Help Volume

System: Setup Assistant

The Setup Assistant

The Setup Assistant simplifies the process of setting up a logic analyzer measurement system. The Setup Assistant asks you for your desired setup, instructs you to connect the necessary hardware, and loads the appropriate software. The topics in this help volume are primarily intended to be accessed from the Setup Assistant.

Main Tasks

- "Setup Assistant Overview" on page 10
- "Identifying System Components" on page 12
- "Setup Information You Will Need" on page 15
- "Selecting a Target & Analysis Probe" on page 17
- "Selecting an Analyzer" on page 20
- "Choosing an Acquisition Mode" on page 21
- "Connecting To Your Target System" on page 22
- "Connecting To Your Analyzer Card" on page 23
- "Setting Up A Network" on page 27
- "Connecting PC Shares" on page 56
- "Saving Settings" on page 28
- "Taking the Next Step" on page 29

Source Correlation Tasks

- "Choosing To Set Up Only Source Correlation" on page 30
- "Licensing the Source Correlation Tool" on page 31
- "Locating Source Code" on page 32
- "Mounting Remote Disks" on page 33
- "Loading Software Symbols" on page 34

- "Using Source Correlation" on page 35

Emulation Tasks

NOTE: For information on how to set up an emulator without using the Setup Assistant, see Setting Up and Starting Emulation Control (see the *Emulation: Setting Up* help volume) or refer to the manual for your emulator. The help topics listed below are designed to be accessed from the Setup Assistant.

- "Choosing to Use Emulation" on page 50 (If you have an emulation probe, and your target processor is supported by both the E5900A and E5900B.)
- "Choosing to Use Emulation" on page 51 (If you have an E5900A emulation probe.)
- "Choosing to Use Emulation" on page 52 (If you have an E5900B emulation probe.)
- "Choosing to Use Emulation" on page 52 (If you have an emulation module.)
- "Installing Emulation Control Software" on page 38
- "Connecting an Emulation Probe to the Network" on page 39 (For emulation probes which have not yet been set up on the LAN.)
- "Entering the Emulation Probe IP Address" on page 47 (For emulation probes which are already set up on the LAN.)
- "Choosing a Firmware Version" on page 48
- "Connecting the Emulation Hardware" on page 49

See Also

- "To decide if you have an E5900A or an E5900B" on page 57
- "To decide if you have an E5901A or an E5901B" on page 57
- Main System Help (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)
- Glossary of Terms (see page 81)

Contents

The Setup Assistant

1 The Setup Assistant

Setup Assistant Overview 10

Identifying System Components 12

Setup Information You Will Need 15

Selecting a Target & Analysis Probe 17

If your target processor is not listed 18
Install Software 18
Manual Configuration 18

Selecting an Analyzer 20

Choosing an Acquisition Mode 21

Connecting To Your Target System 22

Connecting To Your Analyzer Card 23

Setting Up A Network 27

Saving Settings 28

Taking the Next Step 29

Choosing To Set Up Only Source Correlation 30

Licensing the Source Correlation Tool 31

Locating Source Code 32

Contents

Mounting Remote Disks 33

Loading Software Symbols 34

Using Source Correlation 35

Choosing the Type of Emulation Measurement 37

Installing Emulation Control Software 38

Connecting an Emulation Probe to the Network 39
Choosing a Connection Type 39
Deciding if an Emulation Probe is on the LAN 40
Setting Up a LAN Connection 40
Setting Network Parameters 43

Setting Up a Point-to-Point Connection 44
Setting Up a Point-to-Point Connection (E5900A) 44
Setting Up a Point-to-Point Connection (E5900B) 45

Entering the Emulation Probe IP Address 47

Choosing a Firmware Version 48

Connecting the Emulation Hardware 49

Choosing to Use Emulation 50
Choosing to Use Emulation 51
Choosing to Use Emulation 52
Choosing to Use Emulation 52

Connecting Emulation to the Target 54
To Connect Using a Debug Port 54
To Connect Using an Analysis Probe 55

Connecting PC Shares 56

Contents

Measurement Examples 57
To decide if you have an E5900A or an E5900B 57
To decide if you have an E5901A or an E5901B 57

The Symbols Tab 59

Displaying Data in Symbolic Form 60

Setting Up Object File Symbols 61
To Load Object File Symbols 61
Relocating Sections of Code 63
To Delete Object File Symbol Files 64
Symbol File Formats 64
Creating ASCII Symbol Files 65
Creating a readers.ini File 70

Using Symbols In The Logic Analyzer 73
Using Symbols As Trigger Terms 73
Using Symbols as Search Patterns in Listing Displays 74
Using Symbols as Trigger Terms in the Source Viewer 74
Using Symbols as Pattern Filter Terms 74
Using Symbols as Ranges in the Software Performance Analyzer 75

User-Defined Symbols 78
To Create User-Defined Symbols 78
To Replace User-Defined Symbols 78
To Delete User-Defined Symbols 79
To Load User-Defined Symbols 79

Glossary

Index

1

The Setup Assistant

Setup Assistant Overview

The Setup Assistant will help you set up your logic analysis system, eliminating the need to read separate component manuals and manually load configuration files. Follow the Setup Assistant instructions to quickly begin taking data measurements.

See also: “Identifying System Components” on page 12 and Setup Information You Will Need (see page 15)

- If you are connecting an "emulation solution" (an analysis probe or inverse assembler and an emulation probe) to your target system, choose Full Measurement.
- If you are connecting only an analysis probe or inverse assembler to your target system, choose Analysis Only.
- If you are only connecting only an emulation probe or emulation module, choose Emulation only.
- If you have previously set up an analysis probe (required by the Source Viewer), and only want to configure the Source Viewer, choose Source Viewer only.

The Setup Assistant helps with these tasks:

- Setting up an analysis probe
- Setting up an emulation module
- Setting up an emulation probe
- Setting up source correlation

When setting up an analysis probe:

- Selecting the processor or bus, and analysis probe you are using.
- Selecting an analyzer for connecting to the analysis probe.
- Selecting State or Timing data acquisition modes (unless one is required by a certain system feature).
- Connecting to your target system.
- Connecting the analysis probe to your analyzer.

- Selecting among appropriate configuration options.
- Setting configuration switches on the analysis probe.
- Loading appropriate configuration and inverse assembler files.

When setting up an E5901A emulation module:

- Configuring the module for your target processor.
- Starting an emulation session.

When setting up an E5901B emulation module:

- Setting up the interconnnected E5900B emulation probe.
- Configuring the probe for your target processor.
- Starting an emulation session.

When setting up an emulation probe:

- Setting up an emulation probe as a network device.
- Configuring the probe for your target processor.
- Starting an emulation session.

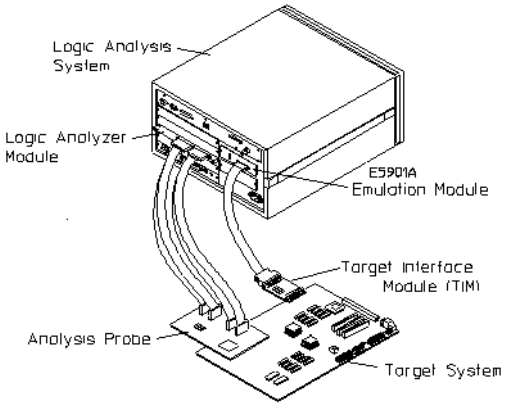
When setting up source correlation:

- Setting up network access to local development machines.
- Specifying a location for symbol files.
- Initiating a Source Viewer display.

Once you have configured your logic analysis system, you can save your settings to disk. The settings can be loaded at any time or configured to be auto-loaded at power-up.

Identifying System Components

The three drawings identify the main components in a logic analysis system set up for emulation and analysis. You may only have some of the components shown.

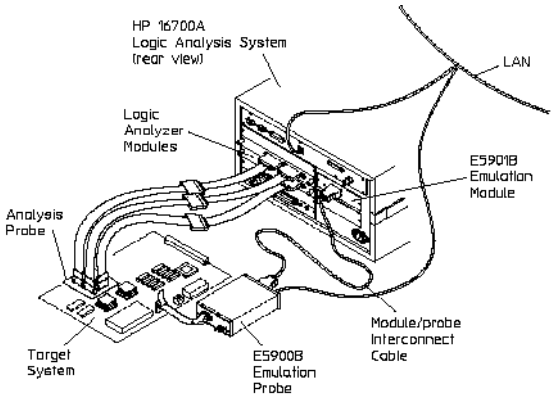


Emulation using an E5901A emulation module

The above drawing shows:

- Logic Analysis System (16700-Series Mainframe) - Connects to keyboard, mouse and monitor. Contains five wide card slots for instrument hardware modules, two narrow card slots for an emulation module, and peripheral connections for LAN and CD-ROM.
- Logic Analyzer Module - An optional product which installs into one or more mainframe card slots allowing various widths and depths of data analysis.
- Analysis Probe - Connects between your target processor and the logic analyzer, and provides processor signals for analysis.
- E5901A Emulation Module - Plugs into an 16600A/700A-series frame to provide emulation for the processor on your target system. Emulation modules are programmed for the processor you are using. Note: In some cases the emulation module connects to an analysis probe instead of directly to your target system.

- Target Interface Module (TIM) - Some emulation setups require this small PC board to connect the emulation probe or module to your target system.
- Probe Adapter (not shown) - Some processors require a special connector that you attach to an analysis probe. You may need to glue your probe adapter to your processor for a reliable electrical connection.

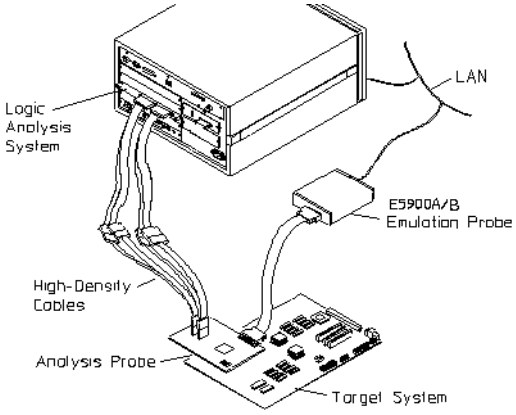


Emulation using an E5901B emulation module

The above drawing shows:

- Logic Analysis System (16700-Series Mainframe) - Connects to keyboard, mouse and monitor. Contains five wide card slots for instrument hardware modules, two narrow card slots for emulation modules, and peripheral connections for LAN and CD-ROM.
- Analysis Probe - Connects between your target processor and the logic analyzer, and provides processor signals for analysis.
- High Density Probe Adapter Cables - These Y-cables connect analysis probe signals to a logic analyzer. The high-density connector minimizes required space near the target processor. The Odd/Even ends of the cable connect to standard logic analyzer connectors.
- E5901B Emulation Module - Plugs into a 16700-series frame. Provides power, cross triggering, and limited communication for the E5900B emulation probe through the module/probe interconnect cable.
- E5900B Emulation Probe - Connects to your target processor to let you access the processor's built-in debugging features, including run control

and access to registers and memory. The emulation probe is controlled from the 16700 frame via LAN. Emulation probes are programmed for the processor you are using. Note: In some cases the emulation probe connects to an analysis probe instead of directly to your target system.



