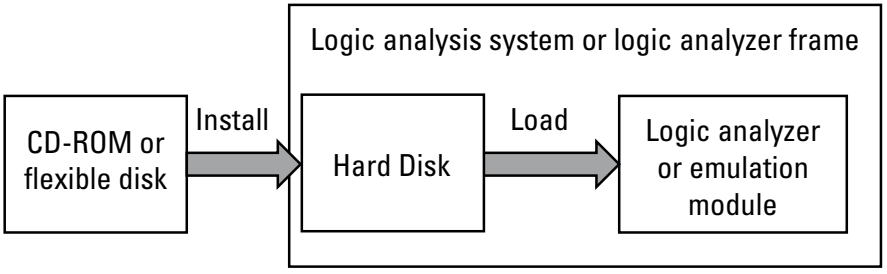
If this is the first time that you have used the emulation module, you should run the built-in performance verification tests before you connect to a target system. Refer to the "Troubleshooting the Emulation Module" chapter on page 211 for information on performance verification.

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 analysis system. Later, you will need to load some of the files into the appropriate measurement module.



What needs to be installed

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.

To install software from CD-ROM

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/16700A-series logic analysis system's operating system, installation may take approximately 15 minutes.

1. Turn on the CD-ROM drive first; then, turn on the logic analysis system.
2. Insert the CD-ROM in the drive.
3. Click the System Admin icon.
4. Click Install... .

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

   A list of the processor support packages on the CD-ROM will be displayed.
7. Click on the "MPC5XX" package.

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

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

   The configuration files are stored in /hplogic/configs/hp/processor.

   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 on-line help for more information on installing, licensing, and removing software.

4

Probing the Target System

Using the Setup Assistant

The Setup Assistant is an on-line 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.

Connecting the Analysis Probe to the Target System

Disconnect power from the logic analyzer and your target system before you make or break connections. If you have a emulation probe, also disconnect its power.

The connection flow is as follows:

- Connect the probe adapter to the target system.
- Connect analysis probe to the probe adapter.

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.

To connect the probe adapter to the target system

The HP E5373A elastomeric probing system consists of a probe adapter that attaches to a 160-pin PQFP microprocessor and a flexible cable that provides a PGA socket for attaching to the HP E2490A analysis probe.

The probe adapter requires a minimum clearance around the microprocessor on the target system that is free of any other components (keep-out area). Refer to the elastomeric probing system *Installation Guide* for information on the required keep-out area.

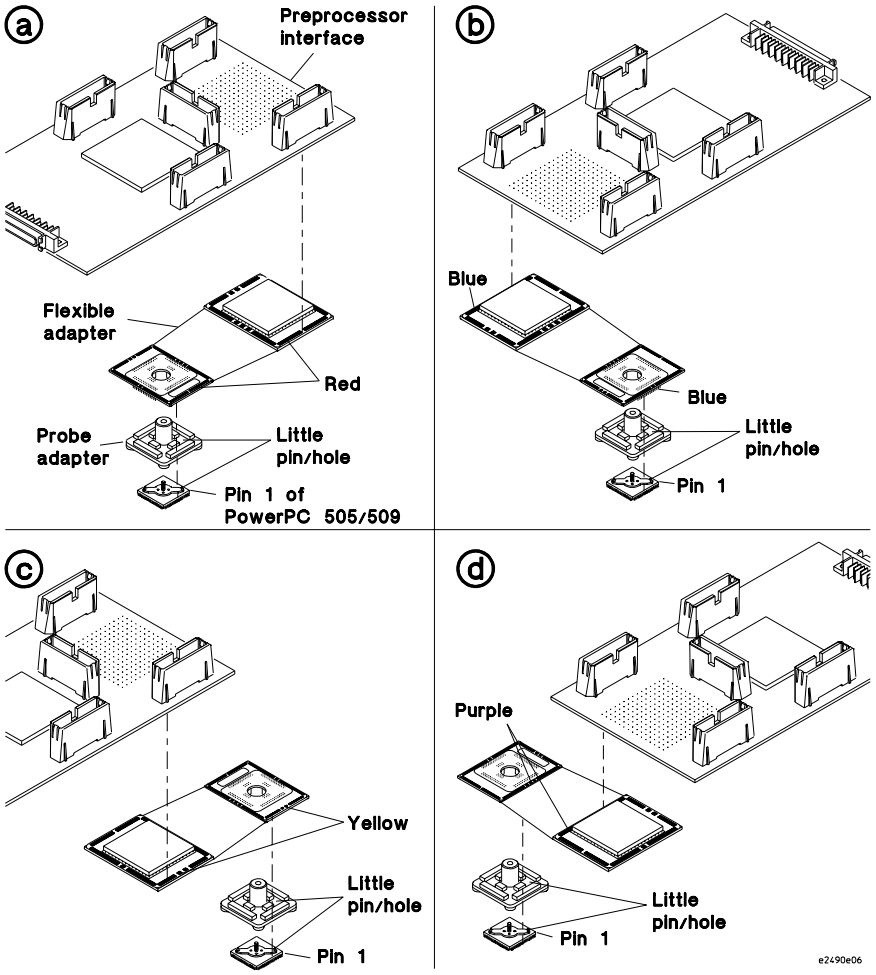
CAUTION: Equipment Damage. To prevent equipment damage, remove power from the target system and make sure no logic analyzer cables are connected to the analysis probe.

1. Turn off the target system and disconnect all logic analyzer cables from the analysis probe.
2. Select the rotation (shown on the next page) that best suits your target system. Note the following indicators on the illustration:
   - Position of Pin 1 on the microprocessor.
   - Position of little pin on the retainer.
   - Position of little hole on the probe adapter.
   - Color code on both ends of the flexible adapter.
   - Position of Pin A1 on the analysis probe.

   Flexible adapters can be installed in one of four rotations as shown in the illustration. This allows flexibility in attaching the analysis probe when target system components interfere.

CAUTION: Serious damage can be done to the target system or analysis probe from incorrect connection. Note the position of pin 1 on the target system and Pin A1 on the analysis probe prior to making any connection. Also, take care to align the pins so that all pins are making contact.

3. Follow the instructions in the elastomeric probing system *Installation Guide* to adhere the retainer and attach the probe adapter to the microprocessor.

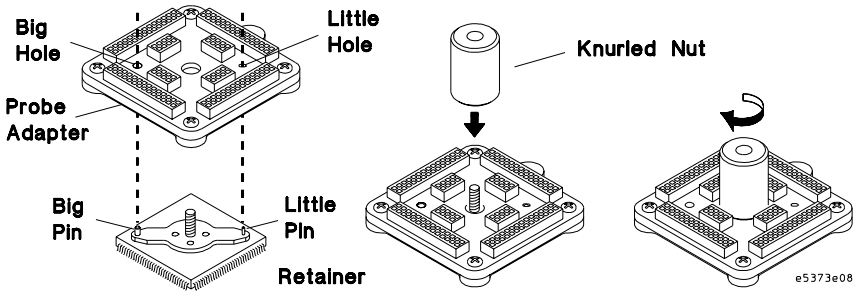


4 Install the probe adapter on the retainer. Use the instructions in the elastomeric probing system *Installation Guide*.

a Place the probe adapter over the retainer so that the threaded stud passes through the center hole of the adapter, and the dowel pins of the retainer enter the holes in the adapter.

b Avoid touching the elastomers to the threaded stud or dowel pins of the retainer as the probe is inserted.

c Make sure the dowels on the retainer are inserted in the holes on the probe adapter and that it is laying flat against the microprocessor before tightening the nut.



5 Turn the knurled nut until it stops.

CAUTION: Do not over-tighten the knurled nut. Over tightening the nut will make it difficult to remove.

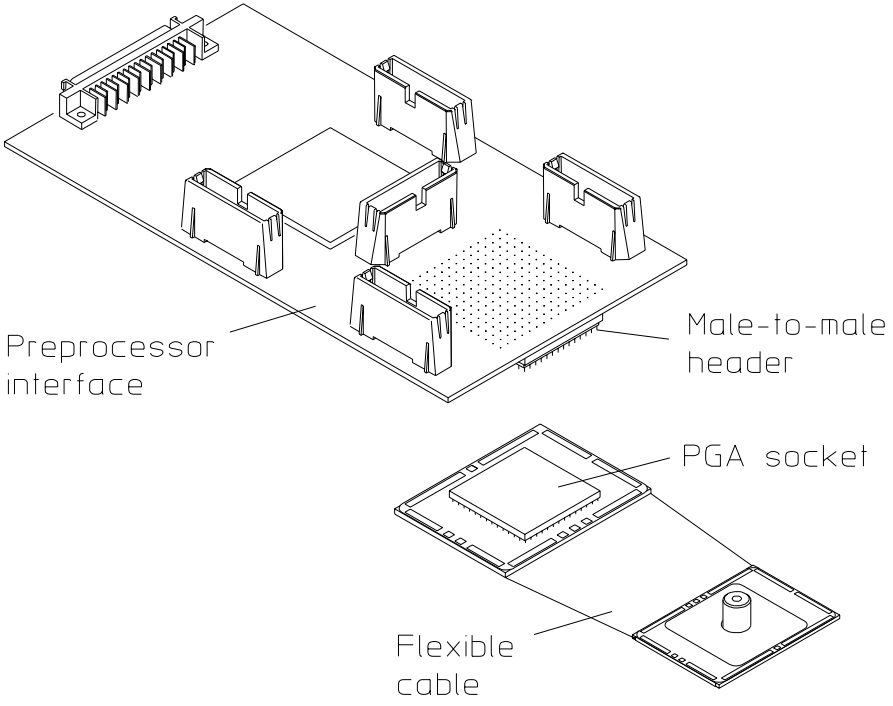
6 Using the rotation selected in step 2 and the illustration on the previous page, attach the flexible adapter to the probe adapter.

7 Using the rotation selected in step 2 and the illustration on the previous page, attach the PGA socket on the analysis probe to the PGA socket on the flexible cable. The male-to-male header provides the connection between the two.

CAUTION: Serious Equipment Damage
Ensure that the analysis probe, pin adapters, flexible adapter, and probe adapter are aligned and seated correctly in the sockets. Serious equipment damage can result from incorrect connection. The final connection must match the rotation selected from the previous page.

To connect the analysis probe to the flexible cable

Using the rotation selected in step 2 of the previous section, attach the male-to-male header on the bottom of the analysis probe to the PGA socket on the flexible adapter.



CAUTION: The above illustration shows one possible rotation. Ensure you use the rotation selected previously, or damage to equipment may result.

Connecting the Analysis Probe to a Logic Analyzer

This section shows you how to connect the analysis probe to the logic analyzer. It consists of the following:

- Connecting the high-density cables to the analysis probe.
- Connecting the high-density cables to the logic analyzer.

This section shows connection diagrams that identify connections to each individual logic analyzer supported by the analysis probe. Since there are different connections and configurations for state and timing analysis, each logic analyzer sections shows connections for both. They are shown in the following order:

- HP 16550A logic analyzer (one or two cards).
- HP 16554/55/56/57 logic analyzers (two or three cards).
- HP 16600A logic analyzer.
- HP 16601A logic analyzer.
- HP 16602A logic analyzer.
- HP 16710/11/12A logic analyzer (one or two cards).

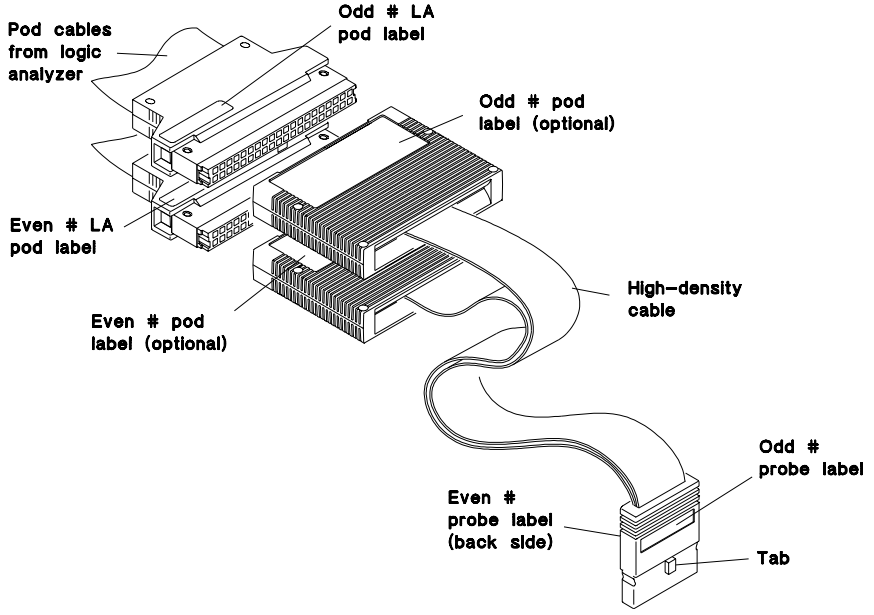
Number of Pods Used/Required

Both state and timing measurements require a minimum of six pods to make a measurement. The logic analyzer configuration files assign signals for eight pods. If fewer than eight pods are used, the logic analyzer will truncate the pods allocated. In this case, viewing the logic analyzer Format menu shows the pod allocations. If the allocations will not acquire the desired signals, the allocations can be altered manually.

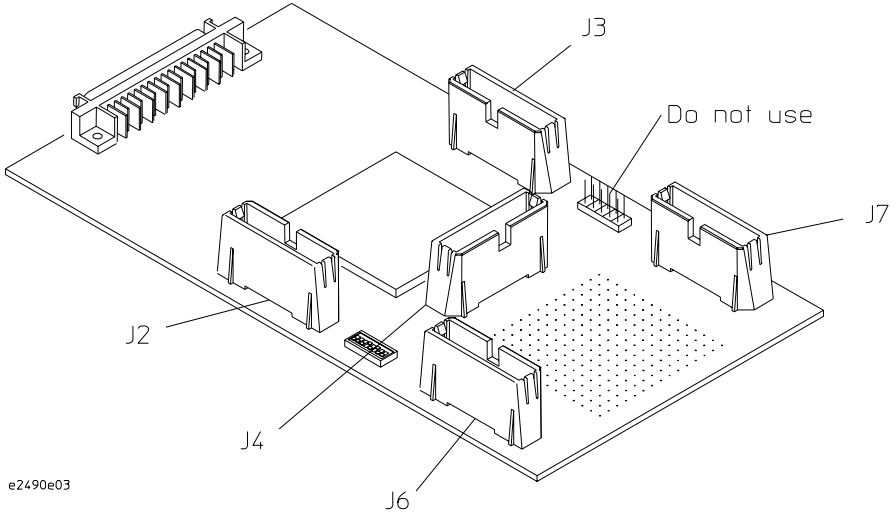
The HP E2490A analysis probe includes three HP E5346A high-density termination cables. If you want to connect eight pods to your logic analyzer, you will need one additional high-density termination cable.

To connect the high-density termination cables to the analysis probe

Three high-density termination cables, and labels to identify them, are included with the HP E2490A analysis probe. You can order additional cables from your HP Sales Office using model number HP E5346A. The labels can be attached to the cables before or after the cables have been connected to the analysis probe and logic analyzer. The illustration below shows the even and odd cable pods. For state analysis, connectors J2, J4, J7, and (optionally) J3 are used. For timing analysis, connectors J2, J6, J7, and (optionally) J3 are used.



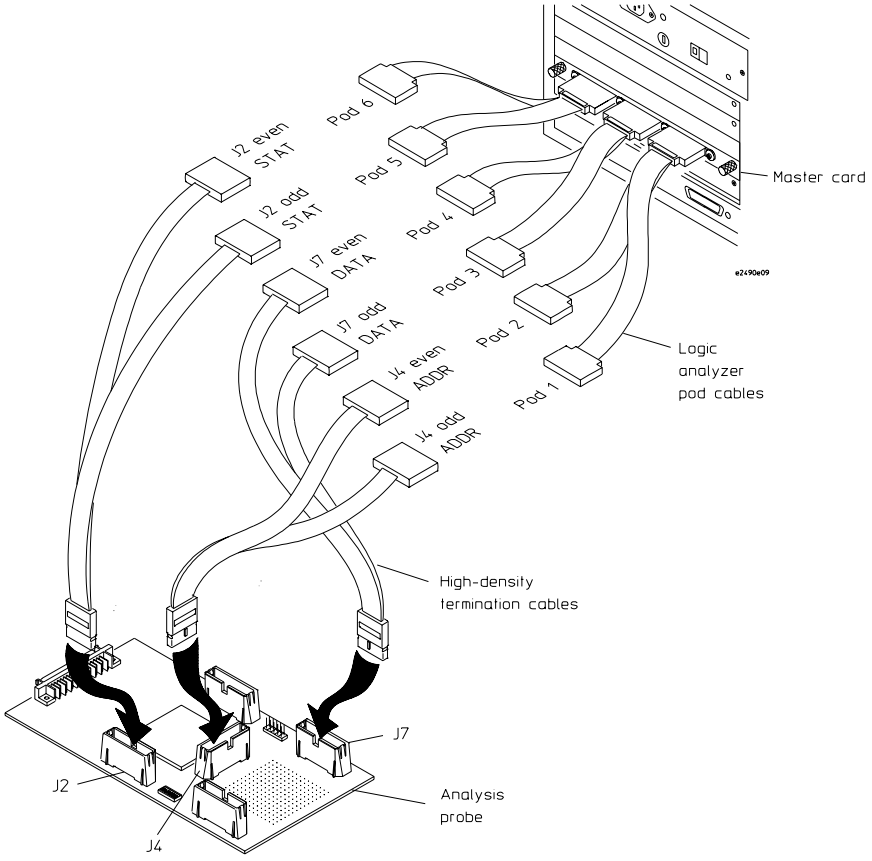
The following pages show the connections between the logic analyzer pod cables and the high-density termination cables of the analysis probe. Note that for each logic analyzer, there are separate connections for state analysis and for timing analysis. Refer to the appropriate pages for your logic analyzer. The configuration file names for each logic analyzer are included with the connection diagrams.



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

Use the following figures to connect the analysis probe to a one-card HP 16550A logic analyzer. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

State Analysis Connection

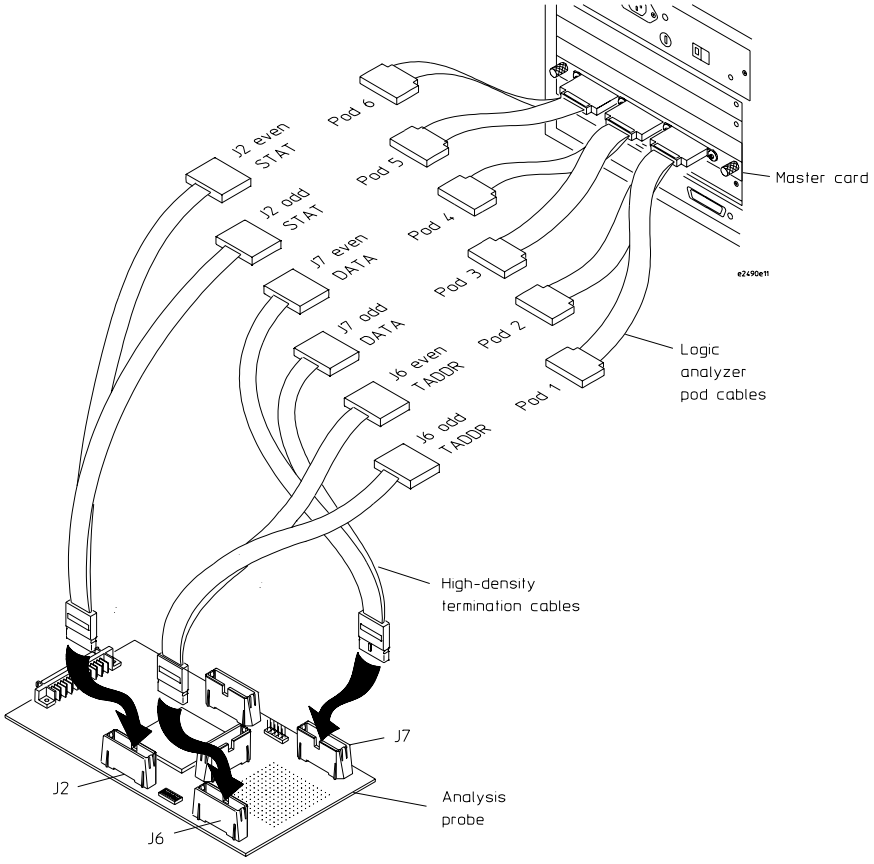


Configuration File (State)

Use configuration file C5051A1 for state analysis with the one-card HP 16550 logic analyzer.

If fewer than eight pods are used, the logic analyzer will truncate the pods allocated. In this case, viewing the logic analyzer Format menu shows the pod allocations. If the allocations will not acquire the desired signals, the allocations can be altered manually.

Timing Analysis Connection



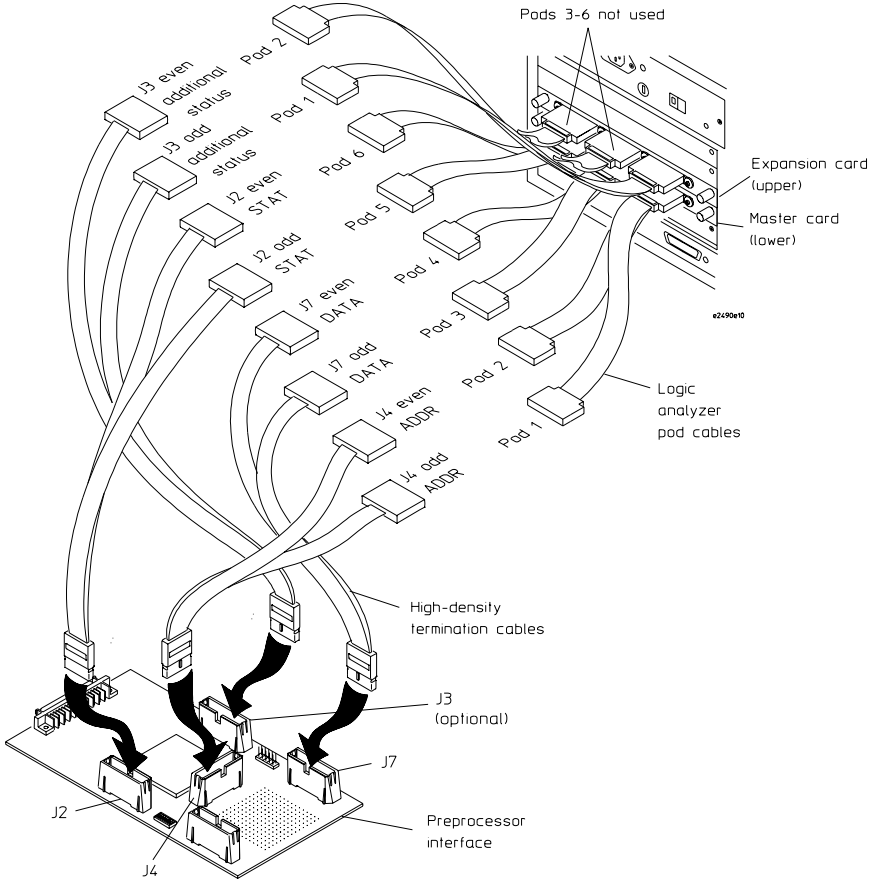
Configuration File (Timing)

Use configuration file C505T1 for timing analysis with the one-card HP 16550 logic analyzer.

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

Use the following figures to connect the analysis probe to a two-card HP 16550A logic analyzer. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

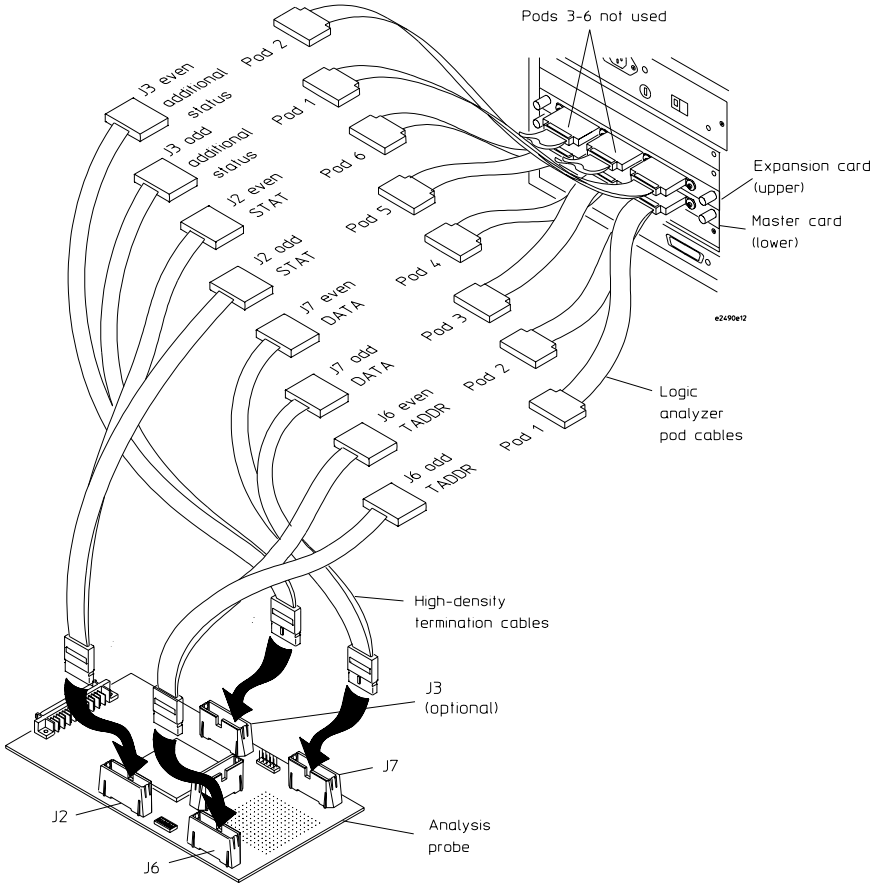
State Analysis Connection



Configuration File (State)

Use configuration file C5051A2 for state analysis with the two-card HP 16550 logic analyzer.

Timing Analysis Connection



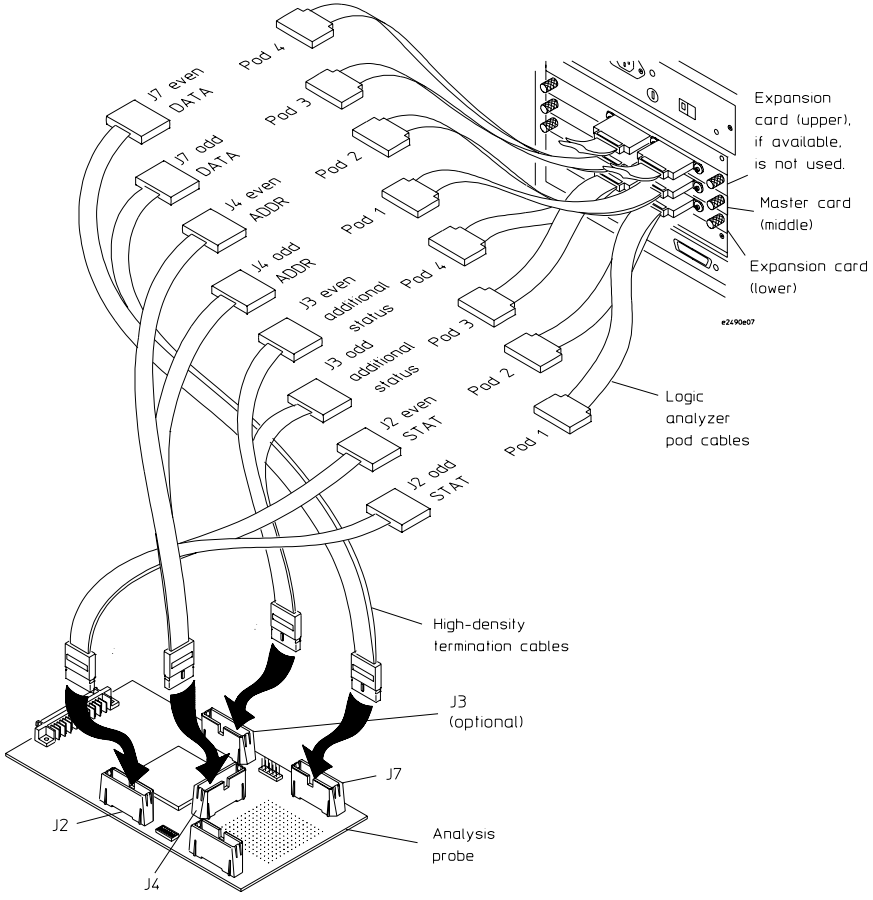
Configuration File (Timing)

Use configuration file C505T2 for timing analysis with the two-card HP 16550 logic analyzer.

To connect to the HP 16554/55/56/57 logic analyzer (two or more cards)

Use the following figures to connect the analysis probe to the HP 16554A/55A/56A and HP 16555D/56D/57D logic analyzers. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

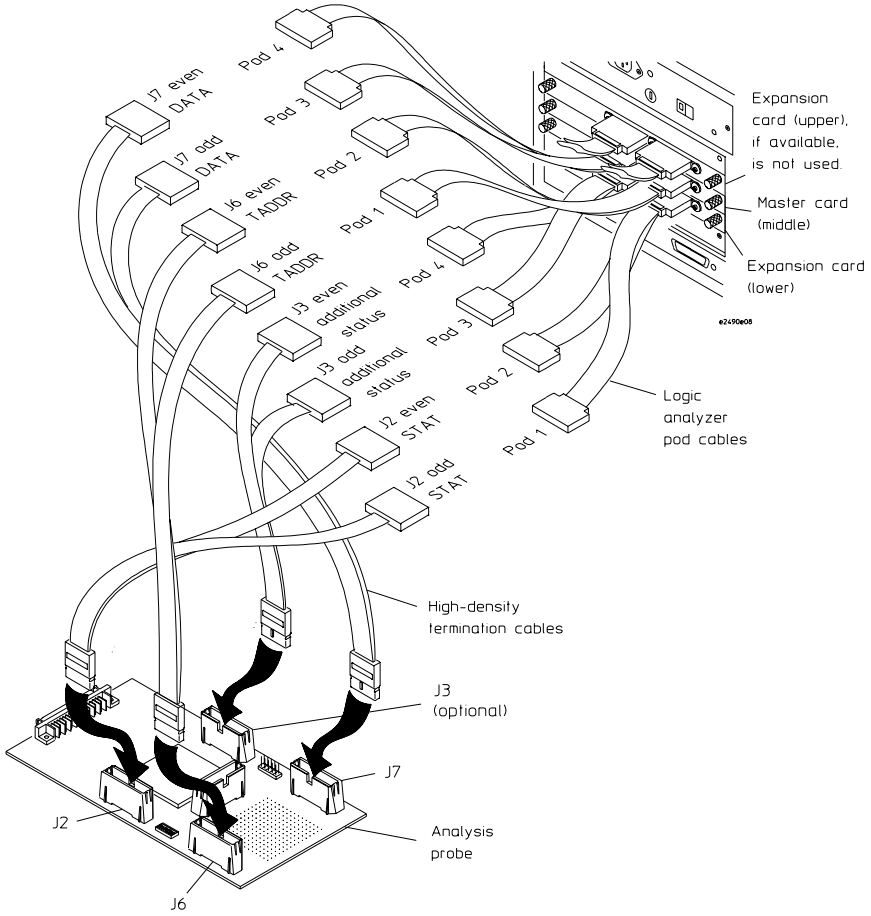
State Analysis Connection



Configuration File (State)

Use configuration file C5051A3 for state analysis with the HP 16554/55/56/57 logic analyzers.

