

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Keysight / Agilent 16517A* at our website: ***Click HERE***

Help Volume

© 1992-2002 Agilent Technologies. All rights reserved.

Instrument: Agilent Technologies 16517A 4GHz Timing/1GHz State Logic Analyzer

Agilent Technologies 16517A/18A 4GHz Timing/1GHz State Logic Analyzer



The Agilent Technologies 16517A provides *timing* analysis at a sample rate of up to 4 GHz. It also offers *state* analysis that uses your target system's clock at speeds of up to 1 GHz. The 16517A can be connected to up to 4 Agilent Technologies 16518A expansion cards, giving 80 channels each with a memory depth of 64 Ksamples.

Getting Started

- "Connecting to Your Target System" on page 10
- "Adjusting Skew" on page 16
- "Setting Up a Measurement" on page 17
- "When Something Goes Wrong" on page 35
- "Error Messages" on page 35

Measurement Examples

- "Making a Basic Timing Measurement" on page 22
- "Making a Basic State Measurement" on page 26
- Advanced Measurement Examples (see the *Measurement Examples* help volume)
- "Interpreting the Data" on page 30

More Features

Correlating with Other Instruments (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

"Using Symbols" on page 85

Using Markers (see the *Markers* help volume)

"Loading and Saving Logic Analyzer Configurations" on page 34

"Testing the Logic Analyzer Hardware" on page 46

Interface Reference

"The Format Tab" on page 47

"The Trigger Tab" on page 55

The Skew Adjust Tab (see page 16)

"The Symbols Tab" on page 94

"Specifications and Characteristics" on page 80

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

Glossary of Terms (see page 105)

Contents

Agilent Technologies 16517A/18A 4GHz Timing/1GHz State Logic Analyzer

1 Agilent Technologies 16517A/18A 4GHz Timing/1GHz State Logic Analyzer

Connecting to Your Target System 10
Recommended Probe Configurations 10
Probing Accessories 11
Probing System Description 13
Requirements of Target Signals 14

Adjusting Skew 16

Setting Up a Measurement 17
Map the Analyzer to the Target System 17
Set Up the Analyzer 18
Define Trigger Conditions 19
Run the Measurement 20
Examine the Data 20

Making a Basic Timing Measurement 22

Making a Basic State Measurement 26

Interpreting the Data 30
Analysis Using Waveform 30
Analysis Using Listing 32

Loading and Saving Logic Analyzer Configurations 34

Contents

When Something Goes Wrong 35
Nothing Happens 35
Error Messages 35
Suspicious Data 44
Interference with Target System 45

Testing the Logic Analyzer Hardware 46

The Format Tab 47
Setting the Acquisition Mode 47
Defining Labels: Mapping Analyzer Channels to Your System 49
Working with Labels 50
Setting the Pod Threshold 53
Activity Indicators 54

The Trigger Tab 55
Setting Up a Trigger 55
Adding and Deleting Sequence Steps 57
Editing Sequence Steps 58
Setting Up Loops and Jumps in the Trigger Sequence 58
Saving and Recalling Trigger Sequences 59
Clearing Part or All of the Trigger 60
Overview of the Trigger Sequence 61
Predefined Trigger Macros 62
Working with User-Defined Macros 68
Defining Resource Terms 71
Trigger Position Control 76
Sample Period (Timing Only) 77
Oversampling: the Samples/Clock Control 77
Arming Control 78

Contents

Specifications and Characteristics 80
Agilent Technologies 16517A/18A Logic Analyzer Specifications 80
Agilent Technologies 16517A/18A Logic Analyzer Characteristics 81
What is a Specification 83
What is a Characteristic 83
What is a Calibration Procedure 84
What is a Function Test 84

Using Symbols 85
To load object file symbols 86
To adjust symbol values for relocated code 87
To create user-defined symbols 88
To enter symbolic label values 89
To create an ASCII symbol file 90
To create a readers.ini file 90

The Symbols Tab 94
Symbols Selector Dialog 96
Symbol File Formats 98
General-Purpose ASCII (GPA) Symbol File Format 99

Glossary

Index

Contents

1

Agilent Technologies 16517A/18A 4GHz Timing/1GHz State Logic Analyzer

Connecting to Your Target System

You can connect the logic analyzer to your target system either directly or through an *analysis probe*. The Agilent Technologies 16517A/18A logic analyzer has two cables ending in pod housings with leads for probing individual lines. Each logic analyzer is shipped with a Probe Accessories Kit, whose components are described below.

You can also connect to the target system through an analysis probe. The Agilent Technologies 16517A/18A logic analyzer requires a clear connection with no termination, so any analysis probe you use must be non-terminated. Additionally, the Agilent Technologies 16517A/18A logic analyzer does not support inverse assembly.

See Also

"Recommended Probe Configurations" on page 10

"Probing Accessories" on page 11

"Probing System Description" on page 13

"Requirements of Target Signals" on page 14

Recommended Probe Configurations

Long leads and improper signal connections can result in inconsistent or erroneous data. In an ideal connection, the probe tip connects directly to the target system without any additional lead lengths. Depending on your measurement, the added inductance caused by additional lead length could cause a problem for the signal rise time.

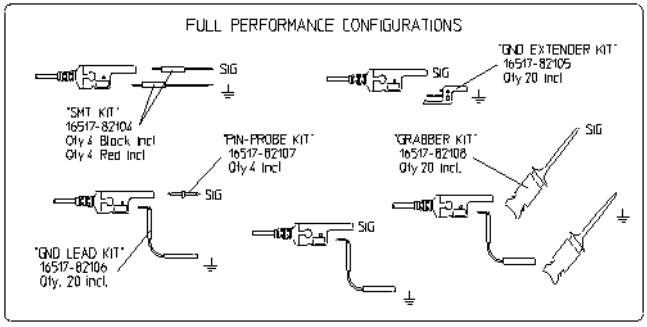
If you must add length to the ground lead, use the following general guidelines:

Target Rise Time Maximum Additional Ground Length

300 ps 1 inch

600 ps 2 inch

1 ns 3 inch



Recommended Configurations

Probing Accessories

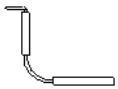
The probe accessories described below are part of the probe Accessory Kits supplied with the 16517A master and 16518A expander *cards*. To order any of these accessories individually, use the part numbers listed below or in the Accessory Kit box.