Emulation using an E5900A or E5900B emulation probe

The above drawing shows:

- Logic Analysis System (16700-Series Mainframe) - Connects to keyboard, mouse and monitor. Contains five wide card slots for instrument hardware modules, two narrow card slots for emulation modules, and peripheral connections for LAN and CD-ROM.
- Analysis Probe - Connects between your target processor and the logic analyzer, and provides processor signals for analysis.
- High Density Probe Adapter Cables - These Y-cables connect analysis probe signals to a logic analyzer. The high-density connector minimizes required space near the target processor. The Odd/Even ends of the cable connect to standard logic analyzer connectors.
- E5900A or E5900B Emulation Probe - Connects to your target processor to let you access the processor's built-in debugging features, including run control and access to registers and memory. The emulation probe is controlled from the 16700 frame via LAN. Emulation probes are programmed for the processor you are using. Note: In some cases the emulation probe connects to an analysis probe instead of directly to your target system.

Setup Information You Will Need

Here is a list of the information you will need to configure your logic analysis system using the Setup Assistant:

To configure an analysis probe:

- The name and manufacturer of the microprocessor, microcontroller or standard bus that you are measuring. (e.g. Intel Pentium® or Motorola PowerPC 603)
- In the case that your processor is supported by more than one analysis probe product, you will need to know which product you are using.
- Which logic analyzers (card sets) you plan to use to connect to the analysis probe.
- The data capture mode (State or Timing) you want to use. (Note: Using inverse assembly or the Source Viewer will require State mode.)
- Depending upon your processor, you may need to reference a hard copy manual for a probe adapter which connects the analysis probe to your target system.

To access source/symbol files on the network:

- An IP address, hostname and directory path for the computer where these files are located.
- A directory path, local to your logic analysis system, where the remote files will be made available.

To connect an emulation module/probe:

- The name and manufacturer of the microprocessor or microcontroller that you are measuring.
- If your processor is supported by more than one emulation module firmware version, you will need to know which firmware version is best for your target system. This is usually based on the detailed version and mask revision information for your processor.
- The IP address, subnet mask, and gateway address of the emulation probe. The IP address may be an address which has already been programmed

into the emulation probe, or you may need to use the Setup Assistant to program the address. (If you have an E5901A emulation module, the emulation module does not have its own IP address because it is accessed through the 16600A/700A-series frame.)

Selecting a Target & Analysis Probe

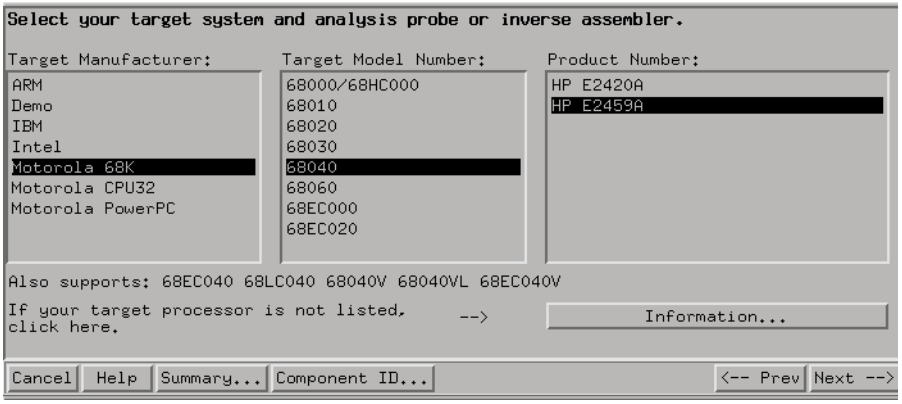
If your target processor is not listed... (see page 18)

To specify a Manufacturer-Processor-Analysis Probe combination:

1. In the left data column, select the manufacturer of your target processor or bus.
2. Scroll among the processors listed in the center column and select the processor used in your target system. If additional processors are supported, their names will appear below the data columns.
3. If your target processor is supported by more than one analysis probe, select the analysis probe that you will be using.

   If you are not sure which product number you have, you can check:

   - The "At a Glance" section of your processor solution manual.
   - The front cover of your analysis probe or inverse assembler *User's Guide*.
   - The packing list in the box for your analysis probe or inverse assembler software.



If your target processor is not listed

If your combination of target manufacturer, processor, and analysis probe is not available, check the following:

- Make sure you are looking in the proper processor family (left column).
- If the "Also supports:" field is visible below the selection windows, see if your processor is listed there.
- Make sure that you “Install Software” on page 18 that was shipped with your order.
- If, after checking the above, you still cannot locate your target processor, cancel out of the Setup Assistant and use the “Manual Configuration” on page 18 process described in your product's User's Guide.

Install Software

For the Setup Assistant to recognize your product, you may need to install the software that was shipped with your analysis probe.

- Locate the System CD-ROM and insert it into your CD-ROM drive.
- Select *Install* from the System Administration dialog.
- Locate and install the fileset for your processor.

Manual Configuration

If your combination of target manufacturer, processor and analysis probe is not available after all current software has been installed, you will not be able to proceed with the Setup Assistant. Refer to the manual for your analysis probe (or optional product) which will outline these configuration steps:

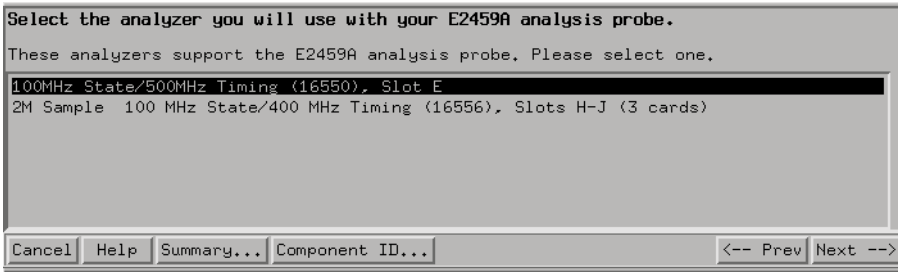
- Connecting the analysis probe to your target system.

- Connecting the analysis probe to your logic analyzer.
- Setting switches on the analysis probe board, if necessary.
- Loading a config file and inverse assembler into the logic analyzer.
- Connecting the emulation probe to the analyzer. (directly or via LAN)
- Connecting the emulation probe to your target system.
- Configuring and programming the emulation probe.
- Connecting the emulation module to your target system.
- Programming the emulation module.
- Establishing a network connection to a code development machine.
- Specifying a network path to source and symbol files.

If your processor was purchased from a manufacturer other than the one listed, check its compatibility in your analysis probe user's guide, or contact Agilent technical support.

Selecting an Analyzer

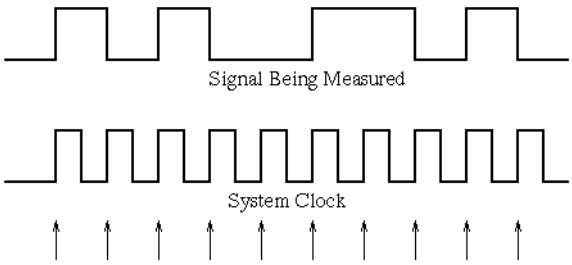
The Setup Assistant has queried your analyzer frame and is displaying the analyzers which are compatible with the chosen analysis probe. If more than one analyzer is available, select the analyzer you wish to connect to your analysis probe.



You will be instructed to connect the analyzer pods to connectors on the analysis probe board.

Choosing an Acquisition Mode

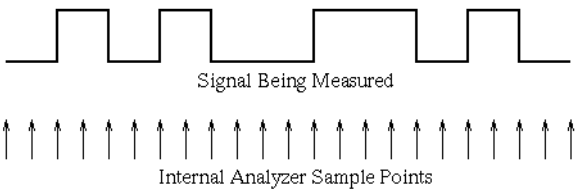
- State Mode - Data is acquired based on a clock signal from your target system. Use this mode when viewing signals controlled by a common clock. State mode is required for inverse assembly and source code correlation.



The rising edge of a clock signal from the circuit being tested is used to measure and display other signals from the system.

State Data Acquisition

- Timing Mode - Data is acquired using an asynchronous clock internal to the analyzer system. Inverse assembly and source code correlation are not available in Timing mode. In some cases, Timing mode requires a different analyzer configuration than State mode.



Signals from the system under test are sampled at regular intervals determined by a sample clock internal to the logic analyzer

Timing Data Acquisition

Connecting To Your Target System

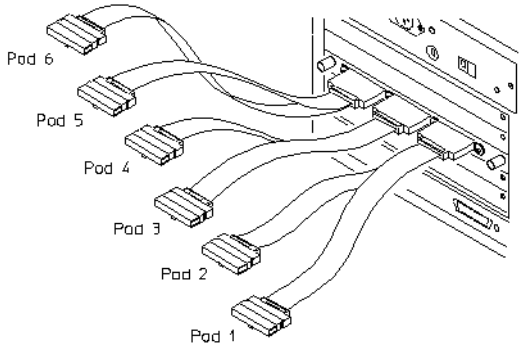
For many of the supported analysis probes, online connection instructions are available, and accessed by the Setup Assistant at the appropriate time.

If you prefer hardcopy manual instructions, or a certain analysis probe is not supported with online connection instructions, refer to "Connecting and Configuring Your System" in the User's Guide for your Analysis Probe.

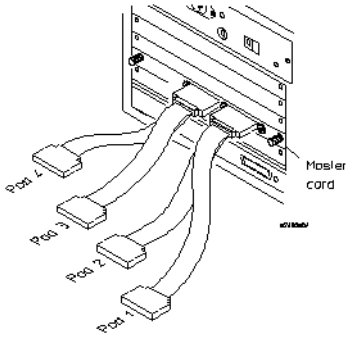
Connecting To Your Analyzer Card

- Connecting to the 16550 and 16710/11/12 (see page 23)
- Connecting to the 16554/55/56/57 and 16715/6/7 (see page 23)
- Connecting to the 16600 Series (see page 24)
- The E5346A High-density probe adapter cable (see page 25)

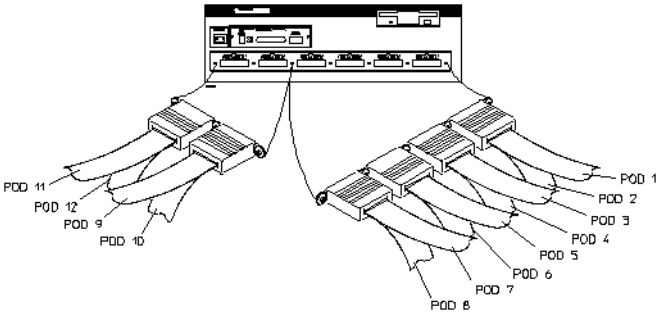
16550 and 16710/11/12 Analyzers



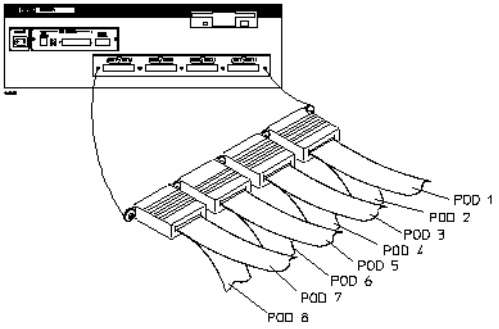
16554/55/56/57 and 16715/16/17-series Analyzers



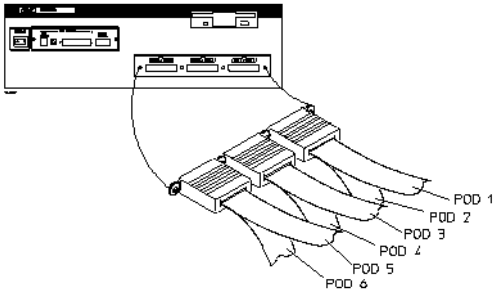
16600 Analyzer



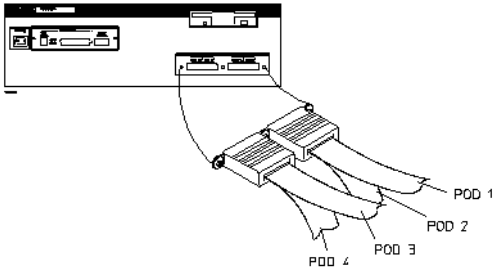
16601 Analyzer



16602 Analyzer

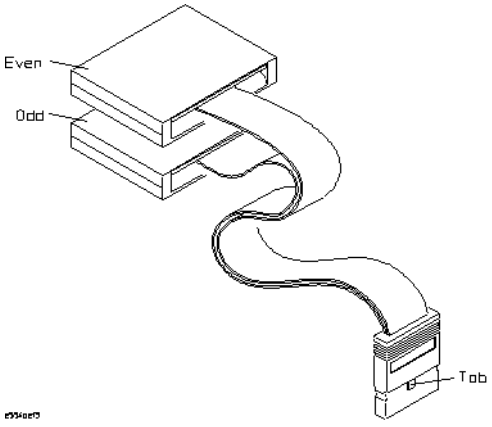


16603 Analyzer



E5346 High-density Probe Adapter Cables

E5346A High-density adapter cables are used to connect certain analysis probes to a logic analyzer. Each high-density connector that attaches to the analysis probe expands to two logic analyzer connectors, labeled Odd and Even.

