User's Guide

Publication number E8137-97000
October 2000

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

Logic Analysis Support for the Texas Instruments TMS320C6000 DSP Family

Logic Analysis Support for the Texas Instruments TMS320C6000 DSP Family—At a Glance

Inverse Assembler

The Agilent Technologies E9617A inverse assembler, in conjunction with an Agilent Technologies 16700-series logic analyzer, allows you to view C6000 assembly instructions that are executing in your target system.

The inverse assembler is identified as "Agilent Technologies E8137A" in the Setup Assistant.

Source Correlation Tool Set

The Agilent Technologies B4620B Source Correlation Tool Set lets you set up logic analyzer triggers based on source code (even in SDRAM), and it lets you view the source code associated with signal values captured by the logic analyzer.

In This Book

This book documents the following products:

Product Ordered	Supports	Includes
E9617A Option 001 inverse assembler	Texas Instruments TMS320C6202/6203/ 6204/6205/6211 DSP	The E8137A inverse assembler and B4620B Source Correlation Tool Set

Tips To Save You Time

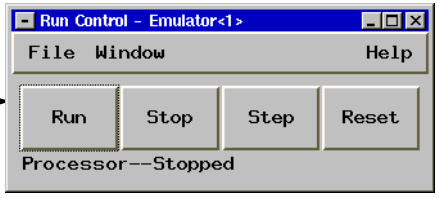
Use the Setup Assistant

Click here to connect the logic analyzer cables, and automatically load the correct configuration files. See page 12.

Use the appropriate Run button

Click here to start a measurement.

If your system includes an emulation probe/module, click here to run the target microprocessor.



Additional Information Sources

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

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

http://www.agilent.com/find/logicanalyzer

The **measurement examples** include valuable tips for making emulation and analysis measurements. You can find the measurement examples under the system help in your Agilent Technologies 16700-series logic analysis system.

Contents

At a Glance 2
In This Book 2
Tips To Save You Time 3

1 Equipment and Requirements 11

Setup Assistant 12

Equipment and Software Supplied 14
Additional equipment required 15
Additional equipment supported 15

Compatible Logic Analyzers 16

2 Preparing the Target System 17

Preparing for Logic Analysis (and Inverse Assembly) 18
Design Considerations 18

Designing Logic Analyzer Connectors into Your Target System 19
Using High-Density Connectors 19
Recommended Connector Layout and Signal Routing 21

3 Setting Up the Logic Analysis System 25

Power-ON/Power-OFF Sequence 26
To power-ON the Agilent Technologies 16700-series logic analysis systems 26

To power-OFF 26

Installing Logic Analyzer Modules 27

Installing Software 28
To install the software from CD-ROM 29

4 Probing the Target System 31

Connecting the Logic Analyzer to the Target System 32

5 Configuring the Logic Analyzer 33

Configuring 16700-series Logic Analysis Systems 35
To load configuration files (and the inverse assembler) from hard disk 36
To load configuration files (and the inverse assembler) from floppy disk 37
To list software packages that are installed 38
Logic analyzer configuration files 38

Using the Inverse Assembler 39
To use the Invasm menu 39
Loading the Inverse Assembler 39

Setting Inverse Assembler Preferences 40
To set the inverse assembler preferences 40
To set the Memory Map preferences 41
To set the C6000 DSP Options preferences 44
To set the Decoding Options preferences 46

Loading Symbol Information 47

To view user-defined symbols 47
To load object file symbols 48
To compensate for relocated code 51

Setting Up Labels for Groups of Signals 52
Predefined Label Descriptions 52
To define additional labels 53

Changing the Analysis Mode 54
To change to state analysis 54
To change to timing analysis 55

6 Capturing DSP Execution 57

Setting Up Logic Analyzer Triggers 59
To set up logic analyzer triggers 59

Triggering on Source Code 61
To avoid capturing library code execution 61

7 Displaying Captured DSP Execution 63

To display the captured state data 64
To display symbols 65
To interpret the inverse assembled data 66
To use the inverse assembler filters 67
To view the source code associated with captured data 69
To display captured timing analysis mode data 72

8 General-Purpose ASCII (GPA) Symbol File Format 73

General-Purpose ASCII (GPA) Symbol File Format 74
GPA Record Format Summary 76
SECTIONS 78
FUNCTIONS 79
VARIABLES 80
SOURCE LINES 81
START ADDRESS 82
Comments 82

9 Specifications and Characteristics 83

Operating Characteristics 84

10 Troubleshooting the Logic Analyzer 85

Solving Logic Analyzer Problems 87
Intermittent data errors 87
Unwanted triggers 87
No activity on activity indicators 88
No trace list display 88

Solving Probing Problems 89
Target system will not boot up 89
Erratic trace measurements 90
Capacitive loading 90

Solving Inverse Assembler Problems 91

Contents

No inverse assembly or incorrect inverse assembly 91
Inverse assembler will not load or run 92

Solving Intermodule Measurement Problems 93
An event wasn't captured by one of the modules 93

Logic Analyzer Messages 94
". . . Inverse Assembler Not Found" 94
"No Configuration File Loaded" 94
"Selected File is Incompatible" 95
"Slow or Missing Clock" 95
"Time from Arm Greater Than 41.93 ms" 95
"Waiting for Trigger" 96

Glossary 97

Index 103

1

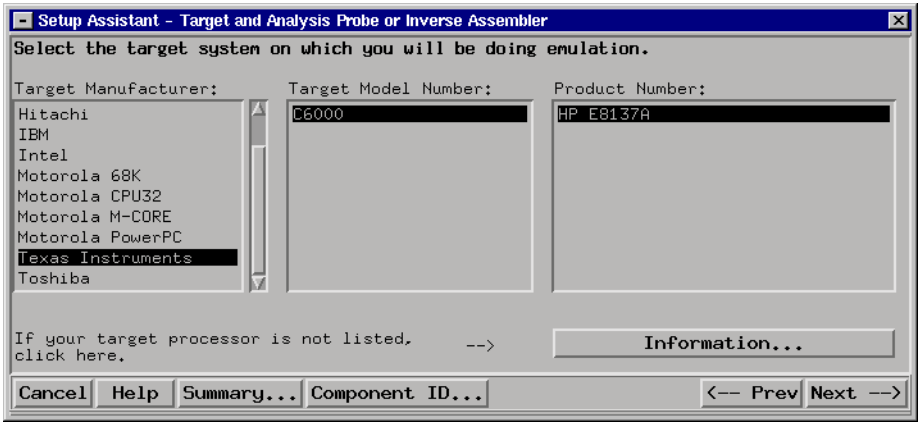
Equipment and Requirements

Setup Assistant

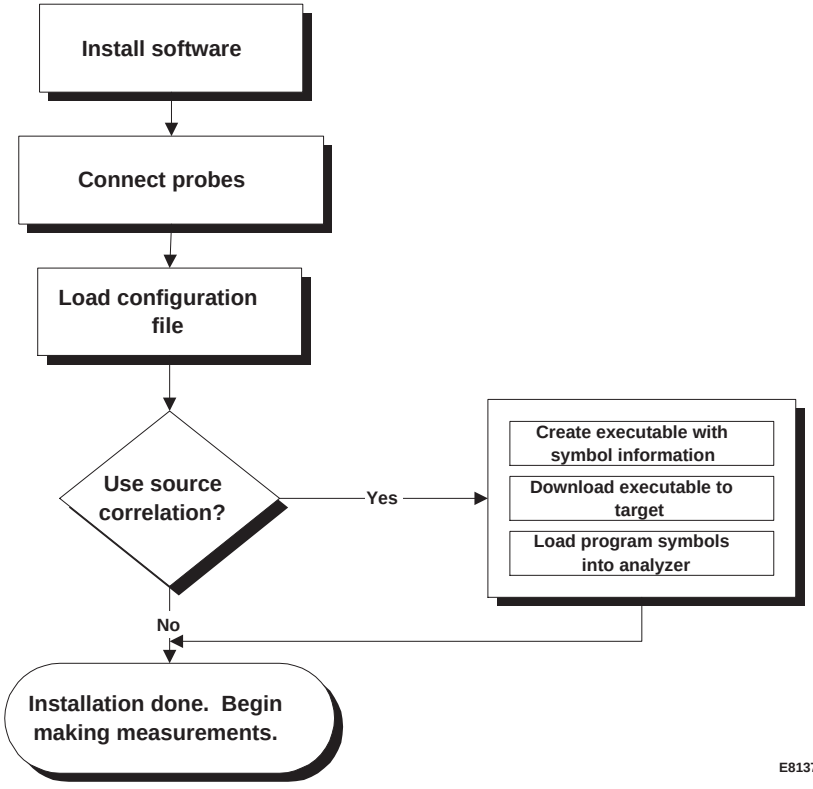
The Setup Assistant is an on-line tool for connecting and configuring your logic analysis system for microprocessor and bus analysis. The Setup Assistant is available on the 16700-series logic analysis systems. You can use the Setup Assistant in place of the connection and configuration procedures provided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

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

The inverse assembler is identified as “Agilent Technologies E8137A” in the Setup Assistant. Start the Setup Assistant by clicking  in the system window.



If you ordered this product with your Agilent Technologies 16700-series logic analysis system, the logic analysis system has the latest software installed, including support for this product.


Install software
Connect probes
Load configuration file
Use source correlation?
Yes
No
Create executable with symbol information
Download executable to target
Load program symbols into analyzer
Installation done. Begin making measurements.
E8137F01.VSD

Equipment and Software Supplied

The inverse assembler (Agilent Technologies E9617A Option 001) includes:

- Logic analyzer configuration files and the inverse assembler software on a CD-ROM (for Agilent Technologies 16700-series logic analysis systems).
- This *User's Guide*.

The inverse assembler is identified as "E8137A" in the Setup Assistant.

Additional equipment required

In addition to the items listed above, the following is required:

- Connector headers on your target system which supply the necessary signals to the logic analyzer. See Chapter 2, “Preparing the Target System,” beginning on page 17 for information on designing the appropriate connectors into the target system.
- Agilent Technologies termination adapter cables to attach your target system to a logic analyzer.

Additional equipment supported

Agilent Technologies B4620B Source Correlation Tool Set. The inverse assembler may be used with the Agilent Technologies B4620B Source Correlation Tool Set. The software is already installed on the Agilent Technologies 16600/16700-series logic analysis system’s disk. All you need is the entitlement certificate for licensing the source correlation tool set software. The CD-ROM is included in case you need to re-install the software.

Compatible Logic Analyzers

A minimum of four logic analyzer pods (68 logic analysis channels) are required for inverse assembly. For inverse assembly of both instructions and data, eight pods are required. If optional signals are used, such as PCI analysis, additional logic analyzer pods are required.

The inverse assembler only works in 16700-series logic analysis systems. The 16500 logic analysis system and 166x/167x portable logic analyzer families are not supported.

The inverse assembler works with logic analysis system software version A.02.10 or greater. The latest logic analysis system software version is on the CD-ROM shipped with this product.

Given these restrictions, the following logic analyzers can be used:

Agilent Technologies Logic Analyzer
16550A
16554A
16555A
16555D
16556A
16556D
16557D
16710A
16711A
16712A
16715A
16716A
16717A
16718A
16719A
16750A
16751A
16752A

2

Preparing the Target System

Preparing for Logic Analysis (and Inverse Assembly)

The inverse assembler requires a minimum of four logic analyzer pods.

Design Considerations

Cache memory

The processor supplies no external information when the cache is enabled. This poses a problem for logic analysis, since there are no linear address and data cycles to reconstruct code flow. Therefore, when the cache is on, the inverse assembler can only track transactions on the visible bus, but cannot provide mnemonic disassembly.

The processor provides no external information for accesses to the internal cache. The inverse assembler does not support accesses to the internal cache.