The probe accessories that plug onto straight pins will fit on 0.63 mm (0.025 in) square pins, or 0.66 mm to 0.84 mm (0.026 in to 0.033 in) diameter round pins.

Ground Leads

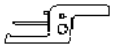
The following probing components connect the probe to ground.

Right Angle Ground Lead (Agilent Technologies 16517-82106)



This flexible lead is 1.5 inches long with a 90 degree bend off the lead tip. It stacks on 0.1 inch centers.

Ground Extender (Agilent Technologies 16517-82105)



If the circuit can tolerate an additional 0.5 to 1 pF capacitance, this socket allows stacking on 0.1 inch centers.

Ground Connector (Agilent Technologies 16515-27601)



This connector creates a 4-to-1 ground connection.

Signal Leads

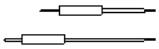
The following components connect the system to the signal.

Probe Pin (Agilent Technologies 16517-82107)



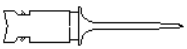
The probe pin allows touch probing.

SMT Tack-on Signal/Ground Wire (Agilent Technologies 16517-82104)



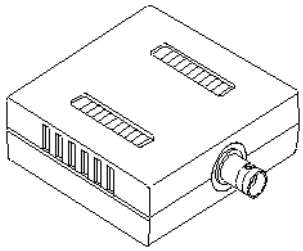
For surface-mount components, PGAs, or cramped areas, this wire can be tacked onto an IC lead for direct connection.

Grabbers (Agilent Technologies 16517-82108)



These 0.05 pitch grabbers attach to IC pins with lead spacing greater than or equal to 0.5 inch.

Calibration Pod (Agilent Technologies 16517-63201)



The calibration pod is only included in the accessory kit for the 16517A. You use the calibration pod during the skew adjust procedure. See “Adjusting Skew” on page 16 for information on performing the skew adjust procedure.

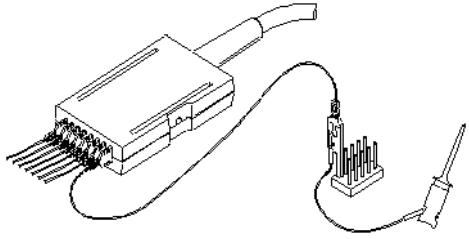
BNC to SMB Cable Adapter (Agilent Technologies 16517-61604)



On the back of the 16517A *card*, there are two SMB connectors used for external ECL arm in/out signals. Use this adapter between the BNC cable and the logic analyzer. The adapter cable is only included in the accessory kit for the 16517A.

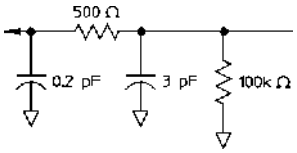
Probing System Description

The Agilent Technologies 16517A/18A logic analyzer probing system consists of shielded cables ending in a pod housing, with 12-inch coaxial cables for the individual probes. The probing accessories are plugged onto the probes.



The pod housing contains termination networks and comparators. The pod housing also shows the equivalent circuit for 4 GHz operation, so

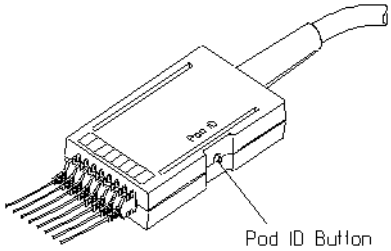
that you can calculate pod loading. It is also shown below.



Each probe has a ground connection, which must be used for acceptable signal quality. The accessory kit contains ground extenders. All probe accessories connect to 0.1-inch centers.

Only Pod 1 of the 16517A card has a *clock channel*. For *state measurements*, this probe must be connected to the target system's clock.

Pressing the Pod ID button on the pod housing causes a message that identifies the pod to appear on the logic analysis system.



See Also

"Requirements of Target Signals" on page 14

Logic Analysis System Installation Guide

Agilent Technologies 16517A/18A 4GHz Timing/1GHz State User's Reference

Requirements of Target Signals

Minimum Signal Amplitude

Any signal line you intend to probe with the logic analyzer probes must supply a minimum voltage swing of 500 mV to the probe tip. If you measure signal lines with a smaller voltage swing, you may not get reliable results. The minimum input overdrive (see page 15) is the

greater of 250 mV or 30% of signal amplitude.

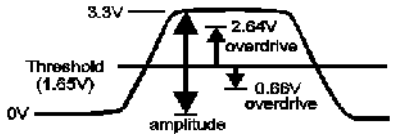
Maximum Probe Input Voltage

The maximum probe input voltage of each logic analyzer probe is 40 volts peak.

Overdrive

Overdrive is the amount a signal must exceed the threshold voltage for the logic analyzer to detect a change in logic level. For the Agilent Technologies 16517A/18A logic analyzer, overdrive is 250 mV or 30% of signal peak-to-peak amplitude, whichever is greater.

For example, given a 3.3 volt CMOS signal (low = 0V, high = 3.3 V) the optimal threshold is 1.65 V (50%). If the threshold is set less than 1.0 V or greater than 2.3 V, then a timing acquisition might show excessive channel-to-channel skew. For a state acquisition, the analyzer's setup and hold requirements might not be met.



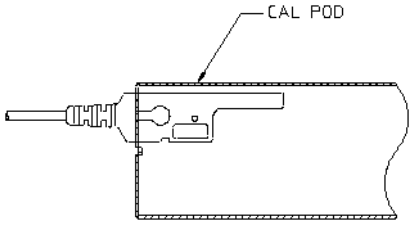
The overdrive amount is specified as the greater of 250 mV or 30% of the signal amplitude because it has two purposes. The 250 mV ensures reliable switching or state detection. The 30% of amplitude ensures the threshold is reasonably centered within the waveform in order to minimize channel-to-channel skew (t_{PHL} vs t_{PLH}).