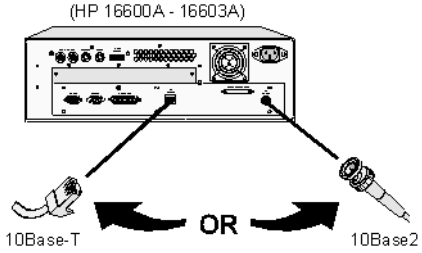

Even
Odd
Tab

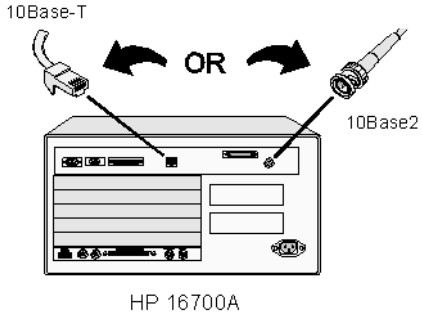
Setting Up A Network

If your logic analysis system is not connected to a network, and you want to access source or symbol files on a separate machine, the setup assistant will invoke the network setup utility.

Make sure your system is connected to the network using a 10BASE2 or 10BASE-T connector.

Network Setup Help (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume).





Saving Settings

The Setup Assistant has completed the process of setting up your logic analyzer system.

To save your system configuration to a file on hard disk:

- Select "Save Setup" in the Setup Assistant dialog.
- From the "File Manager" dialog, enter a path and filename.
- Set the Source to "All" to save the setup for all tools that are configured.
- Select "Save Config and Data" to save everything. (To save your setup but no data, select "Save Config Only")
- Select "Save".

To load your settings at a later time:

- Select a file in the File Manager dialog, then select File->Load.

To specify that these settings should automatically load at power-up:

- Save your settings in a system configuration (workspace) file.
- In the File Manager dialog, select Options->Autoload.
- Enter a filename by typing or selecting the director and file.
- Select "Enable Autoload".

Taking the Next Step

When you have finished setting up your system, press Done.

The Setup Assistant has prepared your system for taking measurements, and will activate a measurement window as it exits.

- After pressing "Run", you can view basic State or Timing measurements in the Lister or Waveform display tools.
- You can capture specific target system data of interest by accessing the Trigger menu belonging to the analyzer you are using.
- You can view inverse assembled code in a Lister window.
- You can load symbols and view correlated source code in a Lister window with the Source Viewer activated.
- If you are using an emulation module/probe, use the emulation control window to configure processor parameters, set breakpoints, start/stop the processor, or download code.

Please refer to

- The user's guide for your analysis probe.
- The user's guide for your emulation module/probe.
- The processor solutions manual if you received one.
- Online help for any of the system components.
- The search utility (see the *Help On Help* help volume) of the online help system.

See Also

- The tutorial booklet "Making Basic Measurements".
- The online Measurement Examples (see the *Measurement Examples* help volume)

Choosing To Set Up Only Source Correlation

The Source Correlation tool requires an analysis probe for gathering data from your target system. Because you have chosen to set up only the Source Correlation tool, it is assumed that you have previously set up an analysis probe (and inverse assembler).

If you have not previously set up an analysis probe, return to the main Setup Assistant window and choose "Full Measurement". Following that path will allow you to set up an analysis probe, an emulation module or probe, and the Source Correlation tool. Without an analysis probe, the Source Correlation tool will not function.

Licensing the Source Correlation Tool

The Source Correlation tool is a licensed product and must be set up with a valid password. If the Setup Assistant determines that your Source Correlation tool contains a valid password, the set up process will proceed.

If your Source Correlation tool has not been set up with a valid password you will be prompted with these choices:

- License Entry - This button accesses the license administrator for you to type in the password that was issued with your product.
- Set Demo License - This button allows you to install a demo license that is valid for 21 days.

To view the license status of your system, select Licensing in the System Administration dialog.

- To obtain a license for the Source Viewer display (see the *Listing Display Tool* help volume)
- More information on licensing (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

Locating Source Code

- On a networked UNIX workstation If you are developing code on a UNIX workstation connected to the network, the source code on that computer can be referenced via an NFS connection from your logic analysis system. The computer containing the sources must be configured as an NFS File Server. The Setup Assistant will prompt you to establish this NFS connection.
- On a networked PC If your source code is on a Windows/NT PC, you can share the PC's hard drive with the logic analysis system. The Setup Assistant will prompt you to connect the PC share.
- On the hard drive of your logic analysis system If you do not have access to a networked computer, sources and a symbol file can be transferred to a directory within your logic analysis system using the File Manager (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume). Specify the directory where you have copied these files for use by the Source Viewer.

Mounting Remote Disks

To access source code or symbols files from a development machine on the network, an an NFS mount must be established to the machine. The Setup Assistant will display the standard system dialog for making this connection.

When you are finished setting up the NFS connection, close the "NFS Client Setup" window and return to the Setup Assistant, then press "Next" to continue the set up process.

Loading Software Symbols

The Source Correlation tool requires access to a symbol file from your compilation process. The symbol file can be loaded when prompted by the Setup Assistant, or loaded/reloaded via the Symbols tab within the Source Viewer window.

See Also

"The Symbols Tab" on page 59

Using Source Correlation

The Source Correlation tool set allows correlating trace data obtained by the analyzer via the analysis probe back to source code.

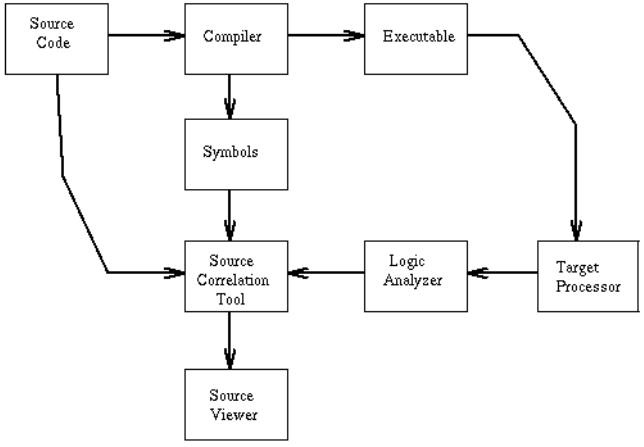
The Setup Assistant will determine if Source Correlation is supported for the processor you have chosen, and if the proper licensing is in place.

Using Source Correlation requires that:

- A symbol file generated during the linking of the executing code must be available.
- The source files used to compile and link the executing code must be accessible.

These required data files can exist on the logic analysis system, or, more likely, on a machine that is NFS mounted to the local machine.

This diagram represents the use of Source Correlation to provide a correlated listing of source code.



Source code is written by the designer and compiled into an executable while creating a by-product symbol file. The executable is downloaded to run on the target processor, and the symbol file is loaded into the

Source Correlation tool. As the executable runs on the target processor, the logic analyzer captures data via an analysis probe. The Source Correlation tool uses trace data, symbol information and the source code files to provide a correlated source listing of the executing program in a listing display.

Choosing the Type of Emulation Measurement

- Select Emulation along with an analysis probe or inverse assembler if you will use an *analysis probe* to trace the activity of your target processor. The emulation module may be connected to your target system via a target interface module or, if the analysis probe has the appropriate connector, it may be connected via the analysis probe.
- Select Emulation by itself if you will use an emulation module without an analysis probe.

Installing Emulation Control Software

If the Emulation Control Interface software for your microprocessor has not already been installed, you need to install it from a CD-ROM.

Install the software that was shipped with your emulation module or emulation probe:

- Insert the CD-ROM into your CD-ROM drive.
- Select *Install* from the Setup Assistant dialog.
- From the software installation menu, install the appropriate package for your processor.

See Also

Software Installation (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

Connecting an Emulation Probe to the Network

You may use a stand-alone *emulation probe* instead of, or in addition to, a built-in *emulation module*.

The Setup Assistant will guide you through the following steps:

- "Deciding if an Emulation Probe is on the LAN" on page 40
- "Choosing a Connection Type" on page 39
- "Setting Up a LAN Connection" on page 40
- "Setting Up a LAN Connection (Interconnected E5900B)" on page 42
- "Setting Up a Point-to-Point Connection" on page 44
- "Setting Network Parameters" on page 43

Choosing a Connection Type

- Choose Site LAN connection if you have an Ethernet LAN available and you plan to connect your logic analysis system and emulation probe to the LAN.
- Choose Point-to-Point if you don't plan to connect your logic analysis system to a LAN. This option is displayed only if your logic analysis system is not connected to a LAN.

  Advantages of a point-to-point connection:

  - No need for a system administrator to assign IP addresses. (You can use any IP address for the emulation probe, and it will not conflict with other devices on the LAN.)
  - The logic analysis system cannot be accessed across the network (required in some high-security environments).
  - Can be used when a LAN is not available.

  Disadvantages of a point-to-point connection:

  - Neither logic analysis system nor the emulation probe can be

connected to a site LAN.

- The emulation probe must be near the logic analysis system.
- A special "point-to-point" LAN cable must be used.
- Remote file systems cannot be mounted for access to source code files, symbol files, or executable files.
- The emulation probe cannot be controlled by a debugger on a host computer, except via a relatively slow serial connection.

Deciding if an Emulation Probe is on the LAN

- Choose Yes if the emulation probe you wish to use is connected to your site LAN and has been configured with a LAN address. The next step will ask you to enter the LAN address so that a connection can be established.
- Choose No if the emulation probe is not set up on the LAN. The Setup Assistant will guide you through the process of connecting the emulation probe to the LAN.

Setting Up a LAN Connection

First, you need to make the physical connection to the LAN. Once you have done this, the Setup Assistant will help you set up a LAN address for the emulation probe.

The process of connecting the emulation probe is different for the E5900A and E5900B.

1. “Setting Up a LAN Connection (E5900A)” on page 40

   If the label on the emulation probe does not say "E5900B" then it is an E5900A.

2. “Setting Up a LAN Connection (E5900B)” on page 42

Setting Up a LAN Connection (E5900A)

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.

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

To physically connect the emulation probe to the network:

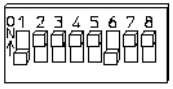
1. Connect a network cable to either the 10BASE-T connector or the 10BASE2 BNC connector.