Electrical requirements

Any probed signal line must be able to supply a minimum of 600 mV to the probe tip and handle a minimum loading of 90 kΩ shunted by 10 pF. The maximum input voltage for the logic analyzer is +/- 40 volts peak.

For all signals, the logic analyzers require a minimum combined setup/hold window:

- For 16715/16/17/18/19/50/51/52A logic analyzers the combined window must be at least 2.5 ns (1.25 ns using eye finder).
- For 16710/11/12A logic analyzers, the combined window must be at least 4.0 ns.
- For 16557 logic analyzers, the combined setup/hold time must be at least 3.0 ns.
- For 16550/54/55/56 logic analyzers, the combined setup/hold time must be at least 3.5 ns.

Designing Logic Analyzer Connectors into Your Target System

The logic analyzer can be connected directly to connectors on your target system. This section describes what kind of connectors to use, and how to connect the correct signals to the connectors.

Using High-Density Connectors

High-density MICTOR (*Matched Impedance ConnecTOR*) connectors are recommended for connecting the target system to the logic analyzer because they require less board space and provide higher signal integrity than medium-density connectors. Each connector carries 32 signals and two clocks.

- Each 32-signal high-density header connector requires approximately 1.1" x 0.4" of printed-circuit board space.
- The part number for the high-density MICTOR connector is: AMP P/N 2-767004-2 or Agilent Technologies 1252-7431.
- Each MICTOR connector requires one Agilent Technologies E5346A high-density termination adapter cable to attach to the logic analyzer. This is a Y-cable where the single end connects to the high-density header connector, and each of the two opposite ends connects to a logic analyzer pod.
- Any probed signal line must be able to supply a minimum of 600 mV to the probe tip and handle a minimum loading of 90 kOhms shunted by 10 pF. The maximum input voltage for the logic analyzer is +/- 40 volts peak.
- If a printed-circuit board already has a header connector attached, but the signal pinouts do not match the requirement, an adapter (Agilent part number E5346-60002) can be used to route the signals to the correct pods.
- A plastic shroud (Agilent part number E5346-44701) is available to secure the mechanical connection of the high-density cable to the MICTOR header connector.

See Also

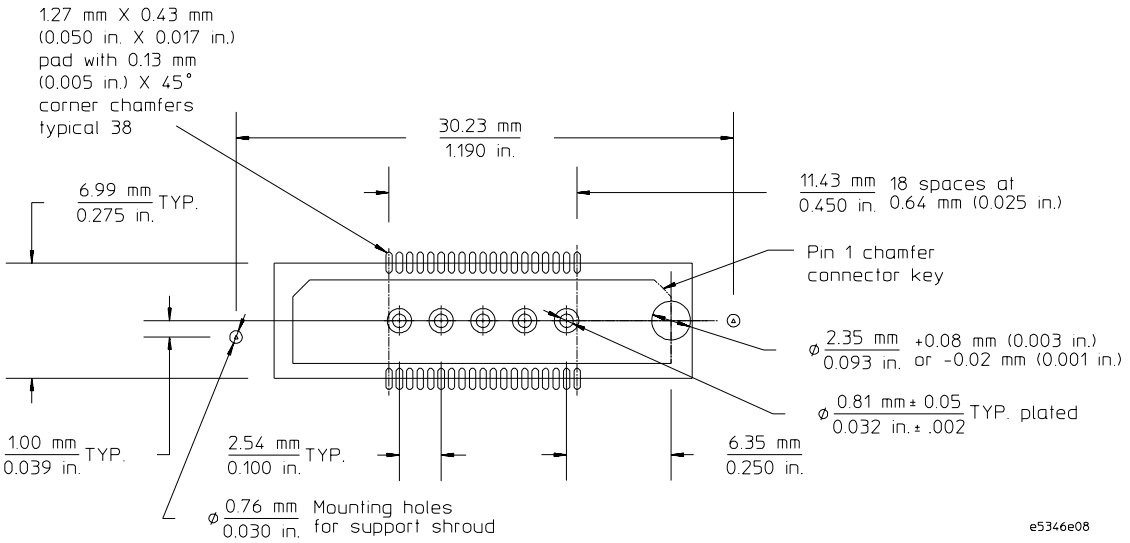
More information on this connector is available in Portable Document Format (PDF) from the web site:

http://www.tm.agilent.com/

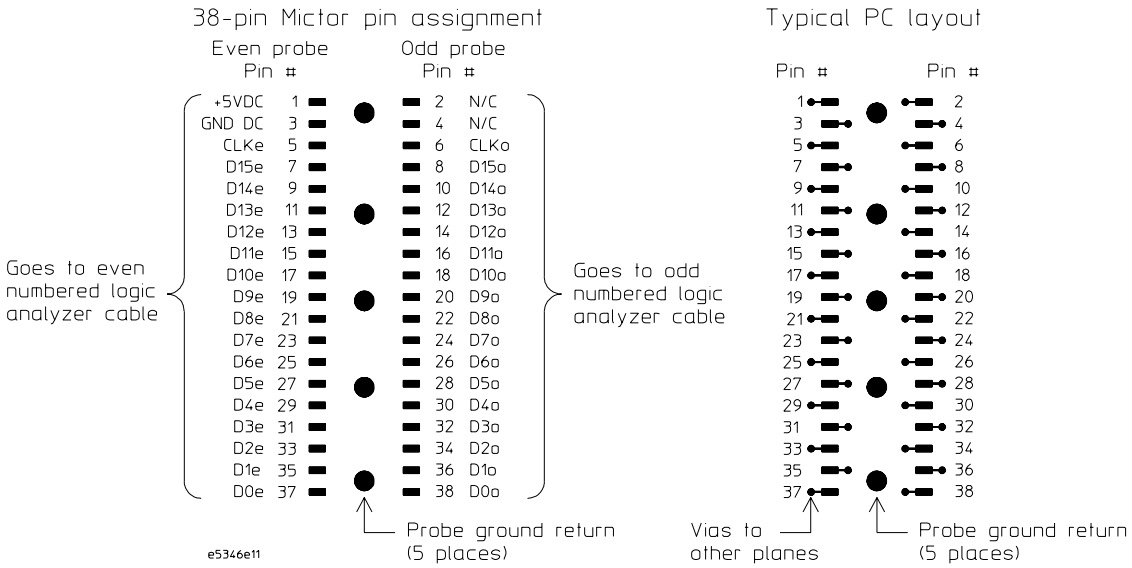
When you reach this web site, search on E5346A.

High-Density Connector Mechanical Specifications

Dimensions of the AMP MICTOR 2-767004-2 surface mount connector are shown below. The holes for mounting a support shroud are off-center to allow 0.40 in (1.20 mm) centers when using multiple connectors.



The high-density connector pin assignment and recommended circuit board routing are shown in the following figure.



Five center inline pins on the connector are the signal ground returns and must be connected to ground.

Recommended Connector Layout and Signal Routing

The following shows the recommended connector layout and signal routing.

Recommended Configuration Connection Notes

- 'nc' pins MUST be a true no-connect on the target. The signals are used for other functions unavailable to target probing.
- Five center inline pins on the connector are the signal ground returns and must be connected to ground.
- Any blank pins can be used for user defined signals.
- '#' or overscore denotes an active low signal.
- J1-J2: Connectors required for inverse assembly.

Recommended Connector Layout

Minimize trace lengths from the DSP to the connectors. Keep trace lengths less than 1/5 of the rise time of the signal.

Recommended Signal Routing

MICTOR Connector J1			
J1 Pin	**Signal**	**J1 Pin**	**Signal**
1	No Connect	2	No Connect
3	No Connect	4	No Connect
5	ECLKOUT/CLKOUT2 (*)	6	$\overline{\text{ARE}}$/$\overline{\text{SDCAS}}$/$\overline{\text{SSADS}}$
7	CE3	8	A15
9	CE2	10	A14
11	CE1	12	A13
13	CE0	14	A12
15	BE3	16	A11
17	BE2	18	A10
19	BE1	20	A9
21	BE0	22	A8
23	ARDY	24	A7
25	$\overline{\text{AWE}}$/$\overline{\text{SDWE}}$/$\overline{\text{SSWE}}$	26	A6
27	A21 (MSB)	28	A5
29	A20	30	A4
31	A19	32	A3
33	A18	34	A2
35	A17	36	
37	A16	38	
Even Cable		**Odd Cable**	
Logic Analyzer Pod 2		**Logic Analyzer Pod 1**	

MICTOR Connector J2			
J2 Pin	**Signal**	**J2 Pin**	**Signal**
1	No Connect	2	No Connect
3	No Connect	4	No Connect
5	$\overline{\text{AOE}}$/$\overline{\text{SDRAS}}$/$\overline{\text{SSOE}}$	6	CLKOUT1(*)
7	D31 (MSB)	8	D15
9	D30	10	D14
11	D29	12	D13
13	D28	14	D12
15	D27	16	D11
17	D26	18	D10
19	D25	20	D9
21	D24	22	D8
23	D23	24	D7
25	D22	26	D6
27	D21	28	D5
29	D20	30	D4
31	D19	32	D3
33	D18	34	D2
35	D17	36	D1
37	D16	38	D0 (LSB)
Even Cable		**Odd Cable**	
Logic Analyzer Pod 4		**Logic Analyzer Pod 3**	

* Use CLKOUT1/2 for C6202/3/4/5

3

Setting Up the Logic Analysis System

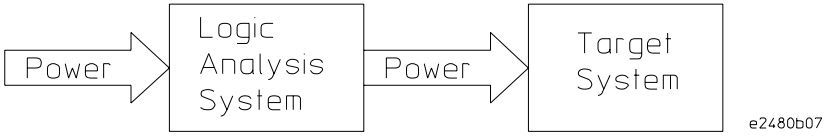
Power-ON/Power-OFF Sequence

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

To power-ON the Agilent Technologies 16700-series logic analysis systems

Ensure the target system is powered off.

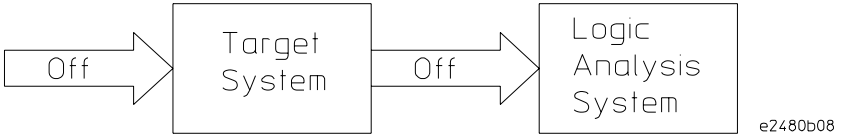
1. Turn on the logic analyzer. The Setup Assistant will guide you through the process of connecting and configuring the logic analyzer.
2. When the logic analyzer is connected to the target system, and everything is configured, turn on your target system.



To power-OFF

Turn off power to your system in the following order:

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



Installing Logic Analyzer Modules

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

CAUTION: Electrostatic discharge can damage electronic components. Use appropriate ESD equipment (grounded wrist straps, etc.) and ESD-safe procedures when you handle and install modules.

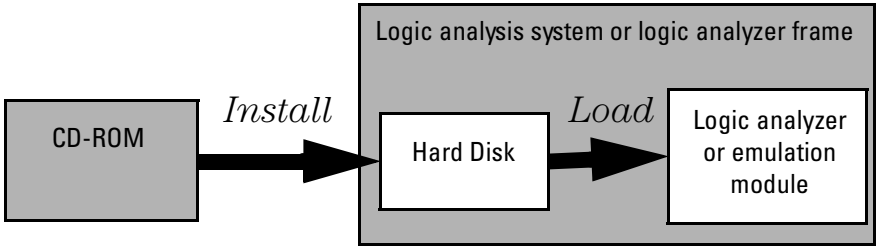
Refer to the Agilent Technologies 16700-series logic analysis system's *Installation Guide* for instructions on installing modules.

Installing Software

This section explains how to install the software you will need for your inverse assembler or emulation solution.

Installing and loading

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



What needs to be installed

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

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

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

The Agilent Technologies B4620B Source Correlation Tool Set is installed with the logic analysis system's operating system.