Adjusting Skew

Skew adjustment minimizes channel-to-channel skew. This is a necessary step to make sure your Agilent Technologies 16517A/18A logic analyzer meets specifications.

This procedure requires the Agilent Technologies 16517-63201 calibration pod and a BNC coax cable.

1. Connect the BNC cable to the Cal port on the back panel of the 16517A *master card*.
2. Connect the other end of the BNC cable to the calibration pod's *Cal input*.
3. In the Agilent Technologies 16517A/18A 4GHz Timing/1GHz State window, select the *Skew Adjust* tab.
4. Select the *Start* button.
5. Follow the instructions on screen. Be sure to plug the probes all the way into the module, as shown below.



The Agilent Technologies 16517A/18A does not require an operational accuracy calibration. To test it against the module specifications, refer to "Testing Performance" in the optional *Agilent Technologies 16517A/18A Service Guide*, available from your Agilent Technologies Sales Office.

Setting Up a Measurement

There are six basic steps for any measurement.

1. "Map the Analyzer to the Target System" on page 17
2. "Set Up the Analyzer" on page 18
3. "Define Trigger Conditions" on page 19
4. "Run the Measurement" on page 20
5. "Examine the Data" on page 20
6. Refine measurement by repeating steps 3 - 5.

If you load a configuration file, it will set up the logic analyzer and trigger sequence. For your particular measurement, you may need to change some settings.

See Also

"Making a Basic Timing Measurement" on page 22

"Making a Basic State Measurement" on page 26

Measurement Examples (see the *Measurement Examples* help volume)

Making Basic Measurements for a self-paced tutorial

Map the Analyzer to the Target System

The first step is to physically connect the logic analyzer to your target system in a way that makes sense.

Connect Pods

The logic analyzer pods detect the signals on your target system. Attach pods in a way that keeps logically related channels together. Be sure to ground each probe.

If you plan on making *state measurements*, be sure to connect the clock probe on Pod 1 of the 16517A *master card* to your target

system's clock line.

Step 2: Set Up the Analyzer (see page 18)

See Also "Connecting to Your Target System" on page 10

Set Up the Analyzer

The next step is to set up the logic analyzer. These controls are grouped under the *Format* tab. If you load a configuration file, this step is taken care of for you.

Set Measurement Type and Clocks

The logic analyzer can perform state or timing measurements. State measurements are also known as synchronous or logic measurements. Timing measurements are also known as asynchronous, internal clock, or periodic measurements. State measurements use the target system's clock to determine when to sample. Timing measurements sample at a fixed rate which you select.

For state measurements, you must specify a clock edge to match the clocking arrangement used by your target system. If the clock is incorrect, the trace data may indicate a problem where there isn't one.

Group Bits With Labels

Pod fields correspond to the cables on the logic analyzer module. The bits in the pod fields correspond to the individual signals monitored by the pods. Bits with activity on them show double-headed arrows, like so . Labels group bits into logical signals; for example, "addr bus". These groupings are then used in the trigger tab and the data displays. A label can have up to 32 channels. Each measurement can define 126 labels.

Step 3: Define Trigger Conditions (see page 19)

See Also

"Setting the State Clock" on page 48

"Assigning Bits to a Label" on page 50

"The Format Tab" on page 47

Define Trigger Conditions

The third step is to define the trigger. Controls for this are located under the *Trigger* tab. Configuration files saved from previous measurements automatically define trigger conditions.

Define Terms

If you think of labels as variables, terms are the specific values the variables can assume. A hardware metaphor would be that labels are bits, and terms are the specific values. Terms can match patterns and edges.

For example, if you wanted to trigger on a write to a specific address, first you'd group the address bus into a label and then you'd set up a term based on that label with the address.

Set Up a Trigger Sequence

The trigger sequence is like a small program that controls when the logic analyzer stores data. There are trigger macros for the common tasks, or you can set up your own. The logic analyzer starts at the first trigger level until either the main branch or the "else" condition becomes true. When that happens, it goes to the next level and follows the instructions there.

Step 4: Run the Measurement (see page 20)

See Also

"Defining Resource Terms" on page 71

"Setting Up a Trigger" on page 55

"The Trigger Tab" on page 55

Measurement Examples (see the *Measurement Examples* help volume)

Run the Measurement

You run the measurement by selecting a button sometimes labeled Run, sometimes Group Run, sometimes Run All. The difference between the three types is that *Run* starts only the instrument you are using, *Group Run* starts all instruments attached to group run in the Intermodule window, and *Run All* starts all instruments currently placed in the workspace.

Runs can be single or repetitive. Single runs gather data until the logic analyzer memory is full, and then stop. Repetitive runs keep repeating the same measurement and are useful for gathering statistics. Repetitive runs on a logic analyzer do not do equivalent time sampling like oscilloscopes do.

If you want to stop a run, select the *Stop* button.

Step 5: Examine the Data (see page 20)

Examine the Data

Data from your measurement can be viewed in various display windows or offline. Some of the things you can do in the display windows are

- Gather statistics
- Search for patterns
- Display time-correlated data

Automatically gather statistics

In the repetitive mode, set markers on data points you are interested in and then select repetitive run.

Search for patterns

Using markers, you can search displays for certain values. There are two global markers which keep their place across all measurement views, even across instruments.

Display correlated data

There are several tools for correlation. The Intermodule window allows you to specify complex triggering configurations using several instruments. It is also useful for starting acquisitions at the same time. Global markers mark the same events in different displays, so you can switch views without having to reorient yourself. The Compare tool lets you compare two different acquisitions to look for changes.