NOTE: Make sure the probe is on the same subnet as the logic analysis system during initial setup; otherwise, probe LAN address setup will fail. After initial setup, you can modify the probe's LAN parameters using the Emulation Control Interface before moving the probe to a different subnet.

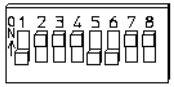
2. Set the configuration switches to indicate the type of connection that is to be made.

   Switch S1 must be set to OPEN (OFF), indicating that a LAN connection is being made. Switch S6 must bet set to OPEN to allow programming of the LAN parameters.

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



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



   If you are using the 10BASE-T connector, see also To set the 10BASE-T configuration switches (see the *Emulation: Setting Up* help volume).

Set all other switches to CLOSED.

3. Cycle power on the emulation probe by unplugging the power cord, then plugging it in again. You must cycle power before switch changes take effect.
4. Wait at least 30 seconds for the emulation probe to connect to the LAN.

When you have made the connection, close this window and select Next to program the emulation probe with an IP address.

Setting Up a LAN Connection (E5900B)

The emulation probe has an IEEE 802.3 Type 10/100Base-TX LAN connector.

To physically connect the emulation probe to the network:

- Connect a network cable to the LAN connector

NOTE: Make sure the probe is on the same subnet as the logic analysis system during initial setup; otherwise, probe LAN address setup will fail. After initial setup, you can modify the probe's LAN parameters using the Emulation Control Interface before moving the probe to a different subnet.

When you have made the connection, close this window and select Next to program the emulation probe with an IP address.

Setting Up a LAN Connection (Interconnected E5900B)

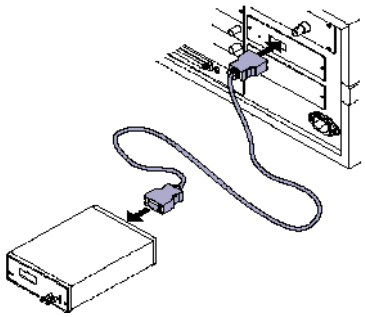
The E5901B emulation module requires an E5900B emulation probe. The emulation probe needs to be connected to the emulation module using the module/probe interconnect cable *and* the emulation probe needs to be connected to the LAN.

Connect to the emulation module

If the emulation module is not already connected to the emulation probe, connect it now:

1. Connect one end of the module/probe interconnect cable to the E5901B emulation module in your logic analysis system.
2. Connect the other end of the module/probe interconnect cable to the

E5900B emulation probe.



3. Turn the emulation probe's power switch ON. Power is supplied by the module/probe interconnect cable.

Remember that the emulation probe also needs to be connected to the LAN.

Connect to the LAN

The emulation probe has an IEEE 802.3 Type 10/100Base-TX LAN connector. To physically connect the emulation probe to the network:

1. Connect a network cable to the LAN connector
2. Cycle power on the emulation probe by turning the power switch OFF then ON again.
3. Wait at least 20 seconds for the emulation probe to connect to the LAN.

When you have made the connection, close this window and select Next to program the emulation probe with an IP address.

Setting Network Parameters

The Setup Assistant will open the Init Probe Lan Addresses window, which is part of the Emulation Control Interface toolset. Help for using this dialog is available from the dialog itself or from the Emulation Control Interface.

Setting Up a Point-to-Point Connection

A point-to-point connection is also called a *stand-alone configuration*. The logic analysis system and the emulation probe form an isolated network with only two nodes.

The process of connecting the emulation probe is different for the E5900A and E5900B.

1. "Setting Up a Point-to-Point Connection (E5900A)" on page 44

   If the label on the emulation probe does not say "E5900B" then it is an E5900A.

2. "Setting Up a Point-to-Point Connection (E5900B)" on page 45

See Also

- "Choosing a Connection Type" on page 39

Setting Up a Point-to-Point Connection (E5900A)

1. Physically connect the emulation probe to the logic analysis system:
   a. Use a point-to-point 10Base-T cable (part number 5061-7342) to connect the emulation probe directly to the logic analysis system.
   b. Set the configuration switches for a LAN configuration:
   c. Set switch 1 to OPEN (OFF), indicating that a LAN connection is being made.
   d. Set switch 5 to OPEN (OFF), indicating a 10BASE-T connection.
   e. Cycle power on the emulation probe. Leave the emulation probe powered on while you configure the network.
   f. While you are near the emulation probe, write down the *link-level address* of the emulation probe. This address is printed on a sticker near the LAN connector on the emulation probe, labelled "LLA".
2. Configure the network on the logic analysis system:

a. In the Setup Assistant window, select *Next*.

b. Select *Network Setup...*.

c. Select *Standard* to turn on networking.

Leave the network parameters with the default values. The IP address should be 192.0.2.231.

d. In the Network Setup dialog, select *OK*.

3. Configure the network on the emulation probe:

a. In the Setup Assistant window, select *Next*.

b. Select *Init LAN...*.

c. Enter the link-level address of the emulation probe, which you wrote down earlier.

d. Enter the following IP address: 192.0.2.233

e. Select *OK* then follow the instructions.

4. In the Setup Assistant window, select *Next* to continue.

Setting Up a Point-to-Point Connection (E5900B)

1. Physically connect the emulation probe to the logic analysis system:

a. Use a point-to-point 10/100Base-T cable (such as part number 5061-7342) to connect the emulation probe directly to the logic analysis system.

b. Leave the emulation probe powered on while you configure the network.

c. If the emulation probe is not connected to an E5901B emulation module, write down the *link-level address* of the emulation probe. This address is printed on a sticker near the LAN connector on the emulation probe, labelled "LLA".

2. Configure the network on the logic analysis system:

a. In the Setup Assistant window, select *Next*.

b. Select *Network Setup...*.

c. Select *Standard* to turn on networking.

   Leave the network parameters with the default values. The IP address should be 192.0.2.231.

d. In the Network Setup dialog, select *OK*.

3. Configure the network on the emulation probe:

   a. In the Setup Assistant window, select *Next*.

   b. Select *Init LAN...*.

   c. If the emulation probe is not connected to an E5901B emulation module, enter the link-level address of the emulation probe, which you wrote down earlier.

   d. Enter the following IP address: 192.0.2.233

   e. Select *Set Addresses* then follow the instructions.

4. In the Setup Assistant window, select *Next* to continue.

Entering the Emulation Probe IP Address

Enter the IP address or LAN name of the emulation probe.

When you select Next, the logic analysis system will attempt to communicate with the emulation probe on the LAN.

If an error message is displayed

If the logic analysis system is unable to communicate with the emulation probe:

- Check that the emulation probe has been plugged in long enough for it to complete its power-on self test (about 30 seconds).
- Check that the emulation probe is physically connected to the LAN.
- Check that the IP address you entered is correct.

If you need to program the LAN address for the emulation module, select Prev then select No to tell the Setup Assistant that the emulation probe is not set up on the LAN.

See Also

Your emulation probe *User's Guide* for more information on LAN communication problems.

Choosing a Firmware Version

For some processors, more than one version of the emulation module firmware may be available. For example, special firmware might be needed to accommodate bugs in certain mask revisions of the target microprocessor.

Choose the firmware version which is appropriate for your processor.

To display information about the specific firmware versions, select *Additional Information*.

Connecting the Emulation Hardware

Now is the time to connect your emulation module or emulation probe to your target system.

Follow the instructions on the screen (if any), OR select *Connection Info...* to see instructions on how to make this connection.

The instructions assume that your emulation module/probe is correctly connected to your logic analysis system, and that your target system has been designed according to the requirements in the manual.

When you are finished, close the Help window and select *Next* to continue.

See Also

The manual for your emulation solution, emulation module, or emulation probe for details on making the connection.

Choosing to Use Emulation

To use the Emulation Control Interface, you will need:

- An *emulation probe*.
- If you are using an E5900A emulation probe, a target interface module (TIM) or an analysis probe which has an emulation connector. If you ordered a "processor solution," you should have all the parts you need.

If you have the necessary parts, and you want to use the Emulation Control Interface, select Yes. The Setup Assistant will guide you through the process of connecting to your microprocessor and configuring the emulation hardware.

If you are using a debugger to control the microprocessor via an emulation probe, select Yes. This will let you configure the emulation probe using the Emulation Control Interface, which is usually much easier than configuring using the emulation probe's built-in terminal interface. You can then end the Emulation Control Interface session and use your debugger.

If you have an emulation module, but it is not installed yet, select |No|. Install the emulation module (instructions are in the manual), then run Setup Assistant again.

If you choose No, you may use the Setup Assistant at a later time to set up emulation. You can also set up emulation using the information provided in your emulation manual.

See Also

"To decide if you have an E5900A or an E5900B" on page 57

Setting Up and Starting the Emulation Control Interface (see the *Emulation: Setting Up* help volume)

Your emulation manual: *Processor Solution User's Guide* or *Emulation Probe Installation and Service Guide/User's Guide*.

Glossary

Choosing to Use Emulation

To use the Emulation Control Interface, you will need:

- An E5900A *emulation probe*.
- A target interface module (TIM) or an analysis probe which has a emulation connector. If you ordered a "processor solution," you should have all the parts you need.

If you have the necessary parts, and you want to use the Emulation Control Interface, select Yes. The Setup Assistant will guide you through the process of connecting to your microprocessor and configuring the emulation hardware.

If you are using a debugger to control the microprocessor via an emulation probe, select Yes. This will let you configure the emulation probe using the Emulation Control Interface, which is usually much easier than configuring using the emulation probe's built-in terminal interface. You can then end the Emulation Control Interface session and use your debugger.

If you have an emulation module, but it is not installed yet, select |No|. Install the emulation module (instructions are in the manual), then run Setup Assistant again.

If you choose No, you may use the Setup Assistant at a later time to set up emulation. You can also set up emulation using the information provided in your emulation manual.

See Also

"To decide if you have an E5900A or an E5900B" on page 57

Setting Up and Starting the Emulation Control Interface (see the *Emulation: Setting Up* help volume)

Your emulation manual: *Processor Solution User's Guide* or *Emulation Probe Installation and Service Guide/User's Guide*.

Glossary

Choosing to Use Emulation

To use the Emulation Control Interface, you will need:

- An E5900B *emulation probe*.
- A target system or an analysis probe which has a debug port.

If you have the necessary parts, and you want to use the Emulation Control Interface, select Yes. The Setup Assistant will guide you through the process of connecting to your microprocessor and configuring the emulation hardware.

If you are using a debugger to control the microprocessor via an emulation probe, select Yes. This will let you configure the emulation probe using the Emulation Control Interface, which is usually much easier than configuring using the emulation probe's built-in terminal interface. You can then end the Emulation Control Interface session and use your debugger.

If you have an emulation module, but it is not installed yet, select |No|. Install the emulation module (instructions are in the manual), then run Setup Assistant again.

If you choose No, you may use the Setup Assistant at a later time to set up emulation. You can also set up emulation using the information provided in your emulation manual.

See Also

"To decide if you have an E5900A or an E5900B" on page 57

Setting Up and Starting the Emulation Control Interface (see the *Emulation: Setting Up* help volume)

Your emulation manual: *Processor Solution User's Guide* or *Emulation Probe Installation and Service Guide/User's Guide*.

Glossary

Choosing to Use Emulation

To use the Emulation Control Interface, you will need:

- The E5901A or E5901B *emulation module* installed in your logic analysis system.
- If you have an E5901A emulation module, a target interface module (TIM) or an analysis probe which has a emulation connector. If you ordered a processor solution package, you should have all the parts you need.
- If you have an E5901B emulation module, an E5900B emulation probe.

