

About this Manual

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

Support for Your Product

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

HP References in this Manual

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

User's Guide

Publication number E3458-97001 October 1998

© Copyright Hewlett-Packard Company 1994-1998

All Rights Reserved

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



Emulation for Motorola CPU32

Motorola CPU32 Emulation—At a Glance

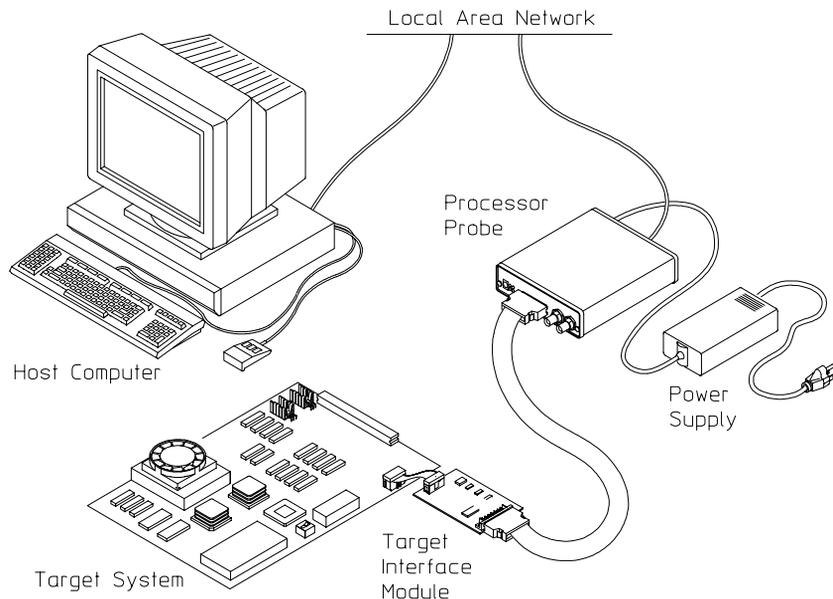
This manual describes how to set up several HP emulation products: an emulation probe, an emulation module, and an emulation migration.

These emulators provide a low-cost way to debug embedded software for Motorola CPU32 microprocessors. The emulator lets you use the target processor's built-in background debugging features, including run control and access to registers and memory. A high-level source debugger can use the emulator to debug code running on the target system.

You can connect the emulator to an analysis probe or you can connect it to a debug port on the target system through the provided target interface module (TIM). The emulator can be controlled by a debugger on a host computer or by the Emulation Control Interface on an HP16600A/700A-series logic analysis system.

Emulation Probe

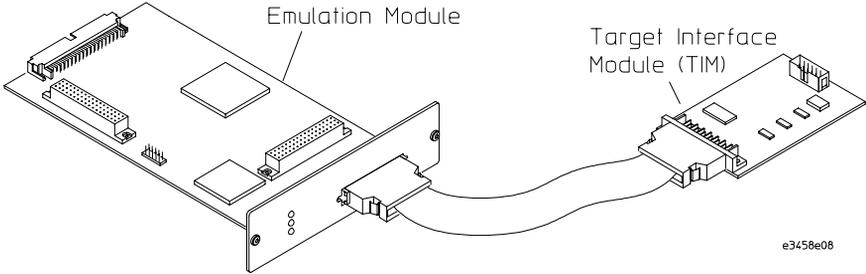
The emulation probe is a stand-alone emulator.



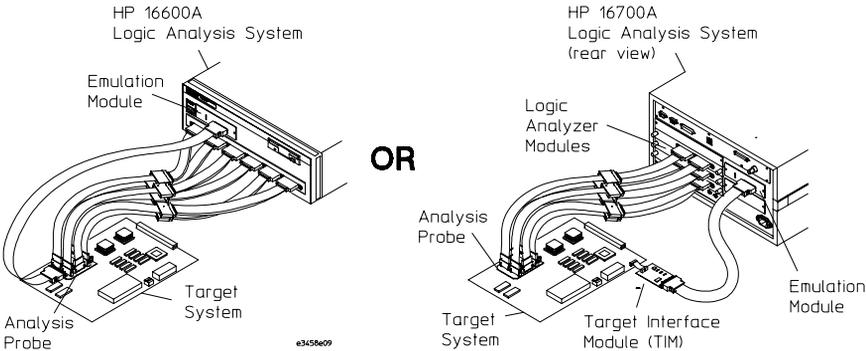
e3458e02

Emulation Module

The emulation module plugs into your HP16600A/700A-series logic analysis system frame.



You can connect the emulation module to an analysis probe or you can connect it to a debug port on the target system through the provided target interface module (TIM).



Emulation Migration

The emulation migration includes a target interface module and firmware. Use the emulation migration if you already have an emulation probe or an emulation module for another processor.

In This Book

This book documents the following products:

Emulation Probe		
Processor supported	Product ordered	Includes
CPU32	HP E5900A Option #030	HP E3458A emulation probe, HP E3458A target interface module (TIM)

Emulation Module		
Processor supported	Product ordered	Includes
CPU32	HP E5901A Option #030	HP 16610A emulation module, HP E3458A target interface module (TIM)

Emulation Migration		
Processor supported	Product ordered	Includes
CPU32	HP E5902A Option #030	HP E3458A target interface module (TIM)

Motorola CPU32 Emulation—At a Glance 2

1 Overview 15

Overview 16

Setup Flowchart 17

Emulation Probe 18

Equipment supplied 18

Minimum equipment required 20

To connect the emulation probe to a power source 20

To power on the system 22

To power off the system 22

Connection Sequence 22

Emulation Module 23

Equipment supplied 23

Minimum equipment required 24

Emulation Migration 25

Equipment supplied 25

Minimum equipment required 26

Additional Information Sources 27

2 Connecting the Emulation Probe to a LAN 29

Connecting the Emulation Probe to a LAN 30

Contents

Setting Up a LAN Connection to a PC or Workstation	31
To obtain an IP address	32
To configure LAN parameters using the built-in terminal interface	34
To configure LAN parameters using BOOTP	37
To set the 10BASE-T configuration switches	40
To verify LAN communications	41
Setting Up a Serial Connection	42
To set the serial configuration switches	42
To connect a serial cable	44
To verify serial communications	46

3 Installing the Emulation Module 47

Installing the Emulation Module	48
To install the emulation module in an HP 16700A-series logic analysis system or an HP 16701A expansion frame	49
To install the emulation module in an HP 16600A-series logic analysis system	51

4 Installing Software on an HP 16600A/700A 53

Installing Software on an HP 16600A/700A	54
Installing and loading	54
What needs to be installed	55
To install the software from CD-ROM (HP 16600A/700A)	56
To list software packages which are installed (HP 16600A/700A)	57

5 Connecting and Configuring the Emulator 59

Connecting and Configuring the Emulator 60

Using the Emulation Control Interface 61

To start the Emulation Control Interface from the main System window 62

To start the Emulation Control Interface from the Workspace window 63

To start the Emulation Control Interface from the Workspace window for an emulation probe 64

Designing a Target System for the Emulator 65

Debug port connections 65

8-pin BDM port 67

10-pin BDM port 67

Target V_{DD} 67

CONFIG2 pin 68

Enabling BDM 68

To test the emulator 68

Connecting the Emulator to the Target System 69

To connect to a target system using a 10-pin debug port 70

To connect to a target system via an 8-pin debug port 71

To connect to a target system using an analysis probe 73

To verify communication between the emulator and target system 74

Configuring the Emulator 75

What can be configured 75

To configure using the Emulation Control Interface 76

To configure using the built-in commands 77

To configure using a debugger 78

To configure the processor type 79

To configure the processor clock speed (BDM communication speed) 80

To set the default clock rate if the processor clock rate is less than 8 MHz 81

Detailed information about processor clock rates 83

To configure restriction to real-time runs 85

Contents

To configure the Trigger Out BNC (Emulation Probe Only)	85
To configure the Trigger In BNC (Emulation Probe Only)	86
Testing the emulator and target system	87
To test memory accesses	87
To test with a running program	87

6 Using Internal Registers (SIM and EMSIM Registers) 89

Internal Registers (SIM and EMSIM Registers)	90
The purpose of SIM Registers	90
The purpose of EMSIM registers	90
Configuring the SIM Registers	91
Summary	91
How SIM Register Values are Set	91
The effect of processor type on the EMSIM registers	92
Using the Emulation Control Interface or built-in commands	92
Configuring EMSIM Register Values	93
To copy target SIM registers to EMSIM registers	93
To manually define EMSIM values	93
Configuring SIM Register Values	95
To copy EMSIM registers to target SIM registers	95
To manually define SIM values	96
Saving and Loading EMSIM Values	97
To save EMSIM values in a configuration file	97
To load EMSIM values from a configuration file	97
Configuring SIM and EMSIM Values Using Built-In Commands	99

Contents

To compare SIM and EMSIM registers	99
Summary of EMSIM-related built-in commands	100
Internal Representation of SIM and EMSIM Registers	101

7 Using the Emulator with a Debugger 103

Using the Emulator with a Debugger	104
------------------------------------	-----

Setting up Debugger Software	107
------------------------------	-----

To connect the logic analysis system to the LAN	108
To change the port number of an emulation module	110
To verify communication with the emulator	111
To export the logic analysis system's display to a workstation	112
To export the logic analysis system's display to a PC	113

Using the Green Hills debugger	114
--------------------------------	-----

Compatibility	114
---------------	-----

Overview	114
----------	-----

Getting started	114
-----------------	-----

To configure the emulator, analysis probe, and target using an initialization script	118
--	-----

To perform common debugger tasks	120
----------------------------------	-----

To send commands to the emulator	120
----------------------------------	-----

To view commands sent by MULTI to the emulator	120
--	-----

To reinitialize the system	121
----------------------------	-----

To disconnect from the emulator	121
---------------------------------	-----

Error conditions	121
------------------	-----

Using the Software Development Systems debugger	123
---	-----

Compatibility	123
---------------	-----

Overview	123
----------	-----

Startup behavior	123
------------------	-----

Getting started	124
To send commands to the emulator	130
Download performance	131
Error conditions	132

8 Using the Analysis Probe and Emulation Module Together 133

Using the Analysis Probe and Emulation Module Together	134
What are some of the tools I can use?	134
Which assembly-level listing should I use?	134
Which source-level listing should I use?	135
Where can I find practical examples of measurements?	135
Triggering the Emulation Module from the Analyzer	136
To stop the processor when the logic analyzer triggers on a line of source code (Source Viewer window)	136
To stop the processor when the logic analyzer triggers (Intermodule window)	137
To minimize the “skid” effect	138
To stop the analyzer and view a measurement	138
Tracing Until the Processor Halts	140
To capture a trace before the processor halts	140
Triggering the Logic Analyzer from the Emulation Module	141
The emulation module trigger signal	141
Group Run	142
To trigger the analyzer when the processor halts	144
To trigger the analyzer when the processor reaches a breakpoint	145

9 Updating Firmware 147

Updating Firmware 148

Emulation Probe Firmware 149

To display current firmware version information 149

To update firmware for an emulation probe 149

Emulation Module Firmware 150

To display current firmware version information 150

To update firmware for an emulation module using the Emulation Control Interface 150

To update firmware for an emulation module using the Setup Assistant 151

10 Specifications and Characteristics 153

Processor compatibility 154

Emulation probe electrical characteristics 155

Emulation Probe and Emulation Module Electrical Characteristics 156

Emulation probe environmental characteristics 157

Emulation module environmental characteristics 157

11 Troubleshooting the Emulator 159

Troubleshooting the Emulator 160

Troubleshooting Guide 161

Status Lights 162

Contents

Emulation Module Status Lights	162
Emulation Probe Status Lights	163
Emulation Probe Status Lights	164
Emulator Built-in Commands	165
To telnet to the emulation module	165
To telnet to the emulation probe	166
To use the built-in commands	167
Problems with the BDM Connection	169
If a user interface behaves erratically	169
Problems with Configuration	170
If you have problems displaying some registers	170
If you have problems initializing some registers	170
Problems with the Target System	171
If boot area accesses fail	171
If the target system does not run	171
Problems with the LAN Interface (Emulation Module Only)	172
If LAN communication does not work	172
If it takes a long time to connect to the network	172
Problems with the Emulation Module	174
To run the built-in performance verification test using the logic analysis system (Emulation Module Only)	174
To run complete performance verification tests using a telnet connection (Emulation Module Only)	175
If a performance verification test fails	176
Problems with the LAN Interface (Emulation Probe Only)	177
If you cannot verify LAN communication	177
If you have LAN connection problems	178
If the "POL" LED is lit	179
If it takes a long time to connect to the network	179

Contents

Problems with the Serial Interface (Emulation Probe Only)	180
If you cannot verify RS-232 communication	180
If you have RS-232 connection problems with the MS Windows Terminal program	181
Problems with the Emulation Probe	182
To run the power up self test	182
To run the emulation probe performance verification tests	184
To run the performance verification tests using the logic analysis system	184
To run complete performance verification tests for an emulation probe	185
If a performance verification test fails	187
Returning Parts to Hewlett-Packard for Service	190
To return a part to Hewlett-Packard	190
To obtain replacement parts	191
To clean the instrument	192
Glossary	193

Contents

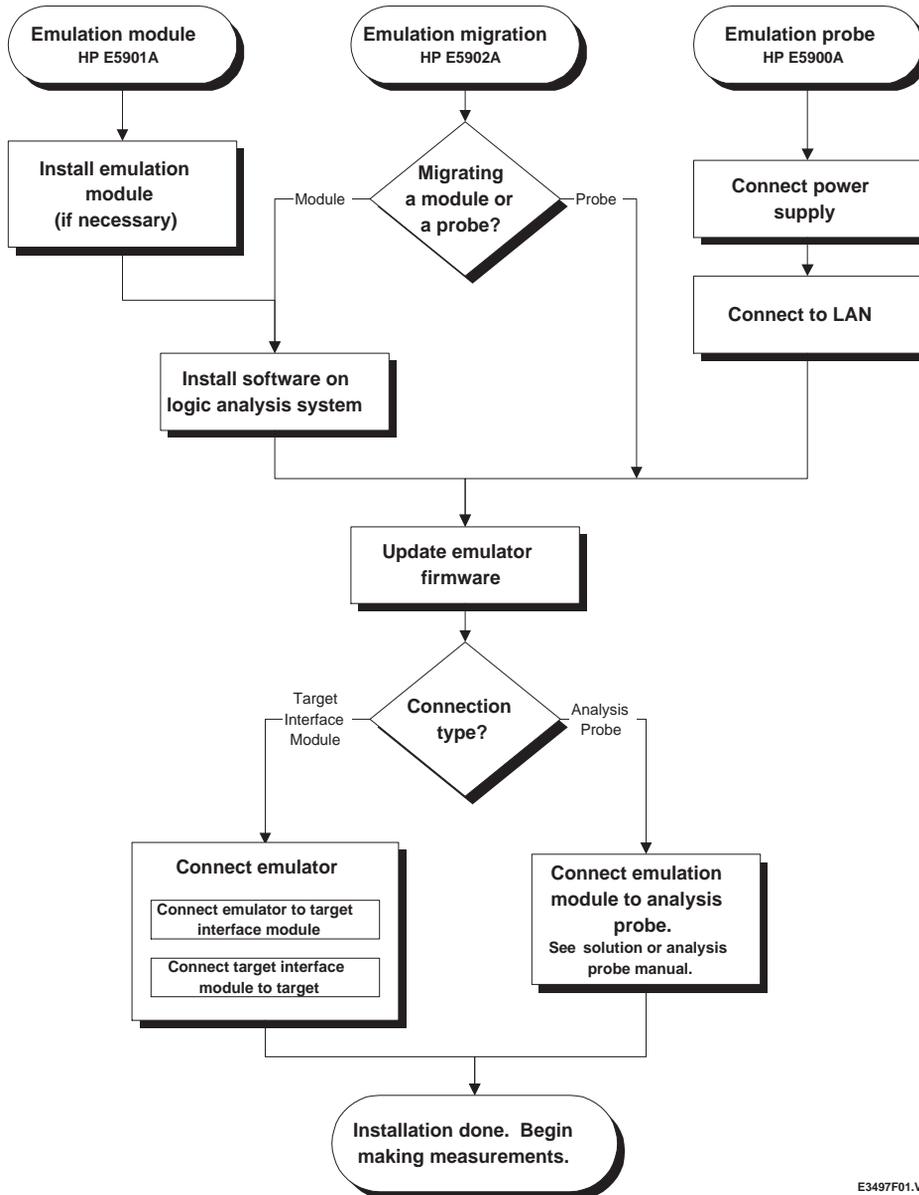
Overview

Overview

This chapter describes:

- Setup flowchart
- Equipment used with the emulation probe
- Connection sequences for the emulation probe
- Equipment used with the emulation module
- Additional information sources

Setup Flowchart

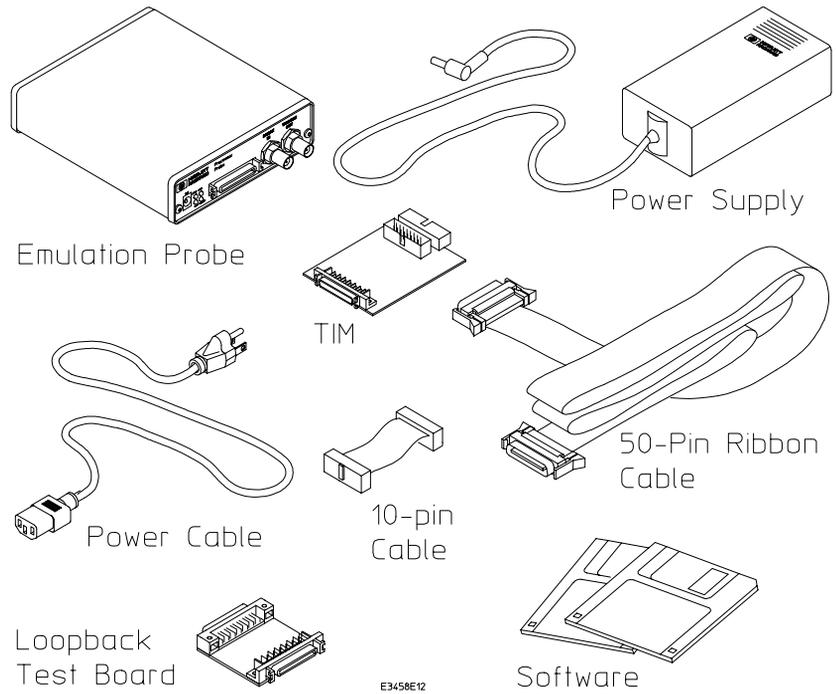


E3497F01.VSD

Emulation Probe

Equipment supplied

- An emulation probe.
- A 12V power supply for the emulation probe.
- A power cord.
- A target interface module (TIM) circuit board.
- An emulation probe loopback test board (HP part number E3496-66502).
- A 50-pin ribbon cable (connects the emulation probe to the target interface module).
- A 10-pin ribbon cable (connects the target interface module to your target system).
- Firmware for the emulation probe on 3.5-inch disks.
- This User's Guide.



Equipment Supplied with the Emulation Probe

Minimum equipment required

The following equipment is required to use the emulation probe:

- A method for connecting the emulator to the target system. You can use an HP analysis probe or you can design a debug port connector on the target system. The target system must meet the criteria on page 65.
- A host computer, such as a PC or workstation. You can also connect the emulation probe to an HP 16600A or HP 16700A logic analysis system.
- A LAN (local area network) to connect the emulation probe to the host computer.
- A user interface on the host computer, such as a high-level source debugger or the logic analysis system's Emulation Control Interface.

To connect the emulation probe to a power source

The emulation probe does not have an On/Off switch. To turn the emulation probe on or off, plug or unplug it from the power supply.

The emulation probe is shipped from the factory with a power supply and cord appropriate for your country. If the cord you received is not appropriate for your electrical power outlet type, contact your Hewlett-Packard sales and service office.