To install the software from CD-ROM

Installing a processor support package from a CD-ROM will take just a few minutes. If the processor support package requires an update to the Agilent Technologies 16700 operating system, installation may take approximately 15 minutes.

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

1. Turn on the CD-ROM drive first and then turn on the logic analysis system.

   If the CD-ROM and analysis system are already turned on, be sure to save any acquired data. The installation process may reboot the logic analysis system.

2. Insert the CD-ROM in the drive.

3. Select the System Administration icon. 

4. Select the Software Install tab.

5. Select Install....

   Change the media type to "CD-ROM" if necessary.

6. Select Apply.

7. From the list of types of packages, double-click "PROC-SUPPORT."

NOTE: For touch screen systems, double select the "PROC-SUPPORT" line by quickly touching it twice.

   A list of the processor support packages on the CD-ROM will be displayed.

8. Select the "TIC6000" package.

   If you are unsure whether this is the correct package, click Details for information about the contents of the package.

9. Select Install.

   The Continue dialog box will appear.

10. Select Continue.

The Software Install dialog will display “Progress: completed successfully” when the installation is complete.

11 If required, the system will automatically reboot. Otherwise, close the software installation windows.

The configuration files are stored in /logic/configs/hp/tic6000. The inverse assemblers are stored in /logic/ia.

See Also

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

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

4

Probing the Target System

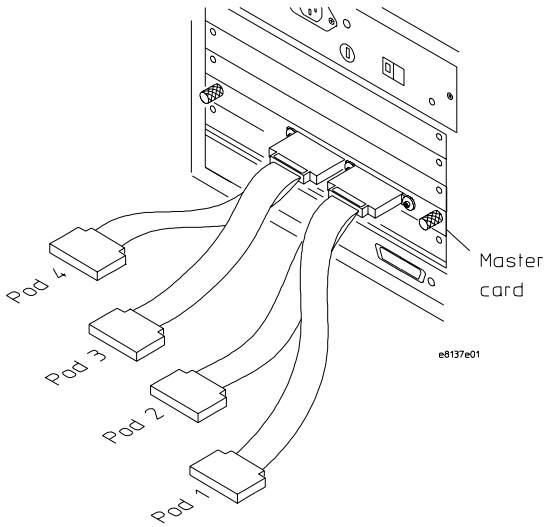
Connecting the Logic Analyzer to the Target System

You can also use the Setup Assistant to guide you through the connection process. See page 12.

CAUTION: Be sure to power down the target system before connecting or disconnecting cables. Otherwise, you may damage circuitry in the analyzer or target system.

Find the labels that were shipped with the high-density cables and use them to help identify the connections. Attach the labels to the cables after connecting the cables to the logic analyzer.

If you have several logic analyzer cards connected together as a single "machine," be sure to use the pods from the master card.



Analyzer Card	Pod 4	Pod 3	Pod 2	Pod 1
Target Connector	J2, Even	J2, Odd	J1, Even	J1, Odd
Signal	ADDR/ STAT	ADDR	DATA	DATA

5

Configuring the Logic Analyzer

The sections of this chapter describe setting up and using the inverse assembler. Because your target system is designed uniquely according to your needs, it is important that you specify the available signals and memory regions to the inverse assembler.

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

- Loading the configuration file and the inverse assembler
- Tables showing configuration file names
- Using the inverse assembler
- Setting the inverse assembler preferences
- Symbols
- Changing analysis mode

Configuring 16700-series Logic Analysis Systems

You configure the logic analyzer by loading a configuration file. Normally this is done using the Setup Assistant (see page 12). If you did not use the Setup Assistant, you can load the configuration and inverse assembler files from the logic analysis system hard disk.

The configuration file to load is determined by what model of logic analyzer card you are using.

The information in the configuration file includes:

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

If the cache is turned off, the inverse assembler decodes captured data into software addresses (SW_ADDR label) and assembly language mnemonics.

To load configuration files (and the inverse assembler) from hard disk

If you use Setup Assistant, it will load configuration files and the inverse assembler for you. This is the preferred method. If you did not use Setup Assistant, you can load the configuration and inverse assembler files from the logic analysis system hard disk.

1. Select the File Manager icon. Use File Manager to ensure that the subdirectory /logic/configs/hp/tic6000/ exists.

   If the above directory does not exist, you need to install the TIC6000 Processor Support Package. Close File Manager, then use the procedure on the CD-ROM jacket to install the TIC6000 Processor Support Package before you continue. See “To install the software from CD-ROM” on page 29 for details.

2. Using File Manager, select the configuration file you want to load in the /logic/configs/hp/tic6000/ directory, then select `Load`. If you have more than one logic analyzer installed in your logic analysis system, use the Target field to select the machine you want to load.

   The logic analyzer is configured for TIC6000 analysis by loading the appropriate configuration file. Loading the indicated file also automatically loads the inverse assembler. The configuration file names are shown in the table on page 38.

3. Close File Manager.

To load configuration files (and the inverse assembler) from floppy disk

If you use Setup Assistant, it will load configuration files and the inverse assembler for you. This is the preferred method. If you did not use Setup Assistant, you can load the configuration and inverse assembler files from the logic analysis system hard disk or floppy disk; however, the preferred method is to install this functionality from the CD-ROM onto the hard disk and load from the hard disk.

To install a configuration and inverse assembler file from a floppy disk:

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

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

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

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

This procedure restores the configuration that was in effect when the configuration file was saved. Because the file was not saved using your system, you may receive error messages about loading the enhanced inverse assembler or about pods that were truncated. Select the Config, Format, and Trigger tabs and modify the configuration to satisfy your measurement desires. Then you can save your customized configuration to DOS format using the File→Save Configuration selection in any of your tool windows, or selecting the Save tab in the File Manager. For details about how to save configuration files, open the Help window.

To list software packages that are installed

- In the System Administration Tools window, select List....

Logic analyzer configuration files

The following table shows the configuration files for the supported logic analyzers.

Analyzer Model	No-data Configuration File
16550A	C6000F_1
16554/5/6/7	C6000M_1
16710/11/12A	C6000F_1
16715/16/17/18/19A	C6000L_1
16750/51/52A	C6000L_1

Using the Inverse Assembler

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

Traditional inverse assembly, in which the external processor bus states are captured and decoded, may be implemented by disabling the target's cache. However, this will slow the target significantly, and may induce timing related problems. The target system's performance will be much better if the cache-on trace reconstruction feature is enabled when using the inverse assembler.

To use the Invasm menu

The Invasm menu provides four choices: Load, Preferences, Filter, and Options. Access the Invasm menu in the listing window.

You must use the Preferences dialog to configure the inverse assembler to match the memory controller configuration. The other dialogs assist in analyzing and displaying data. The following sections describe these dialogs.

Loading the Inverse Assembler

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

Setting Inverse Assembler Preferences

The Invasm Preferences dialog lets you give the inverse assembler information about your target system so that it can properly disassemble signal values captured by the logic analyzer.

To set the inverse assembler preferences

To open the Preferences dialog:

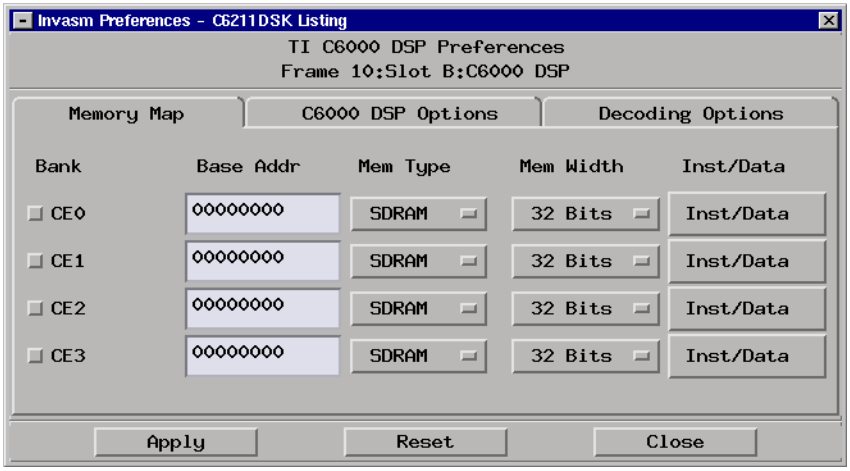
1. Load a configuration file, if needed.
2. Open the Listing window.
3. Select Preferences... from the Invasm menu at the top of the Listing window.
4. Look at each of the tabs in the Preferences dialog and adjust the settings to match your target system. At a minimum, you must set the memory map preferences.

To set the Memory Map preferences

The DSP does not provide all of the signals required to reconstruct physical addresses. It is therefore necessary to reconstruct the address from various pieces of information:

- Memory map configuration (set in the Preferences dialog)
- The RAS and CAS signals
- The address signals

It is necessary to configure the memory map in the Preferences dialog before using the inverse assembler. None of the memory banks are enabled by default.



Bank Enable/Disable. Enable the banks for all of the chip selects that are being used in the system; otherwise, the inverse assembler may not function correctly. These enables are required because chip selects are active low and unconnected logic analyzer channels float low; without the enables, the inverse assembler cannot tell the difference between active and unconnected chip selects.

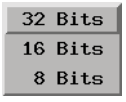
Base Address. Specifies the starting address of the memory bank.

This information is used for upper address bit reconstruction, and is required for proper source correlation and source line triggering.

Memory Type.

Depending upon the type of memory that is accessed, there are different signals that assert when a valid address is on the bus. For this reason, the inverse assembler must know what type of memory is being accessed.

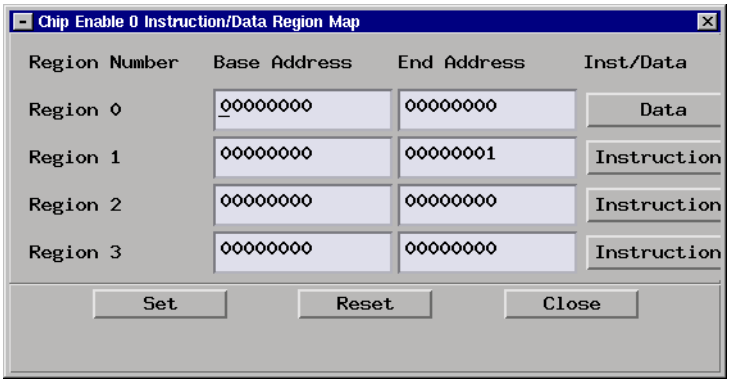
Memory Width.



If your target system uses 8-bit or 16-bit memory, change this from the default of 32 bits.

Inst/Data.

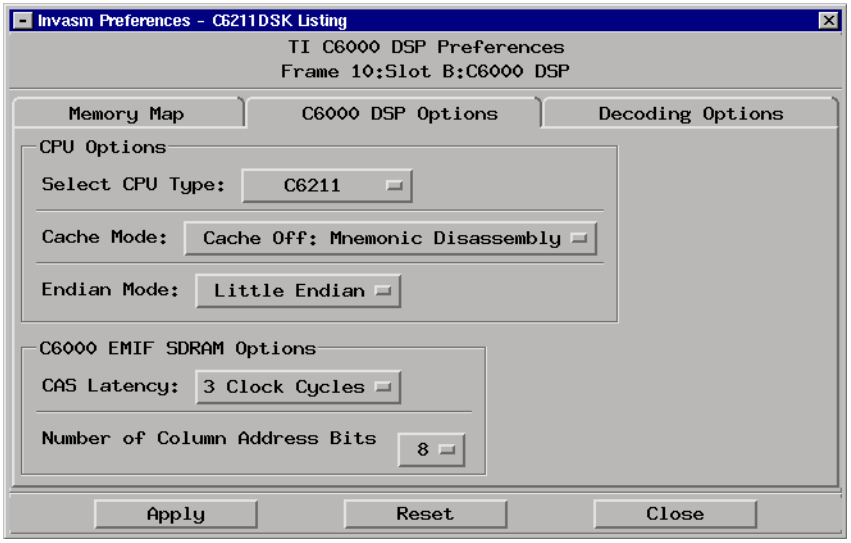
All regions of memory are assumed to contain instructions by default. Set up the Region Map dialog for each memory bank which contains data.