If you have the necessary parts, and you want to use the Emulation Control Interface, select Yes. The Setup Assistant will guide you through the process of connecting to your microprocessor and configuring the emulation hardware.

If you are using a debugger to control the microprocessor via the emulation module, select Yes. This will let you configure the emulation probe using the Emulation Control Interface, which is usually much easier than configuring using the emulation probe's built-in terminal interface. You can then end the Emulation Control Interface session and use your debugger.

If you choose No, you may use the Setup Assistant at a later time to set up emulation. You can also set up emulation using the information provided in your emulation manual.

If your logic analysis system contains an emulation module, but you wish to use a stand-alone emulation probe, choose No. In this case, you will need to set up emulation using the information in your emulation manual.

See Also

"To decide if you have an E5901A or an E5901B" on page 57

Setting Up and Starting the Emulation Control Interface (see the *Emulation: Setting Up* help volume)

Your emulation manual: *Processor Solution User's Guide* or *Emulation Probe Installation and Service Guide/User's Guide*.

Glossary

Connecting Emulation to the Target

Choose the instructions for your configuration:

- To connect an emulation module to your target system using a debug port. (see page 54)
- To connect an emulation module to your target system using an analysis probe. (see page 55)
- If you have a stand-alone emulation probe, refer to the manual.
- If you have a Pentium or Pentium II emulation module, refer to the manual.

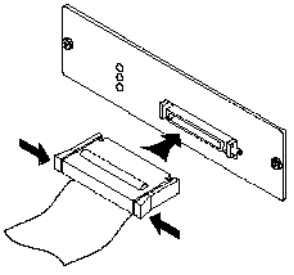
See Also

The *User's Guide* or *Installation/Service Guide* for your processor solution, emulation module, or emulation probe.

To Connect Using a Debug Port

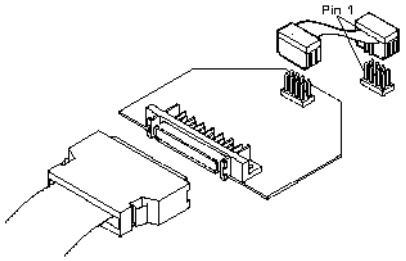
The emulation module can be connected to a target system through a debug port (BDM or JTAG connector). The emulation module should be connected to a header connector on the target system using the cable assembly provided.

1. Plug one end of the 50-pin cable into the emulation module.



2. Plug the other end of the 50-pin cable into the target interface module.
3. Plug one end of the small ribbon cable into the target interface module.
4. Plug the other end of the small ribbon cable into the debug port on the

target system.



The target interface module (emulation adapter) for your processor may look different from the one pictured here.

To Connect Using an Analysis Probe

1. Remove power from the target system.
2. Plug one end of the 50-pin cable into the emulation module.
3. Plug the other end of the 50-pin cable into the connector on the analysis probe.

Connecting PC Shares

To access source code or symbol files from a development PC on the network, you must connect the PC share containing these files to your logic analyzer. The Setup Assistant will display the standard system dialog for making this connection.

For more information on making the PC directory containing the source code or symbol files a "PC share", see your PC's documentation.

When you are finished setting up the PC share connection, close the "Map Windows Drive" dialog and return to the Setup Assistant. Then press "Next" to continue the set up process.

Measurement Examples

This section being developed by Pat White.

To decide if you have an E5900A or an E5900B

There are two kinds of emulation probes: the E5900A and the E5900B. Use the following information to decide which kind you have.

E5900A

- Label: E5900A or no product number
- Target connection: through target interface module (*TIM*) or *analysis probe*

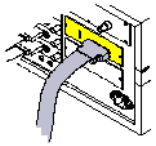
E5900B

- Label: E5900B
- Target connection: directly to background debug port on the target system or on an analysis probe
- Host connection: Often connects to logic analysis system through a LAN connection and through a module/probe interconnect cable attached to a E5901B *emulation module*

To decide if you have an E5901A or an E5901B

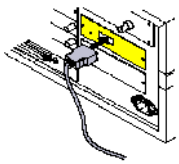
There are two kinds of emulation modules: the E5901A and the E5901B. Use the following information to decide which kind you have.

E5901A



- Label: E5901A or HP 16610A
- Connects to: target interface module (*TIM*) or *analysis probe*

E5901B



- Label: E5901B
- Connects to: E5900B *emulation probe*

The Symbols Tab

The Symbols tab offers control of the *symbols* capabilities. Symbols represent patterns and ranges of values found on labeled sets of bits. Two kinds of symbols are available:

- Object File Symbols. These are symbols from your source code and symbols generated by your compiler.
- User-Defined Symbols. These are symbols you create.

To load symbols, see:

- "To Load Object File Symbols" on page 61
- "To Load User-Defined Symbols" on page 79

Symbols are available for all state and timing analyzers. Each label listed in the Format menu can have its own group of symbols associated with it.

- "User-Defined Symbols" on page 78
- "Setting Up Object File Symbols" on page 61
- "Using Symbols In The Logic Analyzer" on page 73
- "Displaying Data in Symbolic Form" on page 60

Displaying Data in Symbolic Form

You can display data in symbolic form in some of the display tools, such as the Listing display and the Waveform display.

To View Symbolic Values in a Waveform Display

1. Select the label name where you want to display symbolic values.
2. Choose *Properties...*.
3. In the Properties dialog:
   - Set ShowValue to *On*.
   - Set Base to *Symbols* or *Line #s*.
   - Select the *OK* button.

   The symbolic names for the values now appear in the overlayed bus waveform.

To View Symbolic Values in a Listing Display

1. Select the numeric base of the label where you want to display symbolic values.
2. Set the numeric base to *Symbols* or *Line #s*.
   The symbolic names for the values now appear instead of numeric data.

Setting Up Object File Symbols

Object file symbols can include variable names, procedure or function names, and source file names with line numbers. The linkage between symbol names and address or data values comes from one of two sources:

- Object files that are created by your compiler/linker.
- ASCII symbol files you create with a text editor.

To use object file symbols

1. Generate an object file with symbolic information using your software development tools.
2. If your language tools cannot generate object file formats that are supported by the logic analyzer, create an ASCII symbol file (see page 65).
3. Load the object file (see page 61) or ASCII symbol file into the logic analyzer.
4. If necessary, relocate sections of your code (see page 63).

See Also

"Using Symbols In The Logic Analyzer" on page 73

"Symbol File Formats" on page 64

To Load Object File Symbols

1. Select the *Symbol* tab and then the *Object File* tab.
2. Select the label name you want to load object file symbols for.
   In most cases you will select the label representing the address bus of the processor you are analyzing.
3. Specify the directory to contain the symbol database file (*.ns*) in the field under, *Create Symbol File (.ns) in This Directory*. Select the *Browse...* button if you wish to find an existing directory name.
4. In the *Load This Object/Symbol File For Label* field, enter the object file

name containing the symbols. Select the *Browse...* button to find the object file and select the *Load* button in the Browser dialog.
If your logic analyzer is NFS mounted to a network, you can select object files from other servers.

To reload object file symbols

1. Select the object file/symbol file to reload from the *Object Files with Symbols Loaded For Label* field.
2. Select the *Reload* button.

Value update

The values of the object file symbols being used as terms or as SPA state-interval ranges will be updated automatically each time the object file symbols are reloaded.

Configuration file save

The name of the current object file is saved when a configuration file is saved. The object file will be reloaded when the configuration is loaded.

Multiple files

You can load the same symbol file into several different analyzers, and you can load multiple symbol files into one analyzer. Symbols from all the files you load will appear together in the object file symbol selector that you use to set up resource terms.

Object file versions

During the load process, a symbol database file with a *.ns* extension will be created by the system. One *.ns* database file will be created for each symbol file you load. Once the *.ns* file is created, the Symbol Utility will use this file as its working symbol database. The next time you need to load symbols into the system, you can load the *.ns* file explicitly, by placing the *.ns* file name in the *Load This Object/Symbol File For Label* field.

If you load an object file that has been loaded previously, the system will compare the time stamps on the *.ns* file and the object file. If the object file is newer, the *.ns* file will be created. If the object file has not

been updated since it was last loaded, the existing *.ns* file will be used.

See Also

"Using Symbols In The Logic Analyzer" on page 73

"Symbol File Formats" on page 64

Relocating Sections of Code

Use this option to add offset values to the symbols in an object file. You will need this if some of the sections or segments of your code are relocated in memory at run-time. This can occur if your system dynamically loads parts of your code so that the memory addresses that the code is loaded into are not fixed.

To Relocate a Single Section of Code

1. Select the *Symbol* tab and then the *Object File* subtab.
2. Select the *Relocate Sections...* button.
3. In the Address column, select the address you wish to relocate.
4. In the *Edit selected section* field, enter the new address.
5. Select *Apply Edit*.
6. Repeat steps 3 through 5 above for any other sections to be relocated.
7. Select the *Close* button.

To Relocate All Sections of Code

1. Select the *Symbol* tab and then the *Object File* subtab.
2. Select the *Relocate Sections...* button.
3. Enter the value you wish to use in the *Offset all selections by* field.
4. Select the *Apply Offset* button.
5. Select the *Close* button.

To Delete Object File Symbol Files

1. Select the *Symbol* tab, and then the *Object File* subtab.
2. Select the file name you want to delete in the text box labeled, *Object Files with Symbols Loaded For Label*.
3. Select the *Unload* button.

Symbol File Formats

The logic analysis system can read symbol files in the following formats:

- OMF96
- OMFx86
- IEEE-695
- ELF/stabs
- TI COFF

For ELF/stabs, and ELF/stabs/Mdebug files, C++ symbols are demangled so that they can be displayed in the original C++ notation. To improve performance for these ELF symbol files, type information is not associated with variables. Hence, some variables (typically a few local static variables) may not have the proper size associated with them. They may show a size of 1 byte and not the correct size of 4 bytes or even more. All other information function ranges, line numbers, global variables and filenames will be accurate. These behaviors may be changed by creating a readers.ini (see page 70) file.

See Also

“Creating ASCII Symbol Files” on page 65Creating ASCII Symbol Files

“Creating a readers.ini File” on page 70Creating a readers.ini File

Creating ASCII Symbol Files

If your language tool chain does not produce object files in a supported format, you can create an ASCII symbol file to define symbols. You can also use an ASCII symbol file to define symbols that are not included in your object file.

You can create an ASCII symbol file using any text editor that supports ASCII format text. Each entry in the file you create must be a string of ASCII characters consisting of a symbol name followed by an address or address range. The address or address range must be a hexadecimal number. It must appear on the same line of the text file as the symbol name and it must be separated from the symbol name by one or more blank spaces or tabs. Address ranges must be in the following format:

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

Two formats are available for creating ASCII symbol files:

"Simple Format" on page 65

"Record Header Format" on page 65

NOTE: It is possible to generate ASCII symbol files from the symbol or load map output of most language tools.

Simple Format

An ASCII symbol file can be a simple list of name/address pairs.

Example

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

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

Record Header Format

An ASCII symbol file can be divided into records using key words,

called *record headers*. The different records allow you to specify different kinds of symbols, with differing characteristics. An ASCII symbol file can contain any of the following kinds of records:

"Start Address" on page 67

"Sections" on page 67

"Functions" on page 67

"Variables" on page 69

"Source Line Numbers" on page 68

"Comments" on page 69

The record headers must be enclosed in square brackets, like this: [HEADER]. If no record header is specified, the lines following are assumed to be symbol definitions in one of the VARIABLES formats:

```
variable   address
variable   start..end
variable   start address   size
```

Example

Here is an ASCII symbol file that contains several different kinds of records.