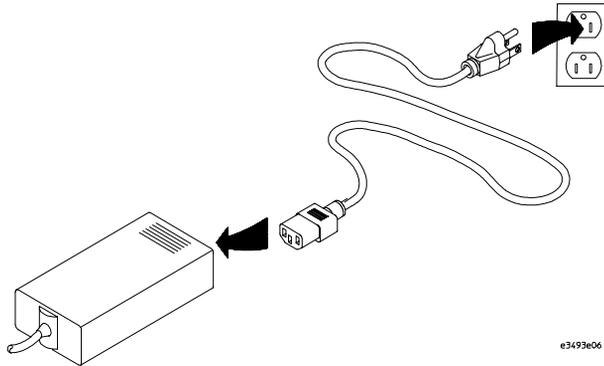
WARNING:

Use only the supplied HP power supply and cord. Failure to use the proper power supply could result in electric shock.

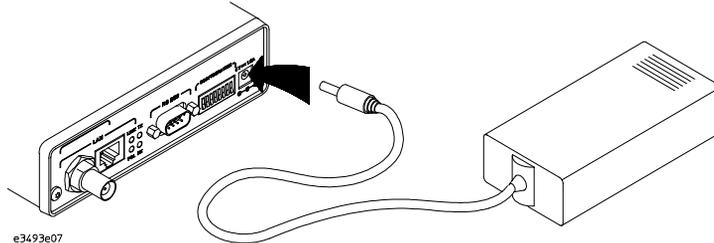
CAUTION:

Use only the supplied HP power supply and cord. Failure to use the proper power supply could result in equipment damage.

- 1 Connect the power cord to the power supply and to a socket outlet.



- 2 Connect the 12V power cord to the back of the emulation probe.



The power light on the target side of the emulation probe will be light.
The emulation probe does not have an On/Off switch.

To power on the system

With all components connected, power on your system as follows:

- 1** Logic analyzer, if you are using one.
 - 2** Emulation probe.
 - 3** Your target system.
-

To power off the system

Power off your system as follows:

- 1** Your target system.
 - 2** Emulation probe.
 - 3** Logic analyzer, if you are using one.
-

Connection Sequence

Disconnect power from the target system, emulation probe, and logic analyzer before you make or break connections.

- 1** Connect the emulation probe to a LAN (page 30).
 - 2** Connect the emulation probe to your target system (page 59).
 - 3** Configure the emulation probe (page 75).
-

Emulation Module

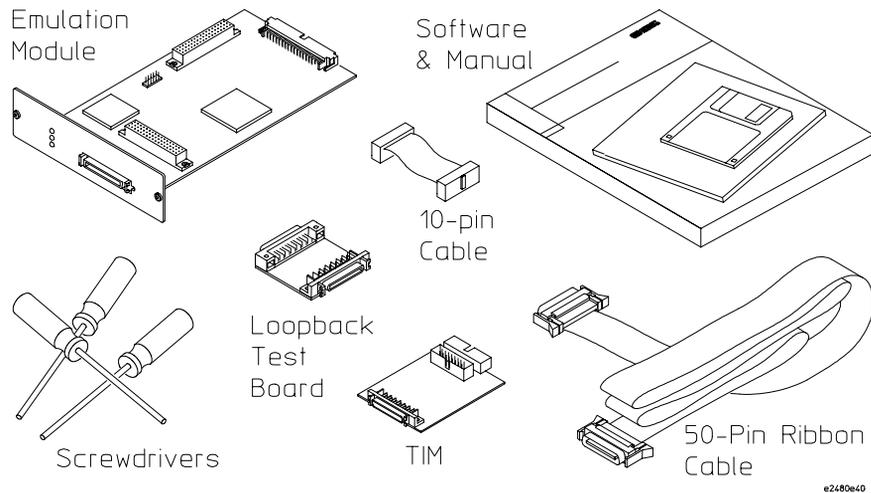
This section lists equipment supplied with the emulation module and lists the minimum equipment required to use the emulation module.

Equipment supplied

The equipment supplied with your emulation module includes:

- An HP 16610A emulation module. If you ordered an emulation module as part of your HP 16600A or HP 16700A logic analysis system, it is already installed in the frame.
- A target interface module (TIM) circuit board.
- An emulation module loopback test board (HP part number E3496-66502).
- Firmware for the emulation module and/or updated software for the Emulation Control Interface on a CD-ROM.
- A 50-pin ribbon cable for connecting the emulation module to the target interface module or the HP E2480A analysis probe.
- A 10-pin ribbon cable for connecting the target interface module to the target system.
- Torx T-8, T-10 and T-15 screwdrivers.
- This User's Guide.

Emulation Module



Equipment Supplied with the Emulation Module

Minimum equipment required

The following equipment is required to use the emulation module:

A method for connecting to the target system. The HP E2480A analysis probe provides a debug port connector. You can also design a debug port connector on the target system (see “Designing a Target System for the Emulator” on page 65).

- An HP 16600A or HP 16700A logic analysis system.
- A user interface, such as a high-level source debugger or the logic analysis system’s Emulation Control Interface.

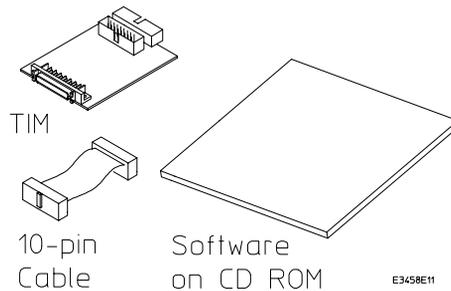
Emulation Migration

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

Equipment supplied

The equipment supplied with your emulation migration includes:

- A target interface module (TIM) circuit board.
- Firmware for the emulation module and/or updated software for the Emulation Control Interface on a CD-ROM.
- A 10-pin ribbon cable for connecting the target interface module to the target system.
- This User's Guide.



Minimum equipment required

The following equipment is required to use the emulation migration:

- An emulation module or emulation probe.
- A 50-pin data cable (supplied with the emulation module or probe).
- A method for connecting to the target system. An HP analysis probe provides a debug port connector. You can also design a debug port connector on the target system.
- A host computer such as a PC, a workstation, or an HP16600A or HP16700A logic analysis system.
- A user interface, such as a high-level source debugger or the logic analysis system's Emulation Control Interface.

Additional Information Sources

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

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

If you have an analysis probe, the instructions for connecting the probe to your target microprocessor are in the analysis probe documentation. The ***Solutions User's Guide*** for your microprocessor provides information on using the analysis probe and emulation module together.

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

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

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

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

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

Connecting the Emulation Probe to a LAN

Connecting the Emulation Probe to a LAN

You can connect your PC, workstation, or logic analysis system to the emulation probe via a serial or LAN connection.

Serial connection

A serial connection allows you to complete all of the performance verification tests and set LAN parameters. Other use of the serial port is not supported.

LAN connection

A LAN connection will allow you to make your measurements quickly and easily. A few of the performance verification tests cannot be run over a LAN.

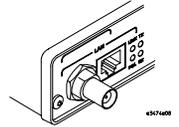
Recommended connection

Use a LAN connection for routine use, and a serial connection for LAN configuration and for troubleshooting.

Setting Up a LAN Connection to a PC or Workstation

The emulation probe has two LAN connectors:

- A BNC connector that can be directly connected to an IEEE 802.3 Type 10BASE2 cable (ThinLAN). When using this connector, the emulation probe provides the functional equivalent of a Medium Attachment Unit (MAU) for ThinLAN.
- An IEEE 802.3 Type 10BASE-T (StarLAN) connector.



Use either the 10BASE2 or the 10BASE-T connector. Do not use both. The emulation probe will not work with both connected at the same time.

You must assign an IP address (Internet address) to the emulation probe before it can operate on the LAN. You can also set other network parameters such as a gateway address. The IP address and other network parameters are stored in nonvolatile memory within the emulation probe.

The emulation probe automatically sets a subnet mask based on the subnet mask used by other devices on the network.

You can configure LAN parameters in any of the following ways:

- Using the built-in terminal interface over a serial connection. This is the most reliable method.
- Using BOOTP, which is part of the HP-UX, SunOS, and Solaris operating systems.

To obtain an IP address

1 Obtain the following information from your local network administrator or system administrator:

- An IP address for the emulation probe.

You can also use a "LAN name" for the emulation probe, but you must configure it using the integer dot notation (such as 127.0.0.1).

- The gateway address.

The gateway address is an IP address and is entered in integer dot notation. The default gateway address is 0.0.0.0, which allows all connections on the local network or subnet. If connections are to be made to workstations on other networks or subnets, this address must be set to the address of the gateway machine.

2 Find out whether port numbers 6470 and 6471 are already in use on your network and if that use constitutes a conflict.

The host computer interfaces communicate with the emulation probe through two TCP service ports. The default base port number is 6470. The second port has the next higher number (default 6471).

In almost all cases, the default numbers (6470, 6471) can be used without change. If necessary the default numbers can be changed if they conflict with some other product on your network.

To change the port numbers, see page 34. If you have already set the IP address, you can use a telnet connection instead of a serial connection to connect to the emulation probe.

3 Write down the link-level address of the emulation probe.

You will need this address if you use BOOTP to set the IP address.

The link-level address (LLA) is printed on a label above the LAN connectors on the emulation probe. This address is configured in each emulation probe shipped from the factory and cannot be changed.

IP Address of Emulation probe _____

LAN Name of Emulation Probe _____

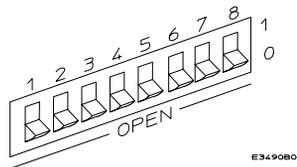
Gateway Address _____

Link-Level Address of Emulation Probe _____

To configure LAN parameters using the built-in terminal interface

- 1 Set configuration switches S1 through S4 to ON, and set the other switches as appropriate for your serial interface.

Switch settings are printed on the bottom of the emulation probe. If you will use a baud rate of 9600 baud, set the switches like this:



- 2 Connect an ASCII terminal (or terminal emulator) to the emulation probe's RS-232 port with a 9-pin RS-232 cable.

Complete instructions for setting up a serial connection begin on page 42.

- 3 Plug in the emulation probe's power cord. Press the terminal's <RETURN> key a couple times. You should see a prompt such as "p>", "?>", or "R>".

At this point, you are communicating with the emulation probe's built-in terminal interface.

- 4 Display the current LAN configuration values by entering the `lan` command:

```
R>  
lan lan is disabled  
lan -i 0.0.0.0  
lan -g 0.0.0.0  
lan -p 6470  
Ethernet Address : 08000903212f
```

The "lan -i" line shows the current IP address (IP address) of the emulation probe.

The Ethernet address, also known as the link level address, is preassigned at the factory, and is printed on a label above the LAN connectors.

- 5 Enter the following command:

```
lan -i > [-g >] [-p >]
```

The lan command parameters are:

- i <internet> The IP address which you obtained from your network administrator.
- g <gateway> The gateway address. Setting the gateway address allows access outside your local network or subnet.
- p <port> This changes the base TCP service port number.

The default numbers (6470, 6471) can be changed if they conflict with some other product on your network. TCP service port numbers must be greater than 1024. If you change the base port, the new value must also be entered in the `/etc/services` file on the host computer. For example, you could modify the line:

```
hp64700      6470/tcp
```

The IP address and any other LAN parameters you change are stored in nonvolatile memory and will take effect the next time the emulation probe is powered off and back on again.

- 6 Disconnect the power cord from the emulation probe, and connect the emulation probe to your network.

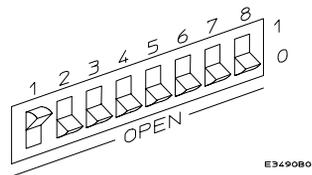
Connecting the Emulation Probe to a LAN

This connection can be made by using either the 10BASE-T connector or the 10BASE2 (BNC) connector on the emulation probe. Do not use both connectors at the same time.

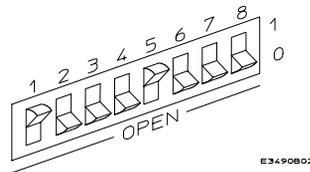
- 7 Set the configuration switches to indicate the type of connection that is to be made.

Switch S1 must be set to OFF, indicating that a LAN connection is being made.

Switch S5 should be ON if you are connecting to the BNC connector:



Switch S5 should be OFF if you are connecting to the 10BASE-T connector:



Set all other switches to ON.

- 8 Connect the power cord to the emulation probe.
- 9 Verify your emulation probe is now active and on the network. See "To verify LAN communications" on page 41.

Once you have set a valid IP address, you can use the telnet utility to connect to the emulation probe, and use the lan command to change LAN parameters.

Example

To assign an IP address of 192.6.94.2 to the emulation probe, enter the following command:

```
R>lan -i 192.6.94.2
```

Now, cycle power on the emulation probe so that the new address will take effect.

See Also

Page 172, if you have problems verifying LAN communication.

To configure LAN parameters using BOOTP

Use this method only on a workstation which is running bootpd, the BOOTP daemon.

- 1 Make sure that BOOTP is enabled on your host computer.

If the following commands yield the results shown below, the BOOTP protocol is enabled:

```
$ grep bootp /etc/services
bootps  67/udp
bootpc  68/udp
$ grep bootp /etc/inetd.conf
bootps  dgram  udp  wait  root  /etc/bootpd  bootpd
```

If the commands did not yield the results shown, you must either add BOOTP support to your workstation or use a different method to configure the emulation probe LAN parameters.

Connecting the Emulation Probe to a LAN

- 2 Add an entry to the host BOOTP database file, /etc/bootptab.

Example

```
# Global template for options common to all HP64700
# emulators and Emulation Probes.
# Use a different gateway addresses if necessary.
hp64700.global:\
    :gw=0.0.0.0:\
    :vm=auto:\
    :hn:\
    :bs=auto:\
    :ht=ether

# Specific emulator entry specifying hardware
# address
# (link-level address) and ip address.
hprobe.div.hp.com:\
    :tc=hp64700.global:\
    :ha=080009090B0E:\
    :ip=192.6.29.31
```

In this example, the "ha=080009090B0E" identifies the link-level address of the emulation probe. The "ip=192.6.29.31" specifies the IP address that is assigned to the emulation probe. The node name is "hprobe.div.hp.com".

- 3 Connect the emulation probe to your network.

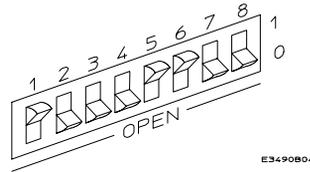
This connection can be made by using either LAN connector on the emulation probe.

- 4 Set the configuration switches to indicate the type of connection that is to be made.

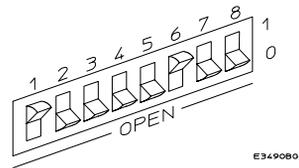
Switch S1 must be set to OFF, indicating that a LAN connection is being made.

Switch S6 must be set to OFF to enable BOOTP mode.

Switch S5 should be set to OFF if you are connecting to the BNC connector



Switch S5 should be set to ON if you are connecting to the 10BASE-T connector.



Set all other switches to ON.

- 5** Connect the power cord to the emulation probe.
- 6** Verify that the power light stays on after 10 seconds.
The IP address will be stored in EEPROM.
- 7** Set switch S6 back to ON.

Do this so that the emulation probe does not request its IP address each time power is cycled. The IP address is stored in EEPROM, so BOOTP does not need to be run again. Leaving this switch on will result in slower performance, increased LAN traffic, and even failure to power up (if the BOOTP server becomes inactive).

- 8** Verify your emulation probe is now active and on the network.
See "To verify LAN communications" on page 41.

See Also

For additional information about using bootpd, refer to the bootpd (1M) man page.

To set the 10BASE-T configuration switches

Set switches S7 and S8 to ON unless one of the following conditions is true:

- If the LAN cable exceeds the standard length, set switch S7 to OFF.

The emulation probe has a switch-selectable, twisted-pair receiver threshold. With switch S7 set to OFF, the twisted-pair receiver threshold is lowered by 4.5 dB. This should allow you to use cable lengths of up to about 200 meters. If you use a long cable, you should consult with your LAN cabling installer to ensure that:

- The device at the other end of the cable has long cable capability, and
- The cable is high-grade, low-crosstalk cable with crosstalk attenuation of greater than 27.5 dB.

When switch S7 is set to ON, the LAN port operates at standard 10BASE-T levels. A maximum of 100 meters of UTP cable can be used.

- If your network doesn't support Link Beat integrity checking or if the emulation probe is connected to a non 10BASE-T network (such as StarLAN) set this switch to LINK BEAT OFF (0 or OPEN).

In normal mode (switch S8 set to ON), a link integrity pulse is transmitted every 15 milliseconds in the absence of transmitted data. It expects to receive a similar pulse from the remote MAU. This is the standard link integrity test for 10BASE-T networks. If your network doesn't support the Link Beat integrity checking or if the emulation probe is used on a non 10BASE-T network (such as StarLAN) set this switch to LINK BEAT OFF (OFF).

NOTE:

Setting switch S8 to OFF when Link Beat integrity checking is required by your network will cause the remote MAU to disable communications.

To verify LAN communications

- 1 Verify your emulation probe is now active and on the network by issuing a telnet to the IP address.

This connection will give you access to the emulation probe's built-in terminal interface.

- 2 To view the LAN parameters, enter the lan command at the terminal interface prompt.

- 3 To exit from this telnet session, type <CTRL>D at the prompt.

The best way to change the emulation probe's IP address, once it has already been set, is to telnet to the emulation probe and use the terminal interface lan command to make the change. Remember, after making your changes, you must cycle power or enter a terminal interface init -p command before the changes take effect. Doing this will break the connection and end the telnet session.

If You Have Problems “Troubleshooting the Emulator” on page 159.

Example

```
$ telnet 192.35.12.6
R>lan
lan is enabled
lan -i 192.35.12.6
lan -g 0.0.0.0
lan -p 6470
Ethernet Address : 08000F090B30
```

Setting Up a Serial Connection

To set up a serial connection to an emulation probe, you will need to:

- Set the serial configuration switches
- Connect a serial cable between the host computer and the emulation probe
- Verify communications

Serial connections on a workstation

If you are using a UNIX workstation as the host computer, you need to use a serial device file. If a serial device file does not already exist on your host, you need to create one. Once it exists, you need to ensure that it has the appropriate permissions so that you can access it. See the system documentation for your workstation for help with setting up a serial device.

Serial connections on a PC

Serial connections are supported on PCs. You must use hardware handshaking if you will use the serial connection for anything other than setting LAN parameters.

If you are using a PC as the host computer, you do not need to set up any special files.

To set the serial configuration switches

- 1** Set switch S1 to ON (RS-232).
- 2** Set switches S2-S4 to ON.
- 3** Set switch S5 to ON (HW HANDSHAKE ON) if your serial

Setting Up a Serial Connection

interface uses the DSR:CTS/RTS lines for flow control. Set S5 to OFF (HW HANDSHAKE OFF) if your serial interface uses software flow control (XON/XOFF).

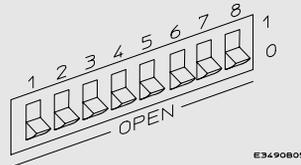
If your serial interface supports hardware handshaking, you should use it (set switch S5 to ON). Hardware handshaking will make the serial connection much more reliable.

- 4 Set switches S6-S8 for the baud rate you will use. These switch settings are listed on the bottom of the emulation probe.

The higher baud rates may not work reliably with all hosts and user interfaces. Make sure the baud rate you choose is supported by your host and user interface.

Example

To use a baud rate of 9600 baud, set the switches as follows:



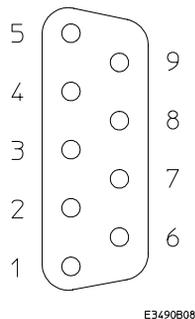
To connect a serial cable

CAUTION:

Use a grounded, shielded cable. If the cable is not shielded, or if the cable is not grounded at the serial controller, the emulation probe may be damaged by electrostatic discharge.