Timing Analysis Connection



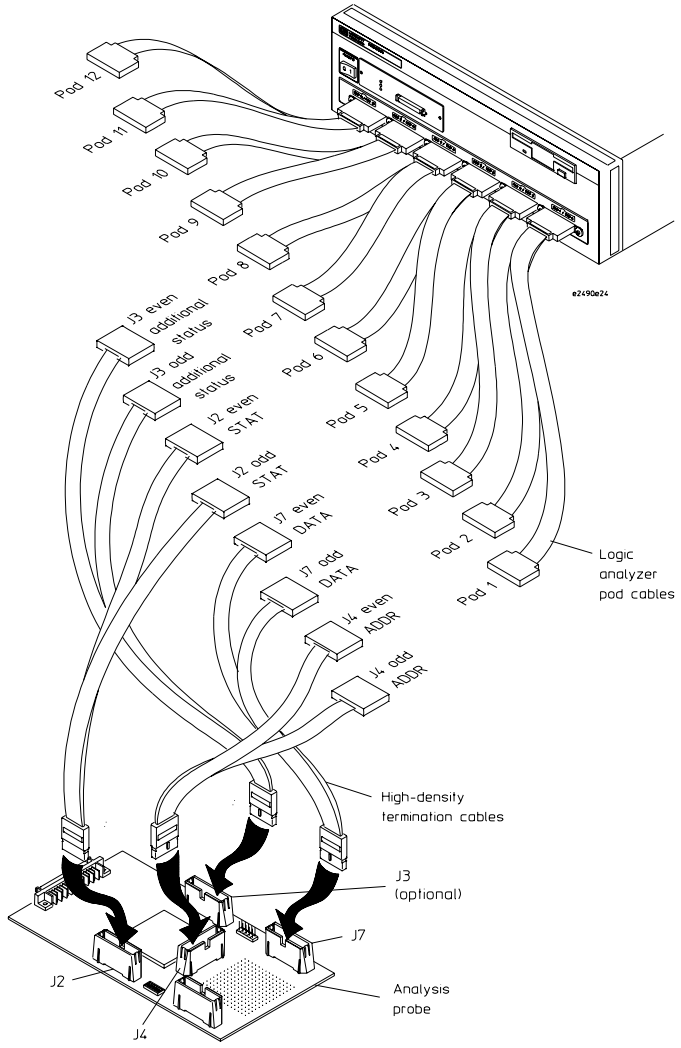
Configuration File (Timing)

Use configuration file C505T3 for timing analysis with the HP 16554/55/56/57 logic analyzers.

To connect to the HP 16600A logic analyzer

Use the following figures to connect the analysis probe to the HP 16600A logic analyzer. Remove the analysis probe from the target system before connecting the high-density cables. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

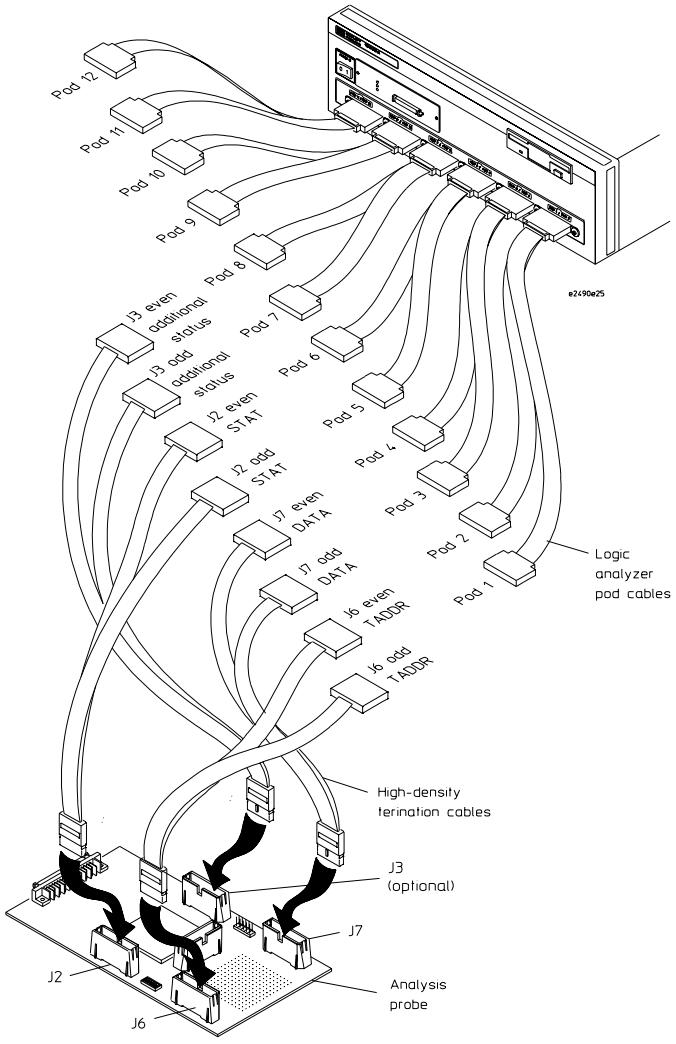
State Analysis Connection



Configuration File (State)

Use configuration file C505IA2 for state analysis with the HP 16600A logic analyzer.

Timing Analysis Connection



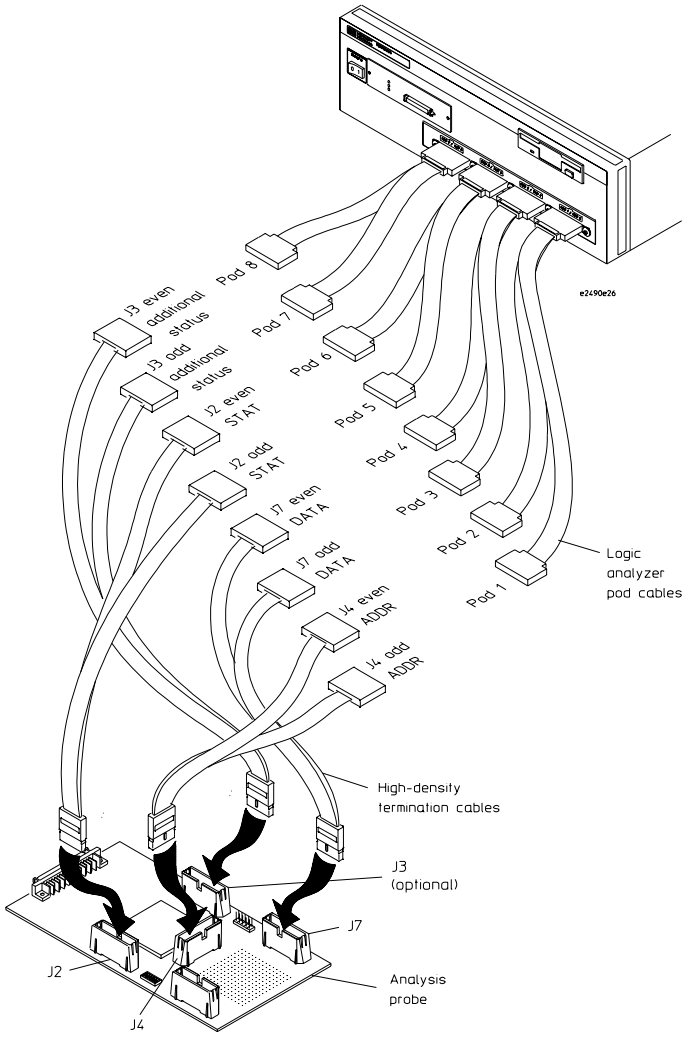
Configuration File (Timing)

Use configuration file C505T2 for timing analysis with the HP 16600A logic analyzer.

To connect to the HP 16601A logic analyzer

Use the following figures to connect the analysis probe to the HP 16601A logic analyzer. Remove the analysis probe from the target system before connecting the high-density cables. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

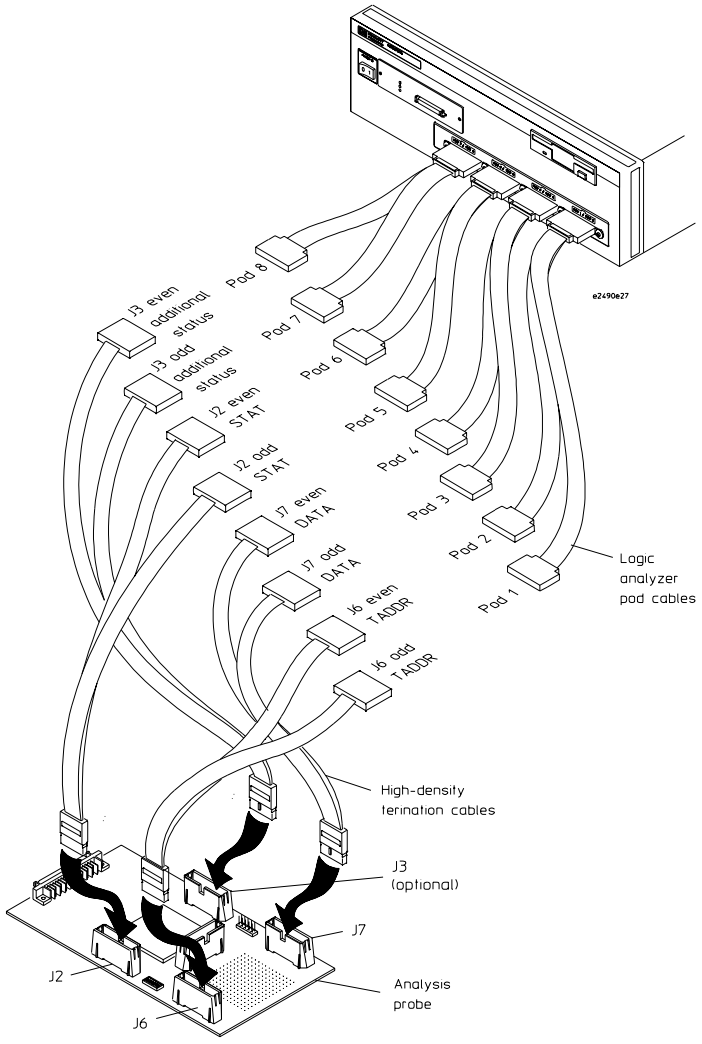
State Analysis Connection



Configuration File (State)

Use configuration file C5051A2 for the HP 16601A logic analyzer.

Timing Analysis Connection



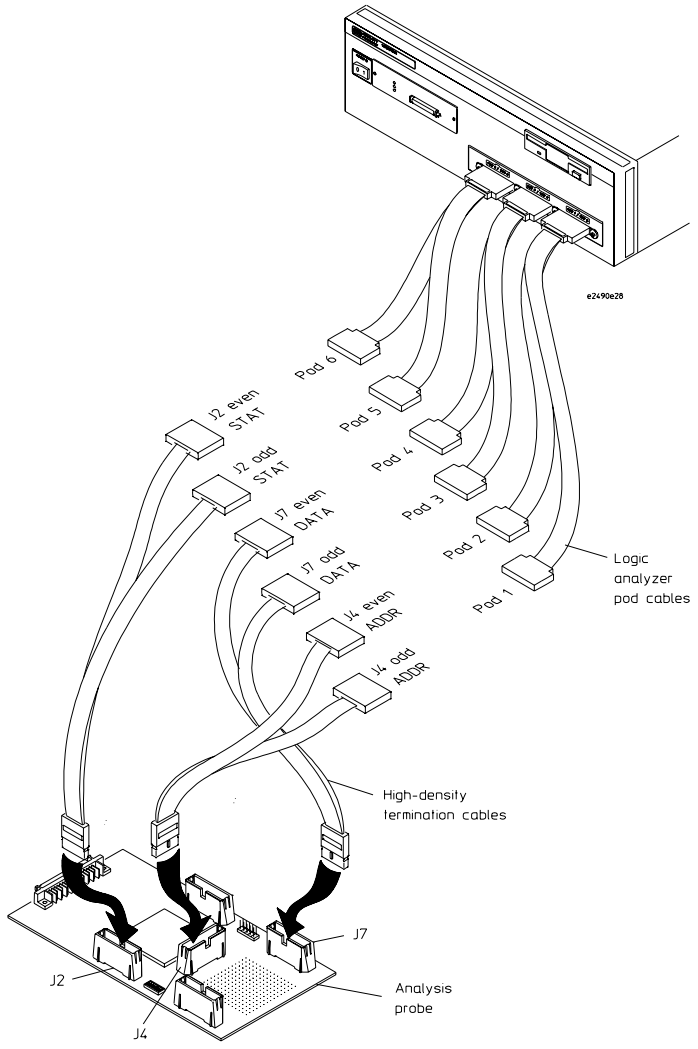
Configuration File (Timing)

Use configuration file C505T2 for the HP 16601A logic analyzer.

To connect to the HP 16602A logic analyzer

Use the following figures to connect the analysis probe to the HP 16602A logic analyzer. Remove the analysis probe from the target system before connecting the high-density cables. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

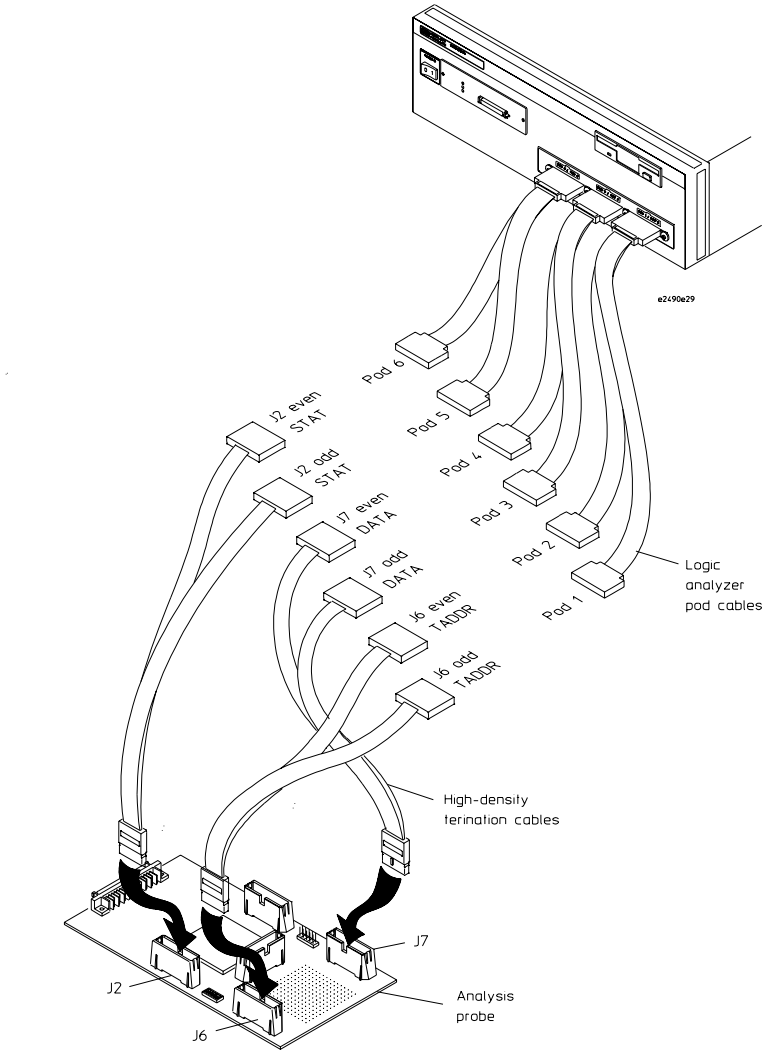
State Analysis Connection



Configuration File (State)

Use configuration file C505IA1 for the HP 16602A logic analyzer.

Timing Analysis Connection



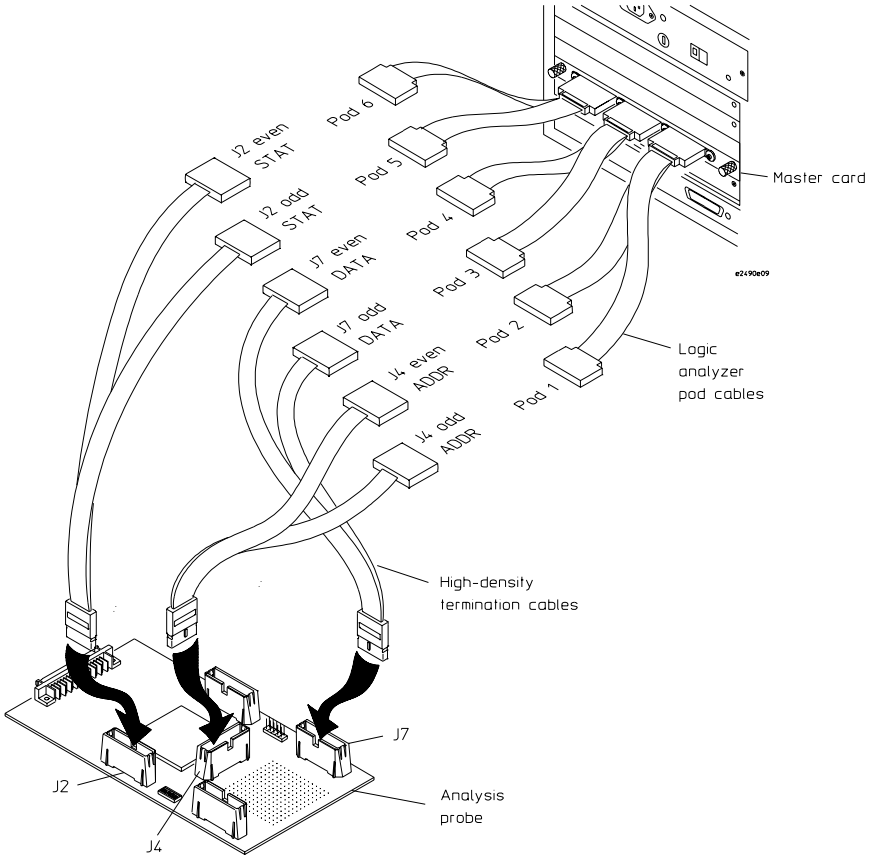
Configuration File (Timing)

Use configuration file C505T1 for the HP 16602A logic analyzer.

To connect to the HP 16710/11/12A logic analyzer (one-card)

Use the following figures to connect the analysis probe to a one-card HP 16710/11/12A logic analyzer. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

State Analysis Connection

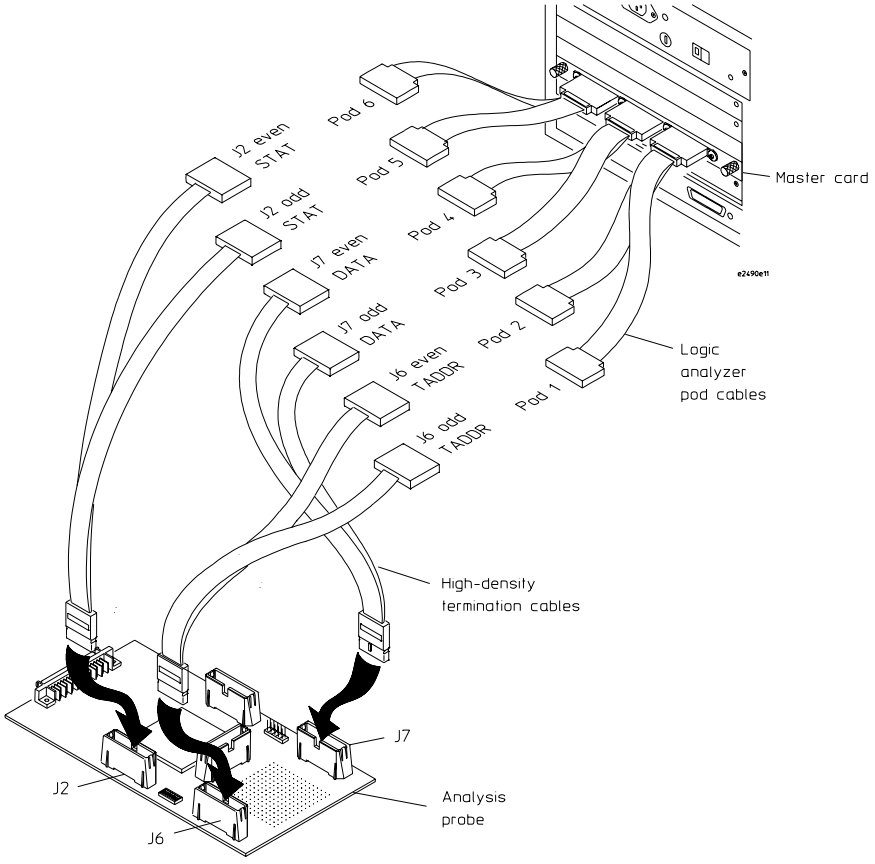


Configuration File (State)

Use configuration file C5051A1 for state analysis with the one-card HP 16710/11/12A logic analyzer.

If fewer than eight pods are used, the logic analyzer will truncate the pods allocated. In this case, viewing the logic analyzer Format menu shows the pod allocations. If the allocations will not acquire the desired signals, the allocations can be altered manually.

Timing Analysis Connection



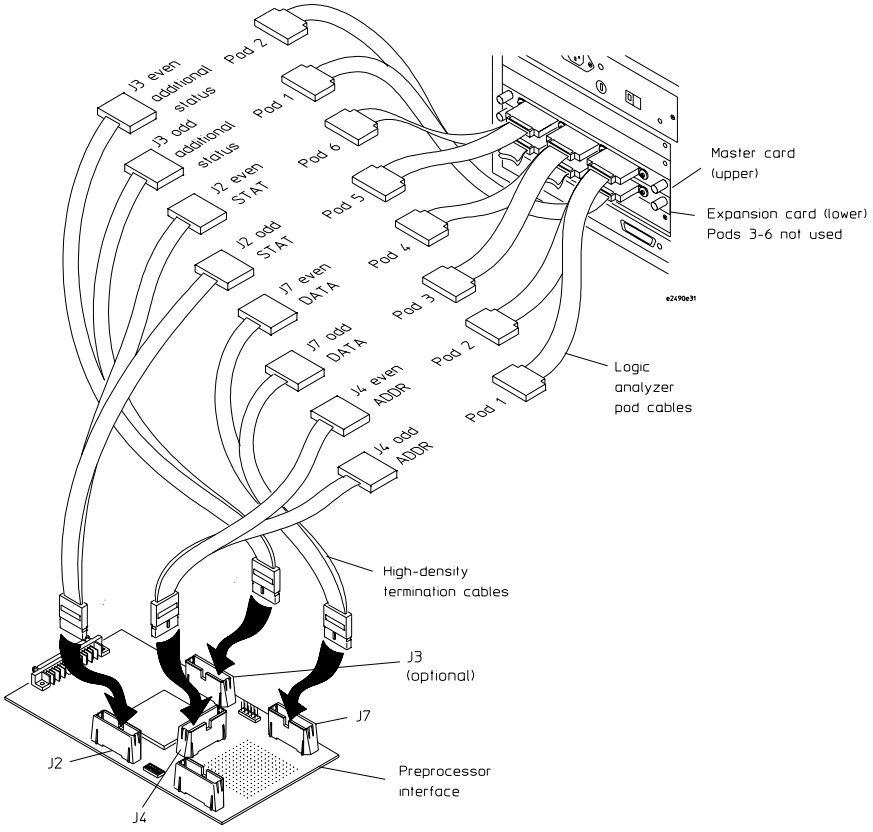
Configuration File (Timing)

Use configuration file C505T1 for timing analysis with the one-card HP 16710/11/12A logic analyzer.

To connect to the HP 16710/11/12A logic analyzer (two-card)

Use the following figures to connect the analysis probe to a two-card HP 16710/11/12A logic analyzer. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

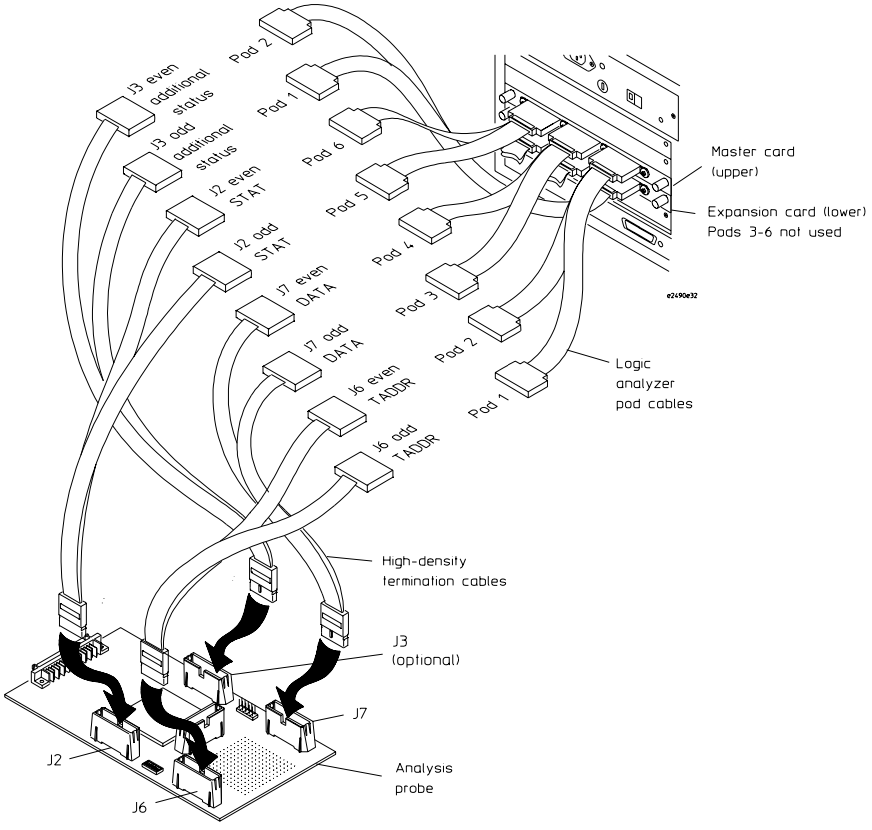
State Analysis Connection



Configuration File (State)

Use configuration file C5051A2 for state analysis with the two-card HP 16710/11/12A logic analyzer.

Timing Analysis Connection



Configuration File (Timing)

Use configuration file C505T2 for timing analysis with the two-card HP 16710/11/12A logic analyzer.

Connecting the Emulation Module to the Target System

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

- Through an HP E2490A analysis probe, which provides a direct connection to the debug port pins.
- Directly through a debug port connector on the target board.

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

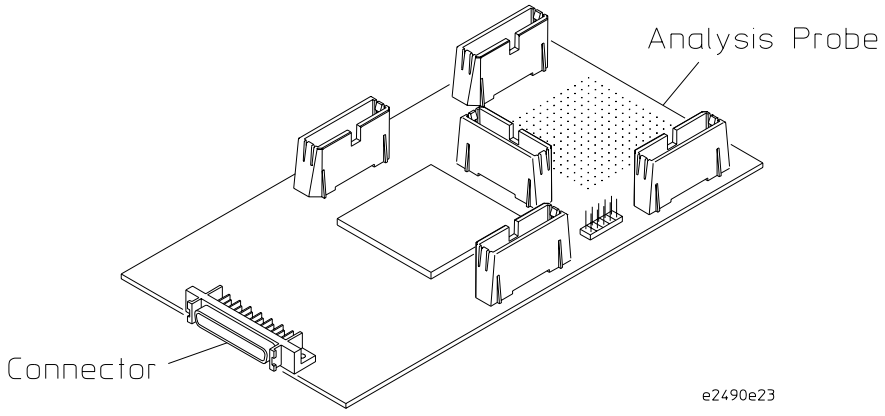
See Also

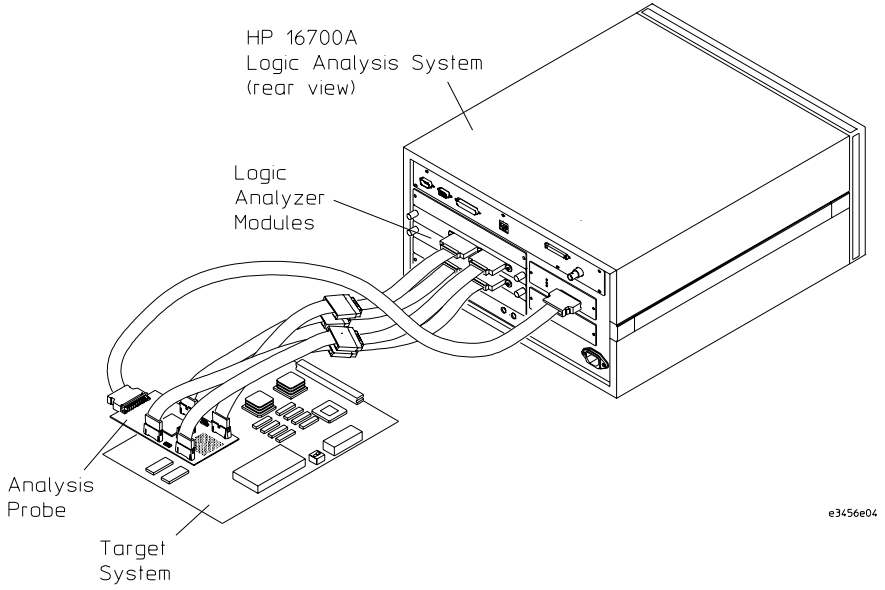
For information on designing a debug port on your target board, see "Debug Port Connection (Optional)" on page 32.

For a list of the parts supplied with the emulation module, see "Emulation Module" on page 21.

To connect to the 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 connector on the analysis probe.