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

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

[VARIABLES]
total    40002000  4
value    40008000  4

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

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

Start Address . Format

```
[START ADDRESS]
address
```

address - The address of the program entry point, in hexadecimal.

Example

```
[START ADDRESS]
00001000
```

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

Format

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

func_name - A symbol representing the function name.

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

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

Example

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

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

Format

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

section_name - A symbol representing the name of the section.

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

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

attribute - (optional) Attribute may be one of the following:

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

NONRELOC - The section contains variables or code that cannot be relocated. In other words, this is an absolute segment.

Example

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

NOTE: If Section definitions are used in an ASCII symbol file, any subsequent Function or Variable definitions must fall within the address ranges of one of the defined Sections. Those Functions and Variables that do not will be ignored by the Symbol Utility.

Source Line Numbers . Use SOURCE LINES to associate addresses with lines in your source files.

Format

```
[SOURCE LINES]
File: file_name
line#  address
```

file_name - The name of a file.

line# - The number of a line in the file, in decimal.

address - The address of the source line, in hexadecimal.

Example

```
[SOURCE LINES]
File: main.c
10       00001000
11       00001002
```

```
14        0000100A
22        0000101E
```

See Also Using the Source Viewer (see the *Listing Display Tool* help volume)

Variables. You can specify symbols for variables using:

- The address of the variable.
- The address and the size of the variable.
- The range of addresses occupied by the variable.

If you give only the address of a variable, the size is assumed to be 1 byte.

Format

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

var_name - A symbol representing the variable name.

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

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

size - (optional) The size of the variable, in bytes, in decimal.

Example

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

Comments . Any text following a # character is ignored by the Symbol Utility. The # can be used to comment a file. Comments can appear on a line by themselves, or on the same line, following a symbol entry.

Format

```
#comment text
```

Example

```
#This is a comment
```

Creating a readers.ini File

You can change how an ELF/Stabs, Ticoff or Coff/Stabs symbol file is processed by creating a reader.ini file.

1. Create the reader.ini file on your workstation or PC.
2. Copy the file to /logic/symbols/readers.ini on the logic analysis system.

Reader options

C++Demangle

```
1= Turn on C++ Demangling (Default)
0= Turn off C++ Demangling
```

C++DemOptions

```
803= Standard Demangling
203= GNU Demangling      (Default Elf/Stabs)
403= Lucid Demangling
800= Standard Demangling without function parameters
200= GNU Demangling without function parameters
400= Lucid Demangling without function parameters
```

MaxSymbolWidth

```
80= Column width max of a function or variable symbol
    Wider symbols names will be truncated.
    (Default 80 columns)
```

OutSectionSymbolValid

```
0= Symbols whose addresses aren't within the
    defined sections are invalid (Default)
1= Symbols whose addresses aren't within the
    defined sections are valid
```

This option must be specified in the Nsr section of the Readers.ini file:

```
[Nsr]
OutSectionSymbolValid=1
```

ReadElfSection

```
2= Process all globals from ELF section (Default)
   Get size information of local variables
```

```
1= Get size information of global and local variables
   Symbols for functions will not be read, and
   only supplemental information for those symbols in
   the Dwarf or stabs section will be read.
0= Do not read the Elf Section
```

If a file only has an ELF section this will have no effect and the ELF section will be read completely. This can occur if the file was created without a "generate debugger information" flag (usually -g). Using the -g will create a Dwarf or Stabs debug section in addition to the ELF section.

StabsType

```
 StabsType=0  Reader will determine stabs type (Default)
 StabsType=1  Older style stabs
              (Older style stabs have individual symbol
              tables for each file that was linked into
              the target executable, the indexes of each
              symbol table restart at 0 for each file.)
 StabsType=2  Newer style stabs
              (New style stabs have a single symbol table
              where all symbols are merged into a large
              symbol array).
```

ReadOnlyTicoffPage

ReadOnlyTicoffPage tells the ticoff reader to read only the symbols associated with the specified page (as an example 'ReadOnlyTicoffPage=0' reads only page 0 symbols). A value of -1 tells the ticoff readers to read symbols associated with all pages.

```
ReadOnlyTicoffPage=-1 Read all symbols associated will all
                      ticoff pages (Default)
ReadOnlyTicoffPage=p  Read only symbols associated with
                      page 'p' (where p is any integer
                      between 0 and n the last page of
                      the object file).
```

AppendTicoffPage

AppendTicoffPage tells the ticoff reader to append the page number to the symbol value. This assumes that the symbol value is 16-bits wide and that that page number is a low positive number which can be ORed into the upper 16 bits of an address to create a new 32-bit symbol address. For example, if the page is 10 decimal and the symbol address is 0xF100 then the new symbol address will be 0xAF100.

```
AppendTicoffPage=1  Append the ticoff page to the symbol
                    address
AppendTicoffPage=0  Do not append the ticoff page to the
                    symbol address (Default)
```

Examples

Example for Elf/Stabs

```
[ReadersElf]
C
C
MaxSymbolWidth=60
StabsType=2
```

Example for Coff/Stabs (using Ticoff reader)

```
[ReadersTicoff]
C
C
MaxSymbolWidth=60
StabsType=2
```

Example for Ticoff

```
[ReadersTicoff]
C
C
MaxSymbolWidth=60
ReadOnlyTicoffPage=4
AppendTicoffPage=1
```

Using Symbols In The Logic Analyzer

The ways symbols can be used in the logic analyzer are listed below:

- "Using Symbols As Trigger Terms" on page 73
- "Using Symbols as Search Patterns in Listing Displays" on page 74
- "Using Symbols as Trigger Terms in the Source Viewer" on page 74
- "Using Symbols as Pattern Filter Terms" on page 74
- "Using Symbols as Ranges in the Software Performance Analyzer" on page 75
- "Displaying Data in Symbolic Form" on page 60

Using Symbols As Trigger Terms

You can use either one or both types of symbols as terms within your trigger sequence:

- *Object File Symbols*.
- *User-Defined Symbols*.

1. At the bottom of the analyzer Trigger window, select the label button next to one of the resource terms, and choose *Replace*.
2. In the Resource selection dialog, select a label to be used in your trigger sequence.
   Use a label that has *symbols* loaded.
3. Set the numeric base of the trigger term to *Symbols* or *Line #s*.
4. Select the button to the right of the numeric base field.
5. In the *Symbol Selector* (see page 75) dialog, select the symbol you want to use.

NOTE: The values of object file symbols used as trigger terms are automatically updated when the object file symbols are reloaded (see page 61).

Using Symbols as Search Patterns in Listing Displays

1. Under the *Search* tab in the Listing display, select the *Advanced searching* button.
2. In the Goto Pattern dialog, select the *Define* button.
3. In the Search Pattern dialog, select the *Symbols* numeric base.
4. Select *Pattern*, *Range*, *Not Pattern*, or *Not Range*.
5. Select the button to the right of the numeric base field.
6. In the *Symbol Selector* (see page 75) dialog, select the symbol you want to use.

See Also Go to an Exact Pattern. (see the *Listing Display Tool* help volume)

Using Symbols as Trigger Terms in the Source Viewer

1. In the Source Viewer menu bar, select *Trace*, and select *Trace Setup*.
2. In the Source Line Trigger dialog, select *Symbols* or *Line #s* in the numeric base field.
3. Select *Pattern*, *Range*, *Not Pattern*, or *Not Range*.
4. Select the button to the right of the numeric base field.
5. In the *Symbol Selector* (see page 75) dialog, select the symbol you want to use.

See Also To modify the trace setup. (see the *Listing Display Tool* help volume)

Using Symbols as Pattern Filter Terms

1. Select the numeric base field beside the selected filter term, and select

Symbols or *Line #s*.

2. Select *Pattern*, *Range*, *Not Pattern*, or *Not Range*.
3. Select *Remove Matching Data* or *Pass Matching Data*, as desired.
4. Select the *Absolute XXXX* button.
5. In the *Symbol Selector* (see page 75) dialog, select the symbol you want to use.

Using Symbols as Ranges in the Software Performance Analyzer

1. In the state interval SPA tool, select the *Symbols* button in the Define Ranges dialog.
2. In the Symbol Selector (see page 76) dialog, select the symbol or group of symbols you want to use as ranges in your measurement.

See Also

Defining State Interval Ranges. (see the *System Performance Analyzer* help volume)

Using the Symbol Selector Dialog

1. In the *Symbol Selector* dialog, select the symbol you want to use. All of your symbols for the current label, regardless of type, will be available in the dialog.
   - Use the Search Pattern (see page 76) field to filter the list of symbols by name. You can use the Recall button to recall a desired Search Pattern.
   - Use the Find Symbols of Type selections to filter the symbols by type.
2. Select the symbol you want to use in the list of *Matching Symbols*.
3. If you are using object file symbols, you may need to:
   - Set *Offset By* (see page 77) to compensate for microprocessor prefetches.
   - Set Align to x Byte (see page 77) to trigger on odd-byte boundaries.

4. Select the Beginning, End, or Range of the symbol.
5. Select the *OK* button.
   The name of your symbol now appears as the value of the resource term.
6. Select the *Cancel* button to exit the *Symbol Selector* dialog without selecting a symbol.

Using the Symbol Selector Dialog

1. In the *Symbol Selector* dialog, select the symbol you want to use. All of your symbols, regardless of type, will be available in the dialog.
   - Use the Search Pattern (see page 76) field to filter the list of symbols by name. You can use the Recall button to recall a desired Search Pattern.
   - Use the Find Symbols of Type selections to filter the symbols by type.
2. Drag to select the symbols you want to use in the list of *Matching Symbols*.
   - Select the *Select All* button to select all symbols in the list.
   - Select the *Unselect All* button to unselect all symbols in the list.
3. Select the *Add Selected Symbols To Range List* button to place the selected symbols into the *Current ranges* list in the Define Ranges dialog.
4. Select the *Close* button to exit the *Symbol Selector* dialog.

Search Pattern

Use this field to locate particular symbols in the symbol databases. To use this field, enter the name of a file or symbol. The system searches the symbol database for symbols that match this name. Symbols that match appear in the list of *Matching Symbols*. You can also use wildcard characters to find symbols.

Asterisk wildcard (*)

The asterisk wildcard represents "any characters." When you perform a search on the symbol database using just the asterisk, you will see a list of all symbols contained in the database. The asterisk can also be added to a search word to find all symbols that begin or end with the

same letters. For example, to find all of the symbols that begin with the letters "st", select the Search Pattern field and enter "st*".

Align to x Byte Option

Most processors do not fetch instructions from memory on byte boundaries. In order to trigger a logic analyzer on a symbol at an odd-numbered address, the address must be masked off. The "Align to x Byte" option allows you to mask off an address.

Example

Assume the symbol "main" occurs at address 100F. The processor being probed is a 68040, which fetches instructions on long-word (4-byte) boundaries. In order to trigger on address 100F, the Align to x Byte option sets the two least-significant address bits to "don't cares". This qualifies any address from 100C through 100F.

Offset By Option

The Offset By option allows you to add an offset value to the starting point of the symbol that you want to use as a term. You might do this in order to trigger on a point in a function that is beyond the preamble of the function, or to trigger on a point that is past the prefetch depth of the processor. Setting an offset helps to avoid false triggers in these situations. The offset specified in the Offset By field is applied before the address masking is done by the "Align to x Byte" option.