Connect an RS-232C modem cable from the host computer to the emulation probe. The recommended cable is HP part number C2932A. This is a 9-pin cable with one-to-one pin connections.

If you want to build your own RS-232 cable, follow the pinout shown in the following figure:



Serial Cable Pinout

Pin Number	Signal	Signal Description
1	DCD	Data Carrier Detect (not used)
2	TD	Transmit Data (data coming from HP emulation probe)
3	RD	Receive Data (data going to HP emulation probe)
4	DTR	Data Terminal Ready (not used)
5	GND	Signal Ground
6	DSR	Data Set Ready (Output from HP emulation probe)
7	RTS	Request to Send (Input to HP emulation probe)
8	CTS	Clear to Send (connected to pin 6)
9	RING	Ring Indicator (not used)

To verify serial communications

1 Start a terminal emulator program on the host computer.

If you are using a PC, the Terminal application in Microsoft Windows will work fine.

If you are using a UNIX workstation, you can use a terminal emulator such as `cu` or `kermit`.

2 Plug the power cord into the emulation probe.

When the emulation probe powers up, it sends a message (similar to the one that follows) to the serial port and then displays a prompt:

```
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.
HPE3499A Series Emulation System
Version:   A.07.53 01Mar98
Location:  Generics

HPE3490A Motorola CPU16/32 BDM Emulator
Version:   A.02.05 01Mar98
R>
```

The version numbers may be different for your emulation probe.

3 Press the Return or Enter key a few times.

You should see a prompt such as `"p>"`, `"R>"`, or `"?>"`.

For information about the commands you can use, enter `?` or `help` at the prompt.

See Also “Troubleshooting the Emulator” on page 160.

Installing the Emulation Module

Installing the Emulation Module

This chapter shows you how to install an emulation module in your HP 16600A/700A-series logic analysis system.

If your emulation module is already installed in your logic analysis system frame, you may skip this chapter.

CAUTION:

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

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

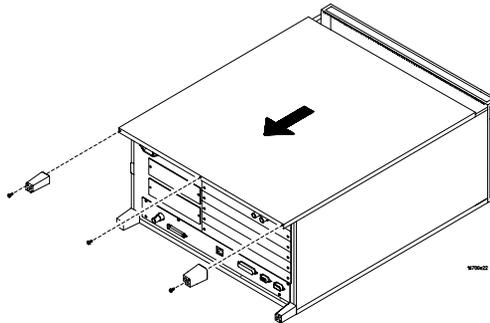
To install the emulation module in an HP 16700A-series logic analysis system or an HP 16701A expansion frame

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

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

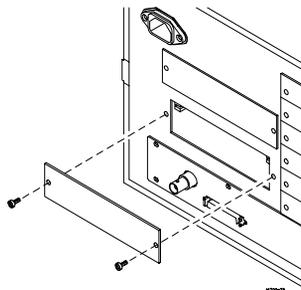
Remove any other cables (including mouse or video monitor cables).

- 2** Turn the logic analysis system frame upside-down.
- 3** Remove the bottom cover.



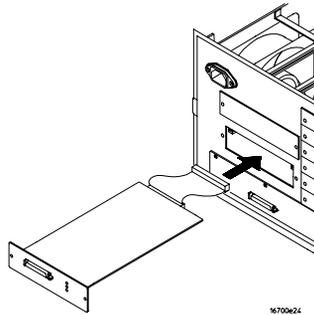
- 4** Remove the slot cover.

You may use either slot.



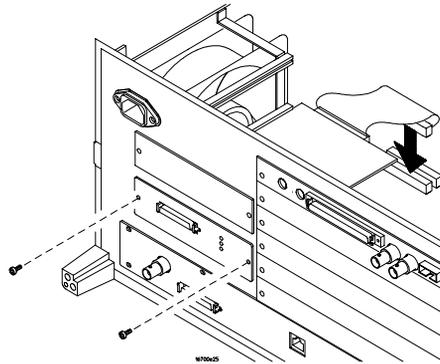
Chapter 3: Installing the Emulation Module
Installing the Emulation Module

- 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.

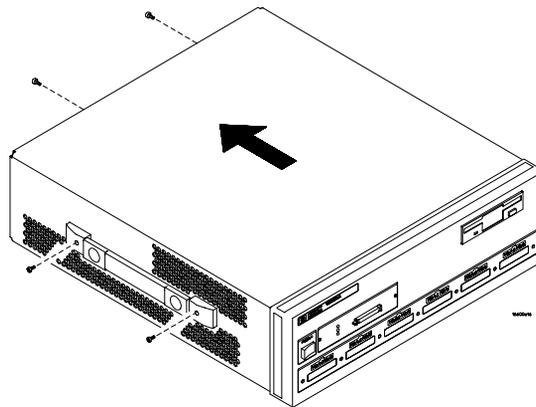
To install the emulation module in an HP 16600A-series logic analysis system

You will need T-8, T-10, and T-15 Torx screw drivers (supplied with the modules).

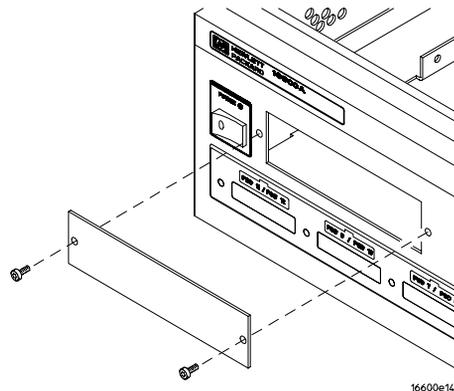
- 1 Turn off the logic analysis system and REMOVE THE POWER CORD.

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

- 2 Slide the cover back.

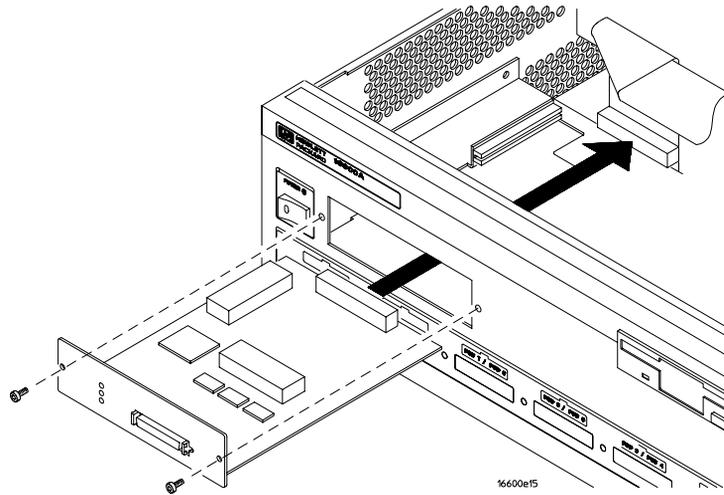


- 3 Remove the slot cover.



Installing the Emulation Module

- 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.

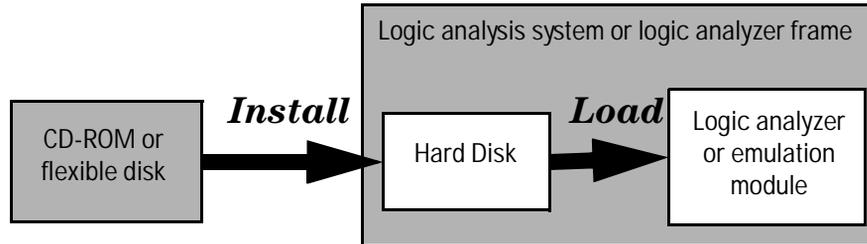
Installing Software on an HP 16600A/
700A

Installing Software on an HP 16600A/700A

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 hardware module.



What needs to be installed

HP 16600A/700A-series logic analysis systems

If you ordered an 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 the software from CD-ROM (HP 16600A/700A)

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

If the CD-ROM drive is not connected, see the instructions printed on the CD-ROM package.

1 Turn on the CD-ROM drive first and 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 available processor support packages will be displayed.

7 Click on the “M683XX” 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 online help for more information on installing, licensing, and removing software.

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

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

Connecting and Configuring the Emulator

Connecting and Configuring the Emulator

This chapter shows you how to connect the emulator to the target system and how to configure the emulator and target.

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

- 1** Make sure the target system is designed to work properly with the emulator. (page 65)
- 2** Install the emulation module in your logic analysis system, if necessary. (page 47)
- 3** Connect the emulator to your target system using the 50-pin cable and the TIM or an analysis probe. (page 69)
- 4** Update the firmware of the emulator, if necessary. (page 147)
- 5** Verify communication between the emulator and the target. (page 74)
- 6** Configure the emulator. (page 75)
- 7** Test the connection between the emulator and the target. (page 87)
- 8** Connect a debugger to the emulator, if applicable. (page 103.)

See Also

Chapter 7, “Using the Emulator with a Debugger,” beginning on page 103 for information on configuration with a debugger, and on configuring LAN port numbers.

Using the Emulation Control Interface

The Emulation Control Interface in your HP16600A/700A-series logic analysis system allows you to control an emulator (an emulation module or an emulation probe).

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

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

The Emulation Control Interface allows you to:

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

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

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

analyzer to cause a break.

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

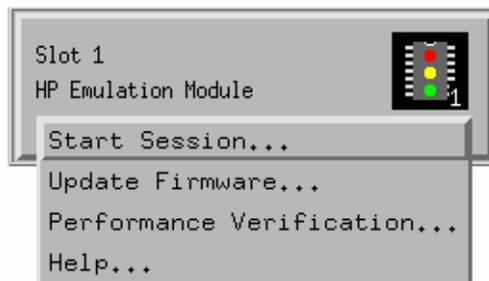
- The interfaces can get out of synchronization when commands are issued from both interfaces. This causes windows to be out-of-date and can cause confusion.
- Most debuggers cannot tolerate another interface issuing commands and may not start properly if another interface is running.

See Also

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

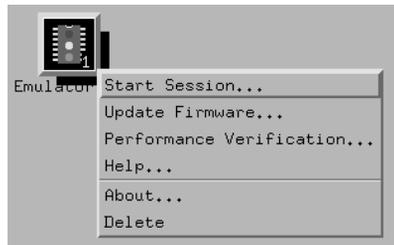
To start the Emulation Control Interface from the main System window

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



To start the Emulation Control Interface from the Workspace window

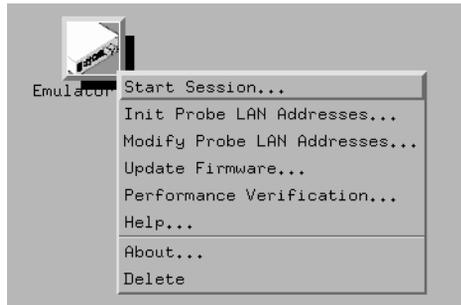
- 1 Open the Workspace window.
- 2 Drag the Emulator icon onto the workspace.
- 3 Right-click on the Emulator icon, then select **Start Session...**



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

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

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



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

Designing a Target System for the Emulator

Debug port connections

If you plan to connect the emulator 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.

Some of the signals at the BDM port share the same lines, as shown in the diagrams on the following pages.

When deciding whether to use an 8-pin or a 10-pin BDM port, consider how often you are likely to encounter "hung" bus cycles. If you are using an 8-pin BDM port, and a target bus cycle fails to terminate, you will need to reset the target system. If you use a 10-pin BDM port, the emulator will detect and terminate the "hung" cycle.

Therefore if your target system does not have a good bus monitor, or if you are not using the built-in bus monitor, you should use a 10-pin BDM port to take advantage of the \overline{DS} and \overline{BERR} signals.

The emulator adds about 40 pF to all target system signals routed to the debug connector. This added capacitance may reduce the rise time of some signals beyond the processor specifications. If so, the target may need to increase the pull-up current on these signal lines.

Designing a Target System for the Emulator

The following signals should be available at the BDM port:

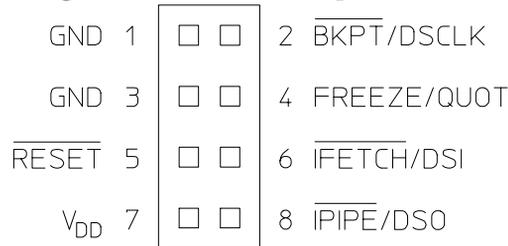
BDM signal definitions

Mnemonic	Name	Direction	Signal Description
GND	Ground		
$\overline{\text{BKPT}}$	Breakpoint	Input (to target)	Signals a hardware breakpoint. Also used to place the CPU32 in background debug mode. Active low
DSCLK	Development system clock	Input	Serial input clock
FREEZE	Freeze	Output	Indicates BDM mode
QUOT	Quotient out	Output	Quotient bit of the polynomial divider. Not used in BDM mode.
$\overline{\text{RESET}}$	Reset	Output	Indicates system reset
$\overline{\text{IFETCH}}$	Instruction fetch	Output	Indicates when the CPU is performing an instruction word prefetch and when the instruction pipeline has been flushed (active low)
DSI	Development serial in	Input	BDM data input
V_{DD}		Output	Target power (+5 V or +3.3 V)
$\overline{\text{IPIPE}}$	Instruction pipe	Output	Used to track the movement of words through the instruction pipeline (active low)
DSO	Development serial out	Output	BDM data output
$\overline{\text{DS}}$	Data strobe	Output	During read, indicates ready to receive valid data; during write
$\overline{\text{BERR}}$	Bus error	Input	Used to terminate target memory cycles (optional)

8-pin BDM port

An 8-pin BDM port should be a dual row header strip ("Berg connector"), 4 pins per row, 10 pins per inch, with 25 mil pins. If you plan to use the 10-pin cable, use a header with 2 rows of 5 pins.

If you plan to make an 8-pin cable, you should use the following pin assignments for the BDM port:



E3490B06

10-pin BDM port

If you plan to use the provided 10-pin cable, you should use the following pin assignments for the BDM port:



E3490B07

Target V_{DD}

The emulator may draw up to 10 mA from target V_{DD} .

See page 162 for more information on current and voltage requirements.

CONFIG2 pin

The CONFIG2 pin on CPU32+ processors shares a pin with the processor FREEZE output. The emulator can pull this line down and cause the CPU32+ to power up in Slave Mode with the internal CPU disabled. To ensure that CONFIG2 is high at reset, the target should supply sufficient pullup current to drive the FREEZE input on the emulator. Loading for the FREEZE pin (and all BDM pins) is described in “Emulation module—electrical characteristics” on page 227.

Enabling BDM

Your target system does not need to enable background debug mode.

If the emulator is connected and turned on before you turn on the target system, the emulator will enable BDM.

If you connect the emulator after you turn on the target system, the emulator will enable BDM when the target is reset.

If you turn on the emulation probe while it is connected to a running target system, the HPE3458A emulation probe will reset the target system.

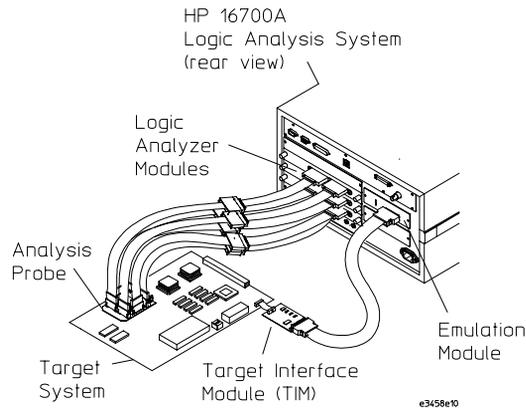
To test the emulator

If this is the first time that you have used the emulator, you should run the built-in performance verification test before you connect to a target system. Refer to page 174 for information on performance verification.

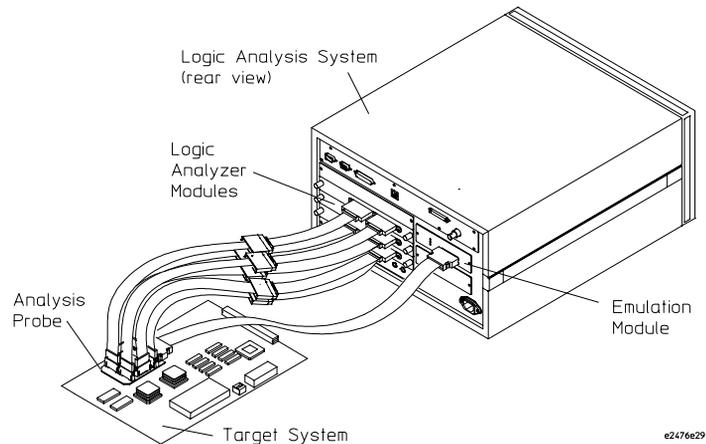
Connecting the Emulator to the Target System

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

- Directly through a debug port connector on the target board.



- Through an HP E2480A analysis probe, which provides a direct connection to the debug port pins.



Connecting the Emulator to the Target System

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

See Also

For information on designing a debug port on your target board, see page 65.

For a list of the parts supplied with the emulator, see page 23.

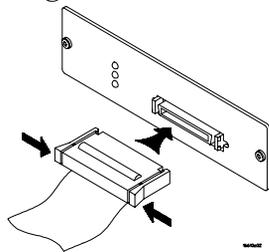
To connect to a target system using a 10-pin debug port

The emulator can be connected to a target system through a 10-pin debug port (BDM connector).

The emulator should be connected to the target system using the 10-conductor cable assembly provided.

In order to connect the emulator to the microcontroller, a 10-pin male 2x5 header connector must be available on the target system.

- 1 Remove power from the target system and the emulator.
- 2 Plug one end of the 50-pin cable into the emulator.

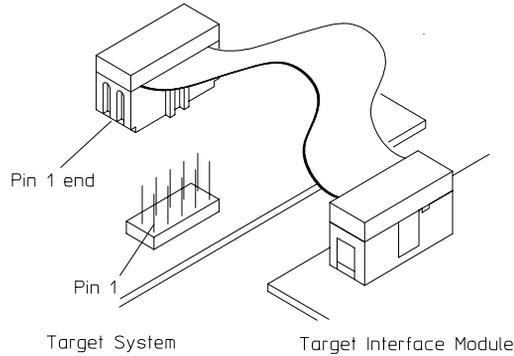


- 3 Plug the other end of the 50-pin cable into the target interface module.
- 4 Plug one end of the 10-pin cable into the target interface module.
- 5 Plug the other end of the 10-pin cable into the target system.

Orient the red wire toward pin 1 of the connector.

CAUTION:

Be careful to orient the connector as shown below. If the connector is rotated, your target system or the emulator may be damaged.



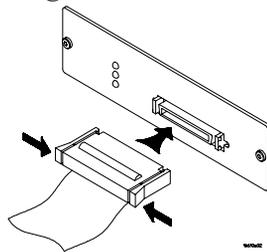
- 6 Turn on the power to the logic analysis system, then turn on the power to the target system.

See Also

“Designing a Target System for the Emulator” on page 65.

To connect to a target system via an 8-pin debug port

- 1 Remove power from the target system and the emulator.
- 2 Plug one end of the 50-pin cable into the emulator.



- 3 Plug the other end of the 50-pin cable into the target interface module.

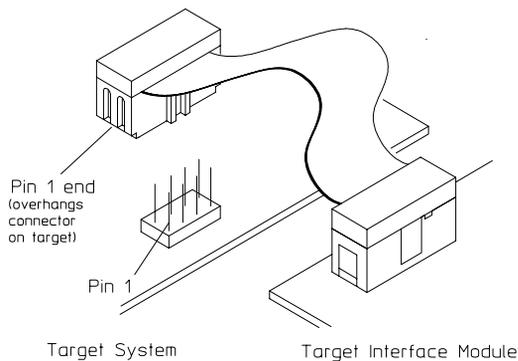
Connecting the Emulator to the Target System

- 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 target system.