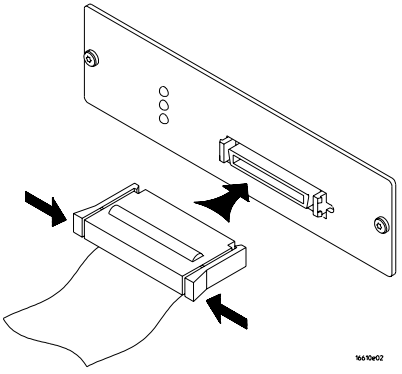

Analysis Probe
Connector
e2490e23


HP 16700A
Logic Analysis System
(rear view)
Logic
Analyzer
Modules
Analysis
Probe
Target
System
e3456e04

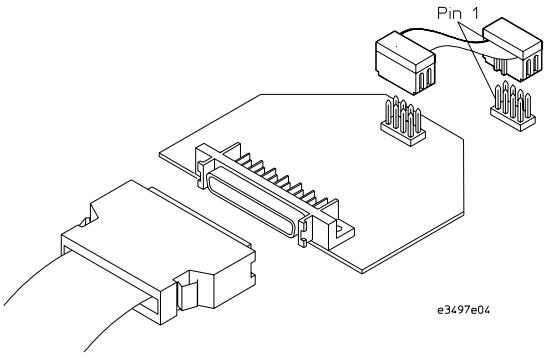
To connect to a target system debug port

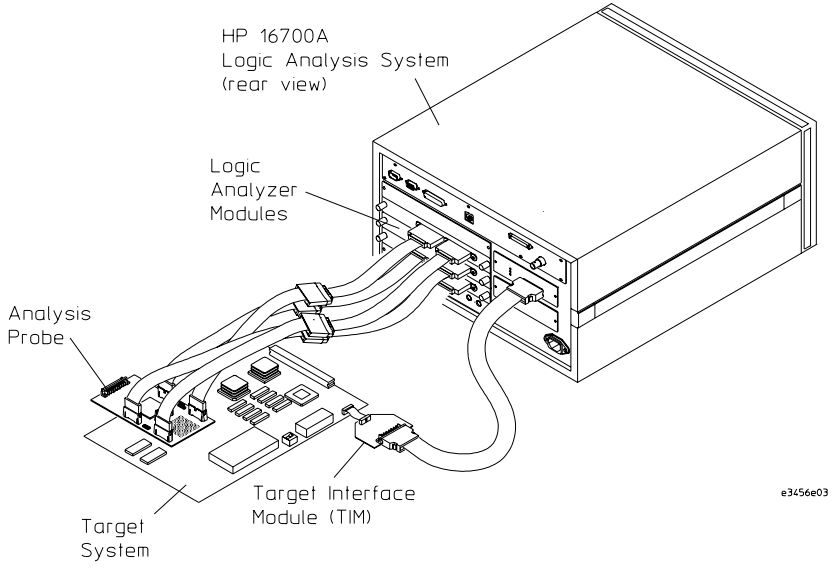
The emulation module can be connected to a target system through a 10-pin debug port (BDM connector). The emulation module should be connected to a 10-pin male 2x5 header connector on the target system using the 10-conductor cable assembly provided.

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 10-pin cable into the target interface module.
5. Plug the other end of the 10-pin cable into the debug port on the target system.





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

See Also

For information on designing a target system for use with the emulation module, see “Preparing for the Emulation Module” on page 32.

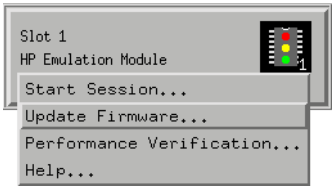
To update emulation module firmware

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 target interface module (TIM).
- The emulation module was not shipped already installed in the logic analysis system.
- You have an updated version of the firmware from HP.

To update the firmware:

1. End any Emulation Control Interface 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 For instructions on how to install the firmware files on the hard disk, see “Installing Software” on page 44.

To display the emulation module firmware version information

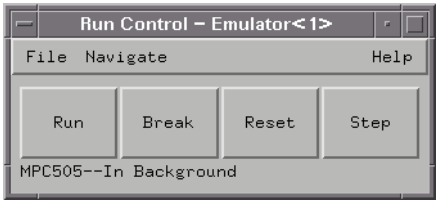
- In the Update Firmware window, click Display Current Version.

  There are usually two firmware version numbers: one for “Generics” and one for the personality of your processor.

To verify communication with the 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:



Part 2

Using the Logic Analyzer

5

Configuring the Logic Analyzer

Configuring the Analysis Probe

Before performing logic analysis measurements on your target system, the analysis probe must be configured in two ways:

- The analysis probe configuration DIP switches must be set to select the type of target system microprocessor (MPC505 or MPC509) and, perhaps, to provide a pullup resistor to the RESET pin.
- The analysis probe must be programmed (by the emulation module) for address reconstruction.

To set the analysis probe switches

The MPC505 and MPC509 use different polarities for the DS pin. The microprocessor select switches on the HP E2490A analysis probe compensate for this difference, so that consistent data is sent to the logic analyzer.

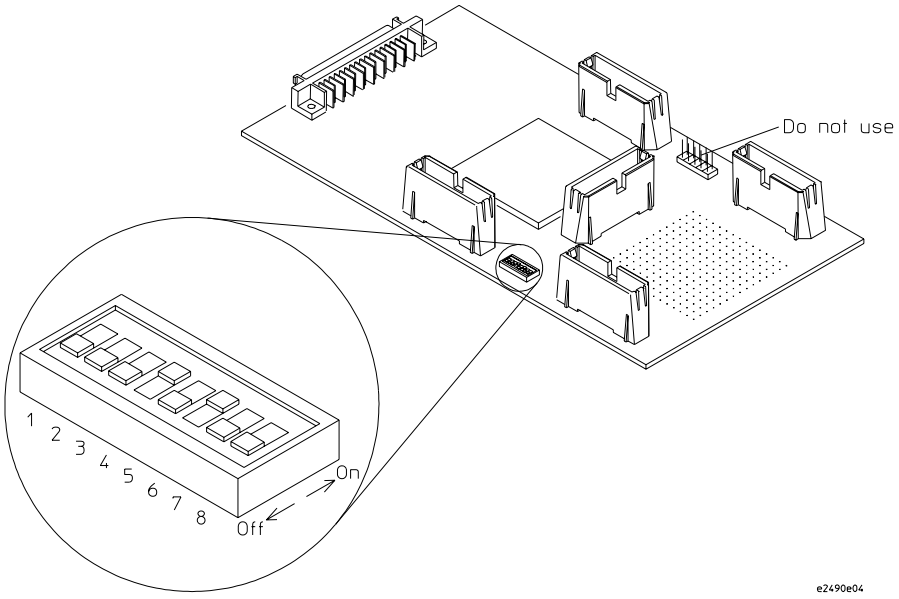
Switches 4 and 6 select the microprocessor. For a MPC505 target system, switches 4 and 6 must be ON. For a MPC509 target system, switches 4 and 6 must be OFF.

Switch 3 provides a 1.78 Kohm pull-up resistor from the target system RESET pin to a 3.3 V source. You can use this circuit if your target system has difficulty breaking from a run control-initiated RESET operation. To implement this circuit, set switch 3 to the ON position.

All other switches are reserved and must be left in the OFF position.

If you have to use the pull-up resistor for RESET, you might want to re-evaluate your design since it may perform differently once the analysis probe is removed.

The figure below shows the microprocessor select switch set for a MPC505 target system, with the pullup resistor disconnected.



To program for address reconstruction

The emulation module is used to configure the HP E2490A analysis probe for address reconstruction. To connect the emulation module to the analysis probe, use the procedure in “To connect to the analysis probe” on page 79.

Use the emulation module to configure the analysis probe as follows:

1 If the target system contains initialization code, run the target system until initialization of the following registers is complete:

- Option registers.
- Base Address registers.
- MEMMAP register.

The target processor can be stopped by using a software breakpoint or with the “break” button.

2 If you are using the Emulation Control Interface in the HP 16600A/16700A-series logic analysis system, use the following steps:

   a Press the “Read Configuration” button in the emulation module Configuration window.

   b Press the “Load Analysis Probe” button in the Configuration window. The analysis probe is now ready for address reconstruction.

   If your target system does not contain initialization code, manually modify the register values in the “Configuration” window according to your target system specification. Then, repeat Step 2 above.

3 If you are using a debugger as the user interface, once the registers are configured, telnet to the emulation module and perform “sync proc” and “pp load.” This will place information needed by the analysis probe for configuration in the analysis probe non-volatile memory.

Loading Configuration Files

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 and whether you are using the state or timing analysis mode.

The MPC5xx inverse assembler decodes captured data into instruction pointer (IP) addresses (also known as software addresses) and assembly language mnemonics.

To load configuration files (and the inverse assembler)

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/mpc5xx/ exists.

> If the above directory does not exist, you need to install the MPC5XX Processor Support Package. Close File Manager, then use the procedure on the CD-ROM jacket to install the MPC5XX Processor Support Package before you continue.

2 Using File Manager, select the configuration file you want to load in the /hplogic/configs/hp/mpc5xx/ 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 MPC505/509 analysis by loading

the appropriate MPC505/509 configuration file. Loading state configuration files 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.

3 Close File Manager.

Logic Analyzer Configuration Files

Analyzer Model	Analyzer Description (full/half channels)	State Configuration File	Timing Configuration File
HP 16550A (one or two cards)	100 MHz state 250/500 MHz timing 4K/8K samples	C505IA1 (one card) C505IA2 (two cards)	C505T1 (one card) C505T2 (two cards)
HP 16554A (two or more cards)	70 MHz state 125/250 MHz timing 0.5M/1M samples	C505IA3	C505T3
HP 16555A/D (two or more cards)	110 MHz state 250/500 MHz timing 2M/4M samples (D)	C505IA3	C505T3
HP 16556A/D (two or more cards)	100 MHz state 200/400 MHz timing 2M/4M samples (D)	C505IA3	C505T3
HP 16557D (two or more cards)	135 MHz state 250/500 MHz timing 2M/4M samples	C505IA3	C505T3
HP 16600A	100 MHz state 125/250 MHz timing 64K/128K samples	C505IA2	C505T2
HP 16601A	100 MHz state 125/250 MHz timing 64K/128K samples	C505IA2	C505T2
HP 16602A	100 MHz state 125/250 MHz timing 64K/128K samples	C505IA1	C505T1
HP 16710A (one or two cards)	100 MHz state 250/500 MHz timing 8K/16K samples	C505IA1 (one card) C505IA2 (two cards)	C505T1 (one card) C505T2 (two cards)

Logic Analyzer Configuration Files			
Analyzer Model	**Analyzer Description (full/half channels)**	**State Configuration File**	**Timing Configuration File**
HP 16711A (one or two cards)	100 MHz state 250/500 MHz timing 32K/64K samples	C505IA1 (one card) C505IA2 (two cards)	C505T1 (one card) C505T2 (two cards)
HP 16712A (one or two cards)	100 MHz state 250/500 MHz timing 128K/256K samples	C505IA1 (one card) C505IA2 (two cards)	C505T1 (one card) C505T2 (two cards)

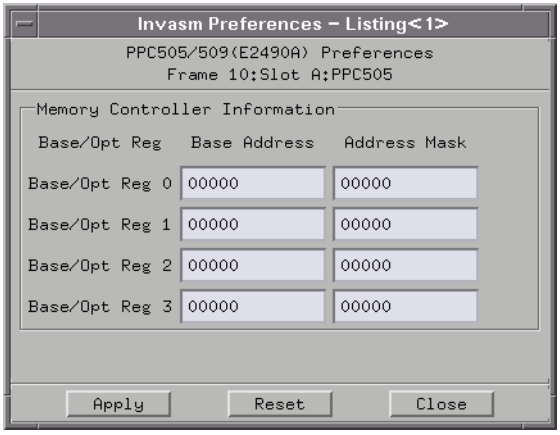
To set the inverse assembler preferences

The Preferences menu allows you to specify an address range for memory banks 0 through 3.

The Base Address field specifies the seventeen most significant bits of the base address (only the most significant bit of the least significant nibble is used).

The Address Mask creates a range within that base address by requiring a match or allowing a don't care for each bit within that base address. A "1" in the Address Mask requires a match, a "0" indicates a don't care which allows a range.

The figure below shows the Preferences dialog.



Loading Symbol Information

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.

To view predefined symbols for the MPC505/509

User-defined symbols are symbols you create in the logic analyzer 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 logic analyzer configurations. The logic analyzer configuration files included with the MPC505/509 analysis probe contain predefined symbols for logic analyzer labels.

To display the predefined symbols for the MPC5xx:

1. Open the logic analyzer's Setup window.
2. Select the Symbols tab.
3. Select the User Defined Symbols tab.
4. Choose a label name from the "Label" list.

   The logic analyzer will display the symbols associated with the label.

Predefined Symbols Description		
Label	**Encoding**	**Symbol**
GDS-	0	CYCLE TERMINATION
	1	
AACK-	0	ADDR ACK
	1	
GAT0-1	00	USER DATA SPACE
	01	USER INST SPACE
	10	SUP DATA SPACE
	11	SUP INST SPACE
BDIP-	0	BURST IN PROGRESS
	1	
BI-	0	BURST INHIBIT
	1	
B16/32	0	16-BIT
	1	32-BIT
GBRST-	0	BURST
	1	
CBOOT-	0	CSBOOT
	1	
PDWU	0	
	1	POWER DOWN WAKE UP
RESET-	0	RESET
	1	
SHOCY-	0	SHOW CYCLE
	1	
TA-	0	XFER ACK
	1	
TEA-	0	XFER ERR ACK
	1	

Predefined Symbols Description		
Label	**Encoding**	**Symbol**
TS-	0	XFER START
	1	
GCT0-3	0000	NORMAL BUS CYCLE
	0001	CHG OF FLOW/RES
	0010	EMUL MEM SELECT
	0011	PRU SELECT
	0100	I-MEM
	0101	L-MEM
	0110	E-MEM NO CS
	0111	INTERNAL REGISTER
	1000	E-MEM CSBOOT
	1001	E-MEM CS1
	1010	E-MEM CS2
	1011	E-MEM CS3
	1100	E-MEM CS4
	1101	E-MEM CS5
	1110	RESERVED
	1111	RESERVED

To load 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. The object file containing symbolic debug information must be in a format the logic analyzer understands.

If your compiler generates object files in a format that the logic analyzer doesn't understand, you can use a General Purpose ASCII (GPA) symbol file (see the "General-Purpose ASCII (GPA) Symbol File Format" chapter on page 251).

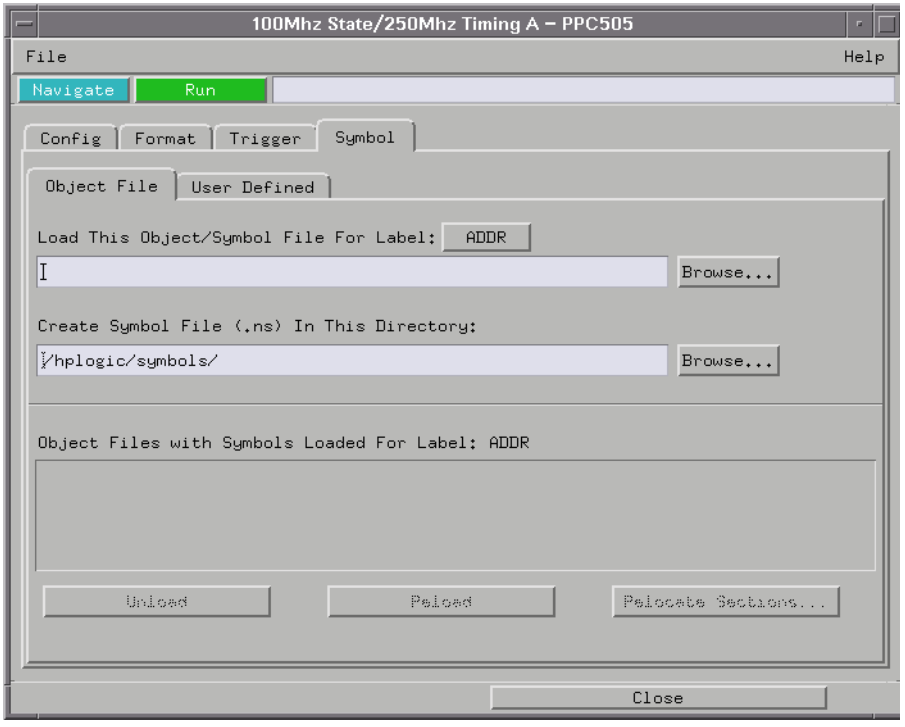
To load symbols in the HP 16600A/16700A-series logic analysis system:

1. Open the logic analyzer module's Setup window.
2. Select the Symbol tab.

3 Select the Object File tab.

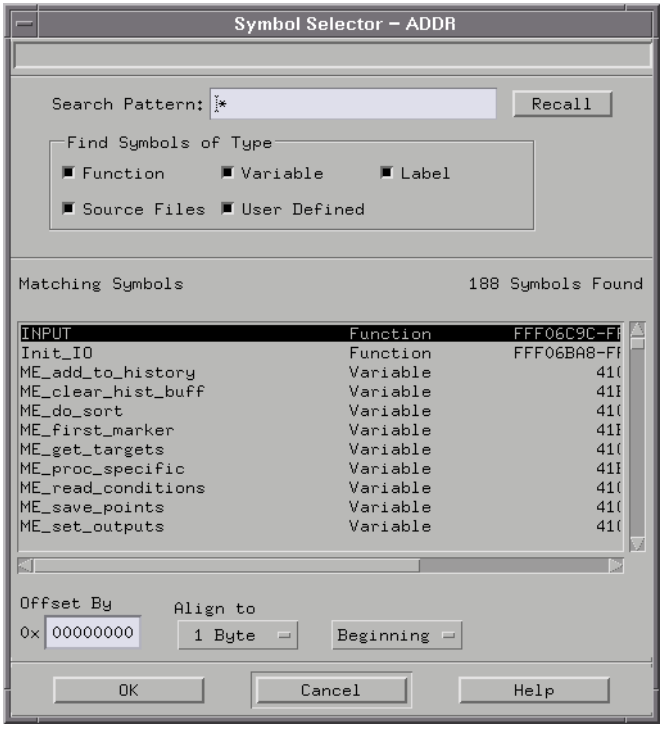
Make sure the label is ADDR.

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 defining trigger patterns, trigger ranges, and so on.


Symbol Selector – ADDR
Search Pattern: *
Recall
Find Symbols of Type
Function
Variable
Label
Source Files
User Defined
Matching Symbols
188 Symbols Found
INPUT Function FFF06C9C-Ff
Init_IO Function FFF06BA8-Ff
ME_add_to_history Variable 41(
ME_clear_hist_buff Variable 41E
ME_do_sort Variable 41(
ME_first_marker Variable 41E
ME_get_targets Variable 41(
ME_proc_specific Variable 41E
ME_read_conditions Variable 41(
ME_save_points Variable 41(
ME_set_outputs Variable 41(
Offset By
0x 00000000
Align to
1 Byte
Beginning
OK
Cancel
Help

Changing the Analysis Mode

The PowerPC 505/509 analysis probe can be used in three different analysis modes:

- State-per-clock.
- State-per-ack.
- Timing.

Inverse assembly is available in the state analysis modes.

To change to state-per-clock analysis

In state-per-clock mode, every microprocessor clock cycle is captured by the logic analyzer, including idle and wait states. (This is the default mode set up by the configuration files.)

Because the VF/VFLS and WP pins are valid on every rising edge of the clock, use the state-per-clock mode to trigger, or to view activity, on these pins.

To configure the logic analyzer for state-per-clock mode:

1 If you are currently using the timing mode connection, change to use the state mode connection (see "Connecting the Analysis Probe to a Logic Analyzer" on page 54).

One of the analysis probe's high-density connectors (J6) contains unbuffered signals for timing analysis. There is an equivalent high-density connector (J4) with buffered signals for state analysis.

2 Load the appropriate logic analyzer configuration file (see "Loading Configuration Files" on page 91).

The state configuration files set up the rising edge of the J clock (J↑) as the master clock signal.

You can change the master clock setting by opening the logic analyzer's

Setup window, selecting the Format tab, and clicking the Mast Clk button to open the master clock dialog.

To change to state-per-ack analysis

In state-per-ack mode, the microprocessor clock is qualified so that only address and data-acknowledge cycles are captured.

This mode will capture show cycles and external bus cycles when DS-, TA-, or TEA- are true (K=0 for any of these events).

To configure the logic analyzer for state-per-ack mode:

1. If you are currently using the timing mode connection, change to use the state mode connection (see "Connecting the Analysis Probe to a Logic Analyzer" on page 54).

   One of the analysis probe's high-density connectors (J6) contains unbuffered signals for timing analysis. There is an equivalent high-density connector (J4) with buffered signals for state analysis.

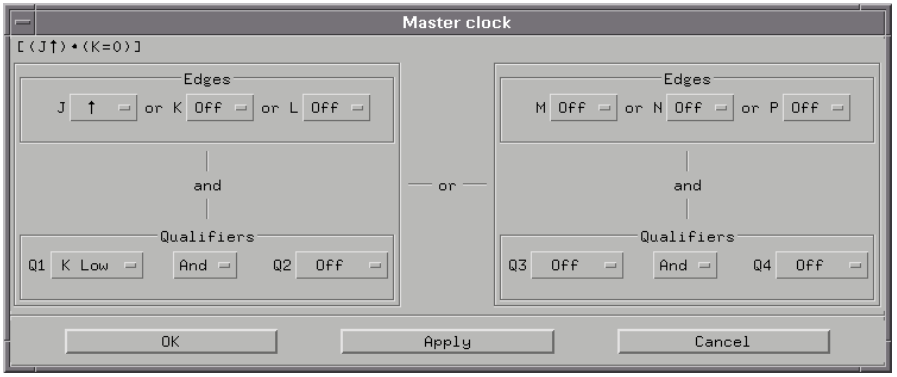
2. Load the appropriate logic analyzer configuration file (see "Loading Configuration Files" on page 91).

3. Open the logic analyzer's Setup window.

4. Select the Format tab.

5. Click the Mast Clk button to open the master clock dialog.

6. Select the rising edge of the J clock AND a low K clock (J↑ · (K=0)), and click OK.



To change to timing analysis

In timing mode, the logic analyzer samples the microprocessor pins asynchronously, according to an internal, adjustable sample rate clock. The minimum sample period for a 250 MHz timing analyzer is 4 ns.

To configure the logic analyzer for timing analysis:

1. If you are currently using the state mode connection, change to use the timing mode connection (see "Connecting the Analysis Probe to a Logic Analyzer" on page 54).

   One of the analysis probe's high-density connectors (J4) contains buffered signals for state analysis. There is an equivalent high-density connector (J6) with unbuffered signals for timing analysis.

2. Load the appropriate logic analyzer configuration file (see "Loading Configuration Files" on page 91).

6

Capturing MPC505/509 Execution

The normal steps in using the logic analyzer are:

1. Configure the logic analyzer.
2. Format labels for the logic analyzer channels (that is, mapping logic analyzer channels to target system signal names).
3. Load symbols from the program's object file.
4. Set up the trigger, and run the measurement.
5. Display the captured data.

With the MPC5xx analysis probe, the logic analyzer is configured, and labels are created (formatted) for the logic analysis channels when configuration files are loaded (see "Loading Configuration Files" on page 91).

You can load program object file symbols into the logic analyzer when configuring it (see "Loading Symbol Information" on page 95).

This chapter describes setting up logic analyzer triggers when using the HP E2490A analysis probe and the HP B4620B source correlation tool set.

See the "Displaying Captured MPC505/509 Execution" chapter on page 113 for information on displaying captured data.

Setting Up Logic Analyzer Triggers

State-Per-Clock Mode Considerations

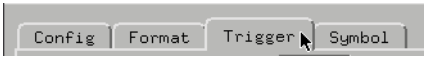
The analysis probe aligns (de-pipelines) the address and data phases of the MPC50X processor bus cycles. The address and data phases are synchronous when the K Clk qualifier is low (K=0). Any triggering sequence that is set up to capture valid cycles must have the term K=0.

The WP[0:5], VFLS[0:1], and VF[0:2] pins are valid on the rising edge of every clock. When you are using these pins for triggering, storing, or counting, no clock qualification is necessary.

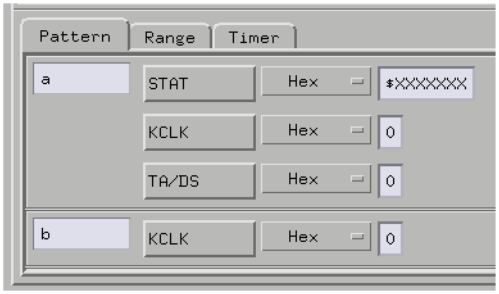
To set up logic analyzer triggers

1 Open the logic analyzer's Setup window.

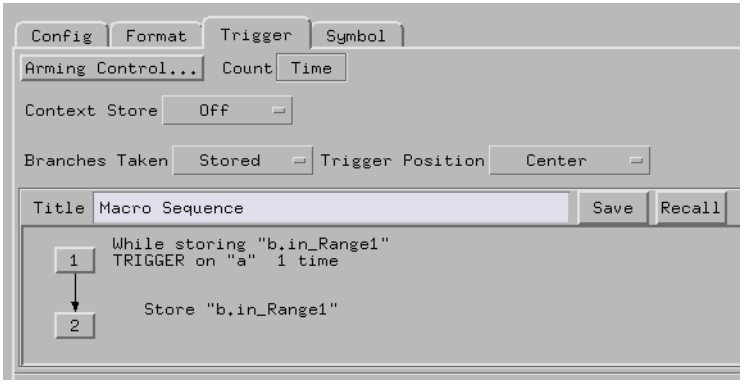
2 Select the Trigger tab.



3 Define the patterns, ranges, and other resources that will be used in the logic analysis measurement.



4 Set up the trigger sequence.

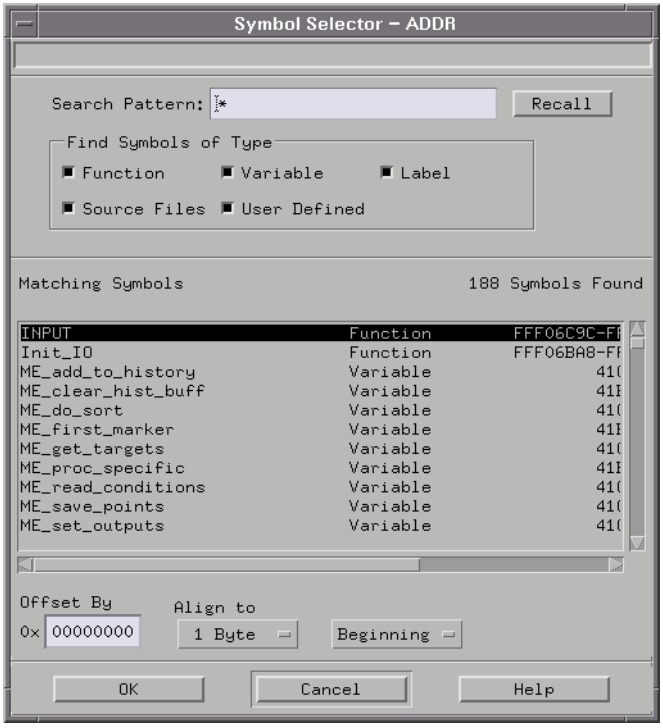


5 Run the measurement.

See Also The HP 16600A/16700A-series logic analysis system's on-line help for more information on setting up logic analyzer triggers.

To compensate for relocated code

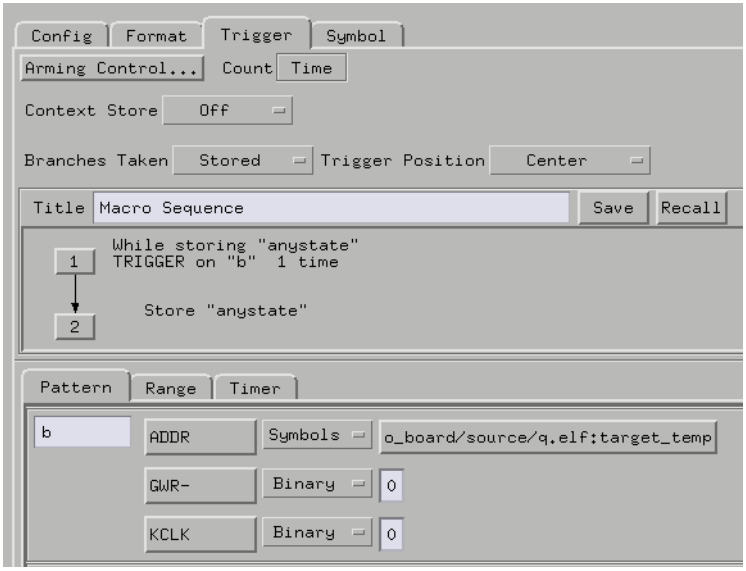
When code segments are relocated, or when memory management units produce fixed code offsets, you can compensate by using the address offset field in the Symbol Selector dialog.



Entering the appropriate address offset will cause the logic analyzer to reference the correct symbol information for the relocatable or offset code.

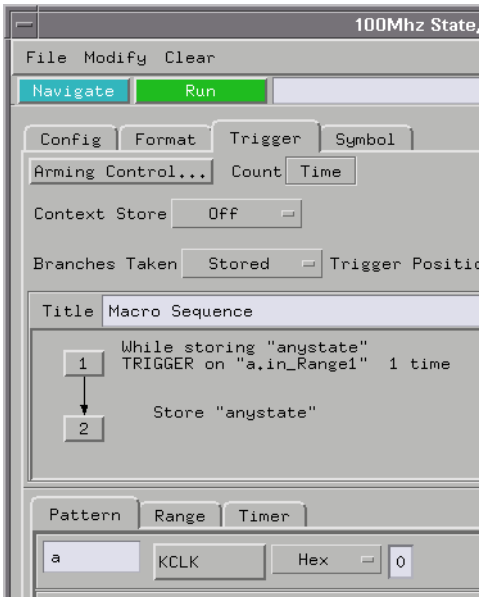
To capture valid write cycles in state-per-clock mode

In the example below, the term “b” consists of an address specified by the symbol target_temp, GWR- = 0, and KCLK = 0. When these conditions are all true, the logic analyzer triggers and the data is stored.

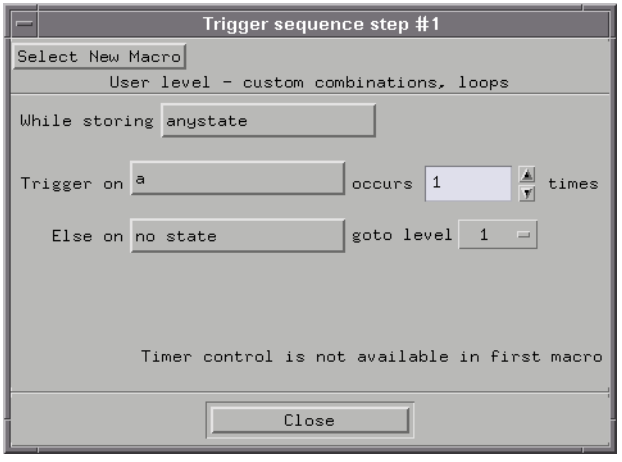


To trigger on an address range access in state-per-clock mode

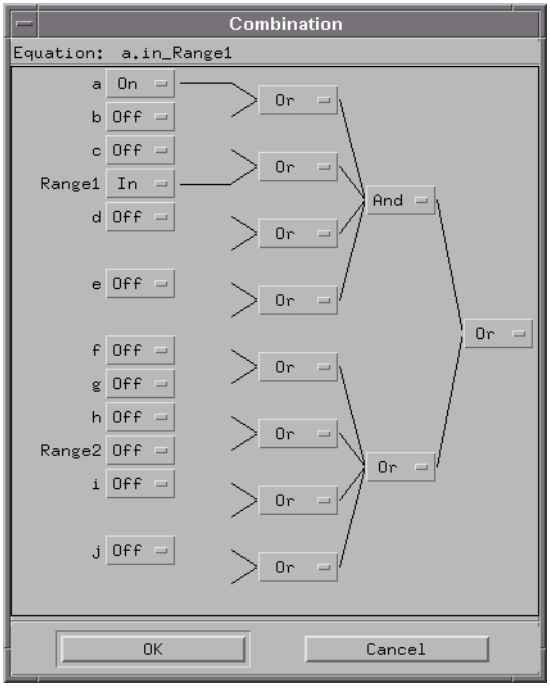
To trigger on an access to an address range, the triggering term must consist of in_range ANDed with K=0. The following illustration shows the Trigger window with this term specified. The illustrations afterwards show how to configure this term.