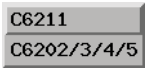
Click Apply when you have finished configuring the memory map.

To set the C6000 DSP Options preferences

The Processor Options tab of the Preferences dialog lets you tell the inverse assembler which mode the DSP is operating in.

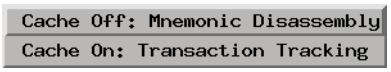


CPU Type



The C6202/3/4/5 can support both synchronous and asynchronous memory by using two clocks, but a logic analyzer card can't use both clocks at once. If your target system uses a mix of synchronous and asynchronous memory, set up a second logic analyzer "machine" to probe the other memory system.

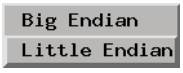
Cache Mode



If the cache of the DSP is turned off, you can view disassembled opcodes in the Listing window.

If the cache is turned on, you can only view bus activity (mostly memory reads). See the documentation for your DSP for information on disabling cache.

Endian Mode Selection



The inverse assembler is designed to support both the big endian mode (most-significant byte has the smallest address) and the little endian mode (least-significant byte has the smallest address) of operation. When operating in little-endian mode, the processor uses a technique known as “address munging” to convert internal little endian addresses into external big endian addresses.

CAS Latency

If your target system uses SDRAM, the inverse assembler needs to know the CAS latency of the memory (the time delay that passes before the SDRAM starts to carry out a read command after receiving it).

Number of Column Address Bits

If your target system uses SDRAM, set the number of column bits strobed by $\overline{\text{CAS}}$ (depends on the size of the SDRAM).

To set the Decoding Options preferences

Display Alias Registers. Displays certain registers in terms of their common names. For example, `b15` will be displayed as `sp`.

Display Simplified Instructions Decoding. Displays certain opcodes in a simplified form. For example, `add 0x00,a10,b4` will be displayed as `mv 10,b4`.

Display Decimal Offsets/Constants. Displays decimal instead of hexadecimal values in the Inverse Assembler column of the listing.

Loading Symbol Information

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

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

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

To view user-defined symbols

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

User-defined symbols are saved with logic analyzer configurations. The logic analyzer configuration files included with this inverse assembler do not contain predefined symbols for logic analyzer labels.

To display any defined symbols:

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

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

To load object file symbols

The most common way to load program symbols into the logic analyzer is from an object file that is created when the program is compiled. The object file containing symbolic debug information must be in a format the logic analyzer understands. To use the Source Correlation Toolset with this inverse assembler, the object file must be in COFF Version 2.0 format.

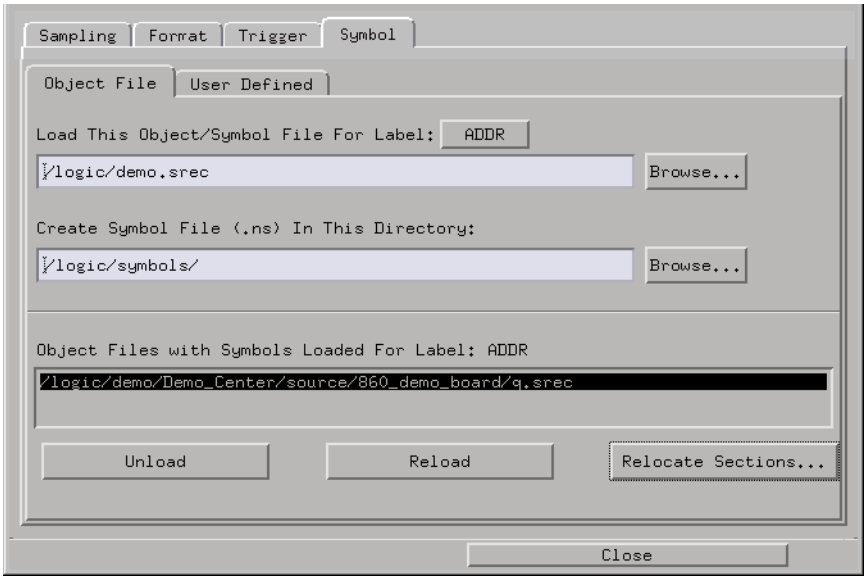
If your compiler generates object files in a format that the logic analyzer doesn't understand, you can use a General-Purpose ASCII (GPA) symbol file (see Chapter 8, "General-Purpose ASCII (GPA) Symbol File Format," on page 73).

To load symbols in the logic analysis system:

1. Open the logic analyzer module's Setup window.
2. Click the Symbol tab.
3. Click the Object File tab.

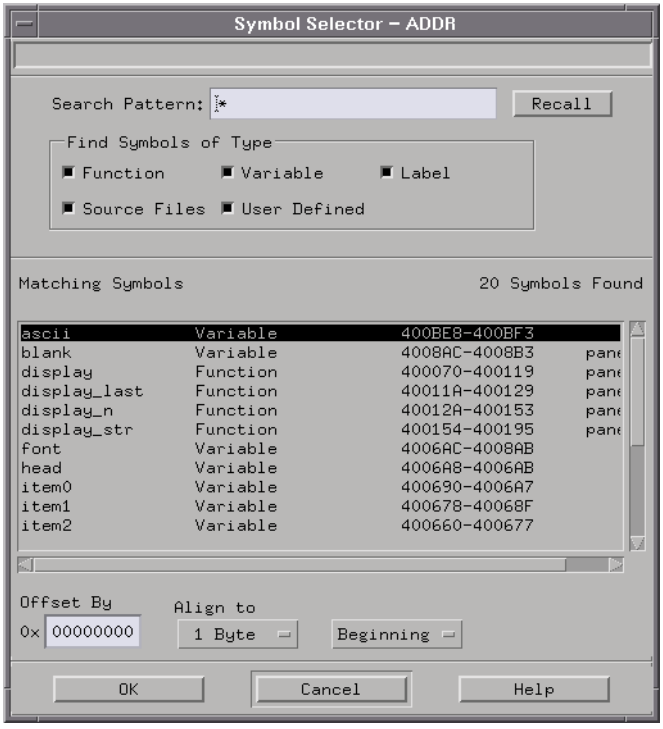
   Make sure the label is ADDR.

From this dialog you can select object files and load their symbol information.



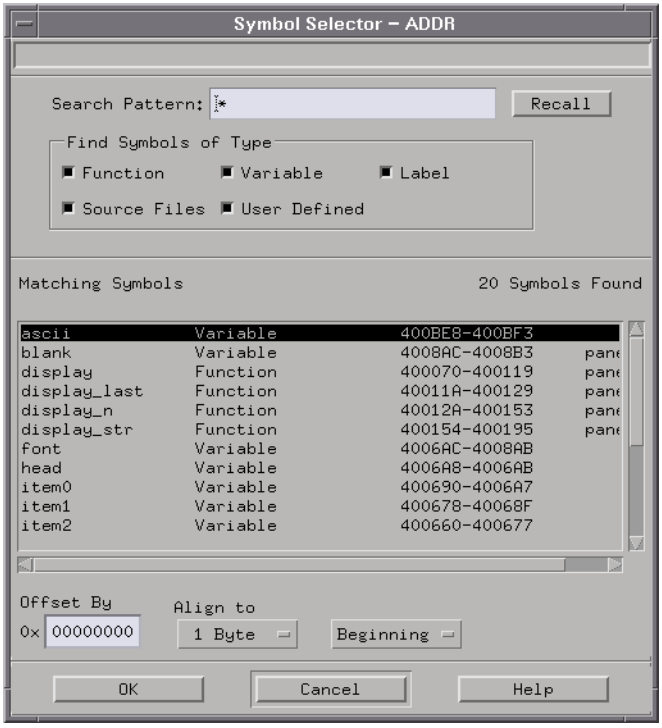
When you load object file symbols into a logic analyzer, a database of symbol/line number to address assignments is generated from the object file.

The Symbol Selector dialog allows you to use a symbol in place of a hexadecimal value when defining trigger patterns, trigger ranges, and so on. To select this dialog, select Trigger tab, then Pattern. Set the numeric base field to Symbols, then select the field to the right of the numeric base field.



To compensate for relocated code

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



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

Setting Up Labels for Groups of Signals

Predefined Label Descriptions

The logic analyzer configuration files automatically set up labels for most C6000 signals. The following tables show some of the predefined labels for the most commonly used signals.

ADDR Label

ADDR	Signal Name
Bit 0	ADDR[0]
..	..
Bit 21	ADDR[21]

DATA Label

DATA	Signal Name
Bit 0	D[31]
...	...
Bit 31	D[0]

EXTRA Label

ADDR	Signal Name
Bit 0	ADDR[0]
Bit 1	ADDR[1]

STAT Label

STAT	Signal Name
Bit 0	$\overline{\text{SDWE}}/\overline{\text{AWE}}$
Bit 1	ARDY
Bit 2	$\overline{\text{BE[0]}}$
Bit 3	$\overline{\text{BE[1]}}$
Bit 4	$\overline{\text{BE[2]}}$
Bit 5	$\overline{\text{BE[3]}}$
Bit 6	$\overline{\text{CE[0]}}$
Bit 7	$\overline{\text{CE[1]}}$

STAT	Signal Name
Bit 8	$\overline{\text{CE[2]}}$
Bit 9	$\overline{\text{CE[3]}}$
Bit 10	CLKOUT1
Bit 11	$\overline{\text{SDRAS}}/\overline{\text{AOE}}$
Bit 12	$\overline{\text{SDCAS}}/\overline{\text{ARE}}$
Bit 13	ECLKOUT

Clock & Qualifier

CLK	Signal Name
J	CLKOUT1
K	$\overline{\text{SDRAS}}/\overline{\text{AOE}}$
L	$\overline{\text{SDCAS}}/\overline{\text{ARE}}$
M	ECLKOUT

To define additional labels

1. Open the Setup window.
2. Click the Format tab.
3. Click a label and select Insert before... or Insert after....
4. Click the signals under the appropriate pod, then select which bits to include in the label.

Changing the Analysis Mode

The logic analyzer can be set up to operate in the following analysis modes:

- State.
- Timing.

Inverse assembly is available in the state analysis mode.

To change to state analysis

In state mode, the logic analyzer uses the CLKIN signal from the target system to capture data synchronously. This mode allows inverse assembly and is the default mode set up by the configuration files.

To configure the logic analyzer for state mode:

- Load the appropriate logic analyzer configuration file (see “To load configuration files (and the inverse assembler) from hard disk” on page 36).

  You can change the master clock setting by opening the logic analyzer’s Setup window, selecting the Format tab, and clicking the Master Clock button to open the master clock dialog.

To change to timing analysis

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

Inverse assembly is not available in the timing analysis mode.

To configure the logic analyzer for timing analysis:

1. Load the appropriate logic analyzer configuration file (see "To load configuration files (and the inverse assembler) from hard disk" on page 36).
2. Open the logic analyzer's Setup window.
3. Select the Sampling tab.
4. Change the type option from State Mode to Timing Mode.

6

Capturing DSP Execution

The normal steps in using the logic analyzer are:

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

The inverse assembler is loaded, the logic analyzer is configured, and labels are created (formatted) for the logic analysis channels when configuration files are loaded (see "To load configuration files (and the inverse assembler) from hard disk" on page 36).

You can load program object file symbols into the logic analyzer when configuring it (see "Loading Symbol Information" on page 47).