Orient the red wire toward pin 1 of the connector. Pins 1 and 2 of the cable should be the ones which are not connected. Connect pin 3 of the cable to pin 1 of the target connector.

CAUTION:

Be careful to orient the connector as shown below. If the connector is rotated, your target system or the emulator may be damaged.



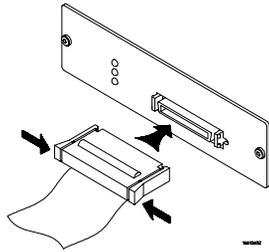
- 6 Turn on the power to the logic analysis system, then turn on the power to the target system.

See Also

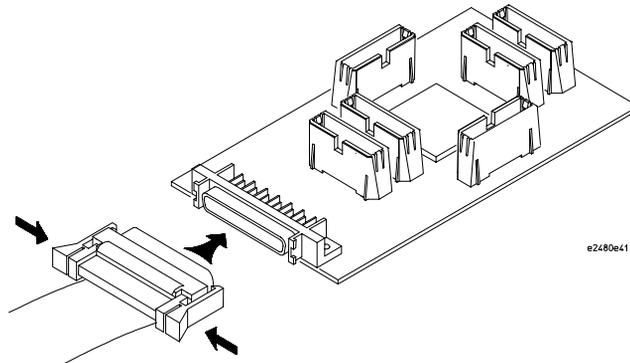
“Designing a Target System for the Emulator” on page 65.

To connect to a target system using an analysis probe

- 1 Remove power from the target system.
- 2 Plug one end of the 50-pin cable into the emulator. The connectors are keyed.



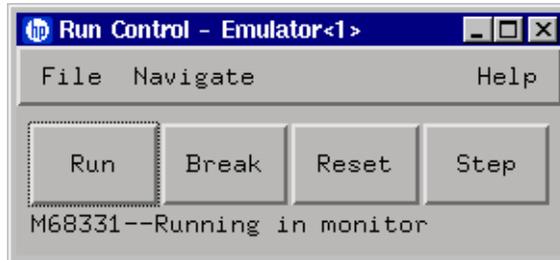
- 3 Plug the other end of the 50-pin cable into the connector on the analysis probe.



To verify communication between the emulator and target system

- 1 Turn on the target system.
- 2 Start the Emulation Control Interface.

If the electrical connections are correct, and if the emulator firmware and analysis probe or TIM match your target processor, the Run Control window should be displayed:



Configuring the Emulator

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

The easiest way to configure the emulator is through the Emulation Control Interface in an HP16600A or HP16700A logic analysis system.

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

Other ways to configure the emulator are by using:

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

What can be configured

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

- Processor type.
- Processor clock speed.
- Initial values for internal registers (see Chapter 6, "Using Internal Registers (SIM and EMSIM Registers)," beginning on page 89).
- BNC break in behavior. (Emulation probe only)
- BNC trigger out behavior. (Emulation probe only)

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

- Restriction to real-time runs.

To configure using the Emulation Control Interface

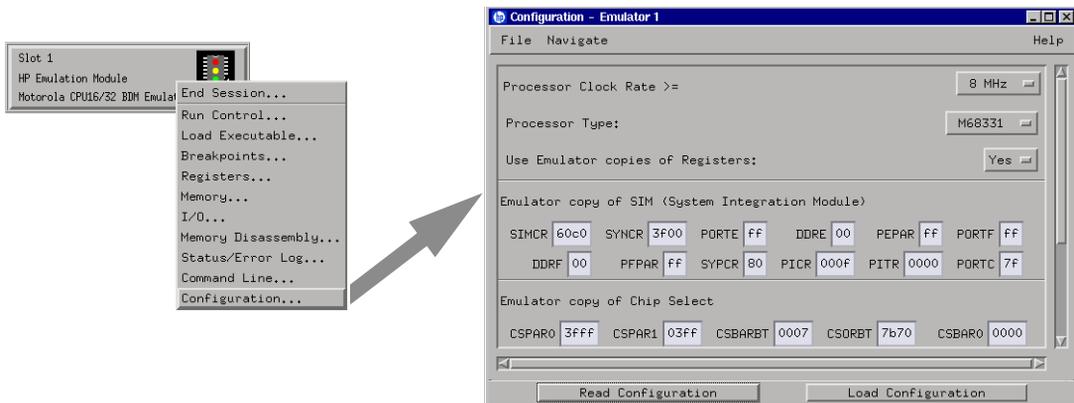
The easiest way to configure the emulator is to use the Emulation Control Interface.

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.

Configuration changes will take effect when you close the configuration window or when you move the mouse pointer outside the window.

4 Save the configuration settings.

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

See Also

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

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

To configure using the built-in commands

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

- 1** Connect a telnet session to the emulator over the LAN.

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

```
telnet LAN_address 6472
```

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

See Also

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

For information on connecting using telnet, and for information on other built-in commands, see page 165.

Example

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

```
M>help cf

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 CPU16/32 Processor
procck              - Set Clock Speed of Processor
dprocck            - Display Default Clock Speed of Processor
rrt                 - Enable or Disable Restriction to Real-Time Runs
breakin            - Select BNC break input option
trigout            - Select BNC trigger output option

M>
```

To configure using a debugger

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

See Also

Information about specific debuggers in Chapter 7, “Using the Emulator with a Debugger,” beginning on page 103.

Your debugger manual.

To configure the processor type

Processor type configuration

Value	Built-in command	Notes
CPU32 Processor Types		
68330	<code>cf proc=68330</code>	
68331	<code>cf proc=68331</code>	
68332	<code>cf proc=68332</code>	
68333	<code>cf proc=68333</code>	
68334	<code>cf proc=68334</code>	
68335	<code>cf proc=68335</code>	
68336	<code>cf proc=68336</code>	
68338	<code>cf proc=68338</code>	
68340	<code>cf proc=68340</code>	
68341	<code>cf proc=68341</code>	
68349	<code>cf proc=68349</code>	
68360	<code>cf proc=68360</code>	
68376	<code>cf proc=68376</code>	
683xx	<code>cf proc=683xx</code>	Use for other CPU32 processors
CPU16 Processor Types		
68hc16z1	<code>cf proc=68hc16z1</code>	
68hc16z2	<code>cf proc=68hc16z2</code>	
68hc16y1	<code>cf proc=68hc16y1</code>	

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

If you are using a processor in your target that is not listed as a choice, the emulator can provide direct access to all the registers defined in the CPU32 architecture programming model but will not have direct access to memory mapped registers in the processor's internal modules.

If you are using a processor that is listed as a choice, the emulator will have knowledge of on-chip peripheral registers and SIM registers and

Configuring the Emulator

will allow display and modification from the user interface. For example, when 68332 is selected as the processor type, the interface will support direct access to the SIM, the QSM, the TPU, and the TPURAM registers.

The emulator does not have explicit support for all CPU32 processors. When using a member of the CPU32 family that is not explicitly supported it may be possible to select a processor that is a formal subset of the unsupported processor. This will provide direct access to all of the internal memory mapped registers that are common. Since the registers in the internal modules are memory mapped, registers in unsupported CPU32 processors are also accessible through the memory commands. For example, on a 68332, SIM registers can be accessed at memory locations 7FFA00 to 7FFA7F or FFFA00 to FFFA7F.

To configure the processor clock speed (BDM communication speed)

The maximum communication rate with the target processor through the BDM port is based upon the target processor type and the target processor clock speed.

For best performance, set the processor clock speed to the highest speed that is equal to or less than the clock speed of the target processor. You may set the processor clock speed to a speed lower than the actual clock speed of your target system. Use the 25 MHz option for microcontrollers running faster than 25 MHz.

If using the internal clock synthesizer, set the EMSYNCR (EMulator copy of the SYNthesizer Control Register, SYNCR) to the same value as set by the initialization code. See Chapter 6, “Using Internal Registers (SIM and EMSIM Registers),” beginning on page 89.

Processor clock speed configuration

Value	Processor clock is at least	Built-in command
33	33 MHz	<code>cf procck=33</code>
25	25 MHz	<code>cf procck=25</code>
20	20 MHz	<code>cf procck=20</code>
16	16 MHz	<code>cf procck=16</code>
8	8 MHz (default)	<code>cf procck=8</code>
4	4 MHz	<code>cf procck=4</code>
2	2 MHz	<code>cf procck=2</code>
1	1 MHz	<code>cf procck=1</code>
512	512 kHz	<code>cf procck=512</code>
32	32 kHz	<code>cf procck=32</code>

Use the `cf dprocck` command to display the default clock speed.

To set the default clock rate if the processor clock rate is less than 8 MHz

If a target system's processor clock rate is less than 8 MHz following powerup, the default clock rate must be set to 131 kHz.

This can occur when the target system has the processor running off of an external clock source that is less than 8 MHz or is using the clock synthesizer with a crystal that is lower in frequency than the standard crystal.

The actual processor clock rate should then be communicated to the emulator through the configuration processor clock rate entry. The emulator will then start communications with the target processor at

Configuring the Emulator

the 131 kHz processor clock rate. When the configuration process is complete, the emulator will change the communication rate to a rate based on the clock speed (procck).

The emulator will not communicate correctly with target systems that have a processor clock rate slower than 131 kHz.

Note

The emulator does not automatically match the communication speed to the actual target speed (SYNCR register). Maintaining consistency is the responsibility of the user.

To maintain consistency, specify a correct SYNCR register value in the configuration process and make sure that the target code does not change the SYNCR register to a value that is slower than what is specified in the configuration.

The emulator will not run correctly if the actual target processor clock rate is slower than the rate specified in the configuration.

Emulation module

For an emulation module, the default clock rate must be set using the Emulation Control Interface.

- 1 End any Emulation Control Interface session for the emulation module.
- 2 Right-click the Emulation Control Interface icon and select **Update Firmware...**
- 3 Click **Modify Lan Port...**
- 4 Select the clock rate.
- 5 Click **Apply**, then click **Close**.

Emulation probe

- If your target system runs at a default clock rate less than 8 MHz but greater than 131 kHz, set switch S2 to OFF.

Detailed information about processor clock rates

Most target systems will communicate with the emulator properly and with excellent performance following the basic guidelines given in the preceding sections. In some target systems, the setting of this parameter requires greater knowledge of the actual clock generation model.

The CPU32 family has two major use models for the processor clock rate which can be used to support the majority of target systems. When using the internal clock synthesizer, the processor will run from reset at a Motorola defined default clock rate which, when using the Motorola recommended crystal is usually 8.38 MHz or 1/2 of the maximum clock rate of the processor. The programmer's initialization code then programs the clock synthesizer to run at the desired clock rate which is usually higher than the default. The second model uses an external clock source to directly control the processor clock rate. The emulator directly supports both processor clock rate models. Users that use a different clock rate model can examine the support of these models to determine the correct settings for supporting their processor clock rate.

The emulator supports the use model of the target processor clock rate being increased through the configuration. When applying the configuration at the start of a user interface or through the configuration process, the emulator communicates with the target processor at a rate based on the default processor clock rate (either 8 MHz or 131 kHz). At this default rate, it copies the EMSYNCR (EMulator copy of the SYNthesizer Control Register) to the SYNCR (SYNthesizer Control Register). The emulator then changes its communications rate to the maximum rate that the processor clock

Configuring the Emulator

rate specified in the configuration can support.

Resetting the target processor also resets the SYNCR to its default value. If the target processor is reset while running user code (as opposed to putting the processor in a reset state from the interface), no communications rate change takes place within the emulator. The initialization code that runs on the target system from reset is required to set the SYNCR to the correct value.

When the target processor clock rate is fixed through the use of an external clock source, the Processor Clock Rate parameter can be set to the highest rate that is equal to or less than the target clock rate. There are no dependencies upon any other configuration parameters.

When the target system is using the internal clock synthesizer to increase the final clock rate but the user leaves the emulator at the default processor type (683xx) the EMSYNCR is not available to the configuration process. The powerup default clock rate for the target system can be entered into the "Processor Clock Rate". This is a failsafe setting but can limit the emulator performance if the target processor is programmed to run at a significantly faster rate than the powerup default.

If the user desires the higher performance available by setting the "Processor Clock Rate" configuration parameter to his final value but does not set his processor type (which means that the configuration process cannot set the SYNCR) the debugger interface will have a number of failures when invoked because it cannot communicate accurately with the target processor. These will show up as either dashes or bad values in the source, backtrace, and memory windows. These errors can be cleared up if the target system can support a "run from reset" with code in the target system that initializes the target SYNCR register. Following the target initializing this register, a break can be requested and the emulator will communicate correctly with the target. Until the emulator and the target system are running at compatible rates, operations such as "run," "load," "modify memory/registers," or "display memory/registers" will either fail or give incorrect information.

If the user loses communication between the emulator and the target system because of incompatible clock rates, control of the target

processor through the emulator can be recovered by applying the configuration to the emulator with the target processor clock rate set to a known good value.

To configure restriction to real-time runs

Real-time runs configuration

Value	Emulator configuration	Built-in command
no	Allows commands which break to the monitor. Examples include commands which display memory or registers. (Default)	<code>cf rrt=no</code>
yes	No commands are allowed which break to the monitor, except "break," "reset," "run," or "step."	<code>cf rrt=yes</code>

To configure the Trigger Out BNC (Emulation Probe Only)

With an emulation module, this configuration item is always set to the default setting and cannot be changed with a `cf` command. The Intermodule window of the logic analysis system must be used instead.

Configuring the Emulator**Trigger out configuration**

Value	The Trigger Out BNC will	Built-in command
fixhigh	Always be high	<code>cf trigout=fixhigh</code>
fixlow	Always be low	<code>cf trigout=fixlow</code>
monhigh	Go high when the processor is running in background (Default)	<code>cf trigout=monhigh</code>
monlow	Go low when the processor is running in background	<code>cf trigout=monlow</code>

To configure the Trigger In BNC (Emulation Probe Only)

With an emulation module, this configuration item is always set to the default setting and cannot be changed with a `cf` command. The Intermodule window of the logic analysis system must be used instead.

Trigger in configuration

Value	Meaning	Built-in command
off	Inputs to the Break In BNC will be ignored.	<code>cf breakin=off</code>
rising	The emulation probe will cause a break on a rising edge. (Default)	<code>cf breakin=rising</code>
falling	The emulation probe will cause a break on a falling edge.	<code>cf breakin=falling</code>

Testing the emulator and target system

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

See Also

Chapter 11, “Troubleshooting the Emulator,” beginning on page 159, for information on testing the emulator hardware.

To test memory accesses

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

The access size is the size of memory access that will be used to write or read the memory values.

- 4 Use the Memory I/O window to stimulate I/O locations by reading and writing individual memory locations.
-

To test with a running program

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

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

RAM or flash memory.

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

The new register or breakpoint values are sent to the processor when you press the Enter key or when you move the cursor out of the selected register field.

- 4** In the Run Control window, click Run.
- 5** Use the Memory Mnemonic window to view the program and use the Memory window to view any output which has been written to memory.

Using Internal Registers (SIM and EMSIM Registers)

Internal Registers (SIM and EMSIM Registers)

The purpose of SIM Registers

The CPU32 family of processors provides a variety of internal peripheral and memory modules that are directly connected to the CPU32 core through an internal bus. These modules are configured through memory mapped register banks. The base address of the register banks as well as the base address of internal memory modules are established through Module Configuration Registers (e.g. MCR) and Base Address Registers (e.g. CSBARx). A common module throughout the family is a System Integration Module (SIM) which controls such things as clock speed and external chip selects.

The purpose of EMSIM registers

The emulator maintains a set of pseudo registers known as EMSIM registers. There is a one to one correspondence between the EMSIM registers and the target SIM registers. The purpose of the EMSIM registers is to provide a stable, known set of registers that can be copied into the SIM to establish an initial SIM state or re-establish a previous known state. This is useful because it allows emulator to communicate with a target without first running initialization code to set up the chip select registers.

The names and values of the EMSIM registers are displayed in the Configuration window of the Emulation Control Interface. The SIM registers can be viewed in the Registers window.

Note

The emulator supports configuration of the internal registers in the System Integration Module (SIM) and other important Module Configuration Registers and Base Address Registers. To simplify the interface, all configurable registers will be referred to as SIM registers even if they are technically part of another module.

Configuring the SIM Registers

Summary

If you have a boot ROM that initializes the SIM registers, you don't have to configure the EMSIM registers in order to load code and run your target. It is a good idea to configure the EMSIM registers anyway, since the EMSIM registers are used to configure an analysis probe. Page 93 discusses how to copy the SIM registers into the EMSIM registers.

If you do not have a boot ROM, then you will need to initialize the EMSIM registers first so that you can communicate with the memory of the processor. Once the EMSIM registers are defined, then every reset followed by a break will write the EMSIM registers to the processor's SIM registers.

Once you have configured the EMSIM registers, it is a good idea to save a configuration. Loading the configuration will restore the values of all configuration options, including the EMSIM registers.

How SIM Register Values are Set

These registers are typically initialized by the CPU32 executing the reset initialization code. During development this code may not be available or may not exist on the target system. To aid in development, the most important of these registers can be set directly by the emulator. This enables such functions as clock speed, chip selects, and location of internal memory to be established prior to executing any user code. Once these registers are set, resources in the target system can be accessed in the same manner as the processor would access them after executing the reset initialization code. Activities such as downloading code into the target system can now be performed through the emulator.

The emulator copy is identified by the prefix "EM" on the register name (e.g. EMSYNCR is the emulator copy of the SYNCR register) and are referred to as the EMSIM. The EMSIM registers are transferred to the processor registers when the target processor is reset while it is running in the BDM monitor.

Configuring the SIM Registers

Based on the previous discussion, it should be clear that the EMSIM values specified during configuration need to match the intended programming and of use of your CPU32 target system. You need to carefully decide how the processor will be configured and the corresponding SIM values.

The effect of processor type on the EMSIM registers

EMSIM registers are valid only if the emulator has been configured with a target processor name other than 683xx. If the processor is 683xx, the emulator does not know what type of SIM exists in the target and will not display correct values for the SIM registers.

Using the Emulation Control Interface or built-in commands

If you are using the logic analysis system's Emulation Control Interface, you should use the Emulation Control Interface to work with SIM and EMSIM registers.

If you are using a debugger, use a telnet window or your debugger's "emulator command" window to enter the emulator's built-in commands.

See Also

See "Emulator Built-in Commands" on page 165 for more information on how to use built-in commands.

Configuring EMSIM Register Values

There are two methods you can use to configure EMSIM register values:

- Copy values from the SIMs into the EMSIM registers, or
- Manually define each of the EMSIM values.

This will not change the value of the SIM registers.

To copy target SIM registers to EMSIM registers

If you have initialization code that properly defines the SIMs, you can copy your values of the SIMs into the EMSIM registers. Then you can save the EMSIM values in a configuration file.

Using the Emulation Control Interface:

- In the Configuration window, click the **Read Configuration** button.

Using the emulator's built-in commands:

- Enter the **sync sim** command.

To manually define EMSIM values

Using the Emulation Control Interface:

- 1 Open the Configuration window.
- 2 Enter the values for the registers.

Configuring EMSIM Register Values

- 3** Open the Workspace window and select **File→Save Configuration...**

The EMSIM values will be saved as part of the configuration. This allows you to restore the EMSIM values by loading the configuration.

Configuring SIM Register Values

There are three ways to configure the values of the SIM registers:

- Using code in your target's boot ROM, or
- Copying values from the EMSIM registers into the SIM registers, or
- Manually entering the value of each SIM register using the Emulation Control Interface

Some registers can only be written once after processor reset.

If you set the EMSIM values, then reset and break, the EMSIM values will be written to the SIM registers. If your initialization code then attempts to write to one of the "write once after reset" registers, the writes will fail. In this case, you must run from reset to correctly execute the initialization code.

To copy EMSIM registers to target SIM registers

You can copy values from the EMSIM registers into the SIM registers in three ways:

Using the Emulation Control Interface:

- In the Configuration window, click the **Load Configuration** button.