See Also

Working with Markers (see the *Markers* help volume)

Using the Chart Display Tool (see the *Chart Display Tool* help volume)

Using the Distribution Display Tool (see the *Distribution Display Tool* help volume)

Using the Listing Display Tool (see the *Listing Display Tool* help volume)

Using the Digital Waveform Display Tool (see the *Waveform Display Tool* help volume)

Using the Compare Analysis Tool (see the *Compare Tool* help volume)

“Interpreting the Data” on page 30

Making a Basic Timing Measurement

This example uses the circuit board that is supplied with the *Making Basic Measurements* kit as the target system. The kit is supplied with every logic analysis system, or can be ordered from your Agilent Technologies Sales Office.

There are nine major steps to making a basic measurement. When using the Agilent Technologies 16517A/18A logic analyzer with the training board, you need to connect pod 1 of an Agilent Technologies 16550, 16554, 16555, or 16556 logic analyzer to J1 of the training board in order to supply power.

Map the Logic Analyzer to the Target System

1. Connect probes.
   a. Using ground extenders, connect the probes of Pod 1 to J2. You need to ground each probe.
   b. Connect Pod 1 of the other logic analyzer to J1 on the target system to provide power.
2. On the Agilent Technologies or 16700A/B logic analysis system, open the 16517 logic analyzer setup window.
   a. In the main window, select the logic analyzer icon.
   b. Choose *Setup...* from the menu.
3. If the logic analyzer is not already set for *Timing*, change the sampling mode to *Timing*.
   a. Select the *Format* tab.
   b. Select the sampling mode option button and choose *Timing - Full Channel*.
4. Group channels with *labels*.
   a. Optional - Insert a second label.
      1. Select *Lab1*.

2. Choose *Insert after...*.
3. In the *Enter Label Name* box, select *OK*.

b. Optional - Rename *Lab1*

1. Select *Lab1*.
2. Choose *Rename...*.
3. Enter a new name in the name field.
4. Select *OK* to dismiss the *Rename Label* box.

c. Select the bit assignment button.
The bit assignment button is to the right of a label name, and under a pod column.

d. Choose `********` from the menu.
If none of the choices match your own system, choose *Individual...* and select the individual bits to assign them (*) or ignore them (.).

5. Define trigger terms for a bus.

a. Select the *Trigger* tab.

b. Optional - Rename pattern term *patt1*.

1. Select the *patt1* field.
2. Enter a new name.

c. Select the appropriate label.

1. Select the label button immediately to the right of the term name.
2. To define the term as a combination of labels, choose *Insert...* To use a different label to define the term, choose *Replace...*
3. In the dialog box, select the label name you want to use and then select the *OK* button.

d. Select the field with *XX* and enter the value you want to trigger on.

6. Define trigger terms for an edge.

a. Select the *Edge* tab in the lower part of the window.

b. Optional - Rename *edge1*.

1. Select the *edge1* field.
2. Enter a new name.

c. Select the appropriate label.

1. Select the label button immediately to the right of the term name.
2. To define the term as a combination of labels, choose *Insert...* To use a different label to define the term, choose *Replace...* Edges within a term are always OR'd together, which means only one of the edges on one of the labels needs to occur for the edge term to become true.
3. In the dialog box, select the label name you want to use and then select the *OK* button.

d. Select the edge assignment button (. . . . . .) and enter the edge or edges you want to trigger on. Remember, if more than one edge is specified, then when the logic analyzer detects any of the edges the term becomes true.

7. Add the edge term to the trigger sequence.

a. Select *Modify* from the menu bar, then *Break down macros*.

b. Select the *1* sequence level button and choose *Edit...*

c. In the dialog box, select the *patt1* button and choose *Combo...*

d. In the Combination box, select *Off* next to *edge1* and choose *On*.

e. Select the *Or* option button where the path from *patt1* and the path from *edge1* come together, and choose *And*.

f. Select the *OK* button.
The analyzer is now set to trigger when it detects edge1 and patt1 is on the bus. The trigger sequence window shows

```
Find "(patt1*edge)" then TRIGGER
```

.

The logic analyzer automatically triggers on the first trigger term. You can set up more complex triggers by editing the sequence levels and defining additional trigger terms.

See Also

"The Trigger Tab" on page 55

1. Select the *Run* button.
2. Examine the data.
   a. Select the *Window* menu.
   b. Select the slot of your high-speed logic analyzer and choose *Waveform*.
   c. To have the waveform display appear automatically when you run the logic analyzer, select Options -> Popup on Run -> On in the menu bar of the waveform display.
   d. To insert additional labels, or expand overlaid signals, select the label name.

See Also

For Connection Information

Logic Analysis System Installation Guide

For Details on the Training Board or More Tutorials

Making Basic Measurements

Examples of Typical Timing Measurements

Hardware Turn-On (see the *Measurement Examples* help volume) measurements.

Firmware Development (see the *Measurement Examples* help volume) measurements.

System Integration (see the *Measurement Examples* help volume) measurements.

For Details on the Logic Analyzer Interface

"The Format Tab" on page 47

"The Trigger Tab" on page 55

Making a Basic State Measurement

This example uses the circuit board that is supplied with the *Making Basic Measurements* kit as the target system. The kit is supplied with every logic analysis system, or can be ordered from your Agilent Technologies Sales Office.

There are ten major steps to making a basic measurement. When using the Agilent Technologies 16517A/18A logic analyzer with the training board, you need to connect pod 1 of an Agilent Technologies 16550, 16554, 16555, or 16556 logic analyzer to J1 of the training board in order to supply power.

Map the Logic Analyzer to the Target System

1. Connect probes.
   a. Using ground extenders, connect the probes of Pod 1 to J2. You need to ground each probe. Be sure to connect the CLK probe to CLK1 of the target system.
   b. Connect Pod 1 of the logic analyzer to J1 on the target system.
2. On the Agilent Technologies 16700A/B logic analysis system, open a logic analyzer setup window.
   a. In the main window, select the logic analyzer icon.
   b. Choose *Setup...* from the menu.
3. If the logic analyzer is not already set for *State*, change the sampling mode to *State*.
   a. Select the *Format* tab.
   b. Select the sampling mode option button and choose *State - Full Channel*.
4. Check that the logic analyzer is detecting a clock signal.