To configure this trigger sequence, click on sequence Step 1. The following dialog box appears.



Click on the “Trigger On” button, then select “Combo” in the pop-up menu. In the Combination window, set the terms as shown below.



Triggering on Source Code

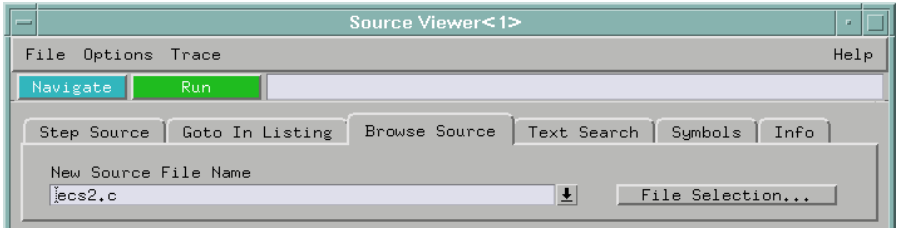
When setting up trigger specifications to capture MPC5xx execution:

- 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).

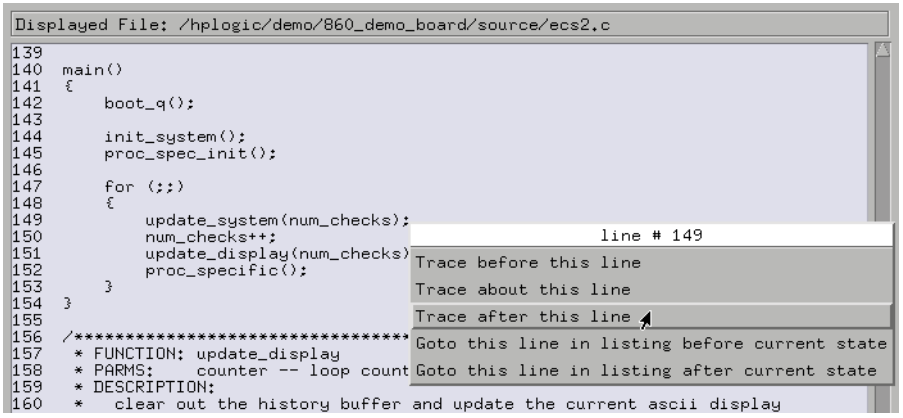
To set up triggers based on source code

1 Open the Source Viewer window.

2 Browse the source file that contains the code you want to trigger on.



3 Click the source code line you want to trigger on and specify whether you want to trace before, about or after the line. Or, use the Source Viewer's Trace menu to trace about a variable, function, or line number.



4 Run the measurement.

To avoid capturing library code execution

When viewing the source code associated with captured data, the source correlation tool set can exhibit long response times 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.

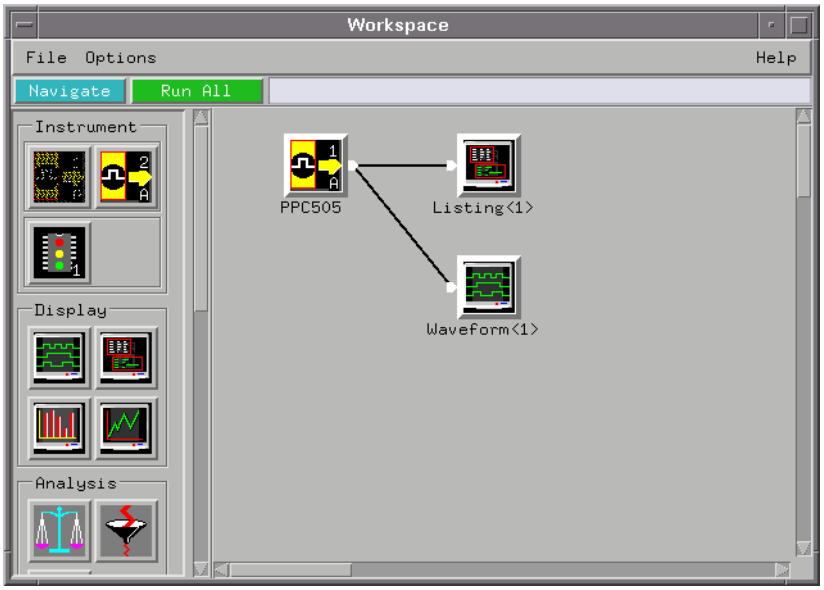
You should also configure the logic analyzer's storage qualification capabilities to store only those cycles that correspond to software execution (non-idle, etc.).

7

Displaying Captured MPC505/509 Execution

Displaying Data from the Logic Analyzer

When you load the configuration files supplied with the inverse assembler, the Workspace window is set up as shown below.

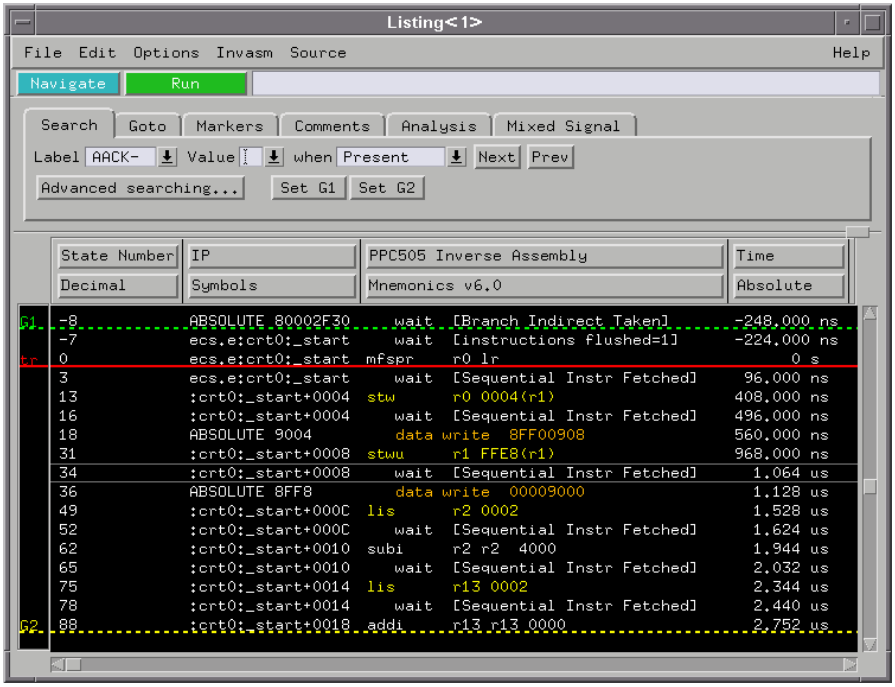


This setup is appropriate for debugging hardware. When you're debugging software, you'll probably want to use the trace reconstruction tool (see "Displaying Data from the Trace Reconstruction Tool" on page 118).

To display the captured state data

1 Open the Listing display window.

The logic analyzer displays captured state data in the Listing display.



The inverse assembler is already loaded when state configuration files are loaded, but it can also be loaded into a Listing display using the Invasm menu.

Because the PowerPC 505/509 presents one address and then reads 16

bytes during a burst, the least-significant digit is synthesized by the disassembler. The entire synthesized address appears under the "IP" label. The actual address bits presented by the PowerPC 505/509 may be observed under the TADDR label (used for timing analysis).

The HP E2490A analysis probe reconstructs and displays the address for all Show cycles:

- Data Show cycles are supported for all processor operating modes supported by the MPC50X processors. This includes the following:

  ✓ All memory accesses.

  ✓ Writes to memory. Reads from memory are not displayed.

  For Data Show cycles, the data is also reported along with the reconstructed address.

- Instruction Show cycles are supported for all processor operating modes supported by the MPC50X processors. This includes the following:

  ✓ All fetch cycles.

  ✓ All change-of-flow (direct and indirect) cycles.

  Instruction Show cycles do not have data associated with them, so only the address is shown.

- Any Show cycles reporting internal accesses to the L-MEM block or to memory-mapped internal registers are also reported, with the accessed address displayed.

> For the inverse assembler to work properly, the CR/DS must be configured for DS mode. For information on configuring this bit, refer to "Boot Code for Inverse Assembly" on page 30.
>
> If the CR/DS pin must be configured in CR mode, show cycle information captured in the trace is still valid.

See Also

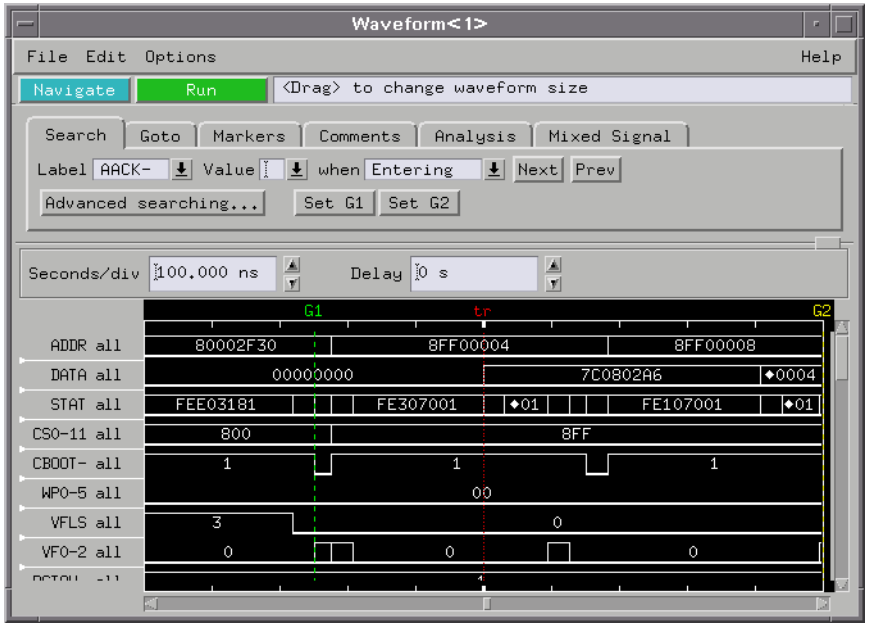
"To use the inverse assembler filters" on page 125 for information on displaying or hiding certain types of microprocessor bus cycles.

The HP 16600A/16700A-series logic analysis system on-line help for information on using the Listing display.

To display captured timing analysis mode data

- Open the Waveform display for your logic analyzer.

You can also use the Waveform display in the state-per-clock analysis mode to display state timing diagrams.



Displaying Data from the Trace Reconstruction Tool

The trace reconstruction tool uses show cycles, the VF/VFLS signals, and program image files to decode captured MPC505/509 execution into complete program traces. When compared to data that comes straight from the logic analyzer, the data from the trace reconstruction tool:

- Contains code that executes out of internal memory.
- Has unexecuted prefetches removed.
- Shows the actual execution times of instructions.

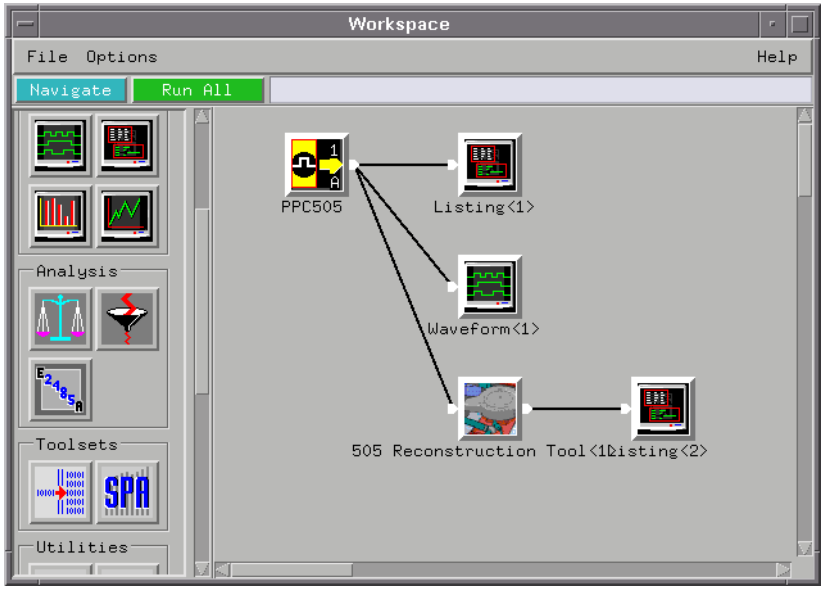
The data from the trace reconstruction tool contains only code that has been executed by the microprocessor. Read and write cycles captured by the logic analyzer are unchanged.

The trace reconstruction tool is appropriate for debugging software. When you're debugging hardware, you'll probably want to look at the actual data captured by the logic analyzer (see "Displaying Data from the Logic Analyzer" on page 114).

NOTE: Don't debug hardware based on the ADDR, DATA, and STAT values output by the trace reconstruction tool because they don't represent the actual signal values captured by the logic analyzer.

To add the trace reconstruction tool to the workspace

1. Open the Workspace window.
2. Drag the 505 Reconstruction Tool from the tool box and drop it on the logic analyzer instrument.
3. Drag a Listing display tool and drop it on the 505 Reconstruction Tool.

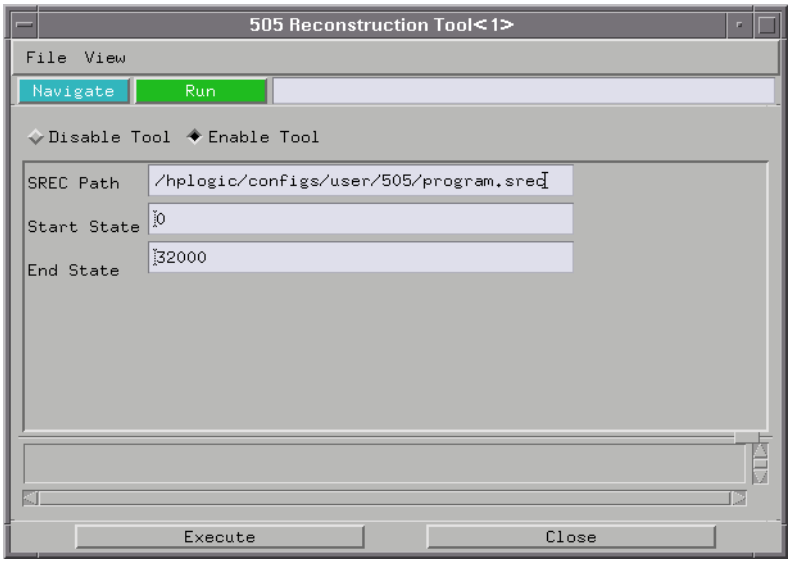


To load program image files into the trace reconstruction tool

Before the trace reconstruction tool can work, you must give it a program image file in S-record format.

1 In the Workspace window, double-click the 505 Reconstruction Tool.

2 Enter the name of your program image S-record file.



3 Enter the starting and ending states of the range to be reconstructed.

Limiting the number of states to be reconstructed results in faster displays.

4 Click Execute.

To use multiple program image files

You can use multiple image files by concatenating the program image files for different parts of the target system code and loading the resulting combined image file.

If undefined opcodes appear in the output

If "Undefined Opcode"s appear in the output of the trace reconstruction tool, it means the opcode for that address could not be found in the program image file.

When this happens, the decoded instructions that follow, up to the next show cycle, may not be correct.

❏ Make sure that all program image files are loaded.

If the trace contains all wait states

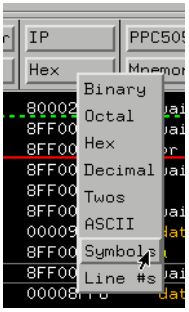
The trace reconstruction tool needs a show cycle to sync up with before it can start decoding data. If there is no show cycle, all data, except reads and writes, are shown as wait states.

Using the Listing and Source Viewer Displays

The Listing and Source Viewer displays (as well as other display tools) can be used with either data straight from the logic analyzer or with data that comes from the trace reconstruction tool.

To display symbols

- Over a Listing display's label base, right-click the mouse button, and select Symbols.



Any symbols that have been defined will be displayed for equivalent captured values.

See Also "To load object file symbols" on page 97.

To interpret the inverse assembled data

The next few paragraphs describe the general output format of the logic analyzer inverse assembler. For information on PowerPC 505/509 modes of operation, refer to "Unsupported Microprocessor Modes" on page 27.

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 0044(r1).”

Bit numbers and shift counts are displayed in decimal with a dot suffix, for example, “cror 31. 31. 31.”

A few instructions display their operands in binary with a “%” prefix, for example, “mtfsfi 4 %0101.”

The disassembler decodes the full PowerPC instruction set architecture, including 64-bit mode instructions and optional instructions not implemented on the PowerPC 505/509. 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”.

Endian Mode

The inverse assembler only supports big-endian mode. Little-endian mode is not supported.

Burst Mode

The analysis probe reconstructs the least significant bits of burst accesses, so that every beat of the burst access has the correct address associated with it. Burst mode is supported for both 16-bit and 32-bit memory ports.

Pipelined Data

For pipelines accesses, the address phase and data phase of the pipelined data are aligned by the analysis probe hardware.

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.

Extended Mnemonics

PowerPC assemblers support a number of extended mnemonics for some popular assembly language instructions as described in the *PowerPC 505/509 User's Manual*. The E2490A analysis probe's disassembler supports the following extensions:

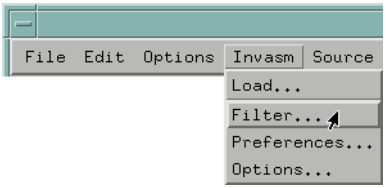
- Conditional traps and branches decode the condition mnemonically when possible. For some conditions which have no conventional mnemonics (for example, "signed less than or unsigned greater than"), the condition field is displayed in binary.
- The L bit is omitted as a compare operand. Instead, compares 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.

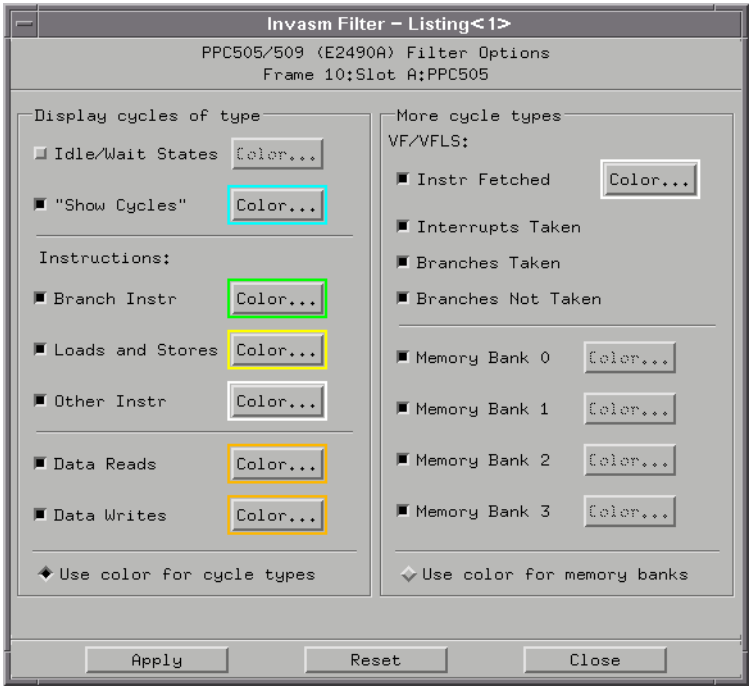
- the cror, crnor, crxor, and creqv instructions map analogously to crmv, crnot, crclr, and crset.
- when the mtcrf instruction field mask specifies the entire cr, it is decoded as mtcr.
- The PowerPC rotate-left instructions have extended mnemonics. The following listing shows the extended mnemonics 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)

To use the inverse assembler filters

- In the Listing display window, choose the Filter command from the Invasm menu.





The inverse assembler filtering options allow you to display or hide certain types of microprocessor bus cycles or memory bank accesses.

Because the filter options do not affect the data that is stored by the logic analyzer (they only affect whether that data is displayed), they let you display the same data in different ways.

Filtering allows faster analysis in two ways:

- Unneeded information can be taken out of the display. For example, suppressing idle states will show only states in which a transaction was completed.
- Particular operations can be isolated by suppressing all other operations. For example, Branch Instr can be shown, with all other states suppressed, allowing quick analysis of branch instructions.

You can also use color to distinguish between cycle types or memory bank accesses (when they are displayed). Color can be used for distinguishing between memory bank accesses or cycle types, but not both at the same time.

You can display or hide the following types of cycles:

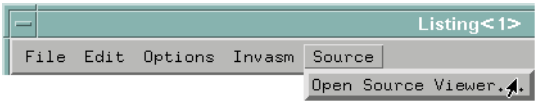
- Idle/Wait States - A state in which there is no valid instruction, data read/write, show cycle, or VF/VFLS activity.
- Show Cycles - (For additional information on the type of information that is displayed for Show Cycles, see "To display the captured state data" on page 115).
- Branch Instructions.
- Loads and Stores.
- All Other Instructions - (Besides branches, loads, and stores).
- Data Reads.
- Data Writes.
- Instructions Fetched - These are VF/VFLS states that are the result of an instruction fetch. The instruction code executed will be hidden/displayed.
- Interrupts Taken - These are VF/VFLS states that are the result of an interrupt or an exception. The inverse assembler will also attempt to decode the type of interrupt based on the interrupt vector address. The interrupt code executed will be hidden/displayed.
- Branches Taken - These are VF/VFLS branches.
- Branches Not Taken - These are any VF/VFLS conditional branches NOT taken.
- Memory Bank 0-3 accesses.

See Also

The address ranges for Memory Banks 0-3 are specified in the Preferences menu (see "To set the inverse assembler preferences" on page 94).

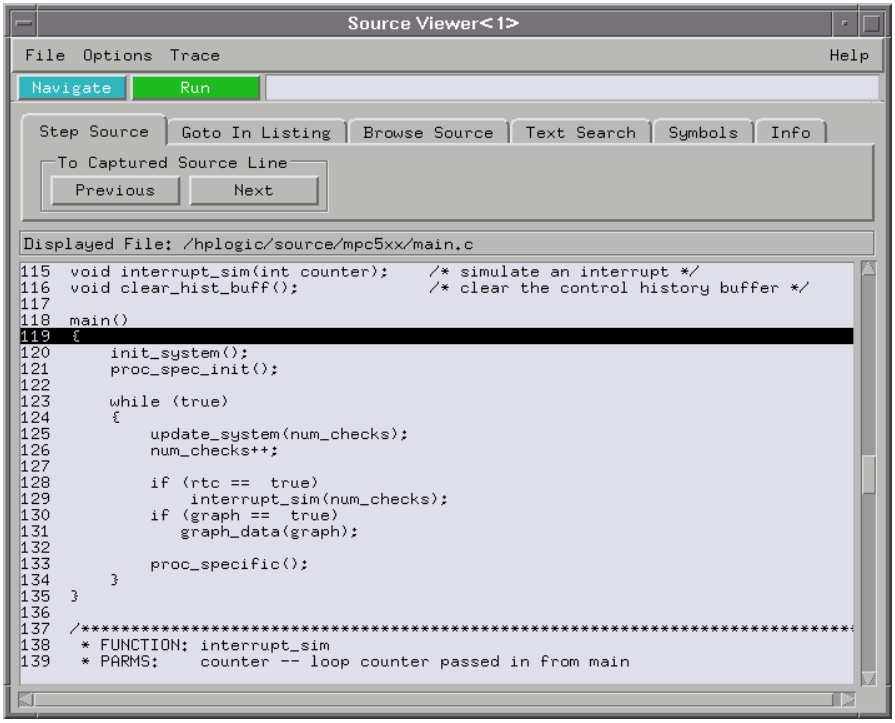
To view the source code associated with captured data

- In the Listing display window, select Source Viewer from the Source menu.



- Or, open the Source Viewer window from the logic analyzer's icon in the main system window.

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 (see "To load object file symbols" on page 97).



Inverse Assembler Generated IP (Software Address) Label

In the HP 16600A/16700A-series logic analysis system, the MPC5xx inverse assembler generates a “IP” label. The IP 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 IP 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 line of source code will generate many assembly instructions. The “Goto this line in listing” commands will not find a given line of source code 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 set up 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.

See Also

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

8

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 logic analyzer, analysis probe, or target system.

Solving Logic Analyzer Problems

This section lists general problems that you might encounter while using the 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 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” on page 138 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.

Solving 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

There are several general problems that can cause erratic variations in trace lists and inverse assembly failures.

❏ 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” on page 138. 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.

Solving 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 “Connecting the Analysis Probe to a Logic Analyzer” on page 54 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 the "Configuring the Logic Analyzer" chapter on page 87 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; however, it may be incorrect because a portion of the execution trace was not visible to the logic analyzer.

❏ Verify that storage qualification has not excluded storage of all the needed opcodes and operands.

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 the "Configuring the Logic Analyzer" chapter on page 87 for details.

Solving 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 the oscilloscope 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 the oscilloscope, 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.

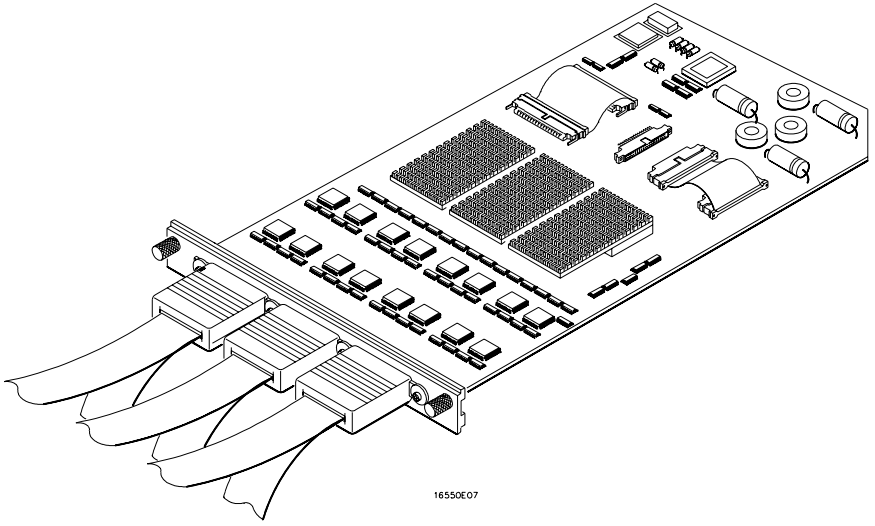
Ensure that the inverse assembler file is not renamed or deleted, and that it is located in the correct directory:

- For HP 16600A/16700A-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.

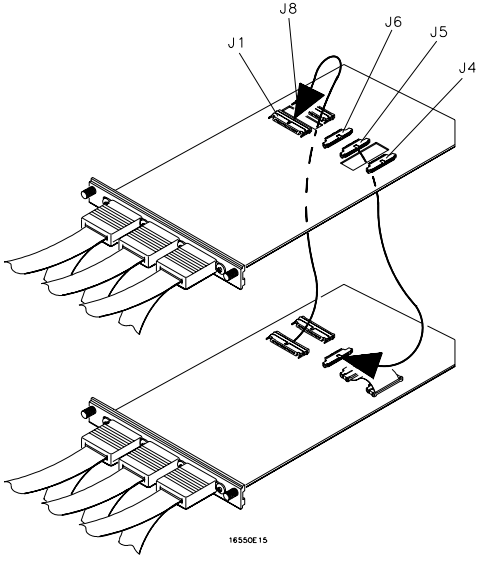
"Measurement Initialization Error"

This error occurs when you have installed the cables incorrectly for one or two HP 16550A logic analyzer cards. The following diagrams show the correct cable connections for one-card and two-card installations. Ensure that your cable connections match the silk-screened labels 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 See the "Configuring the Logic Analyzer" chapter on page 87 for a description of 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 HP 16600A/16700A-series logic analysis system frame or in the HP 16701A expansion 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 "Connecting the Analysis Probe to a Logic Analyzer" on

page 54 to determine the proper connections.

"Time from Arm Greater Than 41.93 ms"

The 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, if the trigger condition is set to look for an opcode fetch at an address not corresponding to a word boundary, the trigger will never be found.

Part 3

Using the Emulation Module

The emulation module and firmware for the Motorola MPC505/509 is also used with the MPC555 processor.

9

Using the Emulation Control Interface

The Emulation Control Interface in your HP 16600A/16700A-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:

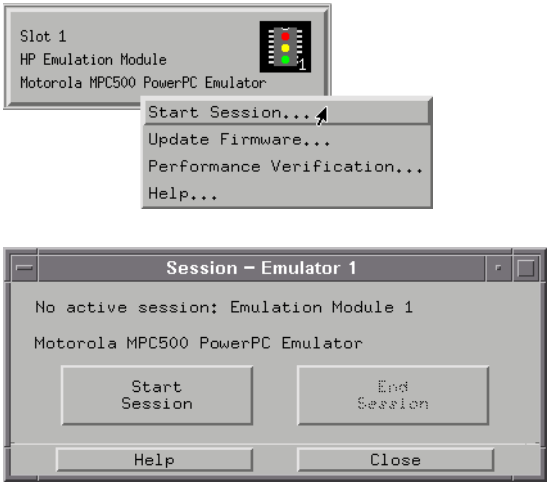
- The interfaces can get out of synchronization when commands are issued from both interfaces. This causes windows to be out-of-date and can cause confusion.
- Most debuggers cannot tolerate another interface issuing commands and may not start properly if another interface is running.

See Also

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

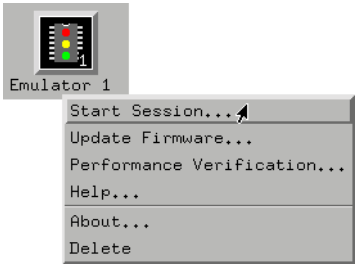
To start from the main System window

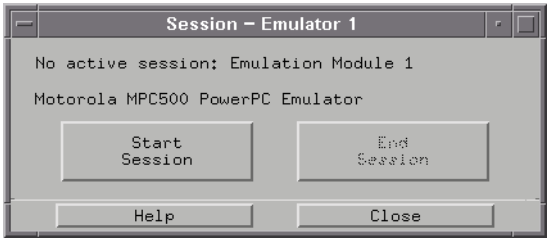
1. In the System window, click the emulation module icon.
2. Select Start Session....