Using the emulator's built-in commands:

- Enter the **sync emsim** command.

By resetting the target:

- 1 Reset the target processor.

Configuring SIM Register Values

- 2 Break the target processor.

Some registers can only be written once after processor reset.

If you set the EMSIM values, then reset and break, the EMSIM values will be written to the SIM registers. If your initialization code then attempts to write to one of the "write once after reset" registers, the writes will fail. In this case, you must run from reset to correctly execute the initialization code.

To manually define SIM values

Using the Emulation Control Interface:

- 1 Open the Registers window.
- 2 Enter the values for the registers.

Once you have entered the values, it is a good idea to copy the SIM values to the EMSIM registers and save a configuration. Then you will be able to reload the SIM registers without typing all the values again:

- 3 In the Configuration window, click the **Read Configuration** button.
- 4 Open the Workspace window and select **File→Save Configuration....**

Saving and Loading EMSIM Values

You can use the Emulation Control Interface to save the EMSIM values to a configuration file then to restore the EMSIM values.

The configuration file contains more than just the EMSIM values.

When you load the configuration, the whole emulator configuration will be restored, including all configuration settings, and the locations of windows. Intermodule measurement configurations will be lost (unless you save and restore with the Source set to **All**).

To save EMSIM values in a configuration file

- 1 Open the Workspace window and select **File→Save Configuration...**
- 2 Set the **Source** field to the emulator (for example, **Motorola CPU16/32 BDM Emulator (Slot 1)**).
- 3 Select a file name and click **Save**.

To load EMSIM values from a configuration file

Once you have saved the configuration, you can specify the saved configuration file and have your EMSIMs set up to the proper values.

- 1 Open the Workspace window and select **File→Load Configuration...**
- 2 Set the **Source** field to the emulator (for example, **Motorola CPU16/32 BDM Emulator (Slot 1)**).

Saving and Loading EMSIM Values

3 Select a file name and click **Load**.

This will not change the SIM registers. To apply the new values to the corresponding SIM registers, see “To copy EMSIM registers to target SIM registers” on page 95.

Configuring SIM and EMSIM Values Using Built-In Commands

To compare SIM and EMSIM registers

Target SIM registers may be compared to the EMSIM to determine if they have changed. The only way to do this is with a built-in command:

- Enter the **sync diff** command.

This will display the differences between the SIM and EMSIM register sets.

The emulator, when comparing SIM and EMSIM registers, will not compare EMSIM registers that have not been set.

When you first turn on the emulator and the target system, **sync diff** will not find any differences. If one register is modified, just that one register may show differences. A complete check of the register differences will occur only if a complete configuration is loaded or the SIM registers are copied to the EMSIM registers.

Summary of EMSIM-related built-in commands

Command	Meaning
<code>sync sim</code>	Copy values from SIM registers to EMSIM registers
<code>sync emsim</code>	Copy values from EMSIM registers to SIM registers
<code>sync diff</code>	Display differences between SIM and EMSIM registers
<code>reset</code> <code>break</code>	Reset, break, and copy values from EMSIM registers to SIM registers

See Also

Use the **help sync** command to display help for these commands.

See “Emulator Built-in Commands” on page 165 for more information on how to use built-in commands.

Internal Representation of SIM and EMSIM Registers

Internal to the emulator the EMSIM and SIM memory spaces are accessed using memory suffixes of the form `offset@emsim` and `offset@reg`. All memory mapped registers in the 683xx family are contained within a contiguous 4k block of memory (8k for the 68360). The base address is determined by the SIM MCR MM bit in the 6833x or the MBAR in the 68340 and 68360 processors. The internal representation within the emulator of these registers is maintained as an offset to the memory mapped register base. Referencing memory as `offset@emsim` will access a SIM copy value in the EMSIM. Referencing `offset@reg` will access a register within the processor. For example, on a 68332 the memory address `00a00@emsim` will access the EMSIM MCR and `0a00@reg` will access the SIM MCR. The memory address `0@reg` within a 68340 will access its SIM MCR.

The command used to compare the values with the EMSIM and SIM will result in memory references using this notation. For example suppose the command is given to compare the SIM and EMSIM within the 68332. Further suppose that the values in the Clock Synthesizer Control differ. This is the SIM register SYNCR. The resulting display may look as follows:

```
0a04@emsim=000, 0a04@reg=03f
0a05@emsim=000, 0a05@reg=008
```

Refer to the processor data book to understand which SIM register is differing.

Using the Emulator with a Debugger

Using the Emulator with a Debugger

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

Benefits of using a debugger

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

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

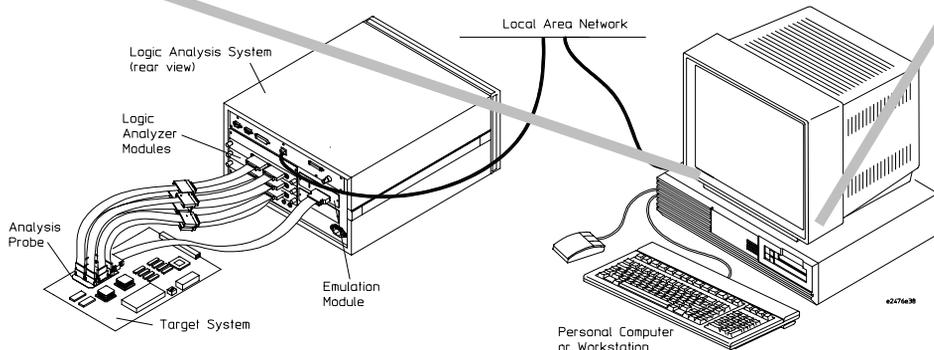
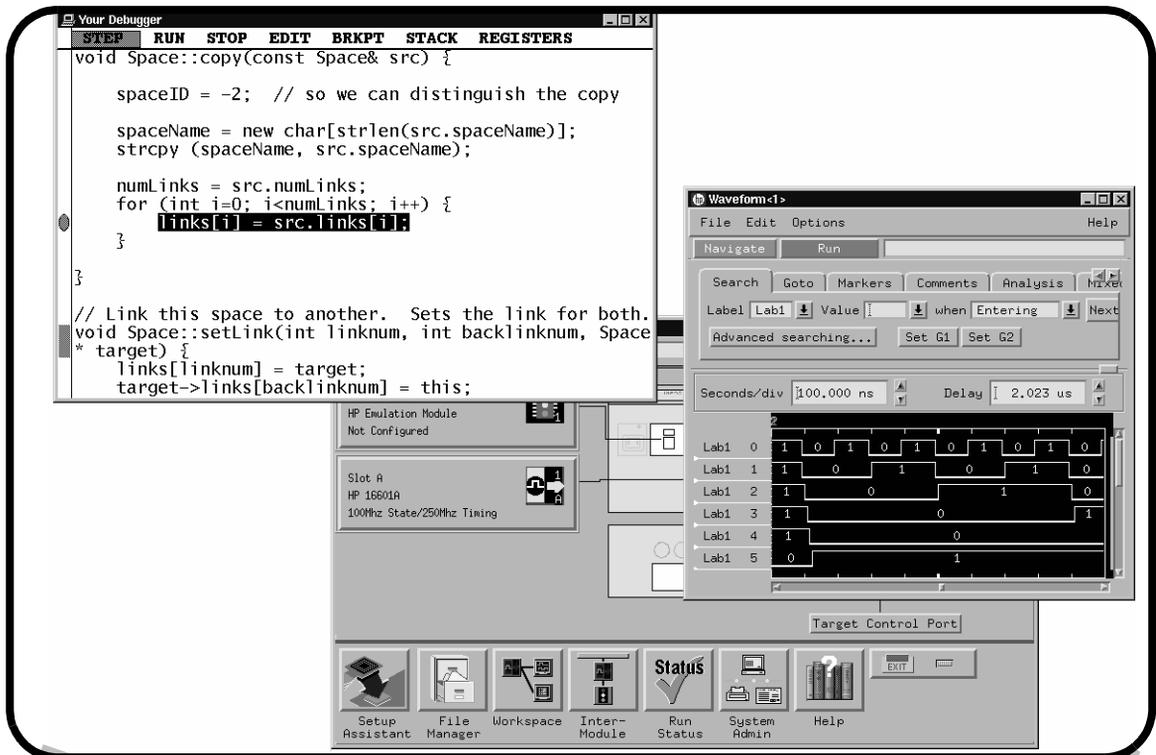
Using a debugger to connect to the emulator allows the entire design team to have a consistent interface from software development to hardware/software integration.

Debugger interfaces must be ordered directly from the debugger vendor.

Compatibility with other logic analysis system tools

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

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



Using the Emulator with a Debugger

Minimum requirements

To use a debugger with the emulator, you will need:

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

Is your debugger compatible with the emulator?

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

LAN connection

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

Compatibility with the Emulation Control Interface

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

Setting up Debugger Software

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

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

- Connect the emulator to your target system (see page 69).
- Connect the emulator or logic analysis system to the LAN (page 108).
- Export the logic analysis system's display to your PC or workstation (page 112).
- Configure the emulator (page 75).
- Begin using your debugger.

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

CAUTION:

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

See Also

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

To connect the logic analysis system to the LAN

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

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

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

Port numbers for emulators

Port number	Use for
Debugger connections	
6470	Slot 1 (First emulator in an HP 1660A/700A-series logic analysis system)
6474	Slot 2 (Second emulator in an HP 16700A-series system)
6478	Slot 3 (Third emulator in an expansion frame)
6482	Slot 4 (Fourth emulator in an expansion frame)
Telnet connections	
6472	Slot 1 (First emulator)
6476	Slot 2 (Second emulator)
6480	Slot 3 (Third emulator)
6484	Slot 4 (Fourth emulator)

Write the information here for future reference:

IP Address of Logic Analysis System _____

LAN Name of Logic Analysis System _____

Gateway Address _____

Port Number of Emulator _____

To change the port number of an emulation module

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

To view or change the port number:

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

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

To verify communication with the emulator

- 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 emulator’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 emulator and firmware version.

- 3 To exit from this telnet session, type <CTRL>D at the prompt.

See Also

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

If you have problems verifying LAN communication, see page 172.

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

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

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

```
xhost +<IP_address>
```

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

```
telnet <IP_address>
```

- 3 Log in as “hplogic”.

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

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

Example

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

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

HP Logic Analysis System

Please Log in as: hplogic [displayname:0]

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

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

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

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

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

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

Log in as "hplogic".

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

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

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

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

Using the Green Hills debugger

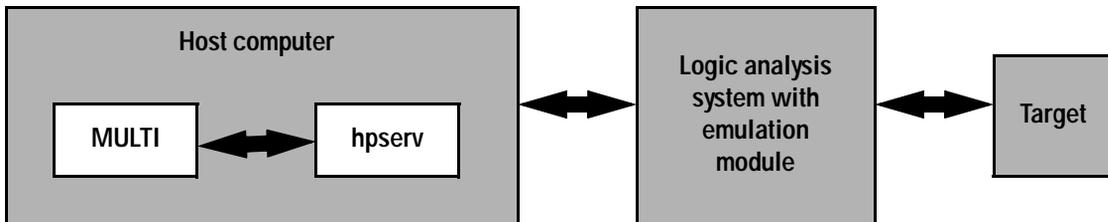
Compatibility

Version 1.8.8.A of the MULTI Development Environment from Green Hills Software, Inc. is one of several debuggers which connect to the emulator.

This section provides information that is specific to using MULTI with the emulator. It is intended to be used in conjunction with the MULTI documentation provided by Green Hills Software.

Overview

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



Getting started

- 1 Check that your emulator is programmed with firmware for a CPU32 processor.

For an emulation module, go to the system window of the logic analyzer interface and verify that the Emulation Module icon is described as a "Motorola CPU16/32 BDM Emulator". If it is not, follow the instructions on page 147 to update the firmware.

- 2 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 68000PC subdirectory where you installed MULTI. Copy the default.lnk file to user.lnk.

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/ecs.bld and press ENTER.
- Make sure the target button on the MULTI window says "cross 68".
- 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 select the default output mode.

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.

g Select **Options→Advanced**, and select the IEEE-695 output mode to generate an IEEE-695 format file.

3 Connect MULTI to the emulator.

There are two ways to connect to the emulator:

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

```
hpserv IP_address
```

OR

Using the Green Hills debugger

- 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 emulator.

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 page 108 for more information on port numbers.

4 Start the debugger.

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

5 Configure the emulator and target system.

Before running the target processor, you must configure the HP emulator 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 emulator, you can use the Configuration window in the logic analysis system's Emulation Control Interface to explore the configuration options.

Once you know the configuration settings needed for your target system, you may use one of the following methods to configure the emulator and target system:

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

See "To configure the emulator, analysis probe, and target using an initialization script" on page 118 for information on saving the configuration commands in a script.

6 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 a7=<address>
```

OR

- * Include the following line in an initialization script:

```
target reg a7=<address>
```

7 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 the emulator, analysis probe, and target using an initialization script

You can use an initialization script to configure the emulator 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 provides an example initialization sequence in the file `hpserv.rc` in the "hpdemo" directory.

- 2 To run the script, enter the following command in the Debugger command pane:

```
<filename
```

Example: simple configuration script

Create a file with the following lines:

```
remote hpserv hplogic1  
target cf proc=68360  
_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.
- Initialize the stack pointer.

Example: script to configure EMSIM and SIM registers

The following script was written for a target which does not have boot ROM and which is connected to an analysis probe. The script sets the EMSIM registers, then copies the EMSIM values to the target processor (**sync emsim**) and to the analysis probe (**pp load**) to enable address reconstruction.

```
target m -d4 0000@emmbar=01000133
target m -d4 1000@emsim=00001438F
target m -d4 1040@emsim=017CD23A0
target m -d4 1034@emsim=000000000
target m -d4 1030@emsim=000000000
target m -d2 1014@emsim=000000000
target m -d2 1016@emsim=0000000A0
target m -d2 1026@emsim=00000070F
target m -d2 102A@emsim=000000000
target m -d2 1010@emsim=000008000
target m -d1 1008@emsim=000000084
target m -d1 100C@emsim=00000008C
target m -d1 1023@emsim=000000000
target m -d1 1022@emsim=00000000E
target m -d4 1050@emsim=000000000
target m -d4 1060@emsim=0A0000001
target m -d4 1070@emsim=000000000
target m -d4 1080@emsim=000000001
target m -d4 1054@emsim=000000000
target m -d4 1064@emsim=02FE00000
target m -d4 1074@emsim=000000000
target m -d4 1084@emsim=04FE00009
target m -d4 1090@emsim=000000000
target m -d4 10A0@emsim=000000000
target m -d4 10B0@emsim=000000000
target m -d4 10C0@emsim=000000000
target m -d4 1094@emsim=0F0000004
target m -d4 10A4@emsim=0F0000004
target m -d4 10B4@emsim=0F0000004
target m -d4 10C4@emsim=0F0000004
target cf proc=68360
target sync emsim
target pp load
```

To perform common debugger tasks

- To display registers, click the **regs** button in the Display window.
- To set a breakpoint, click on the source code line where the breakpoint is to be located.
- To clear a breakpoint, 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 emulator

MULTI communicates to the emulator using the emulator's "terminal interface" commands. MULTI automatically generates and sends the commands required for normal operation. If you want to communicate directly with the emulator 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 these commands from the Debugger window's command pane by preceding the command with the "target" command.

To view commands sent by MULTI to the emulator

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

```
remote hpserv -dc -a -o <filename> <emulator>
```

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 emulator.

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 emulator is out of sync with the MULTI debugger, you may want to reinitialize it. Perform the steps below to accomplish reinitialization:

- 3 In the Target window, type:

```
init -c
```

- 4 Repeat step 5 in the "Getting started" section to configure the emulator.

To disconnect from the emulator

- In the Debugger window, select **Remote→Disconnect**.

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

Error conditions

"!ERROR 800! Invalid command: bcast"

usually means that there is not a target interface module (TIM) connected to the emulator or the emulator does not have firmware for the CPU32 family. Verify that the emulator is connected to the target. Next, go to the system window of the logic analyzer interface and verify that the Emulation Module icon (stop-light) is described as a Motorola CPU16/32 BDM Emulator. If it is not, follow the steps on page 150 to update the firmware in the emulation module.

"command socket connection failed: WSAECONNREFUSED: connection refused"

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 the Green Hills debugger

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

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

“Configuring the Emulator” on page 75 for more information on configuration options and the "cf" command.

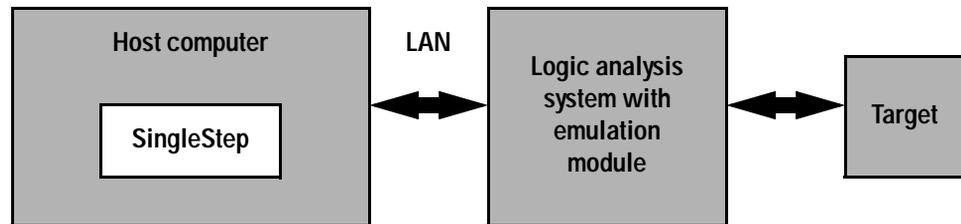
Using the Software Development Systems debugger

Compatibility

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

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

Overview



Startup behavior

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

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

Getting started

- 1 Check that your emulator is programmed with firmware for a CPU32 processor:

For an emulation module, go to the system window of the logic analyzer interface and verify that the Emulation Module icon is described as a "Motorola CPU16/32 BDM Emulator". If it is not, follow the instructions on page 147 to update the firmware.

- 2 Connect to the emulator:
 - a Start SingleStep running on your PC or workstation.
 - b When the small Debug dialog box appears in the middle of the screen, click the Connection tab and then enter the IP address of the HP logic analysis system which contains the emulation module.

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

Note: SingleStep is hard-coded to connect to the emulation module at port 6470 of the logic analysis system frame. See page 110 for more information on port numbers.

- 3 Configure the emulator 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 emulator 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 131) but will be reset to this value each time SingleStep resets the processor.

- 4 Initialize the target system.

The target system must have various registers and memory locations initialized before it can access RAM and before SingleStep can download an application. Normally, code in the target's boot ROM

performs this initialization. However, when SingleStep resets the target, it immediately places the processor in debug mode. Any initialization code which may exist on the target board has not been run.

SingleStep provides a way for target initialization to occur without running application code through the use of the "_config" alias. _config is used to define a list of commands that will be used to initialize the target after a reset. The _config alias should be defined in the sstep.ini file (in the "cmd" directory) and will point to a file of type .cfg which contains the actual initialization commands.

SingleStep provides some workspace files for some standard targets. These files will setup various registers in order to initialize these targets. The files can be found in the *init* directory and are as follows:

- EST SBC 360 - est360.wsp
- Motorola 332 EVS revision a - 332evs_a.wsp
- Motorola 332 EVS revision b - 332evs_b.wsp
- Motorola 340 EVS - 340evs.wsp
- Nohau Trg-332 - trg332.wsp
- Vesta SBC332 - vesta332.wsp

Loading one of these workspace files followed by opening the Debug Dialog from the File Menu and setting the File and Connection options then clicking OK will create a corresponding .cfg file in the *cmd* directory. This file will store the values of the items shown in the Target Configuration tab. You may wish to edit these values before clicking OK and saving them to a .cfg file.