   The *Clock Period* display shows the measured target system clock. For the training board, the clock is about 40 ns.

   If the Clock Period display shows a line, the logic analyzer is not detecting

any clock. Check that the CLK probe of Pod 1 of the *master card* is connected to CLK1 of J2 on the training board, and that the probe is grounded. If there is still no activity, check that the other logic analyzer is connected to J1 to supply power.

5. Group channels with *labels*.

   a. Optional - Insert a second label.

      1. Select the *Lab1* button.
      2. Choose *Insert after...*.
      3. In the *Enter Label Name* box, select the *OK* button.

   b. Optional - Rename *Lab1*.

      1. Select the *Lab1* button.
      2. Choose *Rename...*.
      3. Enter a new name in the name field.
      4. Select *OK* to dismiss the *Rename Label* box.

   c. Select the bit assignment button.
   The bit assignment button is to the right of a label name, and under a pod column.

   d. Choose `********` from the menu.
   If none of the choices match your own system, choose *Individual...* and select the individual bits to assign them (*) or ignore them (.).

6. Define trigger terms for patterns on buses.

   a. Select the *Trigger* tab.

   b. Optional - Rename term *patt1*.

      1. Select the *patt1* field.
      2. Enter a new name.

   c. Select the appropriate label.

      1. Select the label button immediately to the right of the term name.
      2. To define the term as a combination of labels, choose *Insert...* To use a different label to define the term, choose *Replace...*

3. In the dialog box, select the label name you want to use and then select the *OK* button.

d. Select the field with *XX* and enter the value you want to trigger on.

e. Optional - Repeat steps a - d for term *patt2*.

7. Optional - Add additional trigger terms to the trigger sequence.
The logic analyzer automatically triggers on *patt1*, the first trigger term. You can set up more complex triggers by editing the sequence levels and combining trigger terms.

a. Select the *1* sequence level box and choose *Edit...*

b. In the dialog box, select the *patt1* button and choose *Combo...*

c. In the Combination box, select *Off* next to *patt2* and choose *On*.

d. To change the trigger to *patt1 and patt2*, select the *Or* button to the right of the terms and choose *And*.

e. Select the Combination dialog's *OK* button.

f. Select the sequence level dialog's *OK* button.
The analyzer is now set to trigger when it detects both the pattern defined by *patt1* and the pattern defined by *patt2* on the target system's buses. The trigger sequence windows shows

```
Find 1 occurrence of "(patt1*patt2)" then TRIGGER
```

See Also

"The Trigger Tab" on page 55

1. Select the *Run* button.

2. Examine the data.

a. Select the *Window* menu.

b. Select the slot of your high-speed logic analyzer and choose *Listing*.

c. To have the listing display appear automatically when you run the logic analyzer, select Options -> Popup on Run -> On in the menu bar of the listing display.

d. To insert additional labels, select the label name.

See Also

For Connection Information

Logic Analysis System Installation Guide

For Details on the Training Board or More Tutorials

Making Basic Measurements

Examples of Typical Timing Measurements

The "Looking at State Events" group under Hardware Turn-On (see the *Measurement Examples* help volume) measurements.

Firmware Development (see the *Measurement Examples* help volume) measurements.

System Integration (see the *Measurement Examples* help volume) measurements.

For Details on the Logic Analyzer Interface

“The Format Tab” on page 47

“The Trigger Tab” on page 55

Interpreting the Data

After you've acquired a trace with the logic analyzer, you can analyze it in the display tools. The logic analysis system also provides filtering and compare tools for more complex analysis.

The logic analyzer is automatically connected to the Waveform and Listing displays when you set up a measurement. To move to that display,

1. Select the *Window* menu.
2. Move the cursor over the name of the analyzer whose data you want to view.
3. Choose *Waveform* or *Listing*.

- "Analysis Using Waveform" on page 30
- "Analysis Using Listing" on page 32

Analysis Using Waveform

Example: Looking for a Missing Pattern

You can easily use the waveform tool to make timing measurements. For example, if you were triggering when a pattern doesn't follow an edge within a certain time (see the *Measurement Examples* help volume), you would probably want to look at your *data set* to see if the pattern ever did occur. This might be the case when you verifying that the system is responding to an interrupt.

After triggering on an instance where the response did not appear quickly enough, you might take these steps in the Waveform display:

1. Find the edge.
   a. Select the *Search* tab.
   b. Select the down arrow after the Label field, and choose the label containing the edge.

c. Select the value field and enter 1.

d. Select the *Next* button to locate the edge transition.

2. Place a marker on the edge.

   Select *Set G1*. This sets global marker G1 at the location of the edge you just found.

3. Search for the pattern. Searches start at your current location. Since you just set the global marker G1, it indicates where the search starts from.

   a. Select the down arrow after the Label field, and choose the label containing the pattern.

   b. Select the value field and enter the pattern you are searching for.

   c. Select the down arrow after the When field and choose *Entering*.

   d. Select the *Next* button to find the next occurrence of that pattern after G1.

   If the logic analysis system cannot find the pattern, a "Value not found" message pops up.

4. Place a marker on the pattern.

   Select *Set G2*. This will set global marker G2 at the location of the pattern.

5. Find the time between the edge and the pattern.

   a. Select the *Markers* tab.

   b. In the G2 row, select the down arrow after from, and choose *G1*.

   The value after the from field changes to the time between G1 and G2. You can toggle between time and samples by selecting the arrow after the Time or Samples field.