To start 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....





10

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.

Entering Emulation Module Commands

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 on-line 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.

To use the Emulation Control Interface

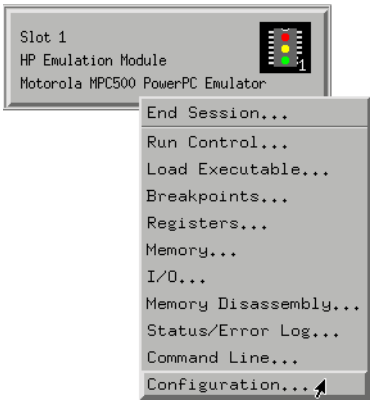
The easiest way to configure the emulation module is to use the Emulation Control Interface.

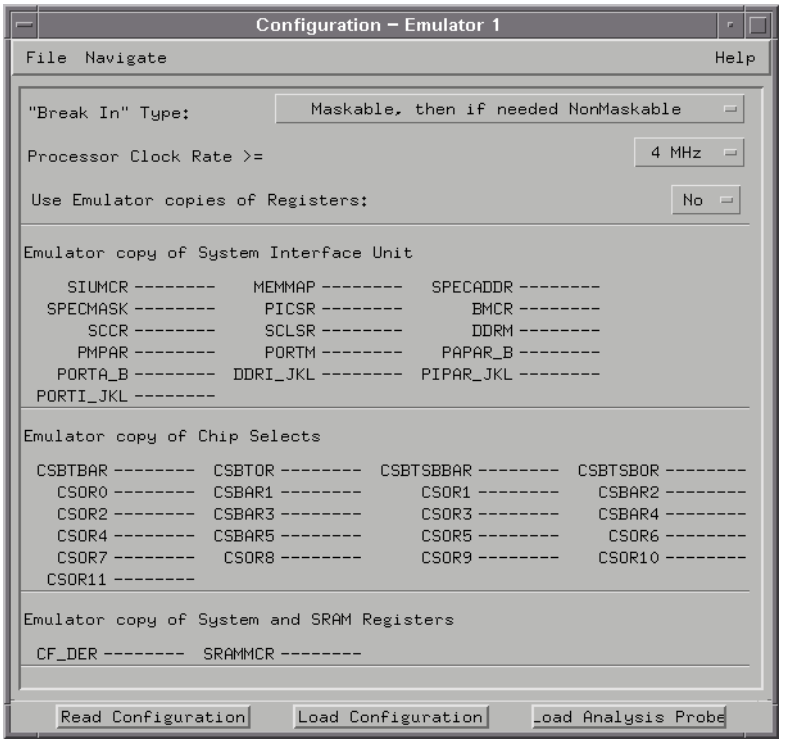
1 Start an Emulation Control Interface session.

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 use the built-in command interface

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 "Built-In Commands" on page 215.

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 ---

  proc     - Set type of target processor
  procck   - Set clock speed of target processor
  dprocck  - Display default clock speed of target processor
  bnchardbrk - Set BNC break type
  breakin  - BNC break in control
  rrt      - Set restriction to real time runs
  trigout  - Trigger out control
M>
```

To use a debugger interface

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 Debuggers (with the Emulation Module)” chapter on page 165.

Your debugger manual.

Setting the MPC5xx Configuration Options

You must configure the emulation module to work with your target system.

The following options can be configured using the Emulation Control Interface or using built-in commands:

- Processor clock speed.
- "Break In" type.

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

- Processor type.
- Restriction to real-time runs.

See Also

See "To program for address reconstruction" on page 89 for information on using the emulation module to configure the analysis probe.

To configure the processor clock speed

The BDM communication speed will be 1/3 of the configured processor clock speed. You may set the processor clock speed to a speed lower than the actual clock speed of your target system.

Processor clock speed configuration		
Value	**Processor clock is at least**	**Built-in command**
25	25 MHz	cf procck = 25
20	20 MHz	cf procck = 20
16	16 MHz	cf procck = 16
8	8 MHz	cf procck = 8
4	4 MHz (default)	cf procck = 4
1	1 MHz	cf procck = 1
512	512 kHz	cf procck = 512
32	32 kHz	cf procck = 32

You can also set the reset clock speed, which controls the BDM communication speed used after a reset, but before the Multiplication Factor in the SCCR is set up:

Reset processor clock speed configuration		
Value	**Processor clock is at least**	**Built-in command**
25	25 MHz	cf dprocck = 25
4	4 MHz	cf dprocck = 4
32	32 kHz	cf dprocck = 32

To configure the "Break In" type

This option affects how the emulation module will react to a trigger in an intermodule measurement.

"Break In" type configuration

Value	What happens when the emulation module is triggered
Maskable	A trigger will immediately cause a maskable break. If the maskable break fails, a non-maskable break will be attempted. The delay between an attempted maskable break and the non-maskable break will allow many instructions to be executed. (Default)
NonMaskable	A trigger will immediately cause a non-maskable break. Use this value if you are trying to halt the processor in an interrupt service routine. The processor may not be able to continue running after the break.

To configure the processor type

Processor type configuration

Value	Emulation module configured for	Built-in command
MPC505	MPC505	cf proc = MPC505
MPC509	MPC509 (Default)	cf proc = MPC509

The cfsave -s command will store the processor type configuration in the emulation module's flash memory. The cfsave -r command will restore this configuration.

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. (Default)	cf rrt = no
yes	No commands are allowed which break to the monitor, except “break,” “reset,” “run,” or “step.”	cf rrt = yes

Testing the Emulation Module 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

See the "Troubleshooting the Emulation Module" chapter on page 211 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 by running a 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.

11

Using Debuggers (with the Emulation Module)

Several companies sell source debuggers which work with the HP emulation module.

Benefits of Using a Debugger

A debugger lets you:

- control (start and stop) the execution of your microprocessor
- step through your code at the source-code level
- set breakpoints
- single-step through source code
- examine variables
- modify source code variables
- download executable code to your target system

Compatibility with Other Logic Analysis System Tools

You can use your logic analysis system to trace and analyze target system execution while you use your debugger.

If the computer running the debugger is also running X Windows server software, you can display logic analyzer windows next to your debugger windows.

Minimum Requirements

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

- A debugger that is compatible with the emulation module.

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

- A LAN connection between the PC or workstation that is running the debugger and the emulation probe or the HP 16600A/16700A-series logic analysis system (which contains the emulation module).

  Emulation probes or emulation modules communicate with debuggers over the LAN.

- To have the logic analysis system user interface displayed on your PC or workstation screen along with the debugger, your computer needs to be running X Windows server software.

  Most UNIX workstations run X Windows server software, but on a PC you may need to install X Windows server software.

Setting Up Debugger Software

The instructions in this section assume that your PC or workstation is already connected to the LAN and that you have already installed the debugger software according its documentation.

To use your debugger with the emulation module:

1. Install the emulation module (see "Installing the Emulation Module" on page 38).
2. Connect the emulation module to your target system (see "Connecting the Emulation Module to the Target System" on page 79).
3. If you are using the debugger with an emulation module in a logic analysis system, you must connect the logic analysis system to the LAN (see "To connect the logic analysis system to the LAN" on page 169).
4. If you want to display logic analysis system windows next to your debugger windows, export the logic analysis system's display to your PC or workstation (see "To view logic analysis system windows next to the debugger" on page 171).
5. Configure the emulation module (see the "Configuring the Emulation Module" chapter on page 153).

   If you use the logic analysis system's 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.

6. Begin using your debugger.

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

See the logic analysis system's installation guide or on-line help for information on setting up a logic analysis system on the LAN.

Debuggers require information about a logic analysis system's LAN connection so they can communicate with an emulation module. They need (write the information here for future reference):

- IP Address of Logic Analysis System _____
- Hostname of Logic Analysis System _____
- Gateway Address _____
- Port Number of Emulation Module _____

Default emulation module port numbers

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 in an HP 16600A/700A-series logic analysis system)
6476	Slot 2 (Second emulation module in an HP 16700A-series system)
6480	Slot 3 (Third emulation module in an expansion frame)
6484	Slot 4 (Fourth emulation module in an expansion frame)

*Port numbers for telnet connections are different than for debugger connections because telnet uses a different service than debuggers, and a telnet port is already set up in order to display the logic analysis system interface remotely.

To change the port number of an emulation module

Some debuggers do not provide a way to specify an emulation module port number. In this case:

- The debugger will always connect to port 6470 (the default port number of an emulation probe, or the port number of the emulation module in slot 1 of an HP 16600A/16700A-series logic analysis system).
- If the port number of the emulation module is not 6470, you must change it.

To view or change the port number of an emulation module:

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 not be 0-1000 and must not already be assigned to another emulation module.

To verify LAN 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

For information on physically connecting the logic analysis system to the LAN and configuring its LAN parameters, see the installation guide or on-line help for your logic analysis system.

To view logic analysis system windows next to the debugger

1 Make sure the computer running the debugger is also running X Windows server software and has telnet software.

2 Give the logic analysis system permission to display on the X Windows server.

3 Connect to the logic analysis system, log in, and start a session, displaying on the X Windows server.

Example, UNIX

On a UNIX workstation:

1. 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, PC

On a Windows 95 PC with Reflection X server software from Walker Richer & Quinn, Inc.:

1. On the PC, start the X Windows server software and connect to the logic analysis system.

   To start Reflection X, click the Reflection X Client Startup icon. Enter the following values in the Reflection X Client Startup dialog:

   a. In the Host field, enter the hostname 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.

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

Using the Green Hills Debugger

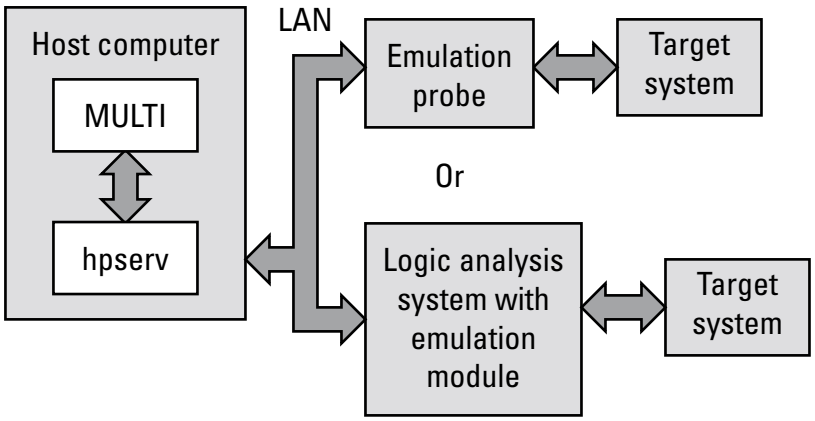
Compatibility

Version 1.8.8.A of the MULTI Development Environment from Green Hills Software, Inc. is one debugger that connects to the HP emulation module.

This information in this section is intended to be used along with the MULTI documentation provided by Green Hills Software.

Overview

MULTI connects to an emulation module through the Green Hills host-resident program (hpserv).



To get started

1 Build the executable.

If you have the demo software shipped with the Green Hills debugger, follow these steps:

a Prepare the executable.

Go to the hpdemo subdirectory where you installed MULTI. Copy the mbx800.lnk file to user.lnk.

You may need to edit the user.lnk file to place the program at a location where target system memory is available.

b Start MULTI.

On Unix, enter "multi".

On Windows, double-click the Green Hills icon.

c Set up the MULTI software environment:

- Replace the project default.bld (in the Builder dialog box next to the project button) with hpdemo/default.bld and press ENTER.
- Make sure the target button on the MULTI window says "PPC".
- In the Builder window, double-click ecs.bld.

The box next to the Debug button should display "ecs". The window should list the names of the source code files.

d In the Builder menu bar, select Options->CPU, then set the processor type.

e In the Builder menu bar, select Options->Advanced, and make sure that "Output DWARF on ELF targets" option is enabled.

f Build the demo program:

- In the Builder window, click the Build icon. (Or, in the menu bar, select Build->Build All.)
- Close the Progress window when the "Build completed" message is displayed.

2 Connect MULTI to the emulation module.

There are two ways to connect to the emulation module:

- In the Remote box in the MULTI Builder window, enter:

```
hpserv IP_address
```

OR

- In the Builder window, click Debug to open the Debugger window; then, in the Debugger window's command pane, enter:

  ```
  remote hpserv IP_address
  ```

Starting hpserv opens two windows: the Target window and the I/O window. Commands entered in the Target window are sent directly to the emulation module.

The I/O window sends input (stdin) to and receives output (stdout) from the target program while it is running.

Note that hpserv connects to the first emulation module (port 6470) in a logic analysis system frame. You may specify another port by using the -p option with hpserv. See "To connect the logic analysis system to the LAN" on page 169 for more information on port numbers.

3 Start the debugger.

If you have not opened the Debugger window yet, click Debug in the Builder window.

4 Configure the emulation module and target system.

Before running the target processor, you must configure the HP emulation module for your target system. For example, you may have to set the BDM clock speed, the reset operation, cache disabling, or other configuration parameters.

If you are unsure of the configuration needed for your emulation module, you can use one of the following methods to explore the configuration options and configure the emulation module and target system:

- Enter "cf" commands in the Target window.
- Use the Configuration window in the logic analysis system's Emulation Control Interface.
- Use an initialization script.

See "To configure using an initialization script" on page 176 for information on saving the configuration commands in a script.

5 Specify an initialization address for the stack pointer.

This is required if the stack pointer is neither initialized when the processor is reset nor set in the start-up code generated by the compiler. If the stack pointer address needs to be initialized:

- In the debugger's command pane, enter:

```
_INIT_SP = <address>
```

OR

- In the Target window, enter:

```
reg r1=<address>
```

OR

- Include the following line in an initialization script:

```
target reg r1= <address>
```

6 Download the code:

In the Debugger window, select Remote->LoadProgram.

The Debugger command pane indicates that the code has been downloaded to the target.

To configure using an initialization script

You can use an initialization script to configure the emulation module and set up your target system. If you will always be using the same configuration, this way will save time and reduce errors.

1 Save the configuration commands in a text file, one command per line.

Green Hills also provides an example initialization sequence in the file MBX800.rc in the "hpdemo" directory.

2 To run the script, enter the following command in the Debugger

command pane:

```
< filename
```

Example

Create a file with the following lines:

```
remote hpserv hplogic1
target cf proc=MPC505
_INIT_SP=0x10000
```

Save the file in the MULTI startup directory and name it hpserv.rc. To run the script, enter the following command in the Debugger command pane:

```
<hpserv.rc
```

When run, this script will:

- Connect to the target through the emulation module in a logic analysis system frame called “hplogic1”.
- Set the processor type to MPC505.
- Initialize the stack pointer.

To perform common debugger tasks

- To display registers, click the regs button in the Display window.
- To set a breakpoint, right-click on the source code line where the breakpoint is to be located.
- To clear a breakpoint, right-click again on the source line.
- To step through code, click next.
- To run from the current PC, click go.
- To toggle the display between source code and source code interlaced with assembly code, click assem.
- To load program symbols, reset the PC, reset the stack pointer,

and run from the start, click restart.

To send commands to the emulation module

MULTI communicates with the emulation module using the emulation module's "terminal interface" commands. MULTI automatically generates and sends the commands required for normal operation.

If you want to communicate directly with the emulation module during a debug session, you may do so using "terminal interface" commands through the Target window (which comes up when hpserv is brought up).

You can also enter terminal interface commands from the Debugger window's command pane by preceding the command with the "target" command.

To view commands sent by MULTI to the emulation module

The communication between MULTI and the emulation module can be viewed by running hpserv in a logging mode:

```
remote hpserv -dc -a -o <filename> <emulation module
name>
```

The options -dc and -da log both asynchronous and console messages and the -o <filename> directs these messages to a log file called <filename>. When using this option, disconnect from hpserv (to flush out the file) and then you may view <filename> to see what commands MULTI sent to the emulation module.

NOTE: Logging commands in this way may result in a VERY large file. Beware of the disk space it may require.

To reinitialize the system

If you suspect that the emulation module is out of sync with the MULTI debugger, you may want to reinitialize it. Perform the steps below to accomplish reinitialization:

1. In the Target window, type:

   ```
   init -c
   ```

2. Repeat steps 4 through 7 in "To get started" on page 173 to configure the emulation module.

To disconnect from the emulation module

- In the Debugger window, select Remote->Disconnect.

  The Debugger command pane indicates that the debugger has disconnected from the emulation module.

Error conditions

"!ERROR 800! Invalid command: bcast"

This message usually means that there is no target interface module (TIM) connected to the emulation module or that the emulation module does not have firmware for the MPC500 family.

1. Verify that the emulation module is connected to the target.
2. Next, check that your emulation module is programmed with firmware for the Motorola MPC500:

   See "To display the emulation module firmware version information" on page 84. If the emulation module is not programmed with the proper firmware, see "To update emulation module firmware" on page 83.

"command socket connection failed: WSAECONNREFUSED: connection refused"

This message usually means the emulation module is not at port #6470 on the logic analysis system.

See Also

Green Hills MULTI Software Development Environment User's Guide.

Using MULTI with the Hewlett-Packard Processor Probe from Green Hills Software, Inc.

The Green Hills web site: http://www.ghs.com

See the "Configuring the Emulation Module" chapter on page 153 for more information on configuration options and the "cf" command.

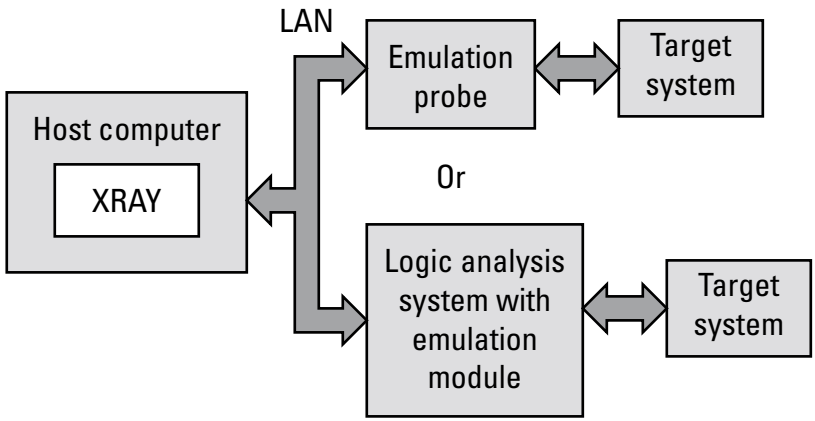
Using the Microtec Research Debugger

Compatibility

Version 4.1 of the XRAY HP Probe debugger from Microtec Research, a Mentor Graphics Company, is another debugger that connects to the HP emulation module.

This information in this section is intended to be used along with the XRAY documentation provided by Microtec Research.

Overview



To get started

1 Edit the gtw.brd file.

The file gtw.brd includes example hostnames, port numbers and initialization information for HP emulation probes/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 module.

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

See "To connect the logic analysis system to the LAN" on page 169 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 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.

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 "To configure the emulation module using an INCLUDE file" on page 184 for more information on using an include file.

If you are unsure of the configuration needed for your emulation module, you can telnet to the emulation module or 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, configuration will be complete and you can proceed to the next step.

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.

2 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 Connect 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.

3 Download the application code.

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 using an INCLUDE file

You can use an include file to configure the emulation module and set up your target system after bringing 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 directory under the xhippchp directory in the file "mo8xxads.inc".
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.

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 proc=MPC505, you can type

  ```
  cf "proc=MPC505"
  ```

  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 window, 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"

This message usually means that there is no target interface module (TIM) connected to the emulation module or the emulation module does not have firmware for the MPC500 family.

1. Verify that the emulation module is connected to the target.
2. Next, check that your emulation module is programmed with firmware for the Motorola MPC500:

   See "To display the emulation module firmware version information" on page 84. If the emulation module is not programmed with the proper firmware, see "To update emulation module firmware" on page 83.

"command socket connection failed: WSAECONNREFUSED: connection refused"

This message 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.

See the “Configuring the Emulation Module” chapter on page 153 for more information on configuration options and the “cf” command.

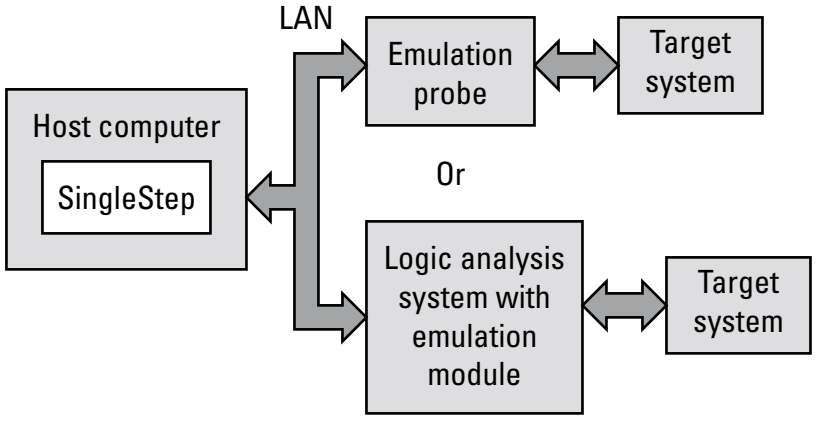
Using the Software Development Systems Debugger

Compatibility

Version 7.3 of the SingleStep debugger from Software Development Systems, Inc. is another debugger that connects to the HP emulation module.

The information in this section is intended to be used along 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 reset target option is selected, the target is reset.
- Hardware breakpoints are disabled.
- Software breakpoints are enabled.
- All breakpoints are cleared.

- main() _exit breakpoints are set, if that option is selected.

To get started

1 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. See "To change the port number of an emulation module" on page 170 for more information on port numbers.

2 Configure the emulation module with the processor clock speed.

In the Debug dialog box, click the Connection tab and then enter a Processor Clock speed which is less than or equal to the speed at which the processor will run out of reset.

The emulation module must know the target clock speed before it can communicate with the target. This value depends on the oscillator or crystal used on your target system and the multipliers applicable at reset. The communications speed can be changed (see "Download performance" on page 192) but will be reset to this value each time SingleStep resets the processor.

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 Target Configuration tab in the "Debug" dialog box.

An alternate way of performing target initialization is by using 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).

The "Debug" dialog method and the sstep.ini method are mutually exclusive. Use one or the other, but not both.

Initialization of the target will not actually occur until the "Debug" dialog is successfully exited.

4 Set up the Loading and Execution options in the Options tab of the Debug dialog.

5 Download the application and run:

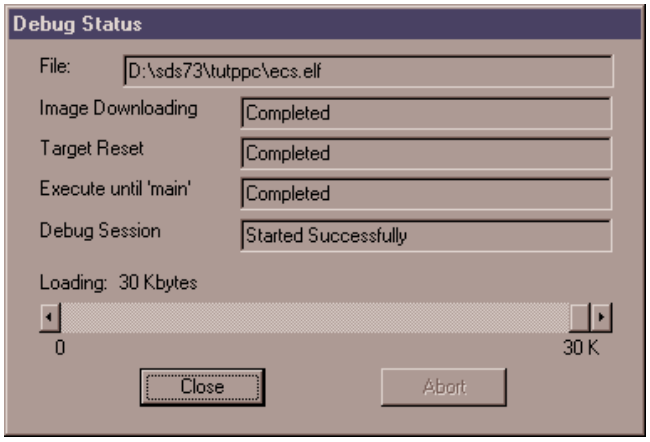
Select the File tab and enter the application file name. Exit the "Debug" dialog box by clicking OK.

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 "load image" was selected, download the application and set the PC based on object module file contents.
- If "reset target" was selected, 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 “execute until main” was selected, 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: Ctrl-c.

Examples

To see the speed that the emulation module is using to communicate with the target system you would issue the following command in the SingleStep command window:

```
control -c "cf procck"
```

To change the speed to match a 25MHz processor clock you would issue the following command in the command window:

```
control -c "cf procck=25"
```

For more information about "terminal interface" commands see "Built-In Commands" on page 215.

Download performance

Downloads are fastest when the emulation module speed is set to match that of the target processor.

The initial speed that the emulation module uses to communicate with the target processor is set by the Processor clock item in the Connection tab of the "Debug" dialog.

You are responsible for specifying this speed to be less than or equal to the initial, reset speed of the processor. Usually a setting in the Target Configuration tab of the Debug dialog or a command in the _config alias will raise the speed of the processor above its initial, reset value.

For maximum download performance, the command to increase the target processor speed should be followed by a command to increase the speed of the emulation module communication.

Example

The mpc505.cfg file contains the following command which writes to the SCCR register to set the processor speed to 28 MHz (assumes a 4 MHz crystal).

```
write -l 0x8007fc50 = 0x041800000
```

The following command, which increases the emulation module communication speed, should be placed immediately after the write command shown above.

```
control -c "cf procck=25"
```

On-chip breakpoints and debugging ROM code

The MPC500 has a built-in hardware breakpoint capability. When SingleStep steps one source line or sets a user defined breakpoint, it will first try to use a software breakpoint. If the breakpoint does not work because the breakpoint address is located in ROM, SingleStep will automatically attempt to use one of the available hardware breakpoints. For more information, see the SingleStep release notes.

To debug ROM based code, unselect “Load Application Image” in the options tab of the “Debug” dialog.

Error conditions

“!ERROR 800! Invalid command: bcast”

This message 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 MPC500 family.

1. Verify that the emulation module is connected to the target.
2. Next, check that your emulation module is programmed with firmware for the Motorola MPC500:

   See “To display the emulation module firmware version information” on page 84. If the emulation module is not programmed with the proper firmware, see “To update emulation module firmware” on page 83.

“command socket connection failed: WSAECONNREFUSED: connection refused”

This message usually means the emulation module is not at port #6470. See “To change the port number of an emulation module” on page 170.

“unrecognized hostname”

This message 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 emulation probe or to the logic analysis system.

  If you are unable to ping the emulator probe or logic analysis system, refer to "Solving LAN Communication Problems" on page 224 or the logic analysis system on-line help, respectively, for more information.

See Also

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

The *SDS SingleStep Users Guide*.

See the "Configuring the Emulation Module" chapter on page 153 for more information on configuration options and the "cf" command.

12

Coordinating Logic Analysis with Processor Execution

This chapter describes how to use an analysis probe, an emulation module, and other features of your HP 16600A/16700A-series 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 provide different views 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. You can 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 on-line help contains quick reminders of how to perform common measurements.

A few of the many things outlined in 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.”

Stopping Processor Execution on a Logic Analyzer Trigger

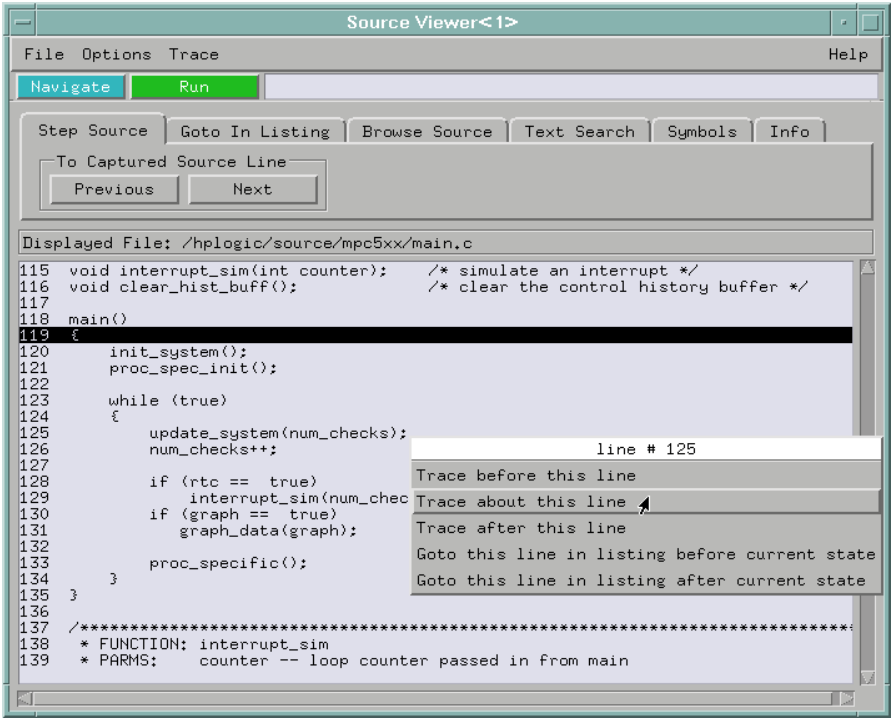
You can trigger the emulation module from the logic analyzer using either the Source Viewer window or the Intermodule window. If you are using the HP B4620B source correlation tool set, using the Source Viewer window is the easiest method.

To stop on a source line trigger (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 Run in the Source window (or other logic analyzer window).

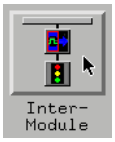
4 If your target system is not already running, click Run in the emulation Run Control window to start your target.

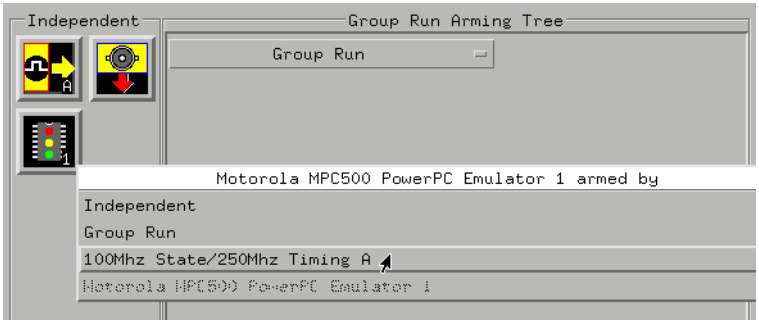
To stop on any trigger (Intermodule window)

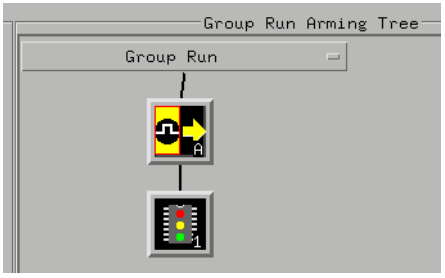
Use the Intermodule window if you do not have the HP B4620B source correlation tool set or if you need to use a more sophisticated trigger than is possible in the Source Viewer window.

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 Run in the emulation Run Control window to start your target.

See Also See the on-line 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 Also "To configure the processor clock speed" on page 160.

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. (You can 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 when Processor Execution Stops

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, it will be much simpler to trace until a processor halts (see "Tracing Until the Processor Halts" on page 203).

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.

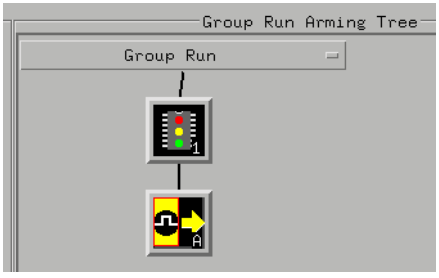
Here are some examples:

- If “Group Run” is armed from “Port In” and an emulation module is connected to Group Run, 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, the first one going into monitor will cause the second one to go into monitor.
- If an emulation module is connected to Port Out, 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 intermodule 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 microsecond and millisecond 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 microsecond and millisecond 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 your 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 trace until a processor halts (see "Tracing Until the Processor Halts" on page 203).

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.

NOTE: 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, it will be much simpler to trace until a processor halts (see "Tracing Until the Processor Halts" on page 203).

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.

NOTE: 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.

13

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.

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 215
Emulation module built-in commands do not work	1 Check that the emulation module has been properly configured for your target system.	page 153
	2 Run the emulation module performance verification tests.	page 226
	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 32, page 218
"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 160
Some commands fail	Check the "restrict to real-time runs" configuration.	page 162

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 debug mode
❍ Reset ❍ Break ● Run	The target processor is executing user code
❍ Reset ● Break ● Run	Only boot firmware is good (other firmware has been corrupted)

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 on-line 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
p	No target power
R	Emulation reset
r	Target reset
?	Unknown state

Examples

To set register R0 and view R0 to verify that it was set, enter:

```
R>rst -m
M>reg r0=ffff
M>reg r0
  reg R0=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.