Example

If you load the `est360.wsp` and set up the Debug Dialog options for the target you are connecting to, when you click OK, the file `68360.cfg` will be created and placed in the `cmd` directory. Comments have been added to this file, in order to explain the items (comments begin with a #).

```
Contents of 68360.cfg:
set vectbase = 0x400000
set vectaddr = 0x400000
write -l CPU:0x3FF00 = 0x08000133      # MBAR
write -l SD:0x08001000 = 0x00014F71   # MCR
write -b SD:0x08001022 = 0x0E        # SYPCR
write -l SD:0x08001040 = 0x17CD23A0   # GIMR
write -l SD:0x08001050 = 0x00000001   # BRO
write -l SD:0x08001054 = 0x3FFC0002   # OR0
write -l SD:0x08001060 = 0x00100001   # BR1
write -l SD:0x08001064 = 0x2FF80000   # OR1
write -l SD:0x08001080 = 0x00400001   # BR3
write -l SD:0x08001084 = 0x2FE00003   # OR3
write -b SD:0x0800100C = 0x8F         # CLKOCR
write -w SD:0x08001010 = 0x4000      # PLLCR
write -w SD:0x08001014 = 0x8000      # CDVCR
write -w SD:0x08001016 = 0x0000      # PEPAR
```

The `.cfg` file sets up the target's chip select registers (SIM registers) needed in order to map memory correctly. If your target is connected to an analysis probe, you will need to execute some additional commands in order to configure the analysis probe. The analysis probe stores its own copy of the chip select registers. In general, just executing the above write commands will configure the target's registers. The emulator also stores a copy of these registers. The emulator command "sync sim" listed below will copy the values of the target's SIM registers into the EMSIM registers. Next, the EMSIM registers must be copied over to the analysis probe. This is accomplished by issuing the built-in command "pp load". Enter the following two commands into the Command window:

```
control -c "sync sim"
control -c "pp load"
```

You should have the appropriate values for your target first written into

the SIM registers before issuing these two commands. “control -c” is used by SingleStep to forward a command to the emulator. For help on either the “sync sim” or “pp load” command, issue the commands, control -c “help sync” or control -c “help pp load” into the Command window. Also, note that the help pp load command will refer to the analysis probe as a preprocessor. They are one in the same.

If your target is not one that SDS ships a workspace for, add a line to the bottom of the file “sstep.ini” (which is supplied with the debugger) that sets up the _config alias to point to a .cfg file to use. This line will look like:

```
alias _config 'source ${cmdpath}68336.cfg'
```

In this case, the name of the config file to use is 68336.cfg. Edit this file specific for your targets need. Keep in mind, that commands beginning with “control -c” followed by a string of quoted text are commands that will be forwarded to the emulator. Commands that do not begin with “control -c” are SingleStep commands.

Example

Here is a configuration file which contains both emulator built-in commands and SingleStep commands:

```
# Config file for 68336 Motorola Modular Board
# First set clock speed to 8 MHz
control -c "cf procck=8"

# Set the processor id to 68336
control -c "cf proc=68336"

set vectbase = 0x000000
set vectaddr = 0x000000
# write -l CPU:0x3FF00 = 0x01000133      # MBAR
# write -l SD:0x01001000 = 0x0001438F   # MCR
# write -b SD:0x01001022 = 0x0E        # SYPCR
# write -l SD:0x01001040 = 0x17CD23A0   # GIMR
# write -l SD:0x01001050 = 0x00000000   # BR0
# write -l SD:0x01001054 = 0x00000000   # OR0
# write -l SD:0x01001060 = 0xA0000001   # BR1
# write -l SD:0x01001064 = 0x2FE00000   # OR1
# write -l SD:0x01001080 = 0x00000001   # BR3
# write -l SD:0x01001084 = 0x4FE00009   # OR3
```

Chapter 7: Using the Emulator with a Debugger

Using the Software Development Systems debugger

```
# write -b SD:0x0100100C = 0x8C           # CLKOCR
# write -w SD:0x01001010 = 0x8000        # PLLCR
# write -w SD:0x01001014 = 0x0000        # CDVCR
# write -w SD:0x01001016 = 0x00A0        # PEPAR
# write -w SD:0x01001026 = 0x070F        # PICR
# write -b SD:0x01001008 = 0x84           # AVR

# Reset, run, break in order to set up chip selects
# We can issue these commands since the target has boot
# code in ROM that sets up the chip select registers
control -c "rst"
control -c "r"
control -c "b"

# Issue the next two commands since there is an
# analysis probe connected to the target

# Copy the sim values to the emsim set
control -c "sync sim"
# Copy the emsim values to the analysis probe
control -c "pp load"
```

The config file shown in this example did not need to set up chip select registers because the target has boot code in ROM that will accomplish this. Therefore, just issuing a “reset, run, break” will execute the boot code in ROM. If your ROM does not contain boot code to set up these registers, add write commands to the .cfg file (similar to what is shown here for the 68360) to set up these values. Reset, run, break would not be used to set up a target that does not contain boot code.

Given that you added a line to the sstep.ini file that points to this configuration file, all you need to do now is to bring up the Debug Dialog and enter the file to download (if desired), and the connection port to use. Click OK when finished. This config file (the one pointed to by the added line in sstep.ini) will get executed if and only if you have not unchecked the “Reset Target” option on the Debug Dialog’s “Options” tab. This is selected by default. Deselecting this option will prevent the config file you specified in the sstep.ini file from being executed. Also, keep in mind that this file will be executed every time the Debug Dialog is terminated via the OK button when “Reset Target” is selected.

In summary, there are two ways for you to configure the emulator and your target. The first method consisted of loading an existing workspace, using the Debug Dialog to modify any items needed followed by clicking the OK button and having it create a corresponding .cfg file. If you have an analysis probe connected to this target, you need to enter the two additional commands into the Command window (control -c "sync sim", control -c "pp load"). The second method consisted of creating a .cfg file and specifying that file in the *_config* alias in the sstep.ini file. This file will automatically be loaded upon clicking OK in the Debug Dialog.

To save time, you may want to do the first method just to generate .cfg file that you can edit and use in method two. If you had an analysis probe connected you would add the two commands mentioned previously to the .cfg file thereby not having to enter them through the Command window.

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

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

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

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

Emulator 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 emulator with the communication speed specified in the "Debug" dialog.
- If "reset target" was selected then execute the commands specified by the *_reset* alias. The *_reset* alias should be used to specify commands that are

specific to initializing the processor. It is executed each time the processor is reset. The value of the `_reset` alias can be viewed by issuing a "alias `_reset`" from the command window.

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

To send commands to the emulator

To view commands sent by SingleStep

SingleStep communicates to the emulator using the emulator's "terminal interface" commands. SingleStep automatically generates and sends the commands required for normal operation. This communication between SingleStep and the emulator 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 emulator from the SingleStep command window or included in SingleStep's `.cfg` or `.dbg` command files.

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

Examples

To see the speed that the emulator 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 page 77.

Download performance

Downloads are fastest when the emulator speed is set to match that of the target processor. The initial speed that the emulator uses to communicate with the target processor is set by the Processor clock item in the connection tab of the "Debug" dialog. The user is responsible for specifying this speed to be less than or equal to the initial, reset, speed of the processor. Usually 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 emulator communication:

Example

After setting the clock rate of the target processor, the following command should be entered to increase the emulator communication speed:

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

Error conditions

"!ERROR 800! Invalid command: bcast" usually means that there is not a target interface module (TIM) connected to the emulator or the emulator does not have firmware for the CPU32 family. Verify that the emulator is connected to the target. Next, go to the system window of the logic analyzer interface and verify that the Emulation Module icon (stop-light) is described as a Motorola CPU16/32 BDM Emulator. If it is not, follow the steps on page 147 to update the firmware in the emulation module.

"command socket connection failed: WSAECONNREFUSED: connection refused"

usually means the emulation module is not at port #6470 on the Logic Analysis System. See step 2 of the getting started section above.

"unrecognized hostname"

usually means that the debugger is unable to establish communication with the emulator. Verify communication to the emulation module by doing a ping to the logic analyzer. If you are unable to ping the logic analyzer refer to page 172 for more information.

See Also

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

The *SDS SingleStep Users Guide*.

The configuration section beginning on page 75 for more information on configuration options and the "cf" command.

Using the Analysis Probe and Emulation Module Together

Using the Analysis Probe and Emulation Module Together

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

What are some of the tools I can use?

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

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

Which assembly-level listing should I use?

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

- The Listing tool shows processor states that were captured during a “Run” of the logic analyzer. Those states are disassembled and displayed in the Listing window.
- The Emulation Control Interface shows the disassembled contents of a

section of memory in the Memory Disassembly window.

- Your debugger shows your program as it was actually assembled, and (if it supports the emulation module) shows which line of assembly code corresponds to the value of the program counter on your target system.

Which source-level listing should I use?

Different tools display source code for different uses:

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

Where can I find practical examples of measurements?

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

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

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

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

Triggering the Emulation Module from the Analyzer

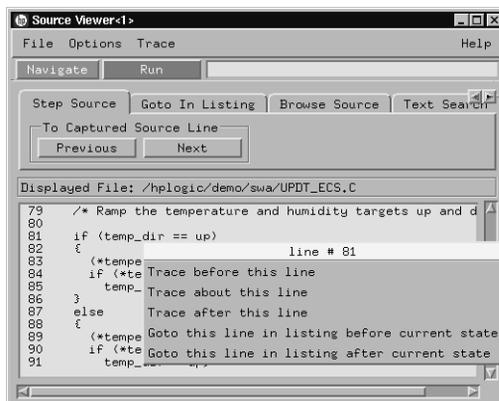
You can trigger the emulation module from the logic analyzer using either the Source Viewer window or the Intermodule window. If you are using the HP B4620B Source Correlation Tool Set, using the Source Viewer window is the easiest method.

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

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

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

The logic analyzer trigger is now set.



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

The emulation module is now set to halt the processor after receiving a

trigger from the logic analyzer.

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

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

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

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

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



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

- 3** Click **Group 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 online help for your logic analysis system for more information on setting triggers.

To minimize the “skid” effect

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

To minimize the skid effect:

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

The amount of skid will depend on the processor’s execution speed and whether code is executing from the cache. See page 80 for information on how to configure the clock speed.

To stop the analyzer and view a measurement

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

When the target processor stops it may cause the analyzer qualified

clock to stop. Therefore most intermodule measurements will have to be stopped to see the measurement.

Example

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

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

If the trigger position is not “End”, the analyzer may continue waiting for more states.

6. The user clicks Stop in a logic analyzer window, which tells the logic analyzer to stop waiting, and to display the trace.

Tracing Until the Processor Halts

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

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

Some possible uses for this measurement are:

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

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

To capture a trace before the processor halts

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

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

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

Triggering the Logic Analyzer from the Emulation Module

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

If you are only using a state analyzer to capture the processor bus then it will be much simpler to use “Tracing until processor halts” as described on page 140.

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

The emulation module trigger signal

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

The “In Monitor” trigger signal can be caused by:

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

Group Run

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

The following setups can operate independently of Group Run:

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

Here are some examples:

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

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

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

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

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



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

Debuggers can cause triggers

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

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

You can often distinguish these additional states because the time tags will be in the ms and ms range. You can use the time tag information to determine when the processor went into monitor. Typically the time between states will be in the nanoseconds while the processor is running and will be in the ms and ms range when the debugger has halted the processor and is reading memory.

Not also that some debugger commands may cause the processor to break temporarily to read registers and memory. These states that the debugger introduces will also show up in you trace listing.

Triggering the Logic Analyzer from the Emulation Module

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

To trigger the analyzer when the processor halts

Remember: if you are only using a state analyzer to capture the processor bus then it will be much simpler to use “Tracing until processor halts” as described on page 140.

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

The logic analyzer is now set to trigger on a processor halt.

- 4 Click **Group Run** to start the analyzer(s).
- 5 Click Run in the Emulation Control Interface or use your debugger to start the target processor running.

Clicking **Group Run** will *not* start the emulation module. The emulation module run, break, step, and 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 previous one, but with the one additional complexity of setting breakpoints. Be aware that setting breakpoints may cause a false trigger and that the breakpoints set may not be valid after a reset.

Remember: if you are only using a state analyzer to capture the processor bus then it will be much simpler to use “Tracing until processor halts” as described on page 140.

- 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.

Triggering the Logic Analyzer from the Emulation Module

If you are going to run the emulation module from Reset you must do a **Reset** followed by **Break** to properly set the breakpoints. The Reset will clear all on-chip hardware breakpoint registers. The Break command will then reinitialize the breakpoint registers. If you are using software breakpoints which insert an illegal instruction into your program at the breakpoint location you will not need to do the Reset, Break sequence. Instead you must take care to properly insert your software breakpoint in your RAM program location.

5 Click **Group Run** to start the analyzer(s).

6 Click **Run** in the Emulation Control Interface or use your debugger to start the target processor running.

Clicking **Group Run** will *not* start the emulation module. The emulation module run, break, step, reset are independent of the Group Run of the analyzers.

7 Wait for the Run Control window in the Emulation Control Interface or the status display in your debugger to show that the processor has stopped.

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

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

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

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

Updating Firmware

Updating Firmware

Firmware gives your emulator a “personality” for a particular processor or processor family.

After you have connected the emulator 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:

- You have an emulation module which was not shipped already installed in the logic analysis system.
- You need to change the personality of the emulator for a new processor.
- You have an updated version of the firmware from HP.

The procedure for updating firmware for an emulation probe is different from the procedure for updating firmware for an emulation module.

Emulation Probe Firmware

To display current firmware version information

- Use `telnet` or a terminal emulator to access the built-in "terminal interface" and use the `ver` command to view the version information for firmware currently in the emulation probe.

To update firmware for an emulation probe

To update the firmware, you must have access to the World Wide Web and a PC or a workstation connected to your emulation probe.

- 1 Download the new firmware from the following World Wide Web site:

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

The firmware will be in the "Technical Support Information" section of this web site.

- 2 Follow the instructions on the web site for installing the firmware.

If HP sends you firmware on a floppy disk, install the firmware from the floppy disk. The README file on the floppy disk contains instructions for installing the firmware using a PC or workstation.

Emulation Module Firmware

Always update firmware by installing a processor support package. This will ensure that the version of the Emulation Control Interface software is compatible the version of the emulator firmware.

To display current firmware version information

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

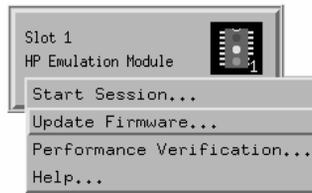
There are usually two firmware version numbers: one for “Generics” and one for the personality of your processor.

To update firmware for an emulation module using the Emulation Control Interface

- 1 End any run control sessions which may be running.
- 2 In the Workspace window, remove any Emulator icons from the workspace.
- 3 Install the processor support package from the CD-ROM, if necessary.

In the system window, click the emulation module and select

4 Update Firmware.



5 In the Update Firmware window, select the firmware to load into the emulation module.

6 Click **Update Firmware**.

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

See also

“Installing Software” beginning on page 53 for instructions on how to install the processor support package from the CD-ROM.

To update firmware for an emulation module using the Setup Assistant

The Setup Assistant is an online tool for connecting and configuring your logic analysis system for microprocessor and bus analysis. The Setup Assistant is available on the HP 16600A and HP 16700A-series logic analysis systems.

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.

Do not use the Setup Assistant to connect an emulation probe if you already have an emulation module installed.

1 Install the processor support package from the CD-ROM.

2 Start the Setup Assistant by clicking its icon in the system

Chapter 9: Updating Firmware
Emulation Module Firmware

window.



- 3** Follow the instructions displayed by the Setup Assistant.

See also

Page page 53 for instructions on how to install a the processor support package from the CD-ROM.

Specifications and Characteristics

Processor compatibility

The emulator supports Motorola 68330, 68331, 68332, 68F333, 68334, 68335, 68336, 68338, 68340, 68341, 68349, 68360, or 68376 microprocessors operating at clock speeds up to 25 MHz.

The emulator supports both 5V and 3.3V operation.

Emulation probe electrical characteristics

BNC, labeled TRIGGER OUT

Output Drive. Logic high level with 50-ohm load ≥ 2.0 V. Logic low level with 50-ohm load ≤ 0.4 V. Output function is selectable, see the configuration section (page 75).

BNC, labeled BREAK IN

Input. Edge-triggered TTL level input (active high), 20 pf, with 2K ohms to ground in parallel. Maximum input: 5V above V_{CC} ; 5 V below ground. The BNC introduces approximately 2.5 ms skid after break-in at 25 MHz. Input function is selectable, see the configuration section (page 87).

Communications

Serial Port. 9-pin female type D subminiature connector. RS-232 DCE to 115.2 kbaud.

10BASE-T LAN Port. RJ-45 connector. IEEE 802.3 10BASE-T (StarLAN).

10BASE 2 LAN Port. 50-ohm BNC connector. IEEE 802.3 10BASE2 (ThinLAN). When using this connector, the HP Emulator provides the functional equivalent of a Medium Attachment Unit (MAU) for ThinLAN.

Accessory Power Out

12 V, 3.0A, center negative

Power Supply

Input. 100-240 V, 9.75 A, 50/60 Hz, IEC 320 connector.

Output. 12 V, 3.3 A

Emulation Probe and Emulation Module Electrical Characteristics

Characteristic	Symbol	Value	Unit
Supply Voltage from Target	V_{DD}	-0.3 to +5.5	V

Characteristic	Symbol	$V_{DD} = 5 \text{ Volts}$		$V_{DD} = 3.3 \text{ Volts}$		Unit
		Min	Max	Min	Max	
Input Current (V_{DD})	I_{il}		10		6	mA
Input Voltage	V_{in}	$V_{SS}-0.5$	$V_{DD}+0.5$	$V_{SS}-0.5$	$V_{DD}+0.5$	V
Input High Voltage	V_{ih}	2	$V_{DD}+0.5$	2	$V_{DD}+0.5$	V
Input Low Voltage	V_{il}	$V_{SS}-0.5$	0.8	$V_{SS}-0.5$	0.8	V
Input High Current	I_{ih}		-20		-15	μA
Input Low Current	I_{il}		0.6		0.35	mA
Input Capacitance	C_{in}		40		40	pF
Output High Voltage ($\overline{\text{BKPT}}$, $\overline{\text{IFETCH/DSI}}$)*	V_{oh}	3.86		3		V
Output High Current ($\overline{\text{BKPT}}$, $\overline{\text{IFETCH/DSI}}$)	I_{oh}	-4		-2		mA
Output Low Voltage ($\overline{\text{BKPT}}$, $\overline{\text{IFETCH/DSI}}$)	V_{ol}	0.4		0.4		V
Output Low Current ($\overline{\text{BKPT}}$, $\overline{\text{IFETCH/DSI}}$)	I_{ol}	3.4		3.4		mA
Output Low Voltage ($\overline{\text{BERR}}$, $\overline{\text{RESET}}$), $I_{ol} = 12 \text{ mA}^*$	V_{ol}	0.4		0.4		V
Output Low Voltage ($\overline{\text{BERR}}$, $\overline{\text{RESET}}$), $I_{ol} = 24 \text{ mA}^*$	V_{ol}	0.5		0.5		V

*The V_{oh} specification for $\overline{\text{BERR}}$ and $\overline{\text{RESET}}$ is not applicable because they are OPEN-collector outputs.

Input-only pins: V_{DD} , $\overline{\text{DS}}$, FREEZE, $\overline{\text{I_PIPE/DSO}}$

Output-only pins: $\overline{\text{BKPT}}$

Input/output pins: $\overline{\text{IFETCH/DSI}}$, $\overline{\text{BERR}}$, $\overline{\text{RESET}}$

Emulation probe environmental characteristics

Temperature	Operating, +0 C to +40 C (+32 to +104 F); nonoperating, -40 to +60 C (-40 to +140 F).
Altitude	Operating/nonoperating 4600 m (15 000 ft).
Relative Humidity	15% to 95%. For indoor use only.

Emulation module environmental characteristics

The HP 16610A emulation module meets the environmental characteristics of the logic analysis system in which it is installed.

For indoor use only.