See Also

Using the Digital Waveform Display Tool (see the *Waveform Display Tool* help volume)

Using the Listing Display Tool (see the *Listing Display Tool* help volume)

Using the Chart Display Tool (see the *Chart Display Tool* help volume)

Using the Distribution Display Tool (see the *Distribution Display Tool* help volume)

Using the Compare Analysis Tool (see the *Compare Tool* help volume)

Using the Pattern Filter Analysis Tool (see the *Pattern Filter Tool* help volume)

Analysis Using Listing

Listing is more useful than Waveform when your target system is running code because it shows the labels as states rather than transitions. Listing is especially useful when you have defined meaningful symbol names for your states.

Example: Examining a Subroutine

Listing is the preferred display tool for state measurements. for example, if you were trying to see if a subroutine were exiting abnormally, you might want to measure the number of states between entering and exiting the subroutine. After acquiring data with the logic analyzer, you could examine the *data set* in the Listing display like this:

1. Find the start of the subroutine.

   Assume the subroutine starts at the address 0x58FC.

   a. Select the *Search* tab.

   b. Select the down arrow after the Label field, and choose *ADDR*.

   c. Select the value field, and enter the starting address, 0x58FC.

   d. Select the down arrow after the When field and choose *Present*.

   e. Select the *Next* or *Prev* buttons to move the display to the address.

2. Place a marker on the start of the subroutine.

   Select the *Set G1* button. This sets global marker G1 at the address you just found.

3. Find the end of the subroutine.

   Assume the end of the subroutine is at address 0x58FF. Searches always start at the current location. Since you just set the global marker G1, it indicates where the search starts from.

a. Select the value field, and enter 58FF.

b. Select the *Next* button to find the next occurrence of 0x58FF after the starting address.

4. Place a marker on the end of the subroutine.

   Select the *Set G2* button to set global marker G2 at this position. This lets you refer to G2 when you want to know where the subroutine ends.

5. Find the number of states between the start and end of the subroutine.

   Since you've placed markers at the start and end of the subroutine, all you have to do is find the number of states between those markers.

   a. Select the *Markers* tab.

   b. In the G2 row, select the second down arrow and choose *Sample*.

   c. Select the down arrow after from, and choose *G1*.

   The value after the from field changes to the number of states between G1 and G2. You can toggle between time and states by selecting the arrow after the Time or Samples field.

Now you know how long the execution stayed in the subroutine, and can also examine the data set between G1 and G2 to look for unusual data.

See Also

Using the Digital Waveform Display Tool (see the *Waveform Display Tool* help volume)

Using the Listing Display Tool (see the *Listing Display Tool* help volume)

Using the Chart Display Tool (see the *Chart Display Tool* help volume)

Using the Distribution Display Tool (see the *Distribution Display Tool* help volume)

Using the Compare Analysis Tool (see the *Compare Tool* help volume)

Using the Pattern Filter Analysis Tool (see the *Pattern Filter Tool* help volume)

Loading and Saving Logic Analyzer Configurations

The Agilent Technologies 16517A/18A logic analyzer settings and data can be saved to a configuration file. You can also save any tools connected to the logic analyzer. Later, you can restore your data and settings by loading the configuration file into the logic analyzer.

The Agilent Technologies 16517A/18A logic analyzer can only load configuration files that it generated. It cannot load configurations from other logic analyzers.

NOTE: The Agilent Technologies and 16700A/B-series logic analysis systems can translate configuration files from Agilent Technologies 16500 and 16505 logic analysis systems.

See Also

Loading Configuration Files (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

Saving Configuration Files (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

When Something Goes Wrong

- "Nothing Happens" on page 35
- "Error Messages" on page 35
- "Suspicious Data" on page 44
- "Interference with Target System" on page 45

Nothing Happens

Look for an error message in the *message bar* at the top of the window. A common message is "Waiting for trigger". Another that appears in the Run Status window is "Cannot Run."

If the *Run* button briefly grayed-out (while the *Stop* or *Cancel* button was briefly active), select the *Window* menu, select the logic analyzer's slot, then choose the Waveform or Listing display.

See Also

"Waiting for Trigger" on page 36

"Cannot Run. External 16517 Clock Period Should be 1 ns to 50 ns" on page 39

"Loss of Sync with External 16517 Clock." on page 40

Error Messages

"Waiting for Trigger" on page 36

"Maximum of 32 Channels Per Label" on page 37

"Using Default 16517/18 Skew Values." on page 37

"16517/18 Skew Values are from a Different Instrument" on page 38

"16517/18 Memory System Failed -- Replace Card(s)" on page 38

"Invalid 16517/16518 Card Configuration." on page 38

"Cannot Run. External 16517 Clock Period Should be 1 ns to 50 ns" on page 39

"Measurement Halted in Sequence Level N." on page 39

"External Clock Out of Spec." on page 40

"Loss of Sync with External 16517 Clock." on page 40

"NOTE: Trigger Sequencer Will Not See Oversampled States." on page 41

"NOTE: 500 MHz Trigger Sequencer Rate Exceeds Sample Rate." on page 41

"NOTE: Sample Rate Exceeds 500 MHz Trigger Sequencer Rate.\" on page 41

"Pod XX identified" on page 41

Skew-related Messages

"Data Channels (X....X..) Are Not Connected Properly" on page 42

"The Clock is Not Connected Properly" on page 42

"16517/18 Skew Values Do Not Match Slot Configuration." on page 43

"16517/18 Skew Values Are From Different Instrument." on page 43

"Using Default 16517/18 Skew Values." on page 43

"Invalid 16517/18 Skew Values." on page 44

Waiting for Trigger

This message indicates that the specified trigger pattern has not occurred. This may be expected, as when you are waiting to trigger on an unusual event.

Possible Causes

- Misaligned boundaries for addresses

  When the target is a microprocessor that fetches only from long-word aligned addresses, if the trigger is set to look for an opcode fetch at an

address that is not properly aligned, the trigger will never be found.

- Trigger set incorrectly

Some strategies you can use when verifying or debugging trigger sequence levels are:

- Look at the run status message line or open the Run Status window. It will tell you what level of the sequence the logic analyzer is in.
- Stop the measurement and look at the data that was captured.
- Save the trigger setup, then simplify it to see what part of the sequence does get captured. When you learn what needs to be changed, you can recall the original trigger setup and make changes to it.