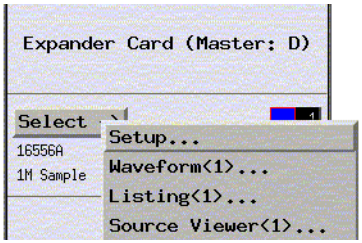
This chapter describes setting up logic analyzer triggers when using the inverse assembler and the Agilent Technologies B4620B Source Correlation Tool Set.

See Chapter 7, "Displaying Captured DSP Execution," beginning on page 63 for information on displaying captured data.

Setting Up Logic Analyzer Triggers

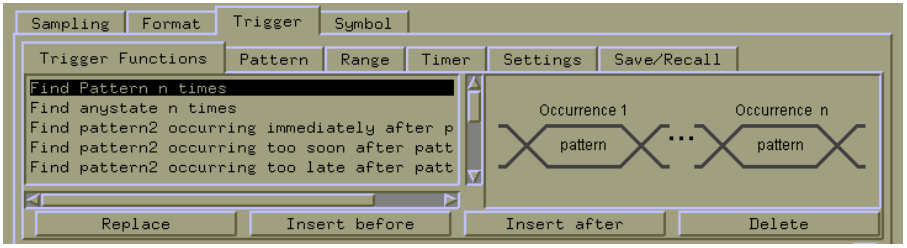
To set up logic analyzer triggers

1 Open the logic analyzer's Setup window.

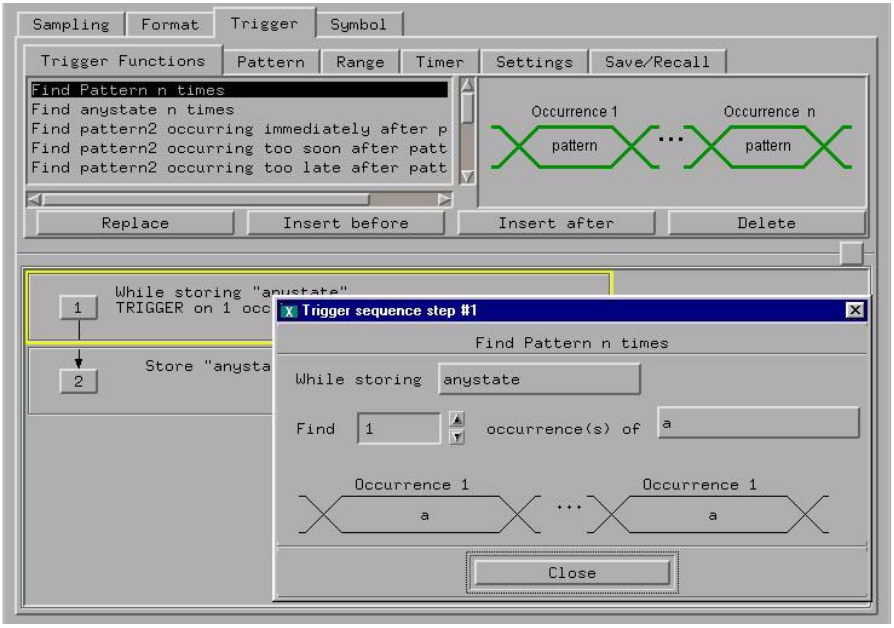


2 Select the Trigger tab.

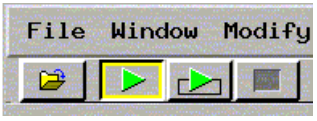
3 Define the patterns, ranges, and other resources that will be used in the logic analysis measurement.



4 Set up the trigger sequence.



5 Run the measurement.



See Also

The 16700-series logic analysis system's on-line help for more information on setting up logic analyzer triggers.

Triggering on Source Code

When setting up trigger specifications to capture program execution:

- Use the COFF 2.0 object file format.
- Be sure to correctly configure the memory type.

  ASYNC and SBSRAM can use straightforward triggering, but SDRAM requires a multi-stage trigger. Configuring the memory type will automatically set up the right kind of source line trigger.
- Remember that triggers will occur on prefetches.

To avoid capturing library code execution

When viewing the source code associated with captured data, the source correlation tool set can exhibit long response times to requests for the next source line if the current trace listing corresponds to code from a library that is not in the source code search path. Logic analyzer storage qualification can be used to avoid capturing library code routines.

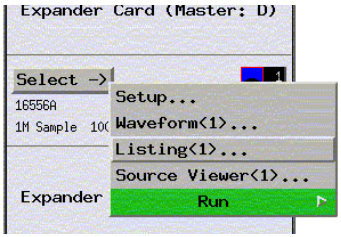
It is not practical to configure the logic analyzer's storage qualification capabilities to exclude idle states, because of prefetching.

7

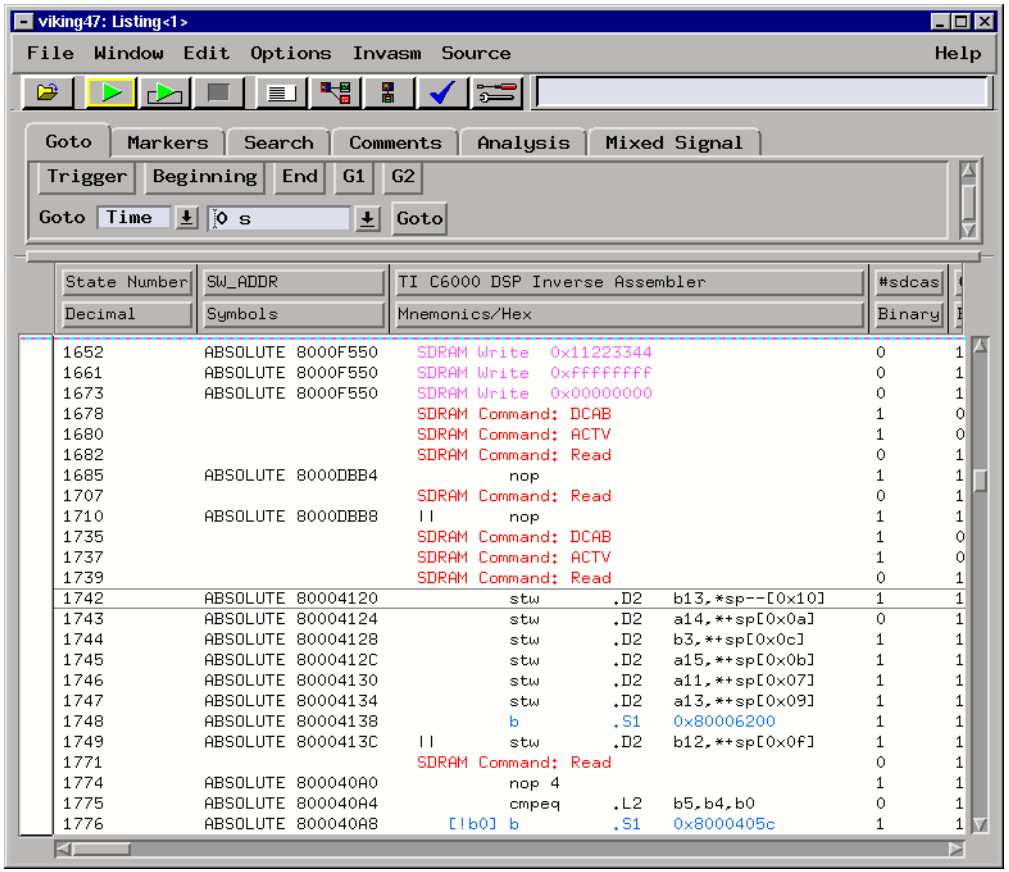
Displaying Captured DSP Execution

To display the captured state data

1 Open the Listing display window.



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



The inverse assembler is already loaded when state configuration files are loaded, but it can also be loaded into a Listing display using the Invasm menu. The name of the inverse assembler file is IC6000E, and it is located in the /logic/ia directory.

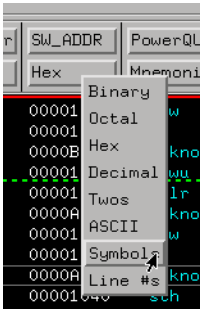
See Also

"To use the inverse assembler filters" on page 67 for information on displaying or hiding certain types of bus cycles.

The Agilent Technologies 16700-series logic analysis system on-line help for information on using the Listing display.

To display symbols

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



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

See Also

"To load object file symbols" on page 48.

To interpret the inverse assembled data

Constants, registers, and instructions are displayed using the usual formats. Here are a few things to be aware of:

Numbers.

Numerical data is displayed in hexadecimal, with a prefix of "`0x`", or decimal, without a prefix.

You can display offsets and constants in the inverse assembly listing in either hexadecimal or decimal (see "To set the Decoding Options preferences" on page 46). You can change the numeric base of the other labels in the listing (such as ADDR) by selecting the label at the top of the display.

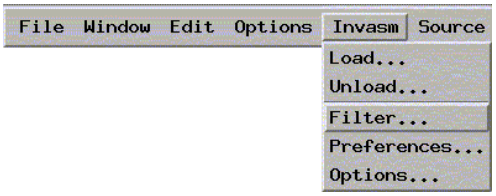
Instructions.

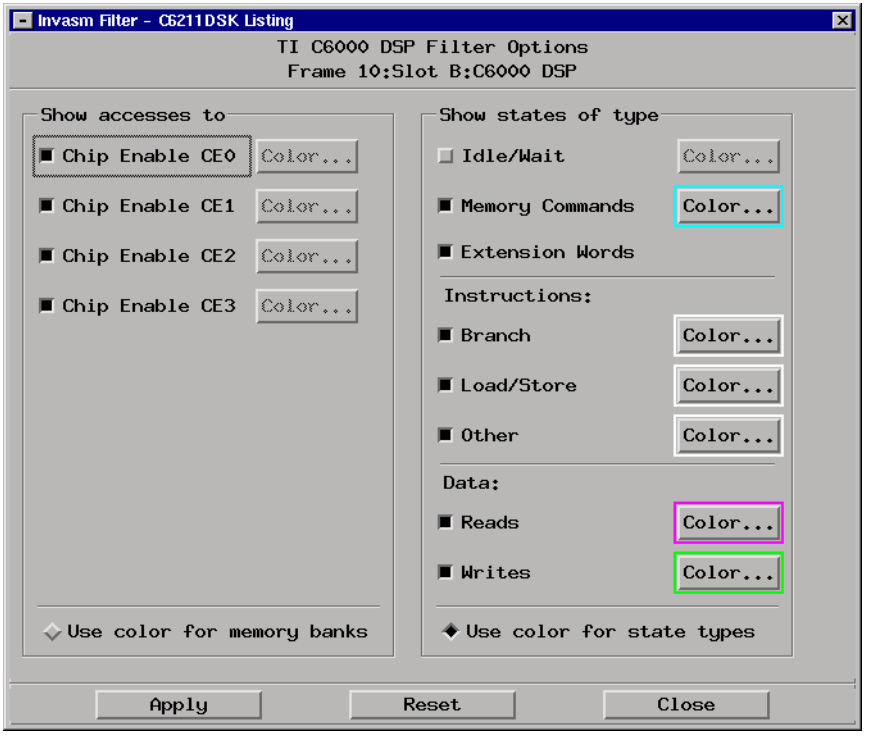
If an opcode is invalid or otherwise not recognized, it is shown as "Unknown Opcode."

Instructions which are executed in parallel are marked with "`||`".

To use the inverse assembler filters

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





The inverse assembler filtering options allow you to display or hide certain

types of bus cycles or memory bank accesses.

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

Filtering allows faster analysis in two ways:

- Unneeded information can be taken out of the display. For example, suppressing idle/wait states will let you view more instruction cycles without scrolling the listing.
- Particular operations can be isolated by suppressing all other operations. For example, Branch instructions can be shown, with all other states suppressed, allowing quick analysis of branch instructions.