Troubleshooting the Emulator

Troubleshooting the Emulator

If you have problems with the emulator, your first task is to determine the source of the problem. Problems may originate in any of the following places:

- The connection between the emulator and your debugger
- The emulation module or emulation probe itself
- The connection between the emulator 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 emulator
- The emulator "performance verification" tests
- The emulator's built-in commands

Troubleshooting Guide

Common problems and what to do about them

Symptom	What to do	See also
Commands from the Emulation Control Interface or debugger have no effect	1 Verify LAN communication.	page 41
	2 Check that you are using the correct firmware.	
	3 Use the Emulation Control Interface or a telnet connection 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 165
Emulator built-in commands do not work	1 Check that the emulator has been properly configured for your target system.	page 75
	2 Run the emulator performance verification tests.	page 174
	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 65 page 171
"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 80
Some commands fail	Check the "restrict to real-time runs" configuration	page 85
Host computer reports LAN connection problems	Follow the checklists in this chapter.	page 177
		page 172

Status Lights

Emulation Module Status Lights

The emulation module uses status lights to communicate various modes and error conditions.

The following table gives more information about the meaning of the power and target status lights.

○ = LED is off

● = LED is on

* = Not applicable (LED is off or on)

Power/Target Status Lights

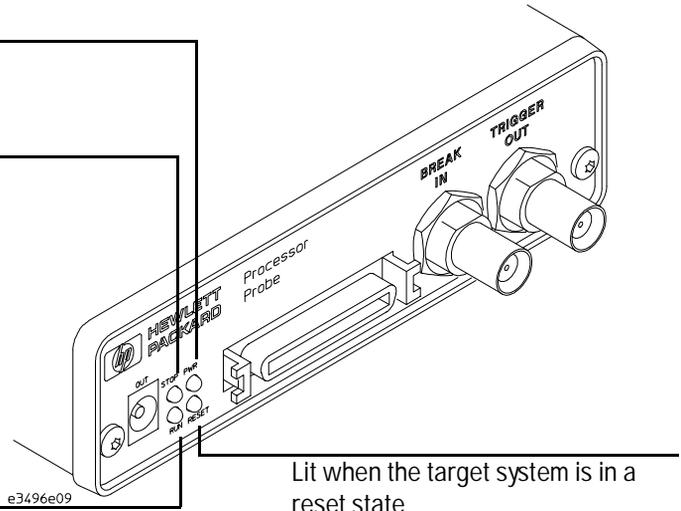
Pwr/Target LEDs	Meaning
○ Reset ○ Break ○ Run	No target system power, or emulation module is not connected to the target system
● Reset ○ Break ○ Run	Target system is in a reset state
○ Reset ● Break ○ Run	The target processor is executing in Debug Mode
○ Reset ○ Break ● Run	The target processor is executing user code
○ Reset ● Break ● Run	Only boot firmware is good (other firmware has been corrupted)

Emulation Probe Status Lights

The following illustration shows the status lights on both sides of the emulation probe and what they mean:

Lit when the power supply is properly connected to the emulation probe

Lit when the target processor is running in debug mode

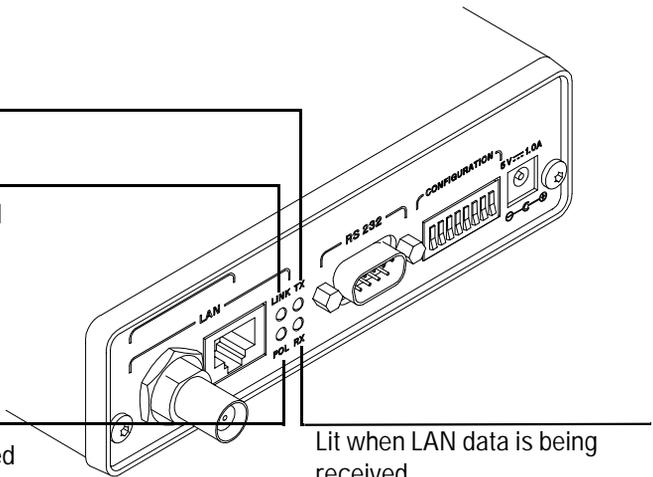


Lit when the target system is in a reset state

Lit when the target processor is running in normal (user program) mode

Lit when LAN data is being transmitted

Lit when 10BASE-T connection has a good link; not used for 10BASE2



Lit when the polarity on the receive twisted pair is reversed for a 10BASE-T connection

Lit when LAN data is being received

Status Lights**Emulation Probe Status Lights**

The emulation probe 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
○ ○ ○ ○	Emulation probe is not connected to power supply
○ ● ○ ○	No target system power, or emulation probe is not connected to the target system
○ ● ○ ●	Target system is in a reset state
● ● ○ ○	The target processor is executing in Debug Mode
○ ● ● ○	The target processor is executing user code
● ● ● ○	Only boot firmware is good (other firmware has been corrupted)

Emulator Built-in Commands

The emulator has some built-in commands (sometimes called the “terminal interface”) which you can use for troubleshooting.

You can enter the built-in commands using:

- A telnet (LAN) connection
- The Command Line window in the Emulation Control Interface
- A “debugger command” window in your debugger
- A serial connection (emulation probe only; see page 42)

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 HP16600A/700A series logic analysis system is 6472. The default port of a second module in an HP16600A-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.

Emulator Built-in Commands

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 telnet to the emulation probe

You can establish a telnet connection to the emulation probe if:

- A host computer and the probe 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 LAN address or LAN name of the emulation probe.
- 2 Start the telnet program.

If the LAN name of the emulation probe is “test2”, the command might look like this:

```
telnet test2
```

- 3 If you do not see a prompt, press the <Return> key a few times.

To exit from this telnet session, type <CTRL>D at the prompt.

To use the built-in commands

Here are a few commonly used built-in commands:

Useful built-in commands

b	Break—go into the background monitor state
cf	Configuration—read or write configuration options
help	Help—display online help for built-in commands
init	Initialize— init -c re-initializes everything in the emulator 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
pp load	Load EMSIM values into preprocessor
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 emulator will wait for you to press the target's RESET button)
s	Step—do a low-level single step
sync diff	Compare EMSIM with SIM registers
sync emsim	Copy EMSIM to SIM registers
sync sim	Copy SIM to EMSIM registers
ver	Version—display the product number and firmware version of the emulator

Emulator Built-in Commands

The prompt indicates the status of the emulator:

Emulator prompts

U	Running user program
M	Running in background monitor
p	No target power
R	Emulation reset
r	Target reset
d	Cable disconnected
?	Unknown state

Examples

To set register R0, then 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 emulator, enter:

```
M>ver
```

See Also

Use the **help** command for more information on these and other commands. Note that some of commands listed in the help screens are generic commands for HP emulators and may not be available for your product.

If you are writing your own debugger, contact HP for more information.

Problems with the BDM Connection

If a user interface behaves erratically

- ❑ Check the orientation of the cable connecting the target interface module to your target system. If the cable is offset or rotated, the emulator will try to interpret the “random” signals with unpredictable results. If the cable is rotated, the emulator or target system may also be damaged.
- ❑ Check that the processor clock speed has been properly configured. See “To configure the processor clock speed (BDM communication speed)” on page 80.
- ❑ If your target system uses a 68360, check that the FREEZE and CONFIG2 lines have sufficient pull-up current. See page 123 for more information.

Problems with Configuration

If you have problems displaying some registers

- ❑ If your user interface can read or write “generic” registers, but cannot access registers that are unique to your target microcontroller, check that the target microcontroller matches the processor type you have configured in the emulator. You can use the Configuration window in the Emulation Control Interface to configure the target processor type.
- ❑ If the value of the SP and PC are displayed as ???, see “If boot area accesses fail” on page 171.

If you have problems initializing some registers

Some registers can only be written once after processor reset.

If you set the EMSIM values, then reset and break, the EMSIM values will be written to the SIM registers. If your initialization code then attempts to write to one of the “write once after reset” registers, the writes will fail.

Problems with the Target System

If boot area accesses fail

When you start a debugger interface or attempt to run from reset, the emulator makes four accesses to the target system boot area to find reset values for the SP and PC. It then attempts to read the values on the stack and the code at the PC.

This problem can result in unknown values for the SP and PC (displayed as ???) and can make the debugger interface respond very slowly.

To avoid this problem, you can do several things:

- ❑ Use the 10-pin connector. The extra two pins on this connector allow the emulator to complete unterminated memory cycles.
- ❑ In your boot-up code, set the initial PC and stack pointer to memory which will be accessible at reset. Be sure to set the chip selects appropriately.

If the target system does not run

- ❑ If your target system uses a 68360, check that the FREEZE and CONFIG2 lines have sufficient pull-up current. See page 123 for more information.

Problems with the LAN Interface (Emulation Module Only)

If LAN communication does not work

If you cannot verify the connection, or if the commands are not accepted by the emulation module:

- ❑ Make sure that you wait for the power-on self test to complete before connecting.
- ❑ Make sure that the LAN cable is connected. Watch the LAN LED's on the back of the logic analysis system to see whether the system is seeing LAN activity. Refer to your LAN documentation for testing connectivity.
- ❑ Check that the host computer or debugger was configured with the correct LAN address. If the logic analysis system is on a different subnet than the host computer, check that the gateway address is correct.
- ❑ Make sure that the logic analysis system's IP address is set up correctly.
- ❑ Refer to the online help in the logic analysis system for more information.

If it takes a long time to connect to the network

- ❑ Check the subnet masks on the other LAN devices connected to your network. All of the devices should be configured to use the same subnet mask.

Subnet mask error messages do not indicate a major problem. You can continue using the emulation module.

The subnet mask is set in the logic analysis system's System Admin window. If it then detects other subnet masks, it will generate error messages.

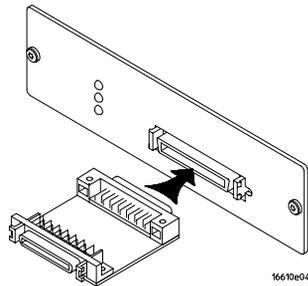
If there are many subnet masks in use on the local subnet, the logic analysis system may take a very long time to connect to the network after it is turned on.

Problems with the Emulation Module

Occasionally you may suspect a hardware problem with the emulation module or target interface module. The procedures in this section describe how to test the hardware, and if a problem is found, how to repair or replace the broken component.

To run the built-in performance verification test using the logic analysis system (Emulation Module Only)

- 1 End any Emulation Control Interface or debugger sessions.
- 2 Disconnect the 50-pin cable from the emulation module, and plug the loopback test board (HP part number E3496-66502) into the emulation module.



- 3 In the system window, click the emulation module and select **Performance Verification**.
- 4 Click **Start PV**.

The results will appear onscreen.

To run complete performance verification tests using a telnet connection (Emulation Module Only)

- 1 Disconnect the 50-pin cable from the emulation module, and plug the loopback test board (HP part number E3496-66502) directly into the emulation module. Do not plug anything into the other end of the loopback test board.

On a good system, the RESET LED will light and the BKG and USER LEDs will be out.

- 2 telnet to the emulation module.
- 3 Enter the `pv 1` command.

See Also

Options available for the “pv” command are explained in the help screen displayed by typing “help pv” or “? pv” at the prompt. Note, however, that some of the options listed may not apply to your emulation module.

Examples

If you are using a UNIX system, to telnet to a logic analysis system named “mylogic”, enter:

```
telnet mylogic 6472
```

Here are some examples of ways to use the `pv` command.

To execute both tests one time:

```
pv 1
```

To execute test 2 with maximum debug output repeatedly until a ^C is entered:

```
pv -t2 -v9 0
```

To execute tests 3, 4, and 5 only for 2 cycles:

```
pv -t3-5 2
```

The results on a good system with the loopback test board connected, are as follows:

```
M>pv 1

Testing: HPE3499C Series Emulation System
Test  1: Powerup PV Results                Passed!
Test  2: Target Probe Feedback Test        Passed!
Test  3: Boundary Scan Master Test         Passed!
Test  4: I2C Test                           Passed!
Test  5: Data Lines Test                    Passed!
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.07.53 26Feb98
Location:  Generics

HPE3490A Motorola CPU16/32 BDM Emulator
Version:  A.02.05 26Feb98

M>
```

If a performance verification test fails

- Details of the failure can be obtained through using a -v option (“verbose” level) of 2 or more.
- Check that the loopback test board is connected.
- If the problem persists, contact HP for assistance.

Problems with the LAN Interface (Emulation Probe Only)

If you cannot verify LAN communication

If you cannot verify connection using the procedure in "To verify LAN communication", or if the commands are not accepted by the emulation probe:

- ❑ Make sure that you have connected the emulation probe to the proper power source and that the power light is lit.
- ❑ Make sure that you wait for the power-on self test to complete before connecting.
- ❑ Make sure that the LAN cable is connected. Watch the LAN LED's to see whether the emulation probe is seeing LAN activity. Refer to your LAN documentation for testing connectivity.
- ❑ Make sure that only one of the LAN ports is connected.
- ❑ Make sure the emulation probe communication configuration switches are set correctly. Unplug the emulation probe power cord, then plug it in again to make sure the switch settings are read correctly by the emulation probe.
- ❑ Check that the Emulation Control Interface or debugger was configured with the correct LAN address. If the emulation probe is on a different subnet than the host computer, check that the gateway address is correct.
- ❑ Make sure that the emulation probe's IP address is set up correctly. Use the RS-232 port to verify this that the IP address is set up correctly. When you are connected to the RS-232 port, run performance verification on the emulation probe's LAN interface with the "pv" command.
- ❑ It's also possible for there to be a problem with the emulation probe firmware while the LAN interface is still up and running. In this case, you must reboot the emulation probe by disconnecting power to the emulation probe and reconnecting it again.

Problems with the LAN Interface (Emulation Probe Only)

- ❑ Use a serial connection to run the LAN performance verification tests (see page 174).

If you have LAN connection problems

- ❑ If the emulation probe does not accept commands from the HP 16600A/700A:
 1. Check that switch S1 is "0" (attached to LAN, not RS-232).
 2. Check that switch S5 is in the correct position for your LAN interface (either 10BASE2 or 10BASE-T).
(Remember: if you change any switch settings, the changes do not take effect until you cycle power.)
- ❑ If the emulation probe still does not respond, you need to verify the IP address and gateway mask of the emulation probe. To do this, connect the emulation probe to a terminal or terminal emulator (see page 42), change the switch settings so it is connected to RS-232, and enter the "lan" command. The output looks something like this:

```
lan -i 15.5.24.116
lan -g 15.5.23.1
lan -p 6470
Ethernet Address : 08000909BAC1
```

"lan -i" shows the internet address is 15.5.24.116 in this case. If the Internet address (IP) is not what you expect, you can change it with the 'lan -i >' command.

"lan -g" shows the gateway address. Make sure it is the address of your gateway if you are connecting from another subnet, or 0.0.0.0 if you are connecting from the local subnet.

"lan -p" shows the port is 6470.

If the "POL" LED is lit

The "POL" LED indicates that the polarity is reversed on the receive pair if you are using a 10BASE-T connection. The emulator should still work properly in this situation, but other LAN devices may not work.

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 probe.

The emulation probe automatically sets its subnet mask based on the first subnet mask it detects on the network. 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 emulation probe may take a very long time to connect to the network after it is turned on.

To "clean up" the network, connect a terminal to the emulation probe. You can then see error messages which will help you identify which devices on the network are using the wrong subnet masks.

Problems with the Serial Interface (Emulation Probe Only)

If you cannot verify RS-232 communication

If the emulation probe prompt does not appear in the terminal emulator window:

- ❑ Make sure that you have connected the emulation probe to the proper power source and that the power light is lit.
- ❑ Make sure that you have properly configured the data communications switches on the emulation probe and the data communications parameters on the host computer. You should also verify that you are using the correct cable.

The most common type of data communications configuration problem involves the configuration of the emulator as a DTE device instead of as a DCE device. If you are using the wrong type of cable, no prompt will be displayed.

A cable with one-to-one connections will work with a PC or an HP Series 700 workstation.

If you have RS-232 connection problems with the MS Windows Terminal program

- ❑ Remember that Windows 3.1 only allows two active RS-232 connections at a time. To be warned when you violate this restriction, choose Always Warn in the Device Contention group box under 386 Enhanced in the Control Panel.
- ❑ Use the "Terminal" program (usually found in the Accessories windows program group) and set up the "Communications..." settings as follows:
 - Baud Rate: 9600 (or whatever you have chosen for the emulation probe)
 - Data Bits: 8
 - Parity: None
 - Flow Control: hardware
 - Stop Bits: 1

When you are connected, hit the Enter key. You should get a prompt back. If nothing echos back, check the switch settings on the emulation probe.

- ❑ If the switches are in the correct position and you still do not get a prompt when you hit return, try turning OFF the power to the emulation probe and turning it ON again.
- ❑ If you still don't get a prompt, make sure the RS-232 cable is connected to the correct port on your PC, and that the cable is appropriate for connecting the PC to a DCE device.

With certain RS-232 cards, connecting to an RS-232 port where the emulation probe is turned OFF (or is not connected) will hang the PC. The only way to get control back is to reboot the PC. Therefore, we recommend that you always turn ON the emulation probe before attempting to connect via RS-232.

Problems with the Emulation Probe

To run the power up self test

- 1 Unplug the emulation probe, then plug it in.
- 2 Watch the status lights. They should show the following pattern:

○ = LED is off

● = LED is on

* = Not applicable (LED is off or on)

Normal sequence during power up self test

	Pwr/Target LEDs	Meaning
1	○● ○○	Initial power up, system reset
2	○● ○○	XILINX array initialized successfully
3	○● ●○	XILINX array tested successfully
4	●● ○○	BOOT ROM space tested successfully
5	○● ●○	GENERIC ROM space tested successfully
6	●● ○○	DRIVER ROM space tested successfully
7	○● ●○	RESERVED ROM space tested successfully

	Pwr/Target LEDs	Meaning
8	●● ○○	RAM tested successfully
9	○● ●○	LAN internal feedback tested successfully
10	○● ○○	Boundary scan master (BSM) test begun
11	●● ●○	BSM test completed, start system, load drivers, initialize LAN

If the power up self test fails, the RESET LED will flash the number of the test, then stay lit.

If any of the LEDs fail to change, or all of them remain on, there is a system failure.

Following power up, the LEDs will enter one of the following states shown on page 164.

Starting a user interface will change the pattern to the one requested by the interface.

If the power up self tests fail, try the following:

- Check and reset the LAN address. LAN powerup failures will occur if the emulation probe does not have a valid link-level address and IP address.
- Disconnect all external connections, including the LAN, serial (RS-232), and BNC break and trigger cables, then cycle power.
- To ensure that the firmware is working as it should, reprogram the firmware, then cycle power.

To run the emulation probe performance verification tests

In addition to the powerup tests, there are several additional performance verification (PV) tests available.

Some of these tests can be performed through the HP 16700A/16700A-series logic analysis system. The LAN tests for an emulation probe can only be executed through the RS-232 port.

To fully test the emulation probe, you will need to run the PV test with several hardware configurations:

- For the BREAK IN, TRIGGER OUT BNC FEEDBACK TEST, connect a coaxial cable between BREAK IN and TRIGGER OUT.

To run the performance verification tests using the logic analysis system

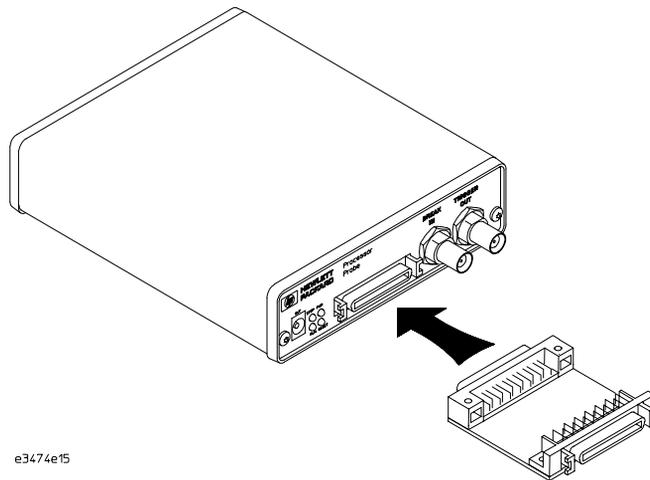
- 1** End any Emulation Control Interface or debugger sessions.
- 2** Disconnect the 50-pin cable from the emulation probe, and plug the loopback test board (HP part number E3496-66502) into the emulation probe.
- 3** In the system window, click the emulation probe and select **Performance Verification**.
- 4** Click **Start PV**.