See Also

"Saving and Recalling Trigger Sequences" on page 59

Maximum of 32 Channels Per Label

The logic analyzer can only assign up to 32 channels for each label. If you need more than 32 channels, assign them to two labels and use the labels in conjunction.

See "Adding and Deleting Labels for Terms" on page 75 for how to use more than one label with trigger *terms*.

Using Default 16517/18 Skew Values.

Possible Causes

The 16517A 4 GHz Timing/1 GHz State logic analyzer functions best when its channels have been deskewed. This message warns you that the logic analyzer has been moved since it was last deskewed and that measurements might be off.

To deskew the channels, go to the logic analyzer window, select the *Skew Adjust* tab, and follow the instructions.

See Also

Using the Agilent Technologies 16517A Logic Analyzer (see the *Agilent Technologies 16517A 4GHz Timing/1GHz State Logic Analyzer* help volume)

16517/18 Skew Values are from a Different Instrument

Possible Causes

- Logic analyzer was last deskewed in a different frame

  The 16517A logic analyzer is very sensitive to changes in its environment. If you move the logic analyzer from an Agilent Technologies 16500C frame to an Agilent Technologies 16700A/B-series frame, or between a 16600A-series and 16700A/B-series frame, you should deskew the channels before making measurements.

To deskew the channels, go to the logic analyzer window, select the *Skew Adjust* tab, and follow the instructions.

See Also

Using the Agilent Technologies 16517A Logic Analyzer (see the *Agilent Technologies 16517A 4GHz Timing/1GHz State Logic Analyzer* help volume)

16517/18 Memory System Failed -- Replace Card(s)

The memory system on the *card* specified has failed. If your 16517A or 16518A is still under warranty, contact your Agilent Technologies Sales Office for repair.

Invalid 16517/16518 Card Configuration.

This message appears on power-up if the logic analyzer *module* is not cabled together properly.

Possible Causes

- An 16518A is in the *frame* but not part of a module

  The 16518A is strictly an expansion *card*. It cannot be used on its own without an 16517A.

- The cables connecting the cards of the module are on the wrong side.
  - When the expansion card is *above* the 16517A master card, use the connector on the *left* side of the cards.
  - When the expansion card is *below* the 16517A master card, use the connector on the *right* side of the cards.
  - When the 16517A master card is *between* expansion cards, use the

connector on the left for the cards above the master card and the connector on the right for the cards below the master card.

"Left" and "right" are when looking at the back of the mainframe.

- You have more than two expansion cards either above or below the 16517A *master card*.

  One end of the interconnect cables must be connected to the 16517A master card. Since the longest cable has three plugs, this means you can have no more than two cards on either side of the master card.

- You are trying to use an 16517A master card as an expansion card.

  The 16517A master card cannot be used as an expansion card. Each module should contain only one 16517A.

See Also *Agilent Technologies 16517A/18A User's Reference*

Cannot Run. External 16517 Clock Period Should be 1 ns to 50 ns

This message only appears in *state* mode.

Possible Causes

- The *clock channel* is not connected properly.
- The target system clock period is greater than 50 ns. (Slower than 20 MHz.)

  If your target system was running near 20 MHz, the capacitive load from the logic analyzer could slow it down further than 20 MHz.

- The target system clock period is less than 1 ns. (Faster than 1 GHz.)

  The Agilent Technologies 16517A/18A logic analyzer is only capable of 1 GHz in state mode. For faster speeds, you need to use timing mode.

Measurement Halted in Sequence Level N.

This message occurs when the Agilent Technologies 16517A/18A logic analyzer was stopped before it triggered.

Possible Causes

- You pressed *Stop*.

  If the message bar was displaying "Waiting for Trigger" when you pressed

Stop, the logic analyzer had not yet found its trigger event. The level is the level in the *trigger sequence* in which it was waiting.

- A fatal error occurred.

  Some error conditions, such as skew values from a different instrument, are serious enough to cause the logic analyzer to halt a run.

See Also

"Using Default 16517/18 Skew Values." on page 37

"16517/18 Skew Values are from a Different Instrument" on page 38

"Cannot Run. External 16517 Clock Period Should be 1 ns to 50 ns" on page 39

External Clock Out of Spec.

This message only occurs in *state* mode.

Possible Causes

- The target system clock is slightly greater than 50 ns.
- The target system clock is slightly less than 1 ns.

The period of the target system clock is not in the preferred range of 1 ns to 50 ns, but is close enough that the Agilent Technologies 16517A/18A logic analyzer can still run. The data may not be valid, however.

See Also

"Cannot Run. External 16517 Clock Period Should be 1 ns to 50 ns" on page 39

Loss of Sync with External 16517 Clock.

This message only occurs in *state* mode when the logic analyzer loses the clock signal.

Possible Causes

- The clock probe has been disconnected.

  Check the target system to make sure the clock probe is still firmly connected to the clock channel of your target system.

- The target system is not running.

  Look for activity indicators for the data channels. If they had been showing activity and no longer do, the target system may need to be reset.

- The voltage threshold level of Pod 1 has been changed.

  The *clock channel* is on Pod 1 of the *master card*. If the voltage threshold level is changed for the data channels, it will also change the voltage threshold for the clock channel. Check the *Format* tab to see if it is set properly.

NOTE: Trigger Sequencer Will Not See Oversampled States.

This message appears when you set the Samples/Clock control (available under the Trigger tab in *state* mode only) to something other than 1.

It is a warning that only states that are synchronous with the clock on the *target system* will be evaluated in the *trigger sequence*.

See Also "Oversampling: the Samples/Clock Control" on page 77

NOTE: 500 MHz Trigger Sequencer Rate Exceeds Sample Rate.

In *timing* mode, you have set the sample period to a value greater than 2 ns (slower than 500 MHz). This message lets you know that you could sample data at a faster rate.