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

You can display or hide the following types of cycles:

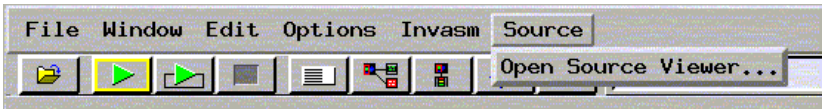
- Idle/Wait States.
- Extension Words.
- Memory Commands.
- Branch Instructions.
- Load/Store Instructions.
- Other Instructions.
- Data Reads.
- Data Writes.

See Also

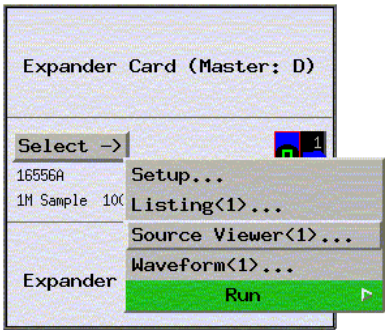
The address ranges for memory banks 0-11 are specified in the Preferences menu (see “To set the inverse assembler preferences” on page 40).

To view the source code associated with captured data

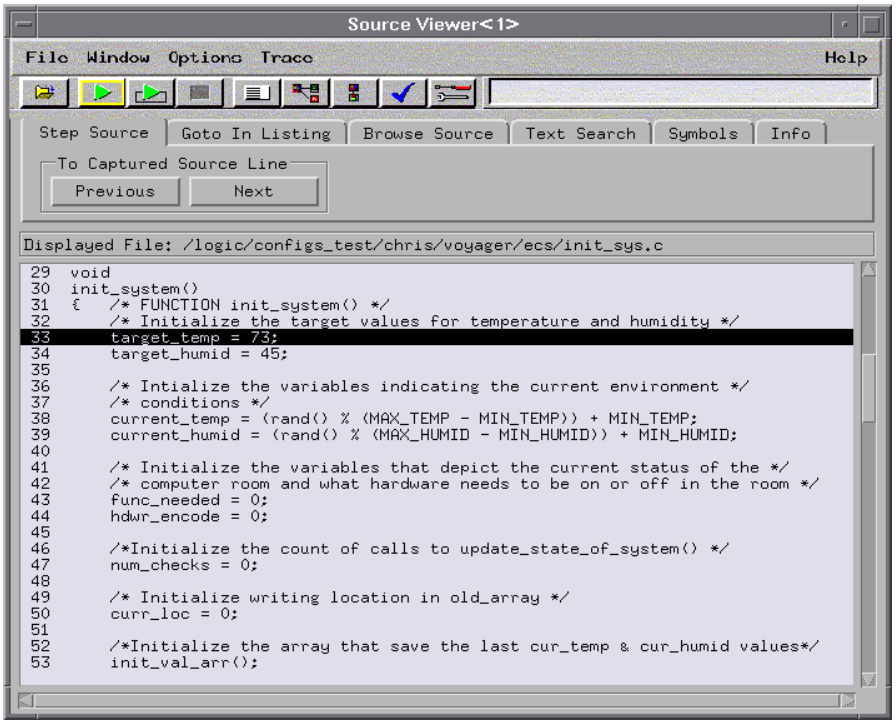
- In the Listing display window, select Source Viewer from the Source menu.



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



The source correlation tool set correlates the logic analyzer's address label with a line of high-level source code whose address, symbol name, file name, and line numbers are described in a symbol file downloaded to the logic analyzer (see "To load object file symbols" on page 48).



Inverse Assembler Generated SW_ADDR (Software Address) Label

In the Agilent Technologies 16700-series logic analysis system, the inverse assembler generates a “SW_ADDR” label. The SW_ADDR label is displayed as another column in the Listing tool. This label is also known as the software address generated by the inverse assembler.

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

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

For example, if the compiler unrolls a loop in the source code, the trace could

begin after the first assembly instruction of the loop has been executed. A "Goto this line in listing" command would not find the source line.

Access to Source Code Files

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

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

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

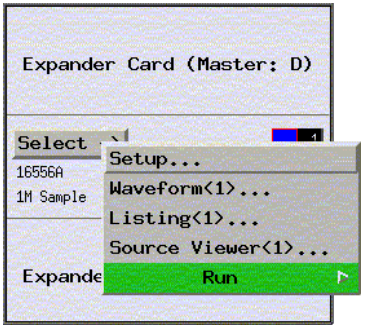
Source File Version Control. If the source code files are under a source code or version control utility, check the file names and paths carefully. These utilities can change source code file paths and file names. If these names are changed from the information contained in the symbol file, the source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

See Also

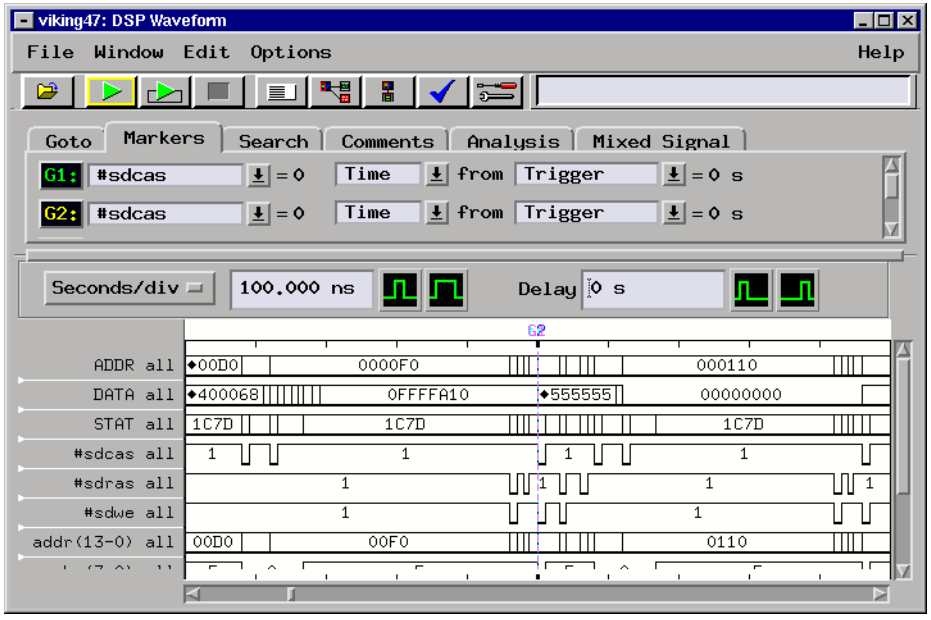
More information on configuring and using the source correlation tool set can be found in the on-line help for your logic analysis system.

To display captured timing analysis mode data

- Open the Waveform display for your logic analyzer.



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



8

General-Purpose ASCII (GPA) Symbol File Format

General-Purpose ASCII (GPA) Symbol File Format

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

If your compiler does not include symbol information in the output, or if you want to define a symbol not in the object file, you can create an ASCII format symbol file.

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

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

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

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

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

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

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

Example

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

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

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

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

GPA Record Format Summary

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

[FUNCTIONS]
func_name start..end

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

[SOURCE LINES]
File: file_name
line# address

[START ADDRESS]
address

#Comments
```

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

Example

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

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

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

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

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

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

SECTIONS

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

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

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

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

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

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

- NORMAL (default)—The section is a normal, relocatable section, such as code or data.
- NONRELOC—The section contains variables or code that cannot be relocated; this is an absolute segment.

Define sections first

To enable section relocation, section definitions must appear before any other definitions in the file.

Example

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

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

FUNCTIONS

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

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

`func_name` A symbol representing the function name.

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

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

Example

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

VARIABLES

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

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

`var_name` A symbol representing the variable name.

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

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

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

Example

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

SOURCE LINES

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

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

`file_name` The name of a file.

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

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

Example

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

START ADDRESS

```
[START ADDRESS]
address
```

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

Example

```
[START ADDRESS]
00001000
```

Comments

`#comment text`

Use the # character to include comments in a file. Any text following the # character is ignored. You can put comments on a line alone or on the same line following a symbol entry.

Example