Example

An 80386 processor has a prefetch depth of 16 bytes. Assume functions *func1* and *func2* are adjacent to each other in physical memory, with *func2* following *func1*. In order to trigger on *func2* without getting a false trigger from a prefetch beyond the end of *func1*, you need to add an offset value to your trigger term. The offset value must be equal to or greater than the prefetch depth of the processor. In this case, you would add an offset of 16 bytes to your trigger term. You would set the value of the "Offset By" field to 10 hex. Now, when you specify *func2* as your trigger term, the logic analyzer will trigger on address *func2+10*.

User-Defined Symbols

"To Create User-Defined Symbols" on page 78

"To Replace User-Defined Symbols" on page 78

"To Delete User-Defined Symbols" on page 79

"To Load User-Defined Symbols" on page 79

To Create User-Defined Symbols

1. Under the *Symbol* tab, select the *User Defined* tab.
2. Select the label name you want to define symbols for.
3. At the bottom of the *User Defined* tab, enter a symbol name in the entry field.
4. Select a numeric base.
5. Select *Pattern* or *Range* type for the symbol.
6. Enter values for the pattern or range the symbol will represent.
7. Select the *Add* button.
8. Repeat steps 3 through 7 for additional symbols.
9. You can edit your list of symbols by using Replace (see page 78) and Delete (see page 79), if desired.

See Also

"Using Symbols In The Logic Analyzer" on page 73

To Replace User-Defined Symbols

1. Under the *Symbol* tab, select the *User Defined* tab.
2. Select the label you want to replace symbols for.

3. Select the symbol to replace.
4. At the bottom of the *User Defined* tab, modify the symbol name, numberic base, Pattern/Range type, and value, as desired.
5. Select the *Replace* button.
6. Repeat steps 3 through 5 to replace other symbols, if desired.

To Delete User-Defined Symbols

1. Under the *Symbol* tab, select the *User Defined* tab.
2. Select the label you want to delete symbols from.
3. Select the symbol to delete.
4. Select the *Delete* button.
5. Repeat steps 3 and 4 to delete other symbols, if desired.

To Load User-Defined Symbols

If you have already saved a configuration file, and the configuration included user-defined symbols, load the file with its symbols, as follows:

1. In the menu bar of your analyzer window, select *File* and then *Load Configuration...*.
2. In the Load Configuration dialog, select the directory and filename to be loaded.
3. Select the target of the load operation.
4. Select the *Load* button.
   User-defined symbols that were resident in the logic analyzer when the configuration was saved are now loaded and ready to use.

See Also

"Using Symbols In The Logic Analyzer" on page 73

absolute Denotes the time period or count of states between a captured state and the trigger state. An absolute count of -10 indicates the state was captured ten states before the trigger state was captured.

acquisition Denotes one complete cycle of data gathering by a measurement module. For example, if you are using an analyzer with 128K memory depth, one complete acquisition will capture and store 128K states in acquisition memory.

analysis probe A probe connected to a microprocessor or standard bus in the device under test. An analysis probe provides an interface between the signals of the microprocessor or standard bus and the inputs of the logic analyzer. Also called a *preprocessor*.

analyzer 1 In a logic analyzer with two *machines*, refers to the machine that is on by default. The default name is *Analyzer<N>*, where N is the slot letter.

analyzer 2 In a logic analyzer with two *machines*, refers to the machine that is off by default. The default name is *Analyzer<N2>*, where N is the slot letter.

arming An instrument tool must be armed before it can search for its trigger condition. Typically, instruments are armed immediately when *Run* or *Group Run* is selected. You can set up one instrument to arm another using the *Intermodule Window*. In these setups, the second instrument cannot search for its trigger condition until it receives the arming signal from the first instrument. In some analyzer instruments, you can set up one analyzer *machine* to arm the other analyzer machine in the *Trigger Window*.

asterisk (*) See *edge terms*, *glitch*, and *labels*.

bits Bits represent the physical logic analyzer channels. A bit is a *channel* that has or can be assigned to a *label*. A bit is also a position in a label.

card This refers to a single instrument intended for use in the Agilent Technologies 16600A-series or 16700A/B-series mainframes. One card fills one slot in the mainframe. A module may comprise a single card or multiple cards cabled together.

channel The entire signal path from the probe tip, through the cable and module, up to the label grouping.

click When using a mouse as the

pointing device, to click an item, position the cursor over the item. Then quickly press and release the *left mouse button*.

clock channel A logic analyzer *channel* that can be used to carry the clock signal. When it is not needed for clock signals, it can be used as a *data channel*, except in the Agilent Technologies 16517A.

context record A context record is a small segment of analyzer memory that stores an event of interest along with the states that immediately preceded it and the states that immediately followed it.

context store If your analyzer can perform context store measurements, you will see a button labeled *Context Store* under the Trigger tab. Typical context store measurements are used to capture writes to a variable or calls to a subroutine, along with the activity preceding and following the events. A context store measurement divides analyzer memory into a series of context records. If you have a 64K analyzer memory and select a 16-state context, the analyzer memory is divided into 4K 16-state context records. If you have a 64K analyzer memory and select a 64-state context, the analyzer memory will be divided into 1K 64-state records.

count The count function records periods of time or numbers of state transactions between states stored in memory. You can set up the analyzer count function to count occurrences of a selected event during the trace, such as counting how many times a variable is read between each of the writes to the variable. The analyzer can also be set up to count elapsed time, such as counting the time spent executing within a particular function during a run of your target program.

cross triggering Using intermodule capabilities to have measurement modules trigger each other. For example, you can have an external instrument arm a logic analyzer, which subsequently triggers an oscilloscope when it finds the trigger state.

data channel A *channel* that carries data. Data channels cannot be used to clock logic analyzers.

data field A data field in the pattern generator is the data value associated with a single label within a particular data vector.

data set A data set is made up of all labels and data stored in memory of any single analyzer machine or

instrument tool. Multiple data sets can be displayed together when sourced into a single display tool. The Filter tool is used to pass on partial data sets to analysis or display tools.

debug mode See *monitor*.

delay The delay function sets the horizontal position of the waveform on the screen for the oscilloscope and timing analyzer. Delay time is measured from the trigger point in seconds or states.

demo mode An emulation control session which is not connected to a real target system. All windows can be viewed, but the data displayed is simulated. To start demo mode, select *Start User Session* from the Emulation Control Interface and enter the demo name in the *Processor Probe LAN Name* field. Select the *Help* button in the *Start User Session* window for details.

deskewing To cancel or nullify the effects of differences between two different internal delay paths for a signal. Deskewing is normally done by routing a single test signal to the inputs of two different modules, then adjusting the Intermodule Skew so that both modules recognize the signal at the same time.

device under test The system under test, which contains the circuitry you are probing. Also known as a *target system*.

don't care For *terms*, a "don't care" means that the state of the signal (high or low) is not relevant to the measurement. The analyzer ignores the state of this signal when determining whether a match occurs on an input label. "Don't care" signals are still sampled and their values can be displayed with the rest of the data. Don't cares are represented by the *X* character in numeric values and the dot (.) in timing edge specifications.

dot (.) See *edge terms*, *glitch*, *labels*, and *don't care*.

double-click When using a mouse as the pointing device, to double-click an item, position the cursor over the item, and then quickly press and release the *left mouse button* twice.

drag and drop Using a Mouse: Position the cursor over the item, and then press and hold the *left mouse button*. While holding the left mouse button down, move the mouse to drag the item to a new location. When the item is positioned where you want it, release the mouse button.

Using the Touchscreen:
Position your finger over the item, then press and hold finger to the screen. While holding the finger down, slide the finger along the screen dragging the item to a new location. When the item is positioned where you want it, release your finger.

edge mode In an oscilloscope, this is the trigger mode that causes a trigger based on a single channel edge, either rising or falling.

edge terms Logic analyzer trigger resources that allow detection of transitions on a signal. An edge term can be set to detect a rising edge, falling edge, or either edge. Some logic analyzers can also detect no edge or a *glitch* on an input signal. Edges are specified by selecting arrows. The dot (.) ignores the bit. The asterisk (*) specifies a glitch on the bit.

emulation module A module within the logic analysis system mainframe that provides an emulation connection to the debug port of a microprocessor. An E5901A emulation module is used with a target interface module (TIM) or an analysis probe. An E5901B emulation module is used with an E5900A emulation probe.

emulation probe The stand-alone equivalent of an *emulation module*. Most of the tasks which can be performed using an emulation module can also be performed using an emulation probe connected to your logic analysis system via a LAN.

emulator An *emulation module* or an *emulation probe*.

Ethernet address See *link-level address*.

events Events are the things you are looking for in your target system. In the logic analyzer interface, they take a single line. Examples of events are *Label1 = XX* and *Timer 1 > 400 ns*.

filter expression The filter expression is the logical *OR* combination of all of the filter terms. States in your data that match the filter expression can be filtered out or passed through the Pattern Filter.

filter term A variable that you define in order to specify which states to filter out or pass through. Filter terms are logically OR'ed together to create the filter expression.

Format The selections under the logic analyzer *Format* tab tell the

logic analyzer what data you want to collect, such as which channels represent buses (labels) and what logic threshold your signals use.

frame The Agilent Technologies 16600A-series or 16700A/B-series logic analysis system mainframe. See also *logic analysis system*.

gateway address An IP address 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 across networks or subnets, this address must be set to the address of the gateway machine.

glitch A glitch occurs when two or more transitions cross the logic threshold between consecutive timing analyzer samples. You can specify glitch detection by choosing the asterisk (*) for *edge terms* under the timing analyzer Trigger tab.

grouped event A grouped event is a list of *events* that you have grouped, and optionally named. It can be reused in other trigger sequence levels. Only available in Agilent Technologies 16715A, 16716A, and 16717A logic analyzers.

held value A value that is held until the next sample. A held value can exist in multiple data sets.

immediate mode In an oscilloscope, the trigger mode that does not require a specific trigger condition such as an edge or a pattern. Use immediate mode when the oscilloscope is armed by another instrument.

interconnect cable Short name for *module/probe interconnect cable*.

intermodule bus The intermodule bus (IMB) is a bus in the frame that allows the measurement modules to communicate with each other. Using the IMB, you can set up one instrument to *arm* another. Data acquired by instruments using the IMB is time-correlated.

intermodule Intermodule is a term used when multiple instrument tools are connected together for the purpose of one instrument arming another. In such a configuration, an arming tree is developed and the group run function is designated to start all instrument tools. Multiple instrument configurations are done in the Intermodule window.

internet address Also called Internet Protocol address or IP address. A 32-bit network address. It

is usually represented as decimal numbers separated by periods; for example, 192.35.12.6. Ask your LAN administrator if you need an internet address.

labels Labels are used to group and identify logic analyzer channels. A label consists of a name and an associated bit or group of bits. Labels are created in the Format tab.

line numbers A line number (Line #s) is a special use of *symbols*. Line numbers represent lines in your source file, typically lines that have no unique symbols defined to represent them.

link-level address Also referred to as the Ethernet address, this is the unique address of the LAN interface. This value is set at the factory and cannot be changed. The link-level address of a particular piece of equipment is often printed on a label above the LAN connector. An example of a link-level address in hexadecimal: 0800090012AB.

local session A local session is when you run the logic analysis system using the local display connected to the product hardware.

logic analysis system The Agilent Technologies 16600A-series or 16700A/B-series mainframes, and all tools designed to work with it. Usually used to mean the specific system and tools you are working with right now.

machine Some logic analyzers allow you to set up two measurements at the same time. Each measurement is handled by a different machine. This is represented in the Workspace window by two icons, differentiated by a *1* and a *2* in the upper right-hand corner of the icon. Logic analyzer resources such as pods and trigger terms cannot be shared by the machines.