Solving Target System Problems

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 "Built-In Commands" on page 215 for information on entering built-in commands.

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 9 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 "U>"	The emulation module is scanning the instruction register correctly.

Now you can do some more tests:

3 Enter the reset command:

```
U>rst
R>
```

The "R>" prompt is a good response that indicates SRESET and HRESET are working.

If interrupts are non-recoverable

❑ Check that interrupt service routines in the target code meet the requirements listed in the PowerPC documentation.

For proper debugging in interrupt service routines, the PowerPC documentation specifies that the exception handlers must do the following:

- As a prologue to the interrupt service routine:
  - Save the SRR0, SRR1, DAR, DSISR registers.
  - Set the RI bit in the MSR (Machine State Register.Recoverable Interrupt Bit).
- As an epilogue to the interrupt service routine:
  - Restore the SRR0, SRR1, DAR, DSISR registers.
  - Issue an RFI (Return from Interrupt) instruction.

Upon entering the interrupt service routine, the processor clears the MSR.RI bit, and copies the IP (Instruction Pointer)->SRR0 and the MSR->SRR1. The SRR0 and SRR1 are the save and restore registers. These contain the information needed to return to the state prior to the interrupt.

The RI bit will prevent the processor from breaking into debug mode with a maskable debug port breakpoint. A non-maskable breakpoint is required to break the processor when the RI bit is cleared, resulting in a possible non-recoverable state.

Software breakpoints place a “Trap” instruction into the breakpoint address. If the trap instruction is executed within an interrupt service routine, a break to background mode will occur. This causes the SRR0 ands SRR1 registers to be written over, causing a non-recoverable state. If the exception handler saves these registers, and sets the MSR.RI bit, the software breakpoint will always be recoverable.

If hardware breakpoints have no effect

Hardware breakpoints by default will not break the processor if they are set within an exception handler which has not saved the SRRs and set the MSR.RI bit. However, these can quite easily be reprogrammed to assert a non-maskable break. Note that the breakpoint will halt the processor, but will cause a non-recoverable state.

To reprogram the hardware breakpoint to assert a non-maskable break:

```
M>bc -e hwbp
M>reg lctrl2
  reg lctrl2=02018000
M>reg lctrl2=02018800 # OR in 0x00000800 with previous value
```

Hardware breakpoints will now cause a non-maskable break, which will halt the processor regardless of the status of the MSR.RI bit. Again, note that in this case the break will be non-recoverable if the exception handler has not saved the SRRs.

If the target resets itself

The most common plug-in issue is the target resetting itself. If the PC is set to some initial location, and then a short time later, the PC=100 or PC=fff00100, the target is resetting itself. In most cases, the chip is causing the reset, not the target hardware.

There are a number of possible causes of the reset. To determine the cause of reset, read the RSR (Reset Status Register):

```
M>m -a2 -d2 288@reg # telnet command which reads the RSR
```

The bits in this register show the cause of the reset:

RSR Bit Encoding		
Bit	**Cause of reset**	**Explanation**
0 (MSB)	External Hard Reset	The emulation module actually uses an external reset when resetting the target.
1	External Soft Reset	
2	Loss of Lock	Caused when the PLL loses the phase lock on the external clock source.
3	SW Watchdog	Make sure the SYPCR register disables the watchdog timer. R > reg cf_sypcr = ffffff88 or M > m -a4 -d4 4@reg = ffffff88
4	Checkstop	Occurs when the processor enters a checkstop state.
5	Debug Port Hard Reset	
6	Debug Port Soft Reset	
7	JTAG Reset	

To clear the RSR, execute the following:

```
M>m -a2 -d2 288@reg=ffff
```

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.

If there are problems with the debug port signals

❏ Check for pull-down resistors on DSDI and DSCK.

Some target systems may have 220 ohm pull downs on these two signals. These signals are series terminated by the analysis probe or target interface module with a 46 ohm resistor. A 220 ohm pull-down would present a 20% drop in signal level when driven high, which could easily cause some malfunctions. There should be a very weak pull down on the target, if any at all. If you want to pull-down DSCK, use a value of 2.2K or greater.

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.

Solving LAN Communication Problems

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 LEDs 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.

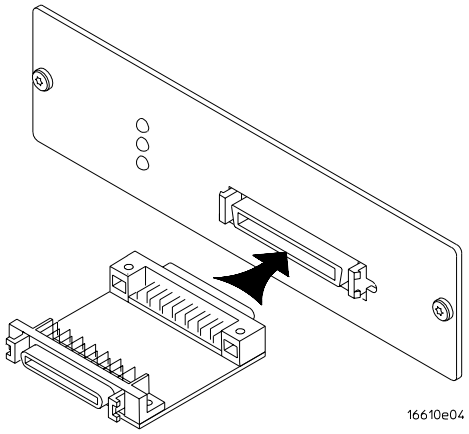
If there are many subnet masks in use on the local subnet, the logic analysis system may take a very long time to connect to the network after it is turned on.

Solving Emulation Module Problems

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 performance verification tests 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 on screen.

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 Ctrl-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.07.51 17Dec97
    Location:  Generics

  HPE3497A Motorola MPC800 Embedded PowerPC Emulator
    Version:   A.01.02 18Dec97
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 self-test 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 5: Target Probe Feedback Test

A verbose output on this test can be extensive. For example, the following is the output of this test if you forget to plug in the loopback test board.

```
p>pv -t5 -v2 1

  Testing: HPE3499A Series Emulation System
    Test # 5: Target Probe Feedback Test                        failed!
      Bad 20 Pin Status Read when unconnected = 0x7fb7
                             Expected Value = 0xffb7
      Bad 20 Pin Status Read when connected = 7fb7
                             Expected Value = 0x7fb7
```

```
     Output 19 Low not received on Input 11
     Output 11 Low not received on Input 19
     Output 13 Low not received on Input 1
     Output 12 High not received on Input 6
     Output 12 and Input 6 not pulled high on release
     Output 8 Low not received on Input 10
     Output 7 Low not received on Input 20
     Output 4 Low not received on Input 14
     Output 2 Low not received on Input 18
FAILED Number of tests: 1           Number of failures: 1
```

If the you get a verbose output like this, check to make sure that the loopback test board was connected properly.

TEST 6: Boundary Scan Master Test
TEST 7: 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.

Part 4

Reference

14

Specifications and Characteristics

Analysis Probe Operating Characteristics

The following operating characteristics are not specifications, but are typical operating characteristics for the HP E2490A analysis probe for the PowerPC 505/509.

Operating Characteristics	
Microprocessor Compatibility	PowerPC 505 and PowerPC 509
Package Supported	160-pin PQFP
Microprocessor Clock Speed	Up to 33 MHz
Logic Analyzers Supported	HP 1660A/AS/C/CP/CS, HP 1661A/AS,C/CP/CS, HP 1670A/D, HP 1671A/D, HP 16550A (one or two cards), HP 16554A/55A/56A (two or three cards), HP 16555D/56/57D (two or three cards)
Accessories Required	The HP 16610A emulation module or the HP E3456A emulation probe is required to initialize the analysis probe.
Probes Required	Six 16-channel probes are required for disassembly of PowerPC 505/509. Four additional 16-channel pods are available, two of which contain additional status signals and two of which contain unbuffered address bus signals for timing analysis. A maximum of eight pods can be used at one time.

Electrical Characteristics	
Power Requirements	500 mA @ 5V, supplied by the logic analyzer. CAT I, Pollution degree 2.
Signal Line Loading	100 KOhms in parallel with 18 pF on the following lines: ADDR[0], ADDR[6:29], ARETRY, AT[0:1], BE[0:3], BDIP, BURST, CSBOOT, RESETOUT, WR. 100 KOhms in parallel with 28 pF on the following lines: ADDR[1:5], CLKOUT, CT[0:3], TS. 100 KOhms in parallel with 23 pF on the following lines: CR/DS, TA, TEA. 100 KOhms in parallel with 8 pF on all other lines.

Environmental Characteristics	
Temperature, Operating	0 to + 50 degrees C +32 to +131 degrees F
Altitude, Operating	4,600 m 15,000 feet
Humidity	Up to 75% noncondensing. Avoid sudden, extreme temperature changes which could cause condensation on the circuit board.

Theory of Operation

The HP E2490A analysis probe programs the on-board programmable logic devices for a specific target configuration by reading the contents of the option and base address registers in the chip-select group. The MEMMAP register is also read to determine the location of the internal L-bus memory block.

The MPC50X can be configured to ignore the contents of the chip-select group of registers by clearing the Data Bus Bit 0. This will cause the Data Bus Reset Configuration Word to program all the potential address lines ADDR[0:29] as address lines, without any chip select support.

If your target requires this mode of operation, be sure to also embed this information into the Option Registers in the chip-select group. The PCON field in the option registers (0 through 11) must be set to the binary value “11” in order to declare that the address pins are not being used as an Output Enable, Write Enable, or Chip Enable. The PCON field is located in bits [19:20] of each Option Register.

Configuration File Label Descriptions

The following table describes the labels that are defined by the analysis probe’s configuration files.

Configuration File Label Descriptions

Label	STAT Bits	Description
AACK-		Address Acknowledge. When asserted, indicates the slave has received the address from the bus master.
ADDR		Address bus signals A0-A31. Signals A31 and A30 are generated by the analysis probe.
ARETR-		When this signal is asserted it indicates the master needs to retry the address phase.
AT0-1		These signals indicate the address type, defining it as user or supervisor, data or instruction.
B16/32	STAT[15]	This signal is generated by the analysis probe. A low indicates that the current access is a 32-bit port, a high indicates a 16-bit port.
BB-		Bus Busy indicates that the bus is in use.
BDIP-	STAT[29]	Burst Data In Progress indicates the duration of a burst transfer.
BE0-3		These signals are the Byte Enables for the data bus.
BG-		The memory system asserts Bus Grant to allow the PowerPC 505/509 onto the address bus.
BI-	STAT[28]	Burst Inhibit indicates that the slave device addressed in the current burst transaction is unable to support burst transfers.
BR-		The PowerPC 505/509 asserts Bus Request to indicate that it has business to conduct on the address bus.
BURST-		This signal indicates that the current initiated transfer is a burst cycle.
CBOOT-	STAT[0]	This signal is the chip select of system boot memory.
CR/DS		Cancel Reservation. When asserted, it instructs the bus master to clear the reservation.
CT0-3		These signals indicate the type of cycle the bus master is initiating.
DATA		Data bus signals D0-D31
ECROUT		The microprocessor's ECROUT signal.
GAT0-3	STAT[26-25]	These analysis probe-generated signals are a modified version of AT0-3 that are used by the inverse assembler.
GBE0-3	STAT[19-16]	These analysis probe-generated signals are a modified version of BE0-3 that are used by the inverse assembler.
GBRST-	STAT[27]	This analysis probe-generated signal is a modified version of BURST- that is used by the inverse assembler.

Configuration File Label Descriptions

Label	STAT Bits	Description
GCT0-3	STAT[24-21]	These analysis probe-generated signals are a modified version of CT0-3 that are used by the inverse assembler.
GDS-		Data Strobe. Asserted by EBI at the end of a chip-select-controlled bus cycle after the chip select unit asserts the internal TA- signal or the bus monitor timer asserts the internal TEA- signal. This signal is also asserted at the end of a Show cycle. The analysis probe inverts the MPC505 signal so that a true low signal is always sent to the logic analyzer.
GWR-	STAT[20]	This signal is generated by the analysis probe. A low indicates a write, a high indicates a read.
IRQ0-6		These signals are interrupt requests.
JCLK		The CLOCKOUT signal
KCLK		A combination of TEA, TA, and DS.
MODCLK		This signal indicates the source of the system clock.
PDWU	STAT[30]	This signal is a power-down wakeup to external power-on reset circuit.
RESET-	STAT[13]	This input signal is a hard reset. When asserted, devices on the bus must reset.
RSTOU-	STAT[12]	Reset output signal. When asserted by the microprocessor, it instructs all other devices monitoring this signal to reset.
SHOCY-	STAT[14]	This signal is generated by the analysis probe. A low indicates that the current state is a Show cycle, a high indicates it is not a show cycle.
STAT		Microprocessor status and control signals.
TA-		The memory system asserts TA to acknowledge a data transaction.
TADDR		Address bus signals A0-A29. These signals are unbuffered and can be used for timing analysis.
TCK		This signal is a test clock input with a pulldown resistor to synchronize the test logic.
TDI		This tri-stateable input is sampled on the rising edge of TCK.
TDO		This tri-stateable signal test data output changes on the falling edge of TCK.
TEA-	STAT[31]	The memory system may assert TEA- to indicate a transfer error, e.g. an unmapped part of the address space.
TMS		This signal is Test Mode Select. It is sampled on the rising edge of TCK to sequence the test controller's state machine.

Configuration File Label Descriptions

Label	STAT Bits	Description
TRST		This signal is an asynchronous active-low test reset that provides initialization of the TAP controller and other logic as required by the JTAG standard.
TS-	STAT[32]	The PowerPC 505/509 asserts TS- for one cycle to commence a transaction.
VFLS	STAT[8-7]	Visible History Buffer Flushes Status tracks program flow.
VF0-2	STAT[11-9]	Visible Instruction Queue Flushes Status tracks program flow.
WP0-5	STAT[6-1]	These signals are I-bus watchpoints (WP0:3) and L-bus watchpoints (WP4:5).
WR-		Read/Write. A high indicates a read, a low indicates a write.

Analysis Probe Signal-to-Connector Mapping

The following tables show the electrical signal-to-connector mapping required by the HP E2490A analysis probe for the PowerPC 505/509.

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J4odd					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J4odd	6	clk1 (J)	28	CLOCKOUT	JCLK
J4odd	8	15	84	A16	ADDR
J4odd	10	14	85	A17	ADDR
J4odd	12	13	86	A18	ADDR
J4odd	14	12	87	A19	ADDR
J4odd	16	11	88	A20	ADDR
J4odd	18	10	89	A21	ADDR
J4odd	20	9	90	A22	ADDR
J4odd	22	8	91	A23	ADDR
J4odd	24	7	92	A24	ADDR
J4odd	26	6	93	A25	ADDR
J4odd	28	5	96	A26	ADDR
J4odd	30	4	97	A27	ADDR
J4odd	32	3	98	A28	ADDR
J4odd	34	2	99	A29	ADDR
J4odd	36	1	*	A30	ADDR
J4odd	38	0	*	A31	ADDR

*** These signals are generated by the analysis probe.**

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J4even					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J4even	5	clk1 (K)	*	TEA/TA/DS	KCLK
J4even	7	15	7	A0 (MSB)	ADDR
J4even	9	14	6	A1	ADDR
J4even	11	13	5	A2	ADDR
J4even	13	12	4	A3	ADDR
J4even	15	11	3	A4	ADDR
J4even	17	10	2	A5	ADDR
J4even	19	9	159	A6	ADDR
J4even	21	8	158	A7	ADDR
J4even	23	7	157	A8	ADDR
J4even	25	6	156	A9	ADDR
J4even	27	5	155	A10	ADDR
J4even	29	4	154	A11	ADDR
J4even	31	3	78	A12	ADDR
J4even	33	2	79	A13	ADDR
J4even	35	1	82	A14	ADDR
J4even	37	0	83	A15	ADDR
*** KCLK is a combination of TEA, TA, and DS.**					

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J7odd					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J7odd	6	clk1 (L)	148	TA	TA-
J7odd	8	15	122	D16	DATA
J7odd	10	14	123	D17	DATA
J7odd	12	13	124	D18	DATA
J7odd	14	12	125	D19	DATA
J7odd	16	11	128	D20	DATA
J7odd	18	10	129	D21	DATA
J7odd	20	9	130	D22	DATA
J7odd	22	8	131	D23	DATA
J7odd	24	7	132	D24	DATA
J7odd	26	6	133	D25	DATA
J7odd	28	5	134	D26	DATA
J7odd	30	4	135	D27	DATA
J7odd	32	3	138	D28	DATA
J7odd	34	2	139	D29	DATA
J7odd	36	1	140	D30	DATA
J7odd	38	0	141	D31 (LSB)	DATA

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J7even					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J7even	5	clk1 (M)	149	AACK	AACK-
J7even	7	15	102	D0 (MSB)	DATA
J7even	9	14	103	D1	DATA
J7even	11	13	104	D2	DATA
J7even	13	12	105	D3	DATA
J7even	15	11	108	D4	DATA
J7even	17	10	109	D5	DATA
J7even	19	9	110	D6	DATA
J7even	21	8	111	D7	DATA
J7even	23	7	112	D8	DATA
J7even	25	6	113	D9	DATA
J7even	27	5	114	D10	DATA
J7even	29	4	115	D11	DATA
J7even	31	3	116	D12	DATA
J7even	33	2	117	D13	DATA
J7even	35	1	118	D14	DATA
J7even	37	0	119	D15	DATA

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J2odd						
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**	**Alt. Label**
J2odd	6	clk1	**	DS for 505 DS- for 509	GDS-	
J2odd	8	15	*	*	STAT	B16/32
J2odd	10	14	*	*	STAT	SHOCY-
J2odd	12	13	30	RESET	STAT	RESET
J2odd	14	12	77	RESETOUT	STAT	RSTOU-
J2odd	16	11	76	VF0	STAT	VF0-2
J2odd	18	10	72	VF1	STAT	VF0-2
J2odd	20	9	71	VF2	STAT	VF0-2
J2odd	22	8	70	VFLS0	STAT	VFLS
J2odd	24	7	69	VFLS1	STAT	VFLS
J2odd	26	6	68	WP0	STAT	WP0-5
J2odd	28	5	65	WP1	STAT	WP0-5
J2odd	30	4	64	WP2	STAT	WP0-5
J2odd	32	3	63	WP3	STAT	WP0-5
J2odd	34	2	62	WP4	STAT	WP0-5
J2odd	36	1	59	WP5	STAT	WP0-5
J2odd	38	0	58	CSBOOT-	STAT	CBOOT-

*** These signals are generated by the analysis probe.**

**** The analysis probe inverts this signal from MPC505 target systems so that a true low (DS-) signal is always sent to the logic analyzer**

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J2even						
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**	**Alt. Label**
J2even	5	clk1	150	TS-	STAT	TS-
J2even	7	15	23	TEA-	STAT	TEA-
J2even	9	14	22	PDWU	STAT	PDWU
J2even	11	13	21	BDIP-	STAT	BDIP-
J2even	13	12	18	BI-	STAT	BI-
J2even	15	11	17	BURST-	STAT	GBRST-
J2even	17	10	16	AT0	STAT	GAT0-1
J2even	19	9	15	AT1	STAT	GAT0-1
J2even	21	8	12	CT0	STAT	GCT0-3
J2even	23	7	11	CT1	STAT	GCT0-3
J2even	25	6	10	CT2	STAT	GCT0-3
J2even	27	5	9	CT3	STAT	GCT0-3
J2even	29	4	8	WR-	STAT	GWR-
J2even	31	3	144	BE0	STAT	GBE0-3
J2even	33	2	145	BE1	STAT	GBE0-3
J2even	35	1	146	BE2	STAT	GBE0-3
J2even	37	0	151	BE3	STAT	GBE0-3

The "G" in front of the Alt. labels indicates these signals are generated by the analysis probe.

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J6odd					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J6odd	6	clk1	25	CR/DS	CR/DS
J6odd	8	15	84	A16	TADDR
J6odd	10	14	85	A17	TADDR
J6odd	12	13	86	A18	TADDR
J6odd	14	12	87	A19	TADDR
J6odd	16	11	88	A20	TADDR
J6odd	18	10	89	A21	TADDR
J6odd	20	9	90	A22	TADDR
J6odd	22	8	91	A23	TADDR
J6odd	24	7	92	A24	TADDR
J6odd	26	6	93	A25	TADDR
J6odd	28	5	96	A26	TADDR
J6odd	30	4	97	A27	TADDR
J6odd	32	3	98	A28	TADDR
J6odd	34	2	99	A29	TADDR
J6odd	36	1		GND	TADDR
J6odd	38	0		GND	TADDR

The “T” in front of the CR/DS and ADDR labels indicates these signals are unbuffered and can be used for timing analysis.

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J6even					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J6even	5	clk1	27	ECROUT	ECROUT
J6even	7	15	7	A0 (MSB)	TADDR
J6even	9	14	6	A1	TADDR
J6even	11	13	5	A2	TADDR
J6even	13	12	4	A3	TADDR
J6even	15	11	3	A4	TADDR
J6even	17	10	2	A5	TADDR
J6even	19	9	159	A6	TADDR
J6even	21	8	158	A7	TADDR
J6even	23	7	157	A8	TADDR
J6even	25	6	156	A9	TADDR
J6even	27	5	155	A10	TADDR
J6even	29	4	154	A11	TADDR
J6even	31	3	78	A12	TADDR
J6even	33	2	79	A13	TADDR
J6even	35	1	82	A14	TADDR
J6even	37	0	83	A15	TADDR

The "T" in front of the ADDR label indicates these signals are unbuffered and can be used for timing analysis.

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J3odd					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J3odd	6	clk1	50	TCK	TCK
J3odd	8	15	24	ARETRY	ARETR-
J3odd	10	14	144	BB-	BB-
J3odd	12	13	145	BG-	BG-
J3odd	14	12	146	BR-	BR-
J3odd	16	11	22	BURST-	BURST
J3odd	18	10	77	AT0	AT0-1
J3odd	20	9	76	AT1	AT0-1
J3odd	22	8	12	CT0	CT0-3
J3odd	24	7	11	CT1	CT0-3
J3odd	26	6	10	CT2	CT0-3
J3odd	28	5	9	CT3	CT0-3
J3odd	30	4	151	WR-	WR-
J3odd	32	3	18	BE0	BE0-3
J3odd	34	2	17	BE1	BE0-3
J3odd	36	1	16	BE2	BE0-3
J3odd	38	0	15	BE3	BE0-3

PowerPC 505/509 Logic Analyzer Interface Signal list - Pod J3even					
Cable Connector	**2x19 Pin**	**LA Bit**	**505/509 Pin**	**505/509 Signal**	**Label**
J3even	5	clk1			
J3even	7	15			
J3even	9	14			
J3even	11	13			
J3even	13	12			
J3even	15	11		TMS	TMS
J3even	17	10		TRST	TRST
J3even	19	9		TDI	TDI
J3even	21	8		TDO	TDO
J3even	23	7		MODCLK	MODCLK
J3even	25	6		IRQ0	IRQ0-6
J3even	27	5		IRQ1	IRQ0-6
J3even	29	4		IRQ2	IRQ0-6
J3even	31	3		IRQ3	IRQ0-6
J3even	33	2		IRQ4	IRQ0-6
J3even	35	1		IRQ5	IRQ0-6
J3even	37	0		IRQ6	IRQ0-6

Emulation Module Operating Characteristics

The following operating characteristics are not specifications, but are typical operating characteristics for the HP 16610A emulation module and MPC505/509 target interface module.

Operating Characteristics	
Microprocessor Compatibility	Motorola MPC505 and MPC509 Embedded PowerPC 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 MPC500 Embedded PowerPC emulation module	**Symbol**	**Min**	**Max**	**Unit**
Input voltage range	Vin	-0.5	5.5	V
Input voltage range	Vtt	1.3	1.7	V
Input High Voltage	Vih	2/3Vtt + 0.2		V
Input Low Voltage	Vil		2/3Vtt - 0.2	V
Input High Current	Iih		-15	µA
Input Low Current	Iil		100	µA
Output High Voltage	Voh	2.4	3.3	V
Output Low Voltage	Vol		0.5	V
Output High Current	Ioh	8		mA
Output Low Current	Iol	-16		mA

15

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 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 that convert compiler or linker map file output that has symbolic information.