```
#This is a comment.
```

9

Specifications and Characteristics

Operating Characteristics

The following operating characteristics are not specifications, but are typical operating characteristics for the E8137A inverse assembler.

Operating Characteristics	
Microprocessor Compatibility	Texas Instruments TMS320C6202/6203/6204/ 6205/6211 DSP. This product does not support the TMS320C6201.
Agilent Technologies Logic Analyzers Supported	16550A 16554/55/56/57 16710/11/12A 16715/16/17/18/19A 16750/51/52A
Accessories Required	For state and timing analysis, the E5346A high-density cables are required.
Probes Required	At least four 16-channel probes are required for disassembly.

Electrical Characteristics
See the documentation for the logic analyzer card you are using

10

Troubleshooting the Logic Analyzer

If you encounter difficulties while making measurements, use this chapter to guide you through some possible solutions. Each heading lists a problem you may encounter, along with some possible solutions.

If you still have difficulty using the analyzer after trying the suggestions in this chapter, please contact your local Agilent Technologies service center.

CAUTION: When you are working with the analyzer, be sure to power down both the analyzer and the target system before disconnecting or connecting cables, and probes. Otherwise, you may damage circuitry in the logic analyzer or target system.

Solving Logic Analyzer Problems

This section lists general problems that you might encounter while using the analyzer.

Intermittent data errors

This problem is usually caused by poor connections, incorrect signal levels, or marginal timing.

- ❏ Remove and reseat all cables and probes, ensuring that there are no bent pins or poor connections.
- ❏ Adjust the threshold level of the data pod to match the logic levels in the system under test.
- ❏ Use an oscilloscope to check the signal integrity of the data lines.

  Clock signals for the state analyzer must meet particular pulse shape and timing requirements. Data inputs for the analyzer must meet pulse shape and setup and hold time requirements.

See Also See “Capacitive loading” on page 90 for information on other sources of intermittent data errors.

Unwanted triggers

Unwanted triggers can be caused by instructions that were fetched but not executed.

- ❏ Set the trigger at an address which is not the target of a conditional branch to avoid this problem. For example, set the trigger in the middle of a subroutine, rather than at the first instruction of a subroutine.

  The logic analyzer captures prefetches, even if they are not executed. When you are specifying a trigger condition or a storage qualification that follows an instruction that may cause branching, an unused prefetch may generate an

unwanted trigger.

No activity on activity indicators

- ❏ Check for loose cables or board connections.
- ❏ Check for bent or damaged pins.

No trace list display

If there is no trace list display, it may be that your trigger specification is not correct for the data you want to capture, or that the trace memory is only partially filled.

- ❏ Check your trigger sequencer specification to ensure that it will capture the events of interest.
- ❏ Try stopping the analyzer; if the trace list is partially filled, this should display the contents of trace memory.

Solving Probing Problems

This section lists probing problems that you might encounter when using a logic analyzer. If the solutions suggested here do not correct the problem, you may have a damaged logic analyzer. Contact your local Agilent Technologies Sales Office if you need further assistance.

Target system will not boot up

If the target system will not boot up after connecting the target system, the processor (if socketed) or the probe cables may not be installed properly, or they may not be making electrical contact.

❏ Ensure that you are following the correct power-on sequence for the logic analyzer and target system.

1. Power up the logic analyzer.
2. Power up the target system.

If you power up the target system before you power up the logic analyzer, interface circuitry may latch up and prevent proper target system operation.

❏ Verify that the logic analyzer cables are in the proper target system connector headers and are firmly inserted.

Erratic trace measurements

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

- ❏ Do a full reset of the target system before beginning the measurement.

  Some designs require a full reset to ensure correct configuration.

- ❏ Ensure that your target system meets the timing requirements of the processor with the logic analyzer connected.

  See “Capacitive loading” on page 90. While logic analyzer probe loading is slight, pin protectors, extenders, and adapters may increase it to unacceptable levels. If the target system design has close timing margins, such loading may cause incorrect processor functioning and give erratic trace results.

- ❏ Ensure that you have sufficient cooling for the processor.

  Ensure that you have ambient temperature conditions and airflow that meet or exceed the requirements of the processor manufacturer.

Capacitive loading

Excessive capacitive loading can degrade signals, resulting in incorrect capture by the logic analyzer, or system lockup in the processor. All logic analyzer probes add additional capacitive loading, as can custom probe fixtures you design for your application.

Careful layout of your target system can minimize loading problems and result in better margins for your design. This is especially important for systems that are running at frequencies greater than 50 MHz.

- ❏ Remove as many pin protectors, extenders, and adapters as possible.

Solving Inverse Assembler Problems

This section lists problems that you might encounter while using the inverse assembler.

When you obtain incorrect inverse assembly results, it may be unclear whether the problem is in the logic analyzer or in your target system. If you follow the suggestions in this section to ensure that you are using the logic analyzer and inverse assembler correctly, you can proceed with confidence in debugging your target system.

No inverse assembly or incorrect inverse assembly

This problem may be due to incorrect synchronization, modified configuration, incorrect connections, or a hardware problem in the target system. A locked status line can cause incorrect or incomplete inverse assembly.

- ❏ Ensure that each logic analyzer pod is connected to the correct connector.

  There is not always a one-to-one correspondence between analyzer pod numbers and probe cable numbers. Probes must supply address (ADDR), data (DATA), and status (STAT) information to the analyzer in a predefined order. The cable connections for each probe are often altered to support that need. Thus, one probe might require that you connect cable 2 to analyzer pod 2, while another will require you to connect cable 5 to analyzer pod 2. See “Connecting the Logic Analyzer to the Target System” on page 32 for connection information.

- ❏ Check the activity indicators for status lines locked in a high or low state.

- ❏ Verify that the STAT, DATA, and ADDR format labels have not been modified from their default values.

  These labels must remain as they are configured by the configuration file. Do not change the names of these labels or the bit assignments within the labels.

  Some inverse assemblers also require other data labels. See “Configuring the Logic Analyzer” on page 33 for more information.

- ❏ Verify that all processor caches and memory managers have been disabled.

  In most cases, if the processor caches and memory managers remain enabled you should still get inverse assembly; however, it will consist of cache line fills.

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

- ❏ Check that the inverse assembler preferences (especially the memory map) have been set correctly.

Inverse assembler will not load or run

You need to ensure that you have the correct system software loaded on your analyzer.

- ❏ Ensure that the inverse assembler is on the same disk as the configuration files you are loading.

  Configuration files for the state analyzer contain a pointer to the name of the corresponding inverse assembler. If you delete the inverse assembler or rename it, the configuration process will fail to load the disassembler.

  See “Configuring the Logic Analyzer” on page 33 for details.

Solving Intermodule Measurement Problems

Some problems occur only when you are trying to make a measurement involving multiple modules.

An event wasn't captured by one of the modules

If you are trying to capture an event that occurs very shortly after the event that arms one of the measurement modules, it may be missed due to internal analyzer delays. For example, suppose you set the oscilloscope to trigger upon receiving a trigger signal from the logic analyzer because you are trying to capture a pulse that occurs right after the analyzer's trigger state. If the pulse occurs too soon after the analyzer's trigger state, the oscilloscope will miss the pulse.

- ❏ Adjust the skew in the Intermodule menu.

  You may be able to specify a skew value that enables the event to be captured.

- ❏ Change the trigger specification for modules upstream of the one with the problem.

  If you are using a logic analyzer to trigger the oscilloscope, try specifying a trigger state one state before the one you are using. This may be more difficult than working with the skew because the prior state may occur more often and not always be related to the event you are trying to capture with the oscilloscope.

Logic Analyzer Messages

This section lists some of the messages that the analyzer displays when it encounters a problem.

". . . Inverse Assembler Not Found"

This error occurs if you rename or delete the inverse assembler file that is attached to the configuration file.

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

- For Agilent 16700-series logic analysis systems it should be in /logic/ia.
- For other logic analyzers it should be in the same directory as the configuration file.

"No Configuration File Loaded"

This is usually caused by trying to load a configuration file for one type of module/system into a different type of module/system.

❑ Verify that the appropriate module has been selected from the Load {module} from File {filename} in the disk operation menu. Selecting Load {All} will cause incorrect operation when loading most configuration files.

See Also

See "Configuring the Logic Analyzer" on page 33 for a description of how to load configuration files.

"Selected File is Incompatible"

This occurs when you try to load a configuration file for the wrong module. Ensure that you are loading the appropriate configuration file for your logic analyzer.

"Slow or Missing Clock"

- ❏ This error message might occur if the logic analyzer cards are not firmly seated in the Agilent Technologies 16700-series logic analysis system frame or in the Agilent Technologies 16701A expansion frame. Ensure that the cards are firmly seated.
- ❏ This error might occur if the target system is not running properly. Ensure that the target system is on and operating properly.
- ❏ If the error message persists, check that the logic analyzer pods are connected to the proper connectors on the target system. See "Connecting the Logic Analyzer to the Target System" on page 32 to determine the proper connections.

"Time from Arm Greater Than 41.93 ms"

The state/timing analyzers have a counter to keep track of the time from when an analyzer is armed to when it triggers. The width and clock rate of this counter allow it to count for up to 41.93 ms before it overflows. Once the counter has overflowed, the system does not have the data it needs to calculate the time between module triggers. The system must know this time to be able to display data from multiple modules on a single screen.

“Waiting for Trigger”

If a trigger pattern is specified, this message indicates that the specified trigger pattern has not occurred. Verify that the triggering pattern is correctly set.

- When analyzing processors that fetch only from word-aligned addresses, if the trigger condition is set to look for an opcode fetch at an address not corresponding to a word boundary, the trigger will never be found.

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

Background Debug Monitor Also called Debug Mode, In Background, and In Monitor. The normal processor execution is suspended and the processor waits for commands from the debug port. The debug port commands include the ability to read and write memory, read and write registers, set breakpoints and start the processor running (exit the Background Debug Monitor).

Debug Mode See *Background Debug Monitor*.

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

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

Emulation Migration The hardware and software required to use an emulation probe with a new processor family.

Emulation Module An emulation module is installed within the mainframe of a logic analysis system. An E5901A emulation module is used with a *target interface module* (TIM) or an analysis probe. An E5901B emulation module is used with an E5900B *emulation probe* and does not use a TIM.

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

Emulation Solution A set of tools for debugging your target system. A solution includes probing, inverse assembly, the B4620B Source Correlation Tool Set, and an emulation module.

Emulator An emulation module or an emulation probe.

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

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

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.

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

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

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

In Background, In Monitor See *Background Debug Monitor*.

Inverse Assembler Software that displays captured bus activity as assembly language mnemonics. In addition, inverse assemblers may show execution history or decode control busses.

IP address Also called Internet Protocol address or Internet address. A 32-bit network address. It is usually represented as decimal numbers separated by periods; for example, 192.35.12.6.

Jumper Moveable direct electrical connection between two points.

JTAG (OnCE) port See *debug port*.

Label Labels are used to group and identify logic analyzer channels. A label consists of a name and an associated bit or group of bits.

Link-Level Address 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. Also known as an LLA, Ethernet address, hardware address, physical address, or MAC address.

Mainframe Logic Analyzer A logic analyzer that resides on one or more board assemblies installed in a 16500, 1660-series, or 16600/700-series mainframe.

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

MICTOR Connector A high-density matched impedance connector manufactured by AMP Corporation. *High-density adapter cables* can be used to connect the logic analyzer to MICTOR connectors on the target system.

Monitor, In See *Background Debug Monitor*.

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

Preprocessor See *Analysis Probe*.

Preprocessor Interface See *Analysis Probe*.

Probe Adapter See *Elastomeric Probe Adapter*.

Processor Probe See *Emulation Probe*.

Run Control Probe See *Emulation Probe* and *Emulation Module*.

Setup Assistant Wizard software program which guides a user through the process of connecting and configuring a logic analyzer to make measurements on a specific microprocessor. The setup assistant icon is located in the main system

window.

Shunt Connector. See *Jumper*.

Solution A set of tools for debugging your target system. A solution includes probing, inverse assembly, the B4620B Source Correlation Tool Set, and (optionally) an emulation module.

Stand-Alone Logic Analyzer A standalone logic analyzer has a predefined set of hardware components which provide a specific set of capabilities. A standalone logic analyzer differs from a mainframe logic analyzer in that it does not offer card slots for installation of additional capabilities, and its specifications are not modified based upon selection from a set of optional hardware boards that may be installed within its frame.