markers Markers are the green and yellow lines in the display that are labeled *x*, *o*, *G1*, and *G2*. Use them to measure time intervals or sample intervals. Markers are assigned to patterns in order to find patterns or track sequences of states in the data. The x and o markers are local to the immediate display, while G1 and G2 are global between time correlated displays.

master card In a module, the master card controls the data acquisition or output. The logic analysis system references the module by the slot in which the master card is plugged. For example, a 5-card Agilent Technologies 16555D

would be referred to as *Slot C: machine* because the master card is in slot C of the mainframe. The other cards of the module are called *expansion cards*.

menu bar The menu bar is located at the top of all windows. Use it to select *File* operations, tool or system *Options*, and tool or system level *Help*.

message bar The message bar displays mouse button functions for the window area or field directly beneath the mouse cursor. Use the mouse and message bar together to prompt yourself to functions and shortcuts.

module/probe interconnect cable

The module/probe interconnect cable connects an E5901B emulation module to an E5900B emulation probe. It provides power and a serial connection. A LAN connection is also required to use the emulation probe.

module An instrument that uses a single timebase in its operation. Modules can have from one to five cards functioning as a single instrument. When a module has more than one card, system window will show the instrument icon in the slot of the *master card*.

monitor When using the Emulation Control Interface, running the monitor means the processor is in debug mode (that is, executing the debug exception) instead of executing the user program.

panning The action of moving the waveform along the timebase by varying the delay value in the Delay field. This action allows you to control the portion of acquisition memory that will be displayed on the screen.

pattern mode In an oscilloscope, the trigger mode that allows you to set the oscilloscope to trigger on a specified combination of input signal levels.

pattern terms Logic analyzer resources that represent single states to be found on labeled sets of bits; for example, an address on the address bus or a status on the status lines.

period (.) See *edge terms*, *glitch*, *labels*, and *don't care*.

pod pair A group of two pods containing 16 channels each, used to physically connect data and clock signals from the unit under test to the analyzer. Pods are assigned by pairs in the analyzer interface. The number of pod pairs avalaible is determined

by the channel width of the instrument.

pod See *pod pair*

point To point to an item, move the mouse cursor over the item, or position your finger over the item.

preprocessor See *analysis probe*.

primary branch The primary branch is indicated in the *Trigger sequence step* dialog box as either the *Then find* or *Trigger on* selection. The destination of the primary branch is always the next state in the sequence, except for the Agilent Technologies 16517A. The primary branch has an optional occurrence count field that can be used to count a number of occurrences of the branch condition. See also *secondary branch*.

probe A device to connect the various instruments of the logic analysis system to the target system. There are many types of probes and the one you should use depends on the instrument and your data requirements. As a verb, "to probe" means to attach a probe to the target system.

processor probe See *emulation probe*.

range terms Logic analyzer resources that represent ranges of values to be found on labeled sets of bits. For example, range terms could identify a range of addresses to be found on the address bus or a range of data values to be found on the data bus. In the trigger sequence, range terms are considered to be true when any value within the range occurs.

relative Denotes time period or count of states between the current state and the previous state.

remote display A remote display is a display other than the one connected to the product hardware. Remote displays must be identified to the network through an address location.

remote session A remote session is when you run the logic analyzer using a display that is located away from the product hardware.

right-click When using a mouse for a pointing device, to right-click an item, position the cursor over the item, and then quickly press and release the *right mouse button*.

sample A data sample is a portion of a *data set*, sometimes just one point. When an instrument samples the target system, it is taking a single

measurement as part of its data acquisition cycle.

Sampling Use the selections under the logic analyzer Sampling tab to tell the logic analyzer how you want to make measurements, such as State vs. Timing.

secondary branch The secondary branch is indicated in the *Trigger sequence step* dialog box as the *Else on* selection. The destination of the secondary branch can be specified as any other active sequence state. See also *primary branch*.

session A session begins when you start a *local session* or *remote session* from the session manager, and ends when you select *Exit* from the main window. Exiting a session returns all tools to their initial configurations.

skew Skew is the difference in channel delays between measurement channels. Typically, skew between modules is caused by differences in designs of measurement channels, and differences in characteristics of the electronic components within those channels. You should adjust measurement modules to eliminate as much skew as possible so that it does not affect the accuracy of your measurements.

state measurement In a state measurement, the logic analyzer is clocked by a signal from the system under test. Each time the clock signal becomes valid, the analyzer samples data from the system under test. Since the analyzer is clocked by the system, state measurements are *synchronous* with the test system.

store qualification Store qualification is only available in a *state measurement*, not *timing measurements*. Store qualification allows you to specify the type of information (all samples, no samples, or selected states) to be stored in memory. Use store qualification to prevent memory from being filled with unwanted activity such as no-ops or wait-loops. To set up store qualification, use the *While storing* field in a logic analyzer trigger sequence dialog.

subnet mask A subnet mask blocks out part of an IP address so that the networking software can determine whether the destination host is on a local or remote network. It is usually represented as decimal numbers separated by periods; for example, 255.255.255.0. Ask your LAN administrator if you need a the subnet mask for your network.

symbols Symbols represent patterns and ranges of values found on labeled sets of bits. Two kinds of symbols are available:

- Object file symbols - Symbols from your source code, and symbols generated by your compiler. Object file symbols may represent global variables, functions, labels, and source line numbers.
- User-defined symbols - Symbols you create.

Symbols can be used as *pattern* and *range* terms for:

- Searches in the listing display.
- Triggering in logic analyzers and in the source correlation trigger setup.
- Qualifying data in the filter tool and system performance analysis tool set.

system administrator The system administrator is a person who manages your system, taking care of such tasks as adding peripheral devices, adding new users, and doing system backup. In general, the system administrator is the person you go to with questions about implementing your software.

target system The system under test, which contains the microprocessor you are probing.

terms Terms are variables that can be used in trigger sequences. A term can be a single value on a label or set of labels, any value within a range of values on a label or set of labels, or a glitch or edge transition on bits within a label or set of labels.

TIM A TIM (Target Interface Module) makes connections between the cable from the emulation module or emulation probe and the cable to the debug port on the system under test.

time-correlated Time correlated measurements are measurements involving more than one instrument in which all instruments have a common time or trigger reference.

timer terms Logic analyzer resources that are used to measure the time the trigger sequence remains within one sequence step, or a set of sequence steps. Timers can be used to detect when a condition lasts too long or not long enough. They can be used to measure pulse duration, or duration of a wait loop. A single timer term can be used to delay trigger until a period of time after detection of a significant event.

timing measurement In a timing measurement, the logic analyzer samples data at regular intervals according to a clock signal internal to the timing analyzer. Since the analyzer is clocked by a signal that is not related to the system under test, timing measurements capture traces of electrical activity over time. These measurements are *asynchronous* with the test system.

tool icon Tool icons that appear in the workspace are representations of the hardware and software tools selected from the toolbox. If they are placed directly over a current measurement, the tools automatically connect to that measurement. If they are placed on an open area of the main window, you must connect them to a measurement using the mouse.

toolbox The Toolbox is located on the left side of the main window. It is used to display the available hardware and software tools. As you add new tools to your system, their icons will appear in the Toolbox.

tools A tool is a stand-alone piece of functionality. A tool can be an instrument that acquires data, a display for viewing data, or a post-processing analysis helper. Tools are represented as icons in the main window of the interface.

trace See *acquisition*.

trigger sequence A trigger sequence is a sequence of events that you specify. The logic analyzer compares this sequence with the samples it is collecting to determine when to *trigger*.

trigger specification A trigger specification is a set of conditions that must be true before the instrument triggers.

trigger Trigger is an event that occurs immediately after the instrument recognizes a match between the incoming data and the trigger specification. Once trigger occurs, the instrument completes its *acquisition*, including any store qualification that may be specified.

workspace The workspace is the large area under the message bar and to the right of the toolbox. The workspace is where you place the different instrument, display, and analysis tools. Once in the workspace, the tool icons graphically represent a complete picture of the measurements.

zooming In the oscilloscope or timing analyzer, to expand and contract the waveform along the time base by varying the value in the s/Div

field. This action allows you to select specific portions of a particular waveform in acquisition memory that will be displayed on the screen. You can view any portion of the waveform record in acquisition memory.

Index

A

acquisition modes, setup assistant, 21
Align to x Byte option, 77
Align to x Byte option for symbols, 77
ASCII format, 65
ASCII format symbols, 65, 67, 68, 69
ASCII symbol file, 68

B

beyond prefetch depth, 77
browser dialog, 73
browser, symbol search, 73
browsing, 76
browsing the symbol database, 76

C

choosing an analyzer, setup assistant, 20
code, assigning address offsets, 63
COFF symbol reader options, 71
comments, 69
configuring, via setup assistant, 2
connecting to analyzer, setup assistant, 23
connecting, via setup assistant, 2
creating a file, 65
creating ASCII symbol files, 65

D

data displayed in symbolic form, 60
debugger, configuring emulation, 50, 51, 52
displaying symbols to represent data, 60

E

E5900A, identifying, 57
E5901A, identifying, 57
ELF symbol reader options, 71
ELF/stabs file format, 64
emulation module, identifying, 12, 57
emulation probe, identifying, 57
emulation probe, setup assistant LAN config, 39
emulation, requirements, 50, 51, 52
example, 65, 76, 77

F

file formats for ASCII symbols, 65
file versions, 61
files, 61
files, loading user-defined symbol files, 79
finding the symbol you want, 76
functions, 67

H

help, symbols, 59
hooking up, via setup assistant, 2

I

identifying system components, 12
IEEE-695 file format, 64
in ASCII format, 67, 68, 69
in symbol browser, 76
interconnect cable, identifying, 12

L

lan, emulation probe, 39
licensing source correlation, setup assistant, 31
line numbers, 68
loading, 61
loading files including symbols, 61
loading object file symbols, 61
loading user-defined symbol files, 79

M

main system help page, 3
masking off addresses of symbols, 77
measurement examples, 57
module/probe interconnect cable, identifying, 12

O

object file symbol browser, 73
object file symbol files, 61
object file symbols, 73, 76
odd-numbered addresses, 77
odd-numbered addresses represented by symbols, 77
offset, 77
offset addresses, assigning, 63
Offset By option of the symbol browser, 77
OMF96 file format, 64
OMFx86 file format, 64

P

Pattern field, 76
prefetch, 77

R

readers.ini file, 70
record headers, 65
relocating sections of code, 63
results, 76

S

Scroll Files field, 73
Search Pattern field, 76
searching the symbol database, 76
sections, 67
setup assistant, overview, 10
simple example, 65
simple form, 65
source line numbers, 68

Stabs symbol reader options, 71
start address, 67
symbol demangling, 70
symbol file formats, 64
symbol file versions, 61
symbol selector dialog, 75, 76
symbol type, 73
symbol types, 73
symbols, 76
symbols, displaying to represent data, 60
symbols, how they are used in the logic analyzer, 73
symbols, loading object file symbols, 61
symbols, loading user-defined symbol files, 79
symbols, outside defined sections, 70
symbols, setting up, 61
symbols, types and use, 59
symbols, user-defined details, 78
system setup assistant, 2

T

tab, symbols, 59
TI COFF file format, 64
trigger term, 73, 77
triggering beyond, 77
triggering on a symbol, 73, 76, 77

U

User Symbols, 73
user-defined symbol files, loading, 79
user-defined symbols, details, 78

V

variables, 69
versions, 61
versions of symbol files, 61

W

wildcard characters, 76