You can typically use 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.
```

16

Using the Analysis Probe with Other Logic Analyzers

Logic Analyzer	Channel Count	State Speed	Timing Speed (Full/Half Channels)	Memory Depth (Full/Half Channels)
HP 1660A/AS/C/CP/CS/E/EP/ES	136	100 MHz	250/500 MHz	4K/8K states
HP 1661A/AS/C/CP/CS/E/EP/ES	102	100 MHz	250/500 MHz	4K/8K states
HP 1670A	136	70 MHz	125/250 MHz	64K or 512K states
HP 1670D/E	136	100 MHz	125/250 MHz	64K/128K states (1M with option)
HP 1671A	102	70 MHz	125/250 MHz	64K or 512K states
HP 1671D/E	102	100 MHz	125/250 MHz	64K/128K states (1M with option)
HP 16550A (one or two cards)	102/card	100 MHz	250/500 MHz	4K/8K states
HP 16554A (two or more cards)	68/card	70 MHz	125/250 MHz	500K/1M states
HP 16555A (two or more cards)	68/card	110 MHz	250/500 MHz	1M/2M states
HP 16555D (two or more cards)	68/card	110 MHz	250/500 MHz	2M/4M states
HP 16556A (two or more cards)	68/card	100 MHz	200/400 MHz	1M/2M states
HP 16556D (two or more cards)	68/card	100 MHz	200/400 MHz	2M/4M states
HP 16557D (two or more cards)	68/card	135 MHz	250/500 MHz	2M/4M 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 E2490A analysis probe.

Logic Analyzer Software Version Requirements	
Logic Analyzer	**Minimum Software Version**
HP 1660A/AS Series	A.02.01
HP 1660C/CS/CP Series	A.02.01
HP 1660E/ES/EP Series	A.01.00
HP 1670A/D Series	A.02.01
HP 1670E Series	A.01.00
HP 16500C Mainframe*	A.01.00
HP 16500B Mainframe*	A.03.40
HP 16505A Prototype Analyzer	A.01.22
* The mainframes are used with the HP 16550 and HP 16554/55/56/57 logic analyzers.	

If your software version is older than those listed, load new system software with the higher version numbers before loading the HP E2490A analysis probe software.

You can get the latest software on the web at:

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

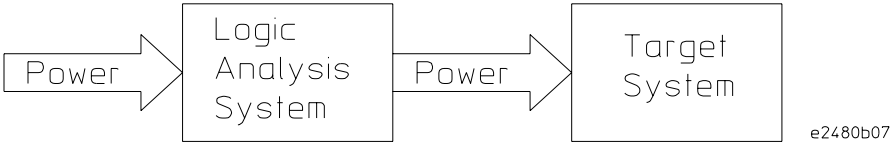
Power-ON/Power-OFF Sequence

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 other HP logic analyzers

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

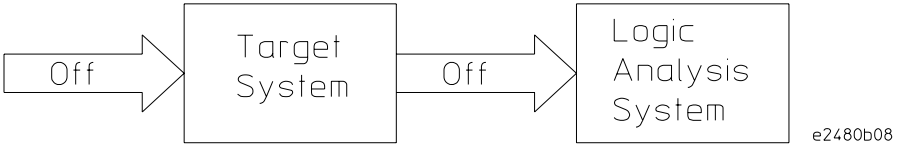
1. Logic analysis system.
2. Your target system.



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.



Installing Software

The following files can be installed from a floppy disk:

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

To install software on other logic analyzers

See the documentation for your logic analyzer.

Connecting the Analysis Probe to the Logic Analyzer

This section shows you how to connect the analysis probe to the logic analyzer. It consists of the following:

- Connecting the high-density cables to the analysis probe.
- Connecting the high-density cables to the logic analyzer.

This section shows connection diagrams that identify connections to each individual logic analyzer supported by the analysis probe. Since there are different connections and configurations for State and Timing analysis, each logic analyzer sections shows connections for both. They are shown in the following order:

- HP 1660 logic analyzers.
- HP 1661 logic analyzers.
- HP 1670 logic analyzers.
- HP 1671 logic analyzers.

Number of Pods Used/Required

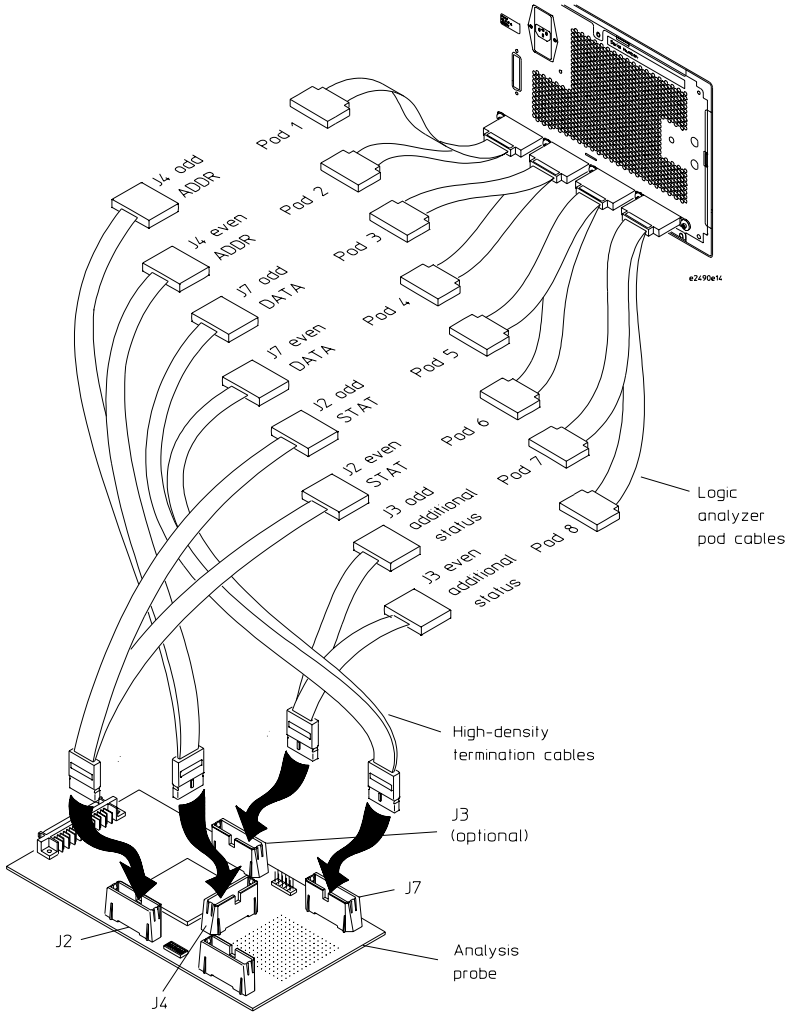
Both state and timing measurements require a minimum of six pods to make a measurement. The logic analyzer configuration files assign signals for eight pods. If fewer than eight pods are used, the logic analyzer will truncate the pods allocated. In this case, viewing the logic analyzer Format menu shows the pod allocations. If the allocations will not acquire the desired signals, the allocations can be altered manually.

The HP E2490A analysis probe includes three HP E5346A high-density termination cables. If you want to connect eight pods to your logic analyzer, you will need one additional high-density termination cable.

To connect to the HP 1660 logic analyzers

Use the following two figures to connect the analysis probe to the HP 1660 logic analyzers. Analysis probe pod J3 (logic analyzer pods 7 and 8) is optional. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

State Analysis Connection

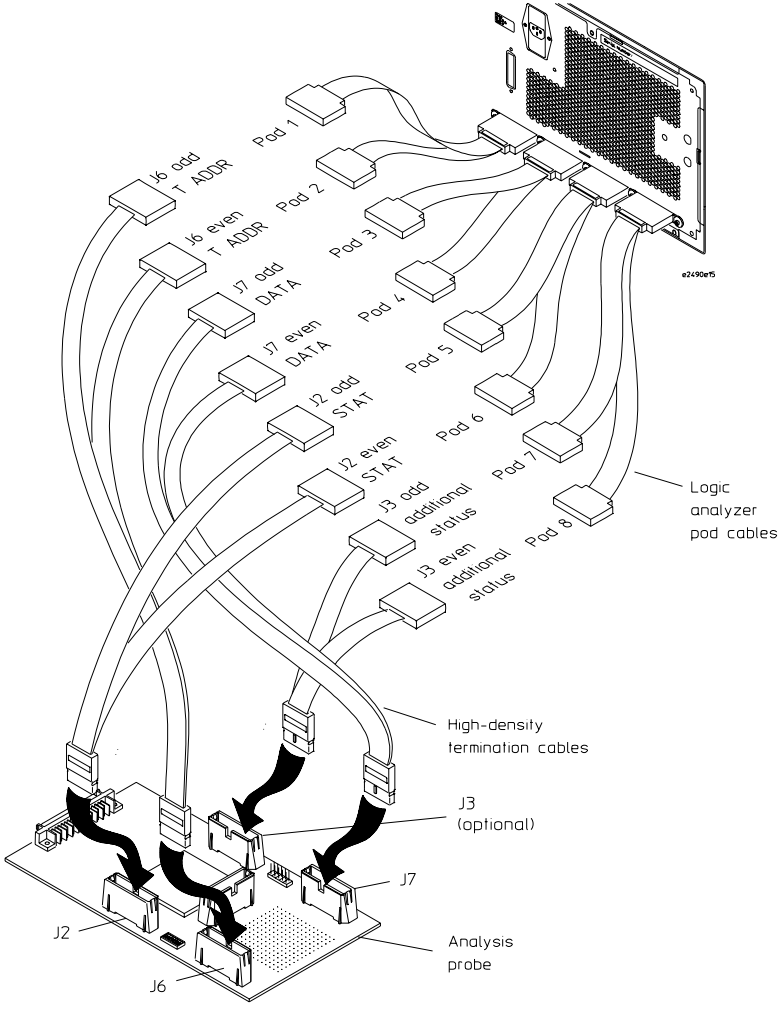


Configuration File (State)

Use configuration file C505IA2 for state analysis with the HP 1660 logic analyzers.

Analysis probe pod J3 (logic analyzer pods 7 and 8) is optional.

Timing Analysis Connection



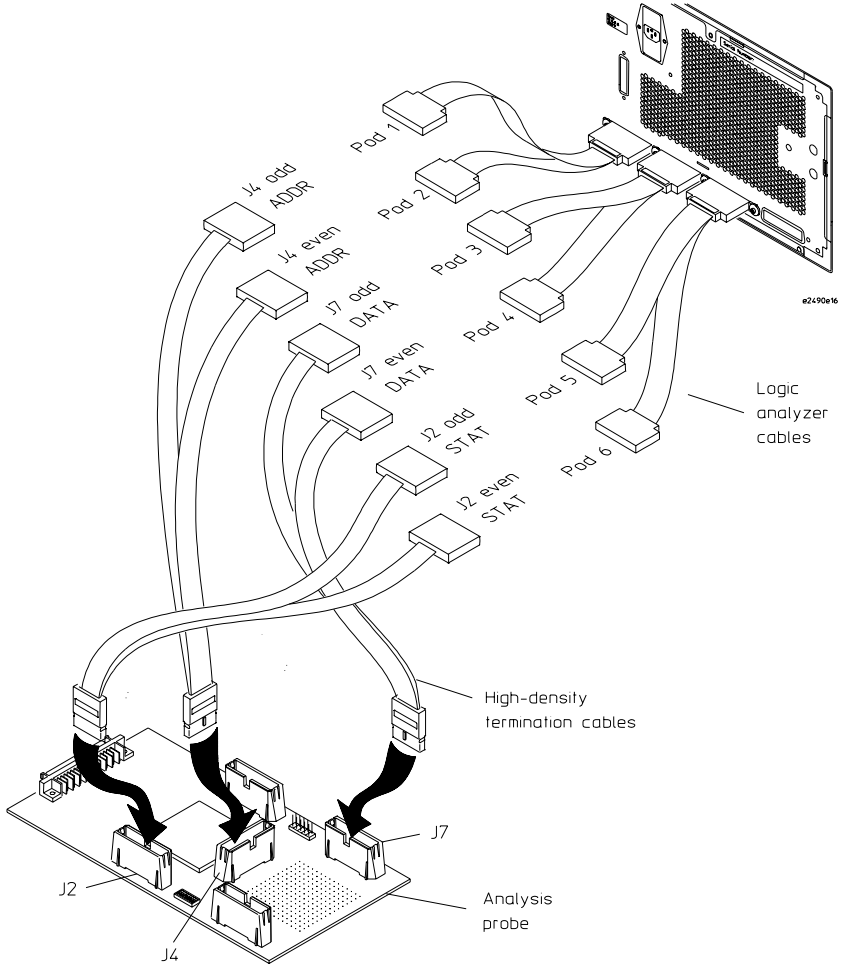
Configuration File (Timing)

Use configuration file C505T2 for timing analysis with the HP 1660 logic analyzers.

To connect to the HP 1661 logic analyzers

Use the following two figures to connect the analysis probe to the HP 1661 logic analyzers. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

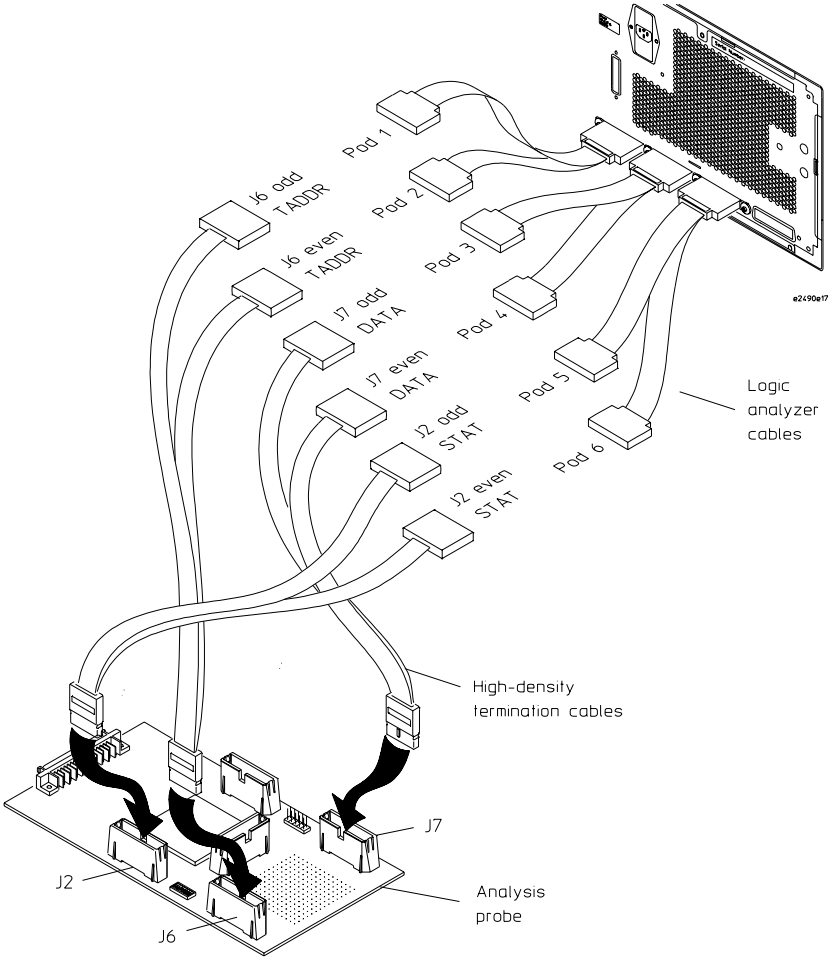
State Analysis Connection



Configuration File (State)

Use configuration file C5051A for state analysis with the HP 1661 logic analyzers.

Timing Analysis Connection



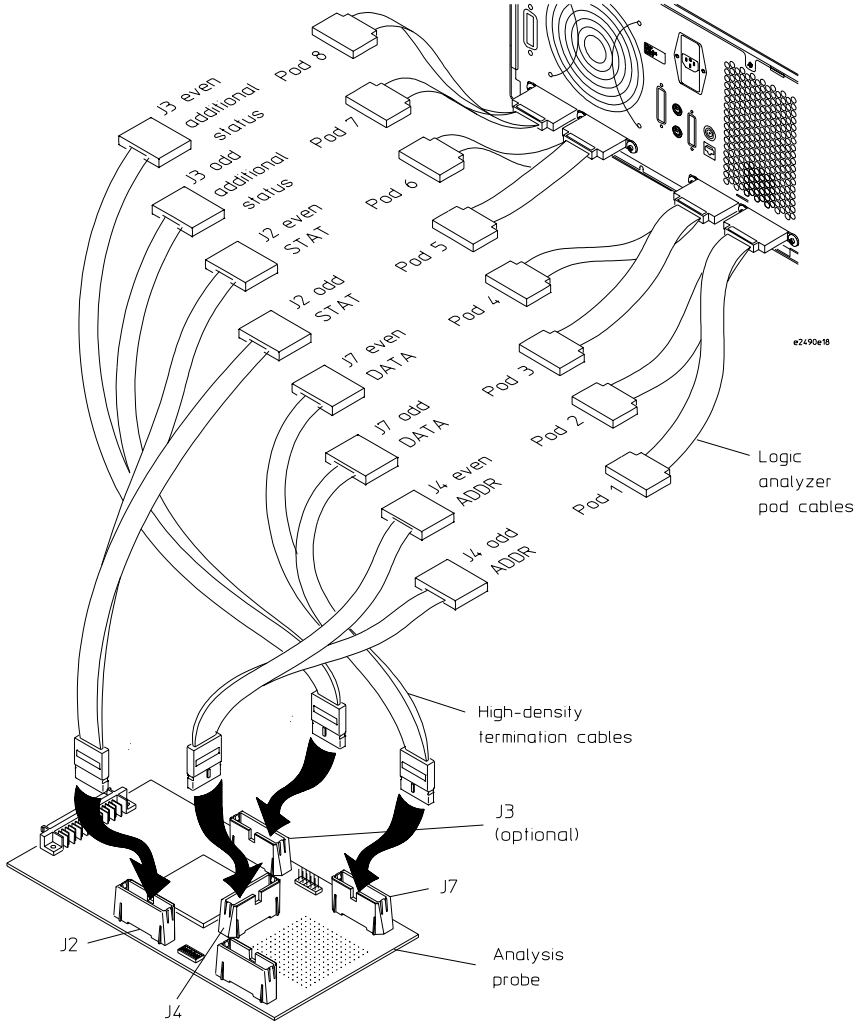
Configuration File (Timing)

Use configuration file C505T1 for timing analysis with the HP 1661 logic analyzers.

To connect to the HP 1670 logic analyzers

Use the following figures to connect the analysis probe to the HP 1670 logic analyzers. Analysis probe pod J3 (logic analyzer pods 7 and 8) is optional. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

State Analysis Connection

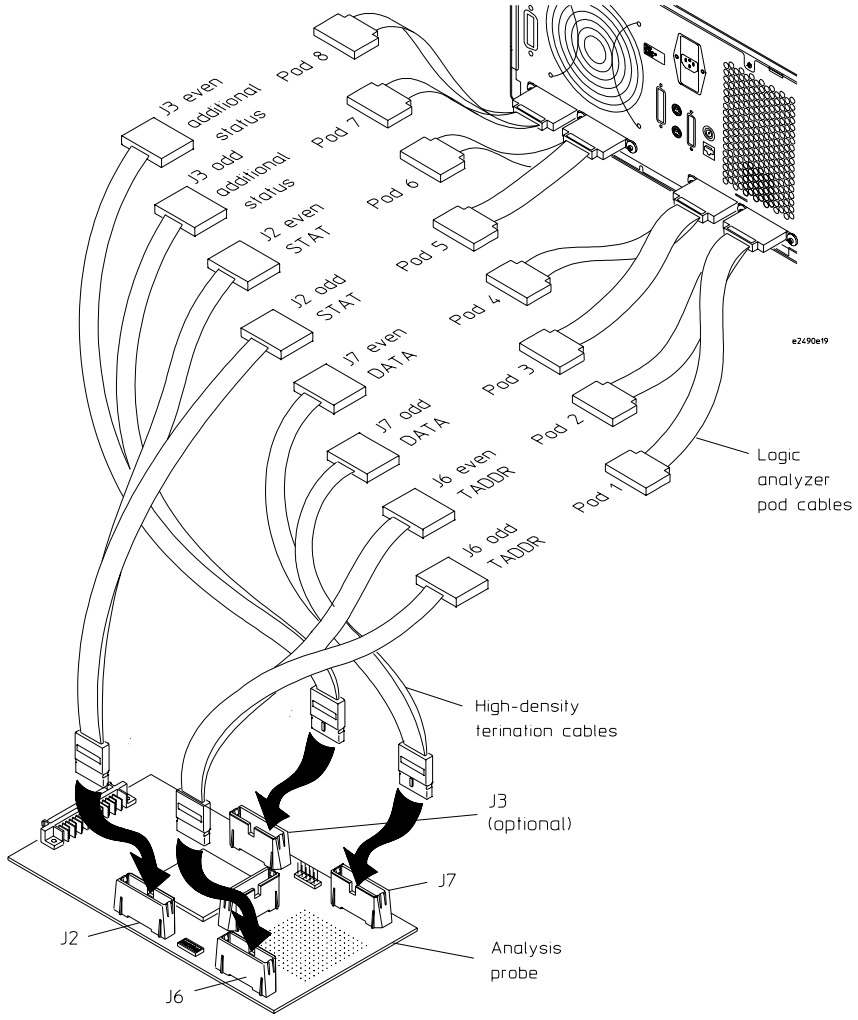


Configuration File (State)

Use configuration file C5051A2 for state analysis with the HP 1670 logic analyzers.

Analysis probe pod J3 (logic analyzer pods 7 and 8) is optional.

Timing Analysis Connection



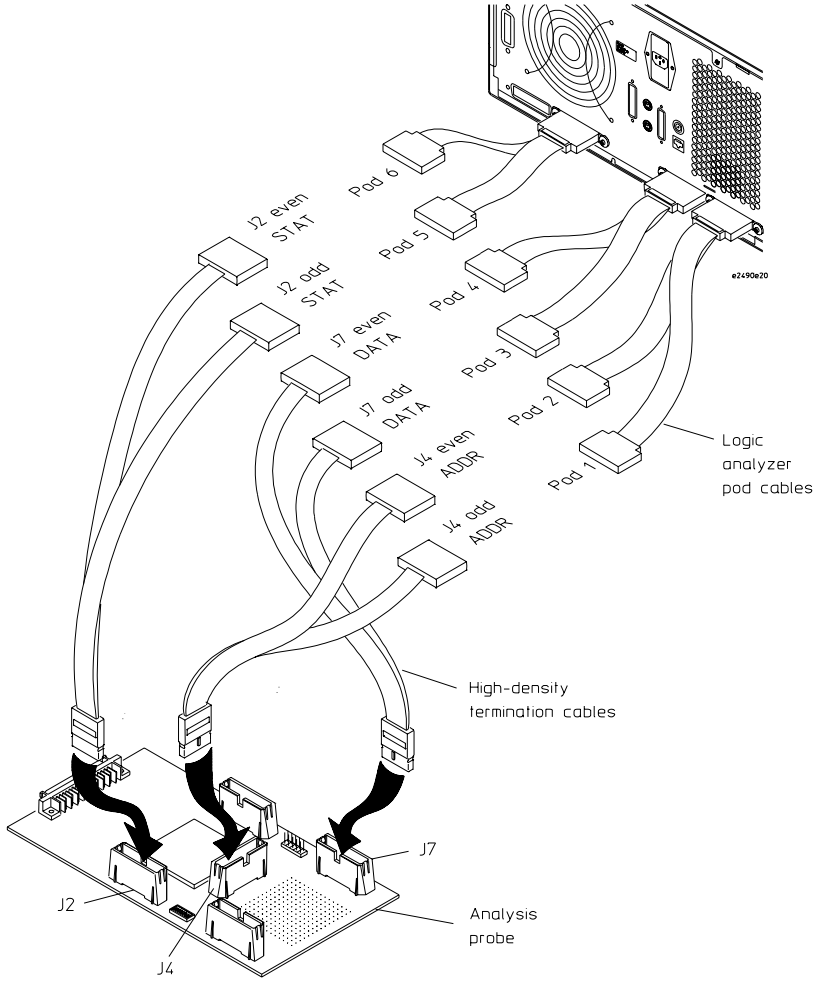
Configuration File (Timing)

Use configuration file C505T2 for timing analysis with the HP 1670 logic analyzers.

To connect to the HP 1671 logic analyzers

Use the following figures to connect the analysis probe to the HP 1671 logic analyzers. Find the labels that were shipped with the high-density cables and use them to help identify the connections.

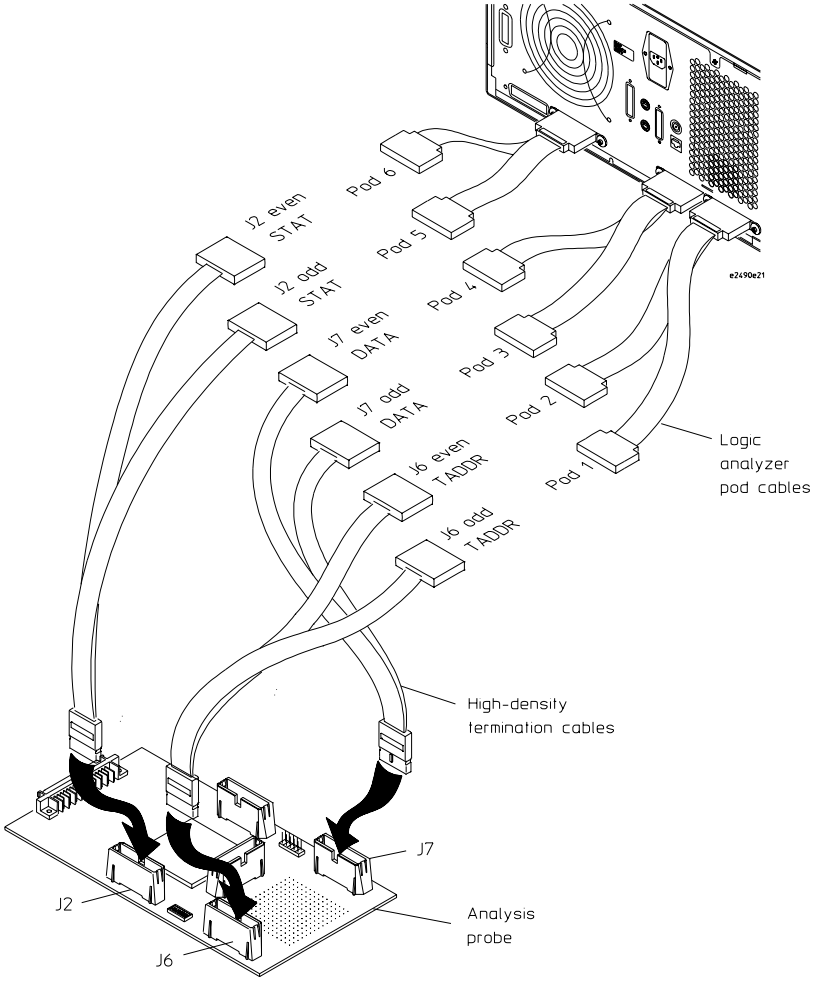
State Analysis Connection



Configuration File (State)

Use configuration file C5051A1 for state analysis with the HP 1671 logic analyzers.

Timing Analysis Connection



Configuration File (Timing)

Use configuration file C505T1 for timing analysis with the HP 1671 logic analyzers.

Configuring the Logic Analyzer

This section describes how to:

- Load configuration files.
- View predefined symbols.

To load configuration files

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 MPC50x 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 appropriate HP E2490A disk (inverse assembler) in the front disk drive of the logic analyzer.
2. Depending on your logic analyzer, select one of the following menus:
   - For the HP 1660-series logic analyzers, select the “System Disk” menu.
   - For the HP 16500B/C mainframe, select the “System 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 the following table.

6 Execute the load operation to load the file into the logic analyzer.

The logic analyzer is configured for MPC505/509 analysis by loading the appropriate MPC505/509 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 appropriate "16505 Prototype Analyzer" flexible disk (inverse assembler) 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 E2490A.

Logic Analyzer Configuration Files

Analyzer Model	Analyzer Description	State Configuration File	Timing Configuration File
HP 16550A (one card)	100 MHz state 250/500 MHz timing 4K/8K samples	C505IA1	C505T1
HP 16550A (two card)	100 MHz state 250/500 MHz timing 4K/8K samples	C505IA2	C505T2
HP 16554A (two or more cards)	70 MHz state 125/250 MHz timing 0.5M/1M samples	C505IA3	C505T3
HP 16555A/D (two or more cards)	110 MHz state 250/500 MHz timing 2M/4M samples (D)	C505IA3	C505T3
HP 16556A/D (two or more cards)	100 MHz state 200/400 MHz timing 2M/4M samples (D)	C505IA3	C505T3

Logic Analyzer Configuration Files

Analyzer Model	Analyzer Description	State Configuration File	Timing Configuration File
HP 16557D (two or more cards)	135 MHz state 250/500 MHz timing 2M/4M samples	C505IA3	C505T3
HP 1660A/AS/C/CP/CS/ E/EP/ES	100 MHz state 250/500 MHz timing 4K/8K samples	C505IA2	C505T2
HP 1661A/AS/C/CP/CS/ E/EP/ES	100 MHz state 250/500 MHz timing 4K/8K samples	C505IA1	C505T1
HP 1670A	70 MHz state 125/250 MHz timing 64K or 512K states	C505IA2	C505T2
HP 1670D/E	100 MHz state 125/250 MHz timing 64K/128K states (1M with option)	C505IA2	C505T2
HP 1671A	70 MHz state 125/250 MHz timing 64K or 512K states	C505IA1	C505T1
HP 1671D/E	100 MHz state 125/250 MHz timing 64K/128K states (1M with option)	C505IA1	C505T1

To view predefined symbols for the MPC5xx

- Select the “Symbols” field on the format specification menu and then choose a label name from the “Label” pop-up. The logic analyzer will display the symbols associated with the label.

Part 5

Service Guide

To return a part to Hewlett-Packard for service

1. Follow the procedures in the "Troubleshooting..." chapters 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 get replacement parts

The repair strategy for the analysis probe and for the emulation module is board replacement. However, the following tables list some mechanical parts that may be replaced if they are damaged or lost. Contact your nearest Hewlett-Packard Sales Office for further information.

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 allows you to 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.

Analysis Probe Replaceable Parts	
HP Part Number	**Description**
E5350-61602	Flex Cable Assembly
E5346A	High-Density Termination Cable
E5322-23801	Locator Tool
E2490-69501	Circuit Board
5081-7798	160-pin QFP Probe Adapter

Emulation Module Exchange Assemblies	
HP Part Number	**Description**
E3456-69401	Programmed emulation probe

Emulation Module Replaceable Parts	
HP Part Number	**Description**
0950-3043	Power Supply
E3496-61603	10-pin target cable
E3496-61601	50-pin cable
E3497-66502	Target Interface Module

These part numbers are subject to change without notice.

To clean 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.

A

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."

D

debug port A hardware interface designed into a microprocessor that allows developers to control microprocessor execution, set breakpoints, and access microprocessor registers or target system memory using a tool like the emulation probe.

development port See *debug port*.

E

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 also *emulation probe*.

emulation probe An emulation probe is a stand-alone 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 also *emulation module*.

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 microprocessor to avoid mechanical contact with other components installed close to the target microprocessor. Sometimes called a "connector board."

F

flexible adapter Two connection devices coupled with a flexible cable. Used for connecting probing hardware on the target microprocessor to the analysis probe.

G

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.

H

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.

J

jumper Moveable direct electrical connection between two points.

L

label A name that you assign to a number of logic analysis channels. Typically, these names map to signal and/or bus names in the target system.

M

mainframe logic analyzer A logic analyzer that resides on one or more board assemblies installed in an HP 16500, HP 1660-series, 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.

P

preprocessor See *analysis probe*.

pod A collection of logic analyzer channels associated with a single cable and connector.

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 interface and powerful analysis capabilities. Replaced by HP 16600A/16700A-series logic analysis systems.

R

run control probe See *emulation probe* and *emulation module*.

S

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 stand-alone logic analyzer has a predefined set of hardware components which provide a specific set of capabilities. It is designed to perform logic analysis. A stand-alone 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.

state analysis When the logic analyzer is configured to capture data synchronously with a clock signal in the target system.

T

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 (TIM) 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*.

timing analysis When the logic analyzer is configured to capture data at a rate determined by an internal sample rate clock, asynchronous to signals in the target system.

trigger specification A set of conditions that must be true before the instrument triggers. See the printed or on-line 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

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.

Index

Symbols

% prefix, 123
? prefix, 123
?cmpd instruction, 124

Numerics

1/4-flexible adapter, 288
10-pin cable, 81
10-pin ribbon cable, 22
10-pin target cable, 283
160-pin PQFP microprocessor, 49
160-pin QFP probe adapter, 283
16-BIT predefined symbol, 96
32-BIT predefined symbol, 96
505 Reconstruction Tool, 119
50-pin cable, 79, 81, 226, 227, 283
64-bit mode instructions, 123

A

AACK- label, 96, 236
access size, 163
access to source code files, 130
accessories required, analysis probe, 234
acquire data, 196
activity indicators, 136, 140
adapters, 138
add immediate instructions, 124
add immediate shifted instructions, 124
added capacitance, 32
ADDR ACK predefined symbol, 96
ADDR label, 98, 124, 140, 236, 245, 246
ADDR values, 118
ADDR[0:29] signals, 235
ADDR[0] signal, 234
ADDR[1:5] signals, 234
ADDR[6:29] signals, 234
address, 252
address mask, 94
address offset field, 107
address phase, 105, 123
address range, 252
address range access, 108
address reconstruction, 23, 27, 88, 89
align, 105
analysis arm, 208
analysis mode
  changing, 100
  state-per-ack, 100
  state-per-clock, 100, 117
  timing, 100
analysis probe, 2, 100, 134, 136, 140, 145, 196, 223, 285
  accessories required, 234
  configuration DIP switches, 88
  configuring, 88
  connecting high-density cables, 264
  connecting target system, 49
  connecting to logic analyzers, 264
  connector, 140
  dimensions, 30
  electrical characteristics, 234
  emulation module connection, 79
  environmental characteristics, 235
  equipment required, 23
  equipment supplied, 20
  flexible cable connection, 53
  high-density cable connection, 55
  humidity, 235
  loading, 138
  logic analyzer connection, 54
  logic analyzers supported, 234
  messages, 143
  microprocessor clock speed, 234
  microprocessor compatibility, 234
  non-volatile memory, 90
  operating altitude, 235
  operating characteristics, 234
  operating temperature, 235
  package supported, 234
  power requirements, 234
  preparing target for, 30
  probes required, 234
  problems, 137
  repair strategy, 282
  replaceable parts, 283
  signal line loading, 234
  signals generated by, 239, 243, 244
  signal-to-connector mapping, 238
  theory of operation, 235
  troubleshooting, 133
  using with other logic analyzers, 259
ARETR- label, 236
ARETRY signal, 234
arm, 142, 146, 208, 210
arm event, 208, 210
arm signal, 207
ASCII format symbol file, 252
assembly code, 184
assembly language mnemonics, 91
assembly-level listing, 196
!ASYNC_STAT 173! error message, 222
asynchronous sampling, 102
AT[0:1] signals, 234
AT0-1 label, 236

B

B16/32 label, 96, 236
background mode, 213, 221
background monitor, 216
base address, 94
Base Address registers, 89
BB- label, 236
BDIP- label, 96, 236
BDIP mode, 27
BDIP signal, 234

BDM communication speed, 160
BDM connector, 81
BE[0:3] signals, 234
BE0-3 label, 236
BG- label, 236
BI- label, 96, 236
big-endian mode, 27, 123
binary, 123
BKG LED, 227
boot code for inverse assembly, 30
boot firmware, 214
Boundary Scan Master Test, 229
BR- label, 236
branch instructions, 124, 127
branch relative instruction, 124
branches not taken, 127
branches taken, 127
Break button, 89
break emulator on trigger, 199
Break In type, 161
break into monitor, 162, 207
break temporarily, 207
breakpoint, 203, 209, 220
  triggering analyzer on, 209
breakpoint address, 221
breakpoint registers, 209
breakpoints, 166, 193, 205
  clearing, 150, 184
  hardware, 193, 221
  setting, 150, 164, 184
  software, 209
Breakpoints window, 164
buffered signals for state analysis, 100, 101, 102
built-in commands, 213, 215, 216, 218
  on-line help, 216
built-in performance verification test, 43
built-in terminal interface, 155, 157, 159, 213
burst, 116
BURST IN PROGRESS predefined symbol, 96
BURST INHIBIT predefined symbol, 96
BURST- label, 236
Burst mode, 123
BURST predefined symbol, 96
BURST signal, 234
bus activity, processor, 203
bus cycles, 105

C

C505IA configuration file, 269
C505IA1 configuration file, 57, 71, 74, 92, 93, 275, 278, 279
C505IA2 configuration file, 60, 65, 68, 77, 92, 93, 266, 272, 278, 279
C505IA3 configuration file, 62, 92, 278, 279
C505T1 configuration file, 58, 72, 75, 92, 93, 270, 276, 278, 279
C505T2 configuration file, 61, 66, 69, 78, 92, 93, 268, 274, 278, 279
C505T3 configuration file, 63, 92, 278, 279
caches, 141
capacitance, added, 32
capacitive loading, 138
captured data, source code associated with, 128
captured execution, displaying, 113
capturing execution, 103
CBOOT- label, 96, 236
cf dprocck, 160
cf proc, 161
cf procck, 160
cf rrt, 162
cfsave -r command, 161
cfsave -s command, 161
change-of-flow (direct and indirect) cycles, 116
channel assignments, 91
characteristics, 233
Chart tool, 196
checkstop, 222
checkstop state, 222
CHG OF FLOW/RES predefined symbol, 97
chip causing reset, 221
chip enable, 235
chip-select registers, 235
circuit board, 283
classes, 254
cleaning the instrument, 284
clearance above target board, 30
CLKOUT signal, 234
clock qualification, 105
clock rate, 213
clock signals, 135
clock speed, 160
  after reset, 160
  processor, 201
clr instruction, 124
clrlslwi instruction, 125
clrlwi instruction, 125
clrrwi instruction, 125
cmpw instruction, 124
color, 126
Command Line window, 215, 218
command script files, 150
Comments, 258
communication speed, 160
condition registers, 123
conditional traps and branches, 124
conductive foam wafer, 49
conductive plastic pin protector, 49
configuration DIP switches, analysis probe, 88
configuration file, 91, 100, 101, 102, 143, 145, 277

label descriptions, 235
configuration file names, 55
configuration files, 20, 44, 45, 92, 95, 104, 115, 141, 154, 263
loading, 91, 277
configuration items, 158
configuration options, 216
setting, 159
Configuration window, 201
configuring the logic analyzer, 277
connection diagrams, 54
control execution, 196
cooling, 138
coordinating logic analysis, 195
counter overflow, 146
CR mode, 31, 116
CR/DS label, 236, 245
CR/DS pin, 30, 31, 116
CR/DS signal, 234
crash, system, 203
crclr instruction, 125
crmv instruction, 125
crnot instruction, 125
crset instruction, 125
CSBOOT predefined symbol, 96
CSBOOT signal, 234
CT[0:3] signal, 234
CT0-3 label, 236
custom probe fixtures, 138
CYCLE TERMINATION predefined symbol, 96
cycle types, 126
cycling power, 216

D

DAR register, 220
data bus bit 0, 235
data bus reset configuration word, 235
data from trace reconstruction tool, 118
DATA label, 140, 236
data phase, 105, 123
data reads, 127
Data Show cycles, 116
DATA values, 118
data writes, 127
data-acknowledge cycles, 101
debug mode, 214, 219, 220
debug port, 3, 21, 32, 81, 285
debug port breakpoint, 220
debug port connection, 32
debug port connector, 79
debug port hard reset, 222
debug port signals, 223
debug port soft reset, 222
debugger, 25, 26, 90, 150, 155, 196, 197, 203, 207, 208, 210, 213, 215, 219, 226
connections, port numbers, 169
tasks, performing common, 177, 184
using, 158
writing your own, 217
decimal, 123
deep memory logic analyzer, 129
default port number, 215
de-pipeline, 105
DER register, 222
development port
See *debug port*, 285
dimensions, analysis probe, 30
DIP switches, analysis probe configuration, 88
disable watchdog freeze, 222
disassembler, 123
display timing analysis mode data, 117
Display tools, 196
displaying captured execution, 113
dot suffix, 123
download performance, 192
download programs, 150, 163
download symbol information, 95
DS- label, 101
DS mode, 31, 116
DS pin, 88
DS signal, 240, 243
DSCK signal, 32, 223
DSDI signal, 32, 223
DSISR register, 220

E

ECROUT label, 236
elastomeric probe adapter, 285
elastomeric probing system, 20
electrical characteristics
analysis probe, 234
emulation module, 249
electrical problem, 213
electrostatic discharge, 38
E-MEM CS1 predefined symbol, 97
E-MEM CS2 predefined symbol, 97
E-MEM CS3 predefined symbol, 97
E-MEM CS4 predefined symbol, 97
E-MEM CS5 predefined symbol, 97
E-MEM CSBOOT predefined symbol, 97
E-MEM NO CS predefined symbol, 97
EMUL MEM SELECT predefined symbol, 97
Emulation Control Interface, 21, 25, 44, 83, 84, 90, 155, 159, 163, 168, 196, 201, 203, 205, 208, 210, 213, 215, 226
start from the Workspace window, 152
starting from main system window, 150, 151
using, 149, 155
emulation module, 2, 3, 88, 89, 90, 150, 196, 204, 208, 226, 227, 282, 285
built-in commands, 213
commands, entering, 155
communication, 84

configuring, 153
connected to Port Out, 206
electrical characteristics, 249
environmental characteristics, 249
equipment required, 25
equipment supplied, 21
exchange assemblies, 283
firmware, 21, 44, 219
firmware version, 84, 216
firmware, updating, 83
flash memory, 161
icon, 200
initialize, 216
installing, 38
loopback test board, 22
microprocessor compatibility, 249
operating characteristics, 249
Port In connected to, 206
port number, 215
preparing target for, 32
problems, 226
repair strategy, 282
replaceable parts, 283
slot 1, 40
status, 216
status lights, 214
stop processor on trigger, 200
target system connection, 79, 81
telnet to, 215
terminal interface, 155, 157
testing, 43, 163
trigger, 198
trigger signal, 205
triggering logic analyzers, 205
troubleshooting, 211
using the, 147
emulation modules
connected in seres, 206
emulation probe, 150, 283, 285
emulation reset, 216
emulation solution, 2
emulator command script files, 150
emulator configuration, reading or writing, 150
Emulator icon, 152
emulator status, 150
endian mode, 27, 123
entitlement certificate, 22
environmental characteristics
analysis probe, 235
emulation module, 249
equipment, 19
equipment and software
optional, 26
required, 23
supplied, 20
eqv instructions, 124
erratic trace measurements, 138
!ERROR 172! Bad status code (0xff) from the hard reset sequence, 228
!ERROR 905! Driver firmware is incompatible with ID of attached device, 219
error conditions, debugger, 179, 186
error messages, subnet mask, 224
Ethernet networks, 130
event wasn't captured, 142
examples of measurements, 197
exception handler, 221
exchange assemblies, 283
emulation module, 283
executing user code, 214
execution times, 118
expansion frame, 215
extended mnemonics, 124
extender, 285
extenders, 138
external clock source, 222