State Analysis A mode of logic analysis in which the logic analyzer is configured to capture data synchronously with a clock signal in the target system.

Subnet Mask A subnet mask blocks out part of an IP address so 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.

Symbol Symbols represent patterns and ranges of values found on labeled sets of bits. Two kinds of symbols are available:
1) 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.
2) User-defined symbols — Symbols you create.

Target Board Adapter A daughter board inside the E5900B emulation probe which customizes the emulation probe for a particular microprocessor family. The target board adapter provides an interface to the ribbon cable which connects to the debug port on the target system.

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

Target Interface Module A small circuit board which connects the 50-pin cable from an E5901A emulation module or E5900A emulation probe

to signals from the debug port on a target system. Not used with the E5900B emulation probe.

TIM See *Target Interface Module*.

Timing Analysis A mode of logic analysis in which the logic analyzer is configured to capture data at a rate determined by an internal sample rate clock, asynchronous to signals in the target system.

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

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

1/4-Flexible Adapter An adapter that obtains one-quarter of the signals from an elastomeric probe adapter (one side of a target microprocessor) and makes them available for probing.

Index

A

access to source code files, 71
accessories required, 84
active low signal, 21
activity indicators, 88, 91
adapter, 19
adapters, 90
ADDR label, 48, 91
address offset field, 51
address range
  memory banks 0-11, 68
addresses
  mask, 39
Agilent Technologies 1252-7431, 19
Agilent Technologies 16550A logic analyzer, 38
Agilent Technologies 16554A logic analyzer, 38
Agilent Technologies 16710A logic analyzer, 38
Agilent Technologies 16715/16/17A logic analyzers, 38
Agilent Technologies B4620B source correlation tool set, 58
Agilent Technologies E5346-44701 shroud, 19
Agilent Technologies E5346-60002 adapter, 19
Agilent Technologies E5346A high-density termination adapter cable, 19
Agilent Technologies E9611A Option 001 inverse assembler, 14
analysis mode
  changing, 54
  state, 54, 72
  timing, 54
analysis probe
  definition, 97
arm, 93, 95
ASCII format (GPA), 74
assembly language mnemonics, 35
asynchronous sampling, 55

B

B4620B source correlation tool set, 2, 28
background debug monitor, 97
bank enable/disable, 42
base address, 42
blank pins, 21
board space, 19
branch instructions, 68
bus cycles, 68

C

cache memory, 18
caches, 92
capacitive loading, 90
captured data, source code associated with, 69
captured execution, displaying, 63
capturing execution, 57
cards
  See logic analyzers
CD-ROM, 14, 16
CD-ROM, installing software from, 29
center inline pins, 21
characteristics, 83
CLKIN signal, 54
clock signals, 87
color, 68
comments, in GPA files, 82
configuration files, 14, 28, 38, 47, 54, 55, 58, 65, 92, 94, 95
configuring the logic analyzer, 35
connection notes
  recommended configuration, 21
connector board, 98
connector layout, 21
  recommended, 21
cooling, 90
counter overflow, 95
creating GPA symbol files, 74
custom probe fixtures, 90
cycle types, 68

D

data cache, 18
DATA label, 91
data reads, 68
data writes, 68
debug mode, 97
debug port, 97
  definition, 97
decoding
  simplified mnemonic, 46
decoding options, 46
deep memory logic analyzer, 70
design considerations, 18
directories
  configuration files, 36
display timing analysis mode data, 72
displaying captured execution, 63
download symbol information, 47

E

E8126A inverse assembler, 2
E9503A emulation solution, 2
E9603A Option 001 inverse assembler, 2
elastomeric probe adapter
  definition, 97
electrical characteristics, 84
Emulation Control Interface, 28
emulation migration
  definition, 97
emulation module
  definition, 97
emulation probe
  definition, 97
emulation solution, 2
  definition, 97

Index

equipment, 11
erratic trace measurements, 90
Ethernet networks, 71
event wasn't captured, 93
extender, 98
extenders, 90
extension words, 68

F

filters, inverse assembler, 67
fixed code offsets, 51
flexible adapter, definition, 98
frequencies greater than 50 MHz, 90
ftp, 71
FUNCTIONS in GPA format, 79

G

gateway address, definition, 98
General-Purpose ASCII (GPA) symbol file, 48
General-Purpose ASCII format, 74
  address format, 74
  comments, 82
  FUNCTIONS, 79
  record format summary, 76
  record headers, 74
  SECTIONS, 78
  simple form, 74
  SOURCE LINES, 81
  START ADDRESS, 82
  VARIABLES, 80
general-purpose flexible adapter definition, 98
ground returns, 21

H

high-density adapter cable definition, 98
high-density connector
  mechanical specifications, 20
  pin assignment, 20
high-density connectors, 19
high-density termination adapter definition, 98
high-level source code, 69

I

idle/wait states, 68
incorrect inverse assembly, 91
incorrect signal levels, 87
information sources, 4
inline pins, 21
input voltage, maximum, 18
installing analyzer modules, 27
installing software, 28
intermittent data errors, 87
intermodule measurement problems, 93
internal analyzer delays, 93
internal data cache, 18
internal instruction cache, 18
internal sample rate clock, 55
invalid opcode, 66
Invasm menu, 65
inverse assembled data, 66
inverse assembler, 18, 28, 35, 39, 65, 70, 92
  definition, 98
  equipment supplied, 14
  file name, 94
  filters, 67
  loading files, 36, 37
  not found, 94
  preferences, 40
  problems, 91
  requirements for, 39
  software, 14
  will not load, 92
inverse assembly, 39, 54, 92
  incorrect, 91
  traditional, 39
IP address
  definition, 98

J

jumper, definition, 99

L

labels
  defining, 53
  definition, 99
  predefined, 52
LAN protocols, 71
LAN system administrators, 71
layout, 90
library code execution, 61
link-level address
  definition, 99
Listing display window, 64
Load menu, 39
load/store instructions, 68
loading configurations, vs. installing, 36
loading object file symbols, 48
loading symbol information, 47
loading, minimum, 18
locked status line, 91
logic analysis
  preparing target for, 18
logic analysis system
  setting up, 25
  software version, 16
logic analyzer, 86
  configuration files, 38, 47
  configuring, 33
  connecting to target system, 32
  deep memory, 70
  maximum input voltage, 18
  messages, 94
  modules, installing, 27
  pods, 91, 95
  solving problems, 87
  storage qualification, 61
  trigger setup, 59
  triggers, 58
  troubleshooting, 85

Index

logic analyzers
  configuring, 36, 37
  supported, 16, 84

M

mainframe logic analyzer
  definition, 99
male-to-male header
  definition, 99
marginal timing, 87
master clock dialog, 54
maximum input voltage, 18
mechanical specifications
  high-density connector, 20
memory bank, 68
  accesses, 68
memory banks, 39
memory management units, 51
memory managers, 92
memory map, 41
memory map information, 41
messages
  logic analyzer, 94
Mictor (Matched Impedance ConnecTOR) connectors, 19
MICTOR connector, definition, 99
minimum loading, 18
mnemonics, assembly language, 35
monitor, background debug, 97

N

network access to source files, 71
NFS client/server, 71
no configuration file loaded, 94
no inverse assembly, 91
no-connect, 21
numerical data, 66

O

object file formats, 47
object file symbols, 58
  loading, 48
opcode fetch, 96
Options menu, 39
oscilloscope, 87, 93
  modules, installing, 27
other instructions, 68
other options, 45, 46

P

pattern generator modules, installing, 27
patterns, 59
PCI analysis, 16
personality files, 28
pin protectors, 90
plastic shroud, 19
pods, logic analyzer, 91, 95, 99
poor connections, 87
power-on sequence, 89
power-ON/OFF sequence, 26
predefined symbols, 47
preferences, inverse assembler, 40
prefetches, 87
preprocessor
  See analysis probe
preprocessor interface
  See *analysis probe*
probe fixtures, custom, 90
probed signal lines, 18
probes required, 84
probing the target system, 31
problems
  intermodule measurement, 93
  inverse assembler, 91
  logic analyzer, 87
  probing, 89
processor caches, 92
processor compatibility, 84
processor options, 44
processor support package, 28, 29
pulse shape requirements, 87

R

ranges, 59
recommended circuit board routing, 20
recommended configuration connection notes, 21
recommended connector layout, 21
recommended signal routing, 21, 22
record format, General-Purpose ASCII, 76
record headers, 74
references, 4
relocated code, compensating for, 51
requirements, 11
resources, 59
run control tool
  See emulation control interface

S

sample period, 55
sample rate clock, internal, 55
search path, source code, 61, 71
Section Format, 74
SECTIONS in GPA format, 78
selected file is incompatible, 95
setup and hold time requirements, 87
Setup Assistant, 12, 26, 28
setup assistant
  definition, 99
Setup window, 47
setup/hold window, 18
shroud, 19, 20
signal ground returns, 21
signal integrity, 19, 87
signal routing
  recommended, 22
simplified instructions, 46
simplified mnemonics, 46

Index

skew, 93
slow or missing clock, 95
software
  list of installed, 38
software addresses, 35, 70
software supplied, 14
software version, 16
software, installing, 28
solution, definition, 100
source code, 69
  associated with captured data, 69
  search path, 61
  triggering on, 61
source correlation tool set, 2, 69, 71
source files
  network access to, 71
  search path, 71
  version control, 71
SOURCE LINES in GPA format, 81
specifications, 83
stand-alone logic analyzer
  definition, 100
START ADDRESS in GPA format, 82
STAT label, 91
state analysis, 100
  definition, 100
state mode, 54, 72
  changing to, 54
status lines, locked, 91
storage qualification, 61, 87, 92
subnet mask
  definition, 100
supplied equipment and software, 14
support shroud, 20
supported logic analyzers, 16
suppressing all other operations, 68
surface mount connector, 20
SW_ADDR label, 35, 70
symbol file, 69
symbol files
  creating, 74
symbol information
  loading, 47
Symbol Selector dialog, 50
symbols
  definition, 100
  predefined, 47
Symbols tab, 47
symbols, displaying, 65
synchronization, 91
system administrators, 71

T

target board adapter
  definition, 100
target control port, 100
target interface module (TIM)
  definition, 100
target system, 86, 91
  connecting logic analyzer to, 32
  preparing, 17
  probing, 31
  won't boot up, 89
TCP/IP protocol, 71
telnet, 71
threshold level, 87
time from arm greater than 41.93 ms, 95
timing analysis, 101
  definition, 101
timing analysis mode, 54
  changing to, 55
  data, displaying, 72
timing requirements, 87, 90
transition board
  definition, 101
trigger condition, 87
trigger pattern, 96
trigger sequence, 60
trigger sequencer specification, 88
trigger specification
  definition, 101
trigger tool, 47
triggering on source code, 61
troubleshooting, 85

U

undefined opcode, 66
unneeded information, 68
unused prefetch, 87
unwanted triggers, 87
user defined signals, 21
User Defined Symbols tab, 47
user-defined symbols, 47

V

VARIABLES in GPA format, 80
version control, source file, 71
version, software, 16

W

waiting for trigger, 96
Waveform display, 72
web sites
  logic analyzers, 4
  See Also under debugger names
word-aligned addresses, 96

X

X-Window client/server, 71

Y

Y-cable, 19

Restricted Rights Legend

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

Document Warranty

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

Agilent Technologies shall not be liable for errors contained herein or for damages in connection with the furnishing, performance, or use of this material.

Safety

This apparatus has been designed and tested in accordance with IEC Publication 1010, Safety Requirements for Measuring Apparatus, and has been supplied in a safe condition. This is a Safety Class I instrument (provided with terminal for protective earthing). Before applying power, verify that the correct safety precautions are taken (see the following warnings). In addition, note the external markings on the instrument that are described under "Safety Symbols."

Warning

- Before turning on the instrument, you must connect the protective earth terminal of the instrument to the protective conductor of the (mains) power cord. The mains plug shall only be inserted in a socket outlet provided with a protective earth contact. You must not negate the protective action by using an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two-conductor outlet is not sufficient protection.
- Only fuses with the required rated current, voltage, and specified type (normal blow, time delay, etc.) should be used. Do not use repaired fuses or short-circuited fuseholders. To do so could cause a shock of fire hazard.
- Service instructions are for trained service personnel. To avoid dangerous electric shock, do not perform any service unless qualified to do so. Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.
- If you energize this instrument by an auto transformer (for voltage reduction), make sure the common terminal is connected to the earth terminal of the power source.
- Whenever it is likely that the ground protection is impaired, you must make the instrument inoperative and secure it against any unintended operation.
- Do not operate the instrument in the presence of flammable gasses or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.
- Do not install substitute parts or perform any unauthorized modification to the instrument.
- Capacitors inside the instrument may retain a charge even if the instrument is disconnected from its source of supply.

Safety Symbols

Instruction manual symbol: the product is marked with this symbol when it is necessary for you to refer to the instruction manual in order to protect against damage to the product.

Hazardous voltage symbol.

Earth terminal symbol: Used to indicate a circuit common connected to grounded chassis.

WARNING

The Warning sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a Warning sign until the indicated conditions are fully understood and met.

CAUTION

The Caution sign denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a Caution symbol until the indicated conditions are fully understood or met.

Agilent Technologies
P.O. Box 2197
1900 Garden of the Gods Road
Colorado Springs, CO 80901-2197, U.S.A.

Product Warranty

This Agilent Technologies product has a warranty against defects in material and workmanship for a period of one year from date of shipment. During the warranty period, Agilent Technologies Company will, at its option, either repair or replace products that prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Agilent Technologies.

For products returned to Agilent Technologies for warranty service, the Buyer shall prepay shipping charges to Agilent Technologies and Agilent Technologies shall pay shipping charges to return the product to the Buyer. However, the Buyer shall pay all shipping charges, duties, and taxes for products returned to Agilent Technologies from another country.

Agilent Technologies warrants that its software and firmware designated by Agilent Technologies for use with an instrument will execute its programming instructions when properly installed on that instrument. Agilent Technologies does not warrant that the operation of the instrument software, or firmware will be uninterrupted or error free.

Limitation of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by the Buyer, Buyer- supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

No other warranty is expressed or implied. Agilent Technologies specifically disclaims the implied warranties of merchantability or fitness for a particular purpose.

Exclusive Remedies

The remedies provided herein are the buyer's sole and exclusive remedies. Agilent Technologies shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products. For any assistance, contact your nearest Agilent Technologies Sales Office.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

About this edition

This is the *Logic Analysis Support for the Texas Instruments TMS320C6000 Family User's Guide*.

Publication number
E8137-97000, October 2000
Printed in USA.

Print history is as follows:

Many product updates do not require manual changes, and manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.