The results will appear on screen.

To run complete performance verification tests for an emulation probe

The LAN tests can only be executed through the RS-232 port. The remainder of this section assumes that the tests are being run from a terminal emulator connected to the RS-232 port.

- 1 Disconnect the 50-pin cable from the emulation probe, and plug the emulator loopback test board (HP part number E3496-66502) directly into the emulator. Do not plug anything into the other end of the emulator loopback test board.



On a good system, the RESET LED will light and the BKG and USER LEDs will be out.

- 2 Connect a coaxial cable between BREAK IN and TRIGGER OUT.
- 3 Set all of the switches to CLOSED.

This is standard RS-232 at 9600 baud which can be connected directly to a 9 pin RS-232 interface that conforms to the IBM PC-AT 9 pin standard.

- 4 Use a terminal emulator to connect to the emulation probe.

5 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.

Examples

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 BNCs connected, and with the loopback test board connected, are as follows:

```
c>pv 1
Testing: HPE3499B Series Emulation System passed!
  Test # 1: Powerup PV Results                passed!
  Test # 2: LAN 10Base2 Feedback Test         passed!
  Test # 3: LAN 10BaseT Feedback Test         passed!
  Test # 4: Break In and Trigger Out BNC Feedback Test passed!
  Test # 5: Target Probe Feedback Test        passed!
  Test # 6: Boundary Scan Master Test         passed!
  Test # 7: I2C Test                           passed!
  Test # 8: 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.

HPE3499B Series Emulation System
Version:  A.07.53 01Mar98
Location:  Generics

HPE3490A Motorola CPU16/32 BDM Emulator
Version:  A.02.05 01Mar98

R>
```

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 value (“verbose level”) of 2 or more.

If the particular failure you see is not listed below, contact HP for assistance.

TEST 2: LAN 10BASE2 Feedback Test failed

For LAN 10BASE2 test, the following is an example of a failure which is not caused by a broken emulator.

```
R>pv -t2 -v2 1

Testing: HPE3499B Series Emulation System
Test # 2: LAN 10Base2 Feedback Test          failed!
FAILED - no lan connection (LAN probably not terminated
FAILED Number of tests: 1          Number of failures: 1
```

Check to see that the port under test has a good cable connected to it and that the cable is properly terminated with a 50 ohm terminator on each end of the overall cable.

```
R>pv -t2 -v2 1

Testing: HPE3499B Series Emulation System
Test # 2: LAN 10Base2 Feedback Test          failed!
FAILED due to excessive collisions
FAILED Number of tests: 1          Number of failures: 1
```

The most common cause of this problem is poor termination of the cable or failure to remove the port under test from the LAN before performing the test. Check to see that the terminators are good (50 ohms) and that you are isolated from any traffic on a system LAN.

```
R>pv -t2 -v2 1

Testing: HPE3499B Series Emulation System
Test # 2: LAN 10Base2 Feedback Test          failed!
FAILED - invalid Ethernet address in EEPROM
FAILED Number of tests: 1          Number of failures: 1
```

First check to see that a correct link-level address and IP address have been set in the virtual EEPROM through the “lan” command. If the “lan” command shows bad information for the link-level address and IP,

Problems with the Emulation Probe

then try to set them to correct values. If you are unable to set them to correct values, there is a failure in the FLASH ROM which requires service from HP.

Test 3: 10BaseT Feedback Test failed here

```
R>pv -t3 -v2 1

Testing: HPE3499B Series Emulation System
Test # 3: LAN 10BaseT Feedback Test failed!
FAILED Number of tests: 1 Number of failures: 1
```

In addition to the internal checks performed in Test 2, this test also checks for shorts on the cable connected to the network. If this test fails, disconnect the cable and run the test again. If it then passes, the cable is faulty. If it still fails, it requires service from HP.

If the emulator passes this “pv” test, additional testing can be performed through exercising the connection to the network. To run this test, set configuration switch 1 and switch 5 to OPEN, all other configuration switches CLOSED (this enables LAN using 10BaseT). Cycle power and wait for 15 to 30 seconds. Then “ping” the emulator from your host computer or PC. See the LAN documentation for your host computer for the location and action of the “ping” utility. If the emulator fails to respond to the “ping” request, verify that the LAN parameters (IP address and gateway address) are set correctly and that your host computer recognizes the IP address of the emulator. If all else is good, then failure to respond to ping indicates a faulty emulator.

Test 4: Break In and Trigger Out BNC Feedback Test

```
R>pv -t4 -v2 1

Testing: HPE3499B Series Emulation System
Test # 4: Break In and Trigger Out BNC Feedback Test failed!
Break In not receiving Break Out HIGH
FAILED Number of tests: 1 Number of failures: 1
```

Before returning to HP, check to ensure that you have connected a good coaxial cable between the two BNCs. If the cable is good, the emulator is bad.

TEST 5: Target Probe Feedback Test

TEST 6: Boundary Scan Master Test

TEST 7: I2C Test

TEST 8: Data Line Test

If these tests are not executed, check that you have connected the emulator loopback test board.

If these tests fail, return the emulator to HP for replacement.

Returning Parts to Hewlett-Packard for Service

The repair strategy for this emulation solution is board replacement.

Exchange assemblies are available when a repairable assembly is returned to Hewlett-Packard. These assemblies have been set up on the “Exchange Assembly” program. This lets you exchange a faulty assembly with one that has been repaired, calibrated, and performance verified by the factory. The cost is significantly less than that of a new assembly.

To return a part to Hewlett-Packard

- 1** Follow the procedures in this chapter to make sure that the problem is caused by a hardware failure, not by configuration or cabling problems.
- 2** In the U.S., call 1-800-403-0801. Outside the U.S., call your nearest HP sales office. Ask them for the address of the nearest HP service center.
- 3** Package the part and send it to the HP service center.

Keep any parts which you know are working. For example, if only the target interface module is broken, keep the emulation module and cables.

- 4** When the part has been replaced, it will be sent back to you.

The unit returned to you will have the same serial number as the unit you sent to HP.

The HP service center can also troubleshoot the hardware and replace the failed part. To do this, send your entire measurement system to the service center, including the logic analysis system, target interface module, and cables.

In some parts of the world, on-site repair service is available. Ask an HP sales or service representative for details.

To obtain replacement parts

The following table lists some parts that may be replaced if they are damaged or lost. The part numbers are subject to change. Contact your nearest Hewlett-Packard Sales Office for further information.

Exchange assemblies

Part number	Description
16600-69515	Emulation module
E3458-69401	Programmed emulation probe assembly

Replacement assemblies

Part number	Description
E3496-61601	50-pin control cable
E3496-61603	10-pin BDM cable
E3496-66502	Loopback test board
E3458-66501	CPU32 target interface module
16700-61608	Expansion cable for emulation module
0950-3043	Power supply for emulation probe

To clean the instrument

If the instrument requires cleaning:

- 1** Remove power from the instrument.
- 2** Clean with a mild detergent and water.
- 3** Make sure that the instrument is completely dry before reconnecting it to a power source.

Analysis Probe A probing solution connected to the target microcontroller. It provides an interface between the signals of the target microcontroller and the inputs of the logic analyzer. Formerly called a “preprocessor.”

Elastomeric Probe Adapter A connector that is fastened on top of a target microcontroller using a retainer and knurled nut. The conductive elastomer on the bottom of the probe adapter makes contact with pins of the target microcontroller and delivers their signals to connection points on top of the probe adapter.

Emulation Module An emulation module is installed within the mainframe of a logic analyzer. It provides run control within an emulation and analysis test setup. See Emulation Probe.

Emulation Probe An emulation probe is a standalone instrument connected via LAN to the mainframe of a logic analyzer or to a host computer. It provides run control within an emulation and analysis test setup. Formerly called a “processor probe” or “software probe.” See Emulation Module.

Emulator An emulation module or

an emulation probe.

Extender A part whose only function is to provide connections from one location to another. One or more extenders might be stacked to raise a probe above a target microprocessor to avoid mechanical contact with other components installed close to the target microcontroller. Sometimes called a “connector board.”

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

General-Purpose Flexible Adapter A cable assembly that connects the signals from an elastomeric probe adapter to an analysis probe. Normally, a male-to-male header or transition board makes the connections from the general-purpose flexible adapter to the analysis probe.

High-Density Adapter Cable A cable assembly that delivers signals from an analysis probe hardware interface to the logic analyzer pod cables. A high-density adapter cable has a single Mictor connector that is installed into the analysis probe, and two cables that are connected to

corresponding odd and even logic analyzer pod cables.

High-Density Termination Adapter Cable Same as a High-Density Adapter Cable, except it has a termination in the Mictor connector.

Jumper Moveable direct electrical connection between two points.

Mainframe Logic Analyzer A logic analyzer that resides on one or more board assemblies installed in an HP16500, HP1660-series, or HP16600A/700A-series mainframe.

Male-to-male Header A board assembly that makes point-to-point connections between the female pins of a flexible adapter or transition board and the female pins of an analysis probe.

Preprocessor See Analysis Probe.

Preprocessor Interface See Analysis Probe.

Probe adapter See Elastomeric Probe Adapter.

Processor Probe See Emulation Probe.

Prototype Analyzer The HP16505A

prototype analyzer acts as an analysis and display processor for the HP16500B/C logic analysis system. It provides a windowed interface and powerful analysis capabilities. Replaced by HP16600A/700A-series logic analysis systems.

Run Control Probe See Emulation Probe and Emulation Module.

Setup Assistant A software program that guides a user through the process of connecting and configuring a logic analyzer to make measurements on a specific microcontroller.

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 HPB4620B Source Correlation Tool Set, and an emulation module.

Stand-alone Logic Analyzer A standalone logic analyzer has a predefined set of hardware components which provide a specific set of capabilities. It is designed to perform logic analysis. A standalone logic analyzer differs from a

mainframe logic analyzer in that it does not offer card slots for installation of additional capabilities, and its specifications are not modified based upon selection from a set of optional hardware boards that might be installed within its frame.

Target Control Port An 8-bit, TTL port on a logic analysis system that you can use to send signals to your target system. It does not function like a pattern generator or emulation module, but more like a remote control for the target's switches.

Target Interface Module A small circuit board which connects the 50-pin cable from an emulation module or emulation probe to signals from the debug port on a target system.

TIM See Target Interface Module.

Trigger Specification A set of conditions that must be true before the instrument triggers. See the printed or online documentation for your logic analyzer for details.

Transition Board A board assembly that obtains signals connected to one side and rearranges them in a different order for delivery at the other side of the board.

1/4-Flexible Adapter An adapter

that obtains one-quarter of the signals from an elastomeric probe adapter (one side of a target microcontroller) and makes them available for probing.

A

address, IP
 IP address, 32
altitude specifications, 157
analysis probe
 connecting to, 73
 definition, 193
 processors supported, 4
 product numbers, 4
assistant
 See setup assistant

B

BDM port
 See debug port
BKG light, 162, 164, 185
BNC
 break in, 86
 trigger out, 85
BNC, LAN, 31, 155
bootp, 37, 38
break in configuration, 86
breakpoints
 tracing until, 145
built-in commands
 configuration, 77
 LAN configuration, 34
 list of commands, 165
bus cycle termination, 65

C

cables
 BDM, 71
 emulator, 70
 LAN, 40
 serial, 44
CD-ROM, installing software from,
 56
cf commands, 77
characteristics
 emulation module, 156
cleaning, 192

clock speed
 configuring, 80
clocks
 qualified, and emulator, 138
 slow, 144, 146
 specifications, 155
configuration
 emulation module
 overview, 75
 using debugger, 78
configuration files
 installing, 53
connecting, 20
connection, 22, 178, 181
 emulation module, 59, 60
 host workstation, 29
 problems, LAN, 172
connector
 10BASE2, 31, 155
 10BASE-T, 31, 155
 BDM, 71
 debug port, 65
 serial, 155
connector board, 193

D

data communications
 specifications, 155
DCE or DTE selection and RS-232
 cable, 180
debug port
 connecting to, 70
 connections, 65
debuggers
 configuration, 78
 Green Hills, 114
 SDS, 123
 writing, 168
development port
 See debug port

E

elastomeric probe adapter
 definition, 193
EMSIM registers, 89
 displaying, 90
 example with debugger, 119
 purpose of, 90
Emulation Control Interface
 configuration, 76
 introduction, 61
 when to use, 134
emulation module
 configuration, 78
 connecting, 60, 69
 definition, 193
 description of, 3
 product numbers, 4
 target system design, 65
emulation probe
 definition, 193
 problems, solving, 182
emulation solution
 See solution
emulator
 definition, 193
 equipment required
 emulation module, 24
 equipment supplied
 emulation migration, 25
 emulation module, 23
 ordering information, 4
 overview, 4
erratic behavior, 169
ethernet address, 35
examples, measurement, 135
extender, 193

F

files
 loading vs. installing, 54
firmware
 version, 149, 150

flexible adapter
definition, 193

G

gateway address, 35, 178
general-purpose flexible adapter
definition, 193
Green Hills
debugger, 114

H

high-density adapter cable
definition, 193
high-density termination adapter
definition, 194
host computer
connecting to, 29
HRESET signal, 65

I

IEEE 802.3, 31
information sources, 27
installation, software, 53
intermodule measurement
creating, 137
intermodule measurement
problems
analyzer doesn't stop, 138
internal registers, 89
internet address, 32
inverse assembly
displays, 134
IP address, 31, 32, 35, 177, 178
IPC
connecting to, 29

J

jumper, definition, 194

L

lan command, 34

LAN connection problems, 172,
178, 187, 188
LAN interface, 31
LAN parameters, configuring
BOOTP, 37
methods, 31
terminal interface, 34
lights
See status lights
link beat, 40
link level address, 35, 38
listing windows, 134
loading configurations
vs. installing, 53
loopback test board, 184

M

mainframe logic analyzer
definition, 194
male-to-male header
definition, 194
mask, subnet, 172, 179
MAU, 31, 155
measurement examples, 135
microprocessors supported, 4
minimum equipment, 20
MULTI debugger, 114

P

performance verification tests,
184, 185
ping command, 178
POL light, 179
port number, emulation module,
165
port number, emulation probe, 35
power supply, 20
power-on self test, 182
preprocessor
See analysis probe
problems, LAN, 178
problems, RS-232, 181

processor support package, 56
processor type
configuring, 79
effect on EMSIM registers, 92
processors supported, 4
prompts, 168
list of, 168
prototype analyzer
definition, 194
PV
See performance verification

R

real-time runs, configuring, 85
references, 27
registers
initializing, 170
internal, 89
problems displaying, 170, 171
unknown values, 171
repair
emulation module, 190
requirements, 20
target system, 65
RESET
light, 162, 164
reset
effects of, SDS debugger, 124
effects on SIM registers, 95
RESET LED, 183, 185
run control tool
See emulation control interface

S

SDS debugger, 123
self test, 182
sequence, 22
serial connection
DCE or DTE selection, 180
number of connections, 181
problems, 181
setting up, 42

verifying, 46
service ports, 35
service, how to obtain, 190
setup
 See configuration
setup assistant, 151
 definition, 194
signals
 debug port, 65
SIM registers
 configuring, 95
 displaying, 90
 introduction, 90
SingleStep debugger, 123
skid, reducing, 138
slow clock message, 144, 146, 161
software
 installing, 53
 list of installed, 57
software probe
 See emulation probe
solution
 at a glance, 2
 definition, 194
 description of, 2
solving problems
 emulation probe, 182
source code
 displays, 134
specifications
 altitude, 157
 clock, 155
 data communications, 155
 See characteristics
 temperature, 157
 trigger in/out, 155
SRESET signal, 65
StarLAN, 31, 40
status lights, 162, 164, 182
subnet mask, 31, 172, 178, 179
switches, 38
 bootp, 38
 LAN configuration, 34, 40

serial configuration, 42

T

target control port
 definition, 195
target interface module (TIM)
 connecting, 70
 definition, 195
target system
 connecting to, 60
 requirements for emulation, 65
TCP, 35
telnet, 41, 165, 166, 177
temperature specifications, 157
terminal (MS Windows program),
 181
terminal interface, 41
 LAN parameters, setting, 34
 See also built-in commands
tests, emulation module, 174
ThinLAN, 31, 155
trace, 140
transition board
 definition, 195
trigger
 emulation module, 136
 in/out specifications, 155
 on break, 141
trigger out configuration, 85
troubleshooting, 161

U

until processor halts, 140
Updating Firmware, 148
USER LED, 185
USER light, 162, 164

V

versions
 emulation module firmware, 150
 emulation probe firmware, 149
voltage

emulation module, 156

W

web sites
 HP logic analyzers, 27
 See Also under debugger names
wizard
 See setup assistant
workstation
 connecting to, 29

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: Processor Probe
Model Number(s): E3458A
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

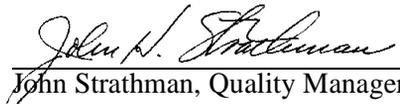
EMC: CISPR 11:1990 / EN 55011:1991 Group 1 Class A
IEC 555-2:1982 + A1:1985 / EN 60555-2:1987
IEC 555-3:1982 + A1:1990 / EN 60555-3:1987 + A1:1991
IEC 801-2:1991 / EN 50082-1:1992 4 kV CD, 8 kV AD
IEC 801-3:1984 / EN 50082-1:1992 3 V/m, {1kHz 80% AM, 27-1000 MHz}
IEC 801-4:1998 / EN 50082-1:1992 0.5 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, 11/01/96


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)

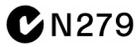
Product Regulations

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.



Emissions EN55011/CISPR 11 (ISM, Group 1, Class A equipment),



Immunity	EN50082-1	Code ¹	Notes ²
	IEC 555-2	1	
	IEC 555-3	1	
	IEC 801-2 (ESD) 4kV CD, 8kV AD	1	1,2
	IEC 801-3 (Rad.) 3 V/m	1	2
	IEC 801-4 (EFT) 0.5 kV, 1kV	1	

¹Performance 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.

²Notes:

1 The target cable assembly is sensitive to ESD events. use standard ESD preventative practices to avoid component damage.

2 The target cable assembly, 10Base-T and 10Base-2, cables were disconnected.

Sound Pressure Level N/A

© Copyright Hewlett-Packard Company 1994-1998
All Rights Reserved.

Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in subparagraph (C) (1) (ii) of the Rights in Technical Data and Computer Software Clause in DFARS 252.227-7013. Hewlett-Packard Company, 3000 Hanover Street, Palo Alto, CA 94304 U.S.A. Rights for non-DOD U.S. Government Departments and Agencies are set forth in FAR 52.227-19 (c) (1,2).

Document Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Hewlett-Packard shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety

This apparatus has been designed and tested in accordance with IEC Publication 1010, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

Warning

- Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.
- Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock or fire hazard.

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

- If you energize this instrument by an auto transformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.

- Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.

- Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

- Do not install substitute parts or perform any unauthorized modification to the instrument.

- Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.

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.

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 *Emulation for Motorola CPU32 User's Guide*.

Publication number
E3458-97001, October 1998
Printed in USA.

The information in this manual previously appeared in:
E3458-97000, Feb 1997

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.

Comments welcome!

Send your comments or suggestions regarding this manual to:

documentation@col.hp.com

Reflection 1 is a U.S. trademark of Walker, Richer & Quinn, Inc.

UNIX is a registered trademark of The Open Group.

Windows, MS Windows, Windows NT, and MS-DOS are U.S. registered trademarks of Microsoft Corporation.

X/Open is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.