external hard reset, 222
external memory-mapped ports, 27
external reset, 222
external soft reset, 222
extlwi instruction, 125
extrwi instruction, 125

F

false trigger, 207
fetch cycle, 116
filter library code execution, 111
filters, inverse assembler, 125
firmware, 213
corrupted, 214
emulation module, 44, 83
updating, 150
firmware version, 216, 217
emulation module, 84
first emulation module, 215
fixed code offsets, 107
flash memory, 161
flex adapter, 20
flex cable assembly, 283
flexible adapter, 52, 285
flexible adapter rotation, 50
flexible cable, 49, 53
floating-point registers, 123
foam wafer, 49
Format menu, 54, 57, 74, 264
frequencies greater than 50 MHz, 138
FRZ line, 33
ftp, 130
function, 111
function calls, 203
FUNCTIONS, 255

G

GAT0-1 label, 96
GAT0-3 label, 236
gateway address, 169, 224
GBE0-3 label, 236
GBRST- label, 96, 236
GCT0-3 label, 97, 237
GDS- label, 96, 237
General Purpose ASCII (GPA)

symbol file, 97
General-Purpose ASCII (GPA) symbol file format, 251
general-purpose flexible adapter, 286
general-purpose registers, 123
generic commands, 217
glitches, 197
glossary, 285
GPA record format summary, 253
GPIO pins, 27
Green Hills debugger, 173
grounded wrist straps, 38
Group Run, 205, 206, 208, 210
GWR- label, 237

H

halt the processor, 161, 199, 221
hardware breakpoint registers, 209
hardware breakpoints, 193, 221
hardware problem, 226
header, male-to-male, 20
hexadecimal, 123
high-density adapter, 286
high-density cables, 54, 264
high-density connectors, 24, 101, 102
high-density termination cables, 20, 55, 264, 283
high-level source code, 128, 196
high-level source debugger, 25
hostname, 169
HP 16500B logic analysis system mainframe, 261
HP 16500B/C logic analysis system, 23
HP 16500B/C mainframe, 277
HP 16500C logic analysis system mainframe, 261
HP 16505A prototype analyzer, 261, 278
HP 16550A logic analyzer, 24, 54, 56, 57, 58, 59, 60, 61, 92, 143, 260, 261, 278
HP 16554/55/56/57 logic analyzers, 54, 61, 62, 63, 261
HP 16554A logic analyzer, 24, 92, 260, 278
HP 16555A logic analyzer, 24, 260
HP 16555A/D logic analyzer, 92, 278
HP 16555D logic analyzer, 24, 260
HP 16556A logic analyzer, 24, 260
HP 16556A/D logic analyzer, 92, 278
HP 16556D logic analyzer, 24, 260
HP 16557D logic analyzer, 24, 92, 260, 279
HP 1660 logic analyzers, 264, 265
HP 16600A logic analyzer, 24, 54, 64, 65, 66, 92
HP 16601A logic analyzer, 24, 54, 67, 68, 69, 92
HP 16602A logic analyzer, 24, 54, 70, 71, 72, 92
HP 1660A/AS/C/CP/CS/E/EP/ES logic analyzers, 23, 260, 279
HP 1660A/AS-series logic analyzer, 261
HP 1660C/CS/CP-series logic analyzer, 261
HP 1660E/ES/EP-series logic analyzer, 261
HP 1660-series logic analyzers, 277
HP 1661 logic analyzers, 264, 268
HP 16610A emulation module, 4, 21, 234, 249
HP 1661A/AS/C/CP/CS/E/EP/ES logic analyzers, 23, 260, 279
HP 1670 logic analyzers, 264
HP 1670A logic analyzer, 260
HP 1670A logic analyzers, 279
HP 1670A/D logic analyzers, 261, 271
HP 1670A/D/E logic analyzers, 23
HP 1670D/E logic analyzer, 260
HP 1670E logic analyzers, 261
HP 1671 logic analyzers, 264, 274
HP 16710/11/12A logic analyzer, 54, 73, 74, 75, 76, 77, 78
HP 16710A logic analyzer, 24, 92
HP 16711A logic analyzer, 24, 93
HP 16712A logic analyzer, 93
HP 1671A logic analyzer, 260
HP 1671A logic analyzers, 279
HP 1671A/D/E logic analyzers, 23
HP 1671D/E logic analyzer, 260
HP 1671D/E logic analyzers, 279
HP B4620B source correlation tool set, 2, 3, 4, 44, 104, 196, 197, 198, 200
HP E2490A analysis probe, 4, 21, 23, 24, 54, 55, 79, 88, 104, 234, 235, 238, 260, 261, 264
HP E3456A emulation probe, 234
HP E5346A high-density termination cables, 20, 54, 264
HP E5348A probe adapter, 20
HP E5350A flex adapter, 20
HP E5373A elastomeric probing system, 20, 49
HP E9485A emulation solution, 2, 4, 23
HP E9585A analysis probe, 4
HP sales office, 282
HP service center, 282
hpserv, 173
HRESET signal, 220
humidity, analysis probe, 235

I

I/O memory, reading or writing, 150
I/O window, 163
I2C Test, 229
idle states, 100
idle/wait states, 127

illegal instruction, 123, 209
image files, 118
  loading, 120
  multiple, 121
I-MEM predefined symbol, 97
In Monitor, 206, 207
In Monitor signal, 205
INCLUDE file, 184
incorrect inverse assembly, 140
incorrect signal levels, 135
initial prompt, 219
initialization code, 89, 90
initialization script, 176
inslwi instruction, 125
insrwi instruction, 125
installation,overview of, 16
installing logic analyzer modules, 37
installing software, 44, 263
installing the emulation module, 38
instruction mnemonic, 123
instruction opcode field, 123
instruction pointer, 91, 220
instruction register, 219
instruction set architecture, 123
Instruction Show cycles, 116
instruction word, 123
instructions fetched, 127
instrument, cleaning the, 284
Intel Hex file, 163
Intel Hex format, 150
intermittent data errors, 135
intermodule bus, 150
intermodule measurement, 201, 205
  problems, 142
  trigger, 161, 203
Intermodule window, 198, 200, 206, 208, 209
internal analyzer delays, 142
internal memory, 2, 118
INTERNAL REGISTER predefined symbol, 97
internal sample rate clock, 102
interrupt service routines, 161, 220
interrupts taken, 127
interrupts, non-recoverable, 220
Invasm menu, 115
inverse assembled data, 122
inverse assembler, 44, 45, 91, 92, 115, 116, 129, 141, 263, 277, 278
  file name, 91, 143
  filters, 125
  loading, 91
  not found, 143
  preferences, 94
  problems, 140
  software, 20
  will not load, 141
inverse assembly, 24, 100, 141
  boot code for, 30
  incorrect, 140
IP address, 169, 224
IP label, 116, 129
IRQ0-6 label, 237

J

J clock, 100, 101
JCLK label, 237
JTAG reset, 222
jumper, 286

K

K clock, 101, 105
KCLK label, 237, 240
keep-out area, 30, 49

L

L bit, 124
label, 286
label name, 279
label names, 91
LAN activity, 224
LAN address, 150, 215, 216, 224
LAN cable, 224
LAN communication
  verifying, 170
LAN interface problems, 224
LAN LEDs, 224
LAN name, 215
LAN port assignment, 150
LAN protocols, 130
LAN system administrators, 130
LAST Burst Mode, 27
latency, 201
latest software, 261
layout, 138
L-bus memory block, 235
li instruction, 124
library code execution, 112
line number, 111
linker map file, 252
lis instruction, 124
Listing display window, 115
Listing tool, 196
little-endian mode, 27, 123
L-MEM block, 116
L-MEM predefined symbol, 97
Load Executable window, 163
load immediate instruction, 124
load immediate shifted instruction, 124
loading configuration files, 91
loading object file symbols, 97
loading symbol information, 95
loads and stores, 127
locator tool, 283
locked status line, 140
logic, 78
logic analysis system, 224, 282
  setting up, 35
logic analysis, coordinating, 195
logic analyzer, 23, 134
  analysis probe connection, 54
  channel assignments, 91
  configuration files, 92, 95
  configuring, 87, 277

connecting analysis probe, 264
connecting high-density cables, 264
deep memory, 129
high-density cable connection, 55
label names, 91
modules, installing, 37
pods, 23, 140, 145
software version requirements, 24, 260
solving problems, 135
stopping, 201
storage qualification, 111, 112
trigger setup, 105
triggers, 104
using the, 85
logic analyzers
other, 259
supported, 23, 234
loop forever, 223
loopback connector, 228
loopback test board, 22, 226, 227, 229
loss of lock, 222

M

mainframe logic analyzer, 286
male-to-male header, 20, 52, 286
marginal timing, 135
maskable break, 161
maskable debug port breakpoint, 220
master clock dialog, 101
master clock signal, 100
measurement examples, 197
measurement initialization error, 143
measurement, viewing, 201
MEMMAP register, 89, 235
memory, 196
memory accesses, 116
testing, 163
memory bank accesses, 126, 127
Memory Disassembly window, 196
memory management units, 107
memory managers, 141
Memory Mnemonic window, 164
Memory window, 163
memory, mnemonic form, 150
memory, reading or writing, 150
memory-mapped internal registers, 116
messages, analysis probe, 143
microprocessor caches, 141
microprocessor clock speed, analysis probe, 234
microprocessor compatibility
analysis probe, 234
emulation module, 249
microprocessor modes, unsupported, 27
microprocessor select switches, 88
Microtec Research debugger, 181
mnemonics, assembly language, 91
MODCLK label, 237
modes, microprocessor, unsupported, 27
Motorola S-Record file, 163
Motorola S-Record format, 150
MPC505, 161
MPC505 target system, 88
MPC505 target systems, 88, 243
MPC509, 161
MPC509 target system, 88
mr instruction, 124
MSR register, 220
MSR.RI bit, 222
mtcr instruction, 125
MULTI Development Environment, 173
Multiplication Factor, 160

N

negative immediate operand, 124
network access to source files, 130
NFS client/server, 130
no configuration file loaded, 145
no inverse assembly, 140
no target system power, 214, 216
non-maskable break, 161, 221, 222
non-maskable breakpoint, 220
non-recoverable interrupts, 220
non-recoverable state, 221
NONRELOC attribute, 254
non-volatile memory, analysis probe, 90
nop instruction, 124
nor instructions, 124
NORMAL attribute, 254
NORMAL BUS CYCLE predefined symbol, 97
nostate, trigger on, 203
not instruction, 124
NULL pointer de-references, 197
null source register, 124

O

object file formats, 95
object file symbols, 104
loading, 97
object files, 252
occurrences remaining in level 1, 208, 210
on-chip hardware breakpoint registers, 209
one-card HP 16550A installations, 144
on-line help for built-in commands, 216
opcode fetch, 146
operating altitude, anaysis probe, 235
operating characteristics
analysis probe, 234

emulation module, 249
operating modes, processor, 116
operating system, 45
operating temperature, analysis probe, 235
option registers, 89, 235
optional equipment and software, 26
or instructions, 124
oscilloscope, 135, 142, 209, 210
  measurement, 203
  modules, installing, 37
other instructions, 127
other logic analyzers, using the analysis probe with, 259
output enable, 235
overview of installation, 16

P

package supported, analysis probe, 234
pattern generator modules, installing, 37
patterns, 105
PCON field in option registers, 235
PDWU label, 96, 237
performance verification tests, 43, 150, 213, 226, 227
  failures, 228
performance, profile system, 197
personality, 84
personality files, 44
PGA socket, 52
phase lock, 222
pin protectors, 49, 138
pipeline depth, 135
pipelined data, 123
plastic pin protector, 49
plug-in issue, 221
pods, logic analyzer, 23, 54, 140, 145, 286
point symbol, 252
poor connections, 135
Port In, 206
port number, 169, 170, 215
Port Out, 206
port replacement unit, 27
POWER DOWN WAKE UP predefined symbol, 96
power requirements, analysis probe, 234
power supply, 283
power up, 136
power-on self test, 224
power-on sequence, 137
power-ON/OFF sequence, 36, 262
pp load, 90
predefined symbols, 95, 279
preferences, inverse assembler, 94
prefetch queue, 135
prefetches, 135
preprocessor
  See *analysis probe*
preprocessor interface
  See *analysis probe*
probe adapter, 20, 51, 52, 283
  See *elastomeric probe adapter*
probe fixtures, custom, 138
probes required, analysis probe, 234
probing the target system, 47
problems
  analysis probe, 137
  emulation module, 226
  intermodule measurement, 142
  inverse assembler, 140
  LAN interface, 224
  logic analyzer, 135
  target system, 218
procedures, 255
processor activity, 203
processor bus, 210
  activity, 203
  cycles, 105
  state analysis of, 203
processor clock speed, 160, 201
processor execution
  coordinating, 195
  stopping on analyzer trigger, 198
  stops trigger the analyzer, 205
processor halt, 209, 210
  trace before, 203
  triggering on, 208
processor operating modes, 116
processor probe, 166
  See *emulation probe*
processor revision, 219
processor support package, 45, 91
processor type, 161
product number, 216
profile system performance, 197
program counter, 196, 197, 207, 218
program entry point, 257
program functions, 255
program image files, 118
  loading, 120
  multiple, 121
program stack, 207
program symbols, loading, 185
program, running a, 163
programmed emulation probe, 283
prompt, initial, 219
prototype analyzer, 287
PRU SELECT predefined symbol, 97
pull-down resistors, 223
pull-up resistor for RESET, 88
pulse shape requirements, 135

Q

qualified processor clock, 208, 210

R

R0 register, 217
R1 register, 218
ranges, 105

Read Configuration button, 90
readme file, 32
reads and writes, 121
reads from memory, 116
real-time runs, 162, 213
reconstructed address, 116
record header, 253
reference, 231
regions of memory, 254
register values, initializing, 164
registers, 196
  displaying, 184
  reading or writing, 150
Registers window, 164
release notes, 32
relocated code, compensating for, 107
repair strategy, 282
repeat a command, 216
replaceable parts, 282
  analysis probe, 283
  emulation module, 283
requirements, 19
  equipment and software, 23
  software version, 24
RESERVED predefined symbol, 97
RESET button, 216
reset clock speed, 160
RESET- label, 96, 237
RESET LED, 227
RESET operation, run control-initiated, 88
RESET pin, 88
RESET predefined symbol, 96
RESET signal, 32
reset state, 214
reset the target processor, 216
reset, running from, 222
RESETOUT signal, 234
resources, 105
restore registers, 220
restrict to real-time runs, 162, 213
retainer, 51, 52
  adhering, 50
return a part, 282
revision, processor, 219
RFI (Return from Interrupt) instruction, 220
RI bit in MSR register, 220
ROM code, 193
rotate-left instructions, 125
rotation (flexible adapter), 50
rotlw instruction, 125
rotlwi instruction, 125
rotrwi instruction, 125
RSR register, 221, 222
RSTOU- label, 237
run control probe
  See *emulation module*
  See emulation probe
Run Control window, 84, 164, 199, 201, 208, 210
running a program, 163
running from reset, 222
running in background, 216
running user program, 216
Running, emulation module state, 206

S

sample period, 102
sample rate clock, internal, 102
save and restore registers, 220
SCCR register, 160
search path, source code, 112, 130
second emulation module, 215
section relocation, 255
SECTIONS, 254
segments, 254
select switches, microprocessor, 88
selected file is incompatible, 145
serial number, 282
service guide, 281
set instruction, 124
setting breakpoints, 209
setup and hold time requirements, 135
Setup Assistant, 36, 44, 48, 91, 287
Setup window, 95
setup, overview of, 16
SHOCY- label, 96, 237
show cycle, 121
show cycle information, 31, 116
SHOW CYCLE predefined symbol, 96
show cycles, 116, 118, 127
shunt connector
  See *jumper*
signal integrity, 135
signal line loading, analysis probe, 234
signal-to-connector mapping, 238
single-step, 166, 184, 216
SingleStep debugger, 188
skew, 142
skid effect, 201
slot 1 (emulation module), 40
slow clock, 213
slow clock error message, 208, 210
slow or missing clock, 145, 213
slwi instruction, 125
sockets, 137
software address, 91, 129
software breakpoints, 89, 193, 209, 221
Software Development Systems debugger, 188
software probe, 166
  See *emulation probe*
software supplied, 20
software version, 261
software version requirements, 24, 260
software watchdog, 222
software, installing, 44, 263
software, latest, 261
solution, 287

source code, 128, 196
  associated with captured data, 128
  search path, 112
  triggering on, 111
source correlation tool set, 3, 25, 128, 130
  included software, 22
source debugger, 25
source files
  network access to, 130
  search path, 130
  version control, 130
source line trigger, stopping processor execution, 198
SOURCE LINES, 257
Source Viewer window, 111, 197, 198, 200
Source window, 198
source-level listing, 197
special-purpose registers, 123
specifications, 233
speed, processor clock, 160, 201
speed, reset clock, 160
S-record format, 120
SRESET signal, 220
SRR0 register, 220, 221
SRR1 register, 220, 221, 223
srwi instruction, 125
stack, program, 207
stand-alone logic analyzer, 287
START ADDRESS, 257
STAT label, 140, 237
STAT values, 118
state analysis, 23, 24, 54, 55, 91, 203, 287
  buffered signals for, 100, 101, 102
state analyzer, 210
state measurements, 264
state of the processor, 196
state-per-ack mode, 100
  changing to, 101
state-per-clock mode, 100, 117
  address range access, 108
  changing to, 100
  considerations, 105
  valid write cycles, 108
status display, 208
status lights, 214
status lines, locked, 140
status signals, 234
  additional, 24
status, emulation module, 216
step a single instruction, 217
stop on any trigger, 200
stop the analyzer, 201
storage qualification, 111, 112, 135, 141
subi instruction, 124
subnet, 224
subnet mask, 224
  error messages, 224
subroutines, 255
subtract from instructions, 124
subtract immediate, 124
SUP DATA SPACE predefined symbol, 96
SUP INST SPACE predefined symbol, 96
supplied equipment and software, 20
supported logic analyzers, 23
suppressing all other operations, 126
suppressing idle states, 126
symbol file, 128
symbol file format, General-Purpose ASCII (GPA), 251
symbol information, 252
  loading, 95
symbol name, 252
Symbol Selector dialog, 98
Symbols tab, 95
symbols, displaying, 122
sync proc, 90
synchronization, 140
synchronous address and data phases, 105
synthesized address, 116
SYPCR register, 222
System Admin window, 224
system administrators, 130
system crash, 203
System Performance Analyzer tool, 196
system performance, profile, 197

T

TA- label, 96, 101, 237
TA signal, 234, 240
TADDR label, 116, 237
target board, 79
  clearance above, 30
  keep-out area, 30
target control port, 287
target hardware, 221
target interface module, 3, 4, 21, 226, 282, 283, 287
target power, 216
Target Probe Feedback Test, 228
target processor, reset, 216
target reset, 216
target resets itself, 221
target system, 134, 140, 196
  connecting analysis probe, 49
  emulation module communication, 84
  emulation module connection, 79, 81
  MPC505, 88, 243
  MPC509, 88
  preparing, 29
  probing, 47
  problems, 218
  testing, 163, 223
  won't boot up, 137
TCK label, 237

TCP/IP protocol, 130
TDI label, 237
TDO label, 237
TEA- label, 96, 101, 237
TEA signal, 234, 240
telnet, 90, 130, 215, 218, 227
  port numbers, 169
temporary breaks, 207
terminal interface, 155, 157, 213, 215
termination cables, high-density, 20
TEST 5: Target Probe Feedback Test, 228
TEST 6: Boundary Scan Master Test, 229
TEST 7: I2C Test, 229
test the target system, 223
test, performance verification, 43
testing, emulation module, 43
theory of operation, analysis probe, 235
threshold level, 135
TIM
  See *target interface module*
time from arm greater than 41.93 ms, 146
timing analysis, 23, 24, 54, 55, 91, 116, 203, 234, 245, 246, 288
  unbuffered signals for, 100, 101, 102
timing analysis mode, 100
  changing to, 102
  data, displaying, 117
timing analyzer, 209, 210
timing measurements, 264
timing requirements, 135, 138
TMS label, 237
tools, 196
Torx T-10 screwdriver, 22
Torx T-15 screwdriver, 22
trace about this line, 198
trace reconstruction tool, 2, 118
  adding, 119
trace until processor halt, 205
transition board, 288
trap instruction, 221
trigger condition, 135
trigger in an intermodule measurement, 161
trigger on anystate, 209
trigger on nostate, 203
trigger pattern, 146
trigger point, 203, 208, 209
trigger position, 203
trigger sequence, 106, 109
trigger sequencer specification, 136
trigger specification, 288
triggering on source code, 111
troubleshooting, 133
troubleshooting guide, 213
TRST label, 238
TS- label, 97, 238
TS signal, 234
two-card HP 16550A installations, 144

U

unbuffered address bus signals, 234
unbuffered signals for timing analysis, 100, 101, 102, 245, 246
undefined opcode, 123
undefined opcodes, 121
unexecuted prefetches, 118
unimplemented opcodes, 123
unknown state, 216
unneeded information, 126
unsupported microprocessor modes, 27
unused prefetch, 136
unwanted triggers, 135
updating emulation module firmware, 83
USER DATA SPACE predefined symbol, 96
User Defined Symbols tab, 95
USER INST SPACE predefined symbol, 96
USER LED, 227
user-defined symbols, 95

V

valid pins on every clock, 105
valid write cycles, 108
variable, 111
VARIABLES, 256
verify emulation module communication, 84
version control, source file, 130
version information, emulation module firmware, 84
version requirements, 260
  software, 24
VF/VFLS branches, 127
VF/VFLS conditional branches, 127
VF/VFLS pins, 100
VF/VFLS signals, 118
VF/VFLS states, 127
VF[0:2] pins, 105
VF0-2 label, 238
VFLS label, 238
VFLS[0:1] pins, 105
VFLS0 signal, 33
VFLS1 signal, 33
view a measurement, 201

W

wait states, 100, 121
waiting for trigger, 146
watchdog timer, 222
Waveform display, 117
web site, 261
word-aligned addresses, 146

WP pin, 100
WP[0:5] pins, 105
WP0-5 label, 238
WR- label, 238
WR signal, 234
write enable, 235
writes to memory, 116

X

X Windows server software, 166, 171
XFER ACK predefined symbol, 96
XFER ERR ACK predefined symbol, 96
XFER START predefined symbol, 97
xor instructions, 124
XRAY HP Probe, 181
X-Window client/server, 130

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:	PowerPC 505/509 Analysis Probe
Model Number(s):	HP E2490A
Product Option(s):	All

conforms to the following Product Specifications:

Safety:	IEC 1010-1:1990+A1 / EN 61010-1:1993	
	UL3111	
	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 801-4:1998 / 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, 10/22/97

Ken Wyatt, Product Regulations 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
UL3111
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

Immunity

EN50082-1	Code[1]	Notes[2]
IEC 801-2 (ESD) 8kV AD	3	1
IEC 801-3 (Rad.) 3 V/m	1	
IEC 801-4 (EFT) 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:
1 The target cable assembly is sensitive to ESD events. Use standard ESD preventative measures to avoid component damage.

Sound Pressure Level

NA

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 U.S.A.

declares, that the product

Product Name:	Emulation Probe
Model Number(s):	HP E3456A
Product Option(s):	All

conforms to the following Product Specifications:

Safety:	IEC 1010-1:1990+A1 / EN 61010-1:1993 UL3111 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 801-4:1998 / 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, 03/22/97

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
UL3111
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)

Immunity

EN50082-1	Code[1]	Notes[2]
IEC 801-2 (ESD) 8kV AD	1	1
IEC 801-3 (Rad.) 3 V/m	1	
IEC 801-4 (EFT) 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:
1. The target cable assembly is sensitive to ESD events. Use Standard ESD preventative practices to avoid component damage.

Sound Pressure Level

N/A

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 of 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.

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 80901-2197, U.S.A.

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 MPC505/509 User's Guide*.

Publication number
E2490-97003, May 1999
Printed in USA.

Print history is as follows:
First edition, July 1998
Second edition, December 1998
Third edition, May 1999

New editions are complete revisions of the manual. Many product updates do not require manual changes, and manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.