NOTE: Sample Rate Exceeds 500 MHz Trigger Sequencer Rate.\

Trigger May Not Be Included In Acquired Data

In *timing* mode, you have set the sample period to a value less than 2 ns (faster than 500 MHz).

Because data will be sampled faster than the trigger sequencer can evaluate it, the *trigger* may not be included in the acquired data.

See Also "Suspicious Data" on page 44

Pod XX identified

This message appears in the logic analyzer window when the Pod ID button on the side of a pod is pressed. The "XX" is replaced with the

letter of the slot the *card* is in, and 1 or 2 to identify the pod.

The message goes away when the logic analysis system detects a mouse movement.

See Also "Probing System Description" on page 13

Data Channels (X....X..) Are Not Connected Properly

This message only appears when you run the *Skew Adjust* procedure, and one or more *data channels* of the pod being tested are not connected.

The *X* in the message refers to the channel that is not connected, with the leftmost position referring to channel 1 of the pod being adjusted.

Make sure that all pod probes are plugged completely into the calibration pod. (The top of the probe should completely fill the slot showing on top of the calibration pod.) Select the *Continue* button in the Pod Connections dialog. If the warning message still appears and the referenced probe is completely plugged in, the probe hardware is broken.

See Also "Adjusting Skew" on page 16

The Clock is Not Connected Properly

This message appears if you attempt to adjust skew without connecting the clock probe to the calibration pod.

There is only one probe labeled *Clk* per 16517A/18A *module*, and it is on pod 1 of the 16517A card. The clock probe must remain plugged into the calibration pod through the entire skew adjust procedure.

If the clock probe is completely plugged into the calibration pod and you still get this message, the clock channel hardware may be bad.

Contact your Agilent Technologies sales office for information on repairing the 16517A *card*.

See Also "Adjusting Skew" on page 16

16517/18 Skew Values Do Not Match Slot Configuration.

This message appears when the logic analysis system is turned on if the 16517A/18A skew values do not match. The skew values are stored in NV-RAM on the 16517A *card*.

Possible causes

- The arrangement of 16518A expansion cards changed.
- The *module* was moved to another slot of the *frame*.

To correct this situation, run the Skew Adjust procedure.

See Also "Adjusting Skew" on page 16

16517/18 Skew Values Are From Different Instrument.

This message appears when the logic analysis system is turned on if the 16517A/18A skew values are not valid for the current instrument (Agilent Technologies 16700A/B-series frame).

Possible causes

- The 16517A/18A module was moved from one frame to another.

The skew values are stored in NV-RAM on the 16517A *card* and are preserved when the module is moved from one type of analysis frame to another. Because the different frame types have different internal wiring, the skew factors need to be readjusted.

To correct this situation, run the Skew Adjust procedure.

See Also "Adjusting Skew" on page 16

Using Default 16517/18 Skew Values.

This warning message appears when you start a run if the skew values are set to factory defaults. The factory default skew adjustments are good enough to allow the module to run, but may cause inaccuracies in data correlation.

Possible Causes

- The skew factors in NV-RAM aren't valid.

  You could have invalid skew factors if the *module* were moved within the *frame* or if expansion cards were added or removed.

- There were no skew factors in NV-RAM.

  Until you perform the skew adjust procedure, no skew factors are stored in NV-RAM.

To correct this situation, run the Skew Adjust procedure.

See Also "Adjusting Skew" on page 16

Invalid 16517/18 Skew Values.

This message appears when the skew values in NV-RAM fail a checksum test.

Possible Causes

- The module was last deskewed in an Agilent Technologies 16500 frame.
- The skew factors have been corrupted.

To correct this situation, run the Skew Adjust procedure.

See Also "Adjusting Skew" on page 16

Suspicious Data

Intermittent Data Errors

Check for poor connections, incorrect signal levels on the hardware, incorrect logic levels under the logic analyzer's Format tab, or marginal timing for signals.

Unexpected State at Trigger Point

If you are using oversampling in state mode, only the states synchronous with the external clock are evaluated by the trigger sequence. The event you expected may have occurred but not been evaluated. Try setting up the same trigger in timing mode.

Unwanted Triggers

If you are using an inverse assembler or a pipeline, triggers can be caused by instructions that were fetched but not executed. To fix, add the prefetch queue or pipeline depth to the trigger address.

The depth of the prefetch queue depends on the processor that you are analyzing, and can be quite deep.

Another solution which is sometimes preferred with very deep prefetch queues is to add writes to dummy variables to your software. Put the instruction just before the area you want to trigger on, then trigger on the actual write to this variable. Although the instruction is prefetched, the analyzer can be set to only trigger when the write is executed.

See Also

"Oversampling: the Samples/Clock Control" on page 77

Interference with Target System

Capacitive Loading on the Target System

Excessive capacitive loading can degrade signals, resulting in suspicious data or even system lockup. All analysis probes add capacitive loading, as can custom probes you design for your target. To reduce loading, remove as many pin protectors, extenders, and adapters as possible.

Careful layout of your target system can minimize loading problems and result in better margins for your design.

Testing the Logic Analyzer Hardware

In order to verify that the logic analyzer hardware is operational, run the Self Test utility. The Self Test function of the logic analysis system performs functional tests on both the system and any installed modules.

1. If you have any work in progress, save it to a configuration file. (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)
2. Select the *System Admin* button.
3. Select the *Admin*i tab, then select the *Self Test...* button.
   The system closes all windows before starting up Self Test.
4. Select *Master Frame*.
   If the module is in an expansion frame, select *Expansion Frame*.
5. Select the *16517A 4GHz Timing/1 GHz State* logic analyzer that you want to test.
6. In the Self Test dialog box, select *Test All*.
   You can also run individual tests by selecting them. Tests that require you to do something must be run this way.

If any test fails, contact your local Agilent Technologies Sales Office or Service Center for assistance.

See Also

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

Agilent Technologies 16517A/18A 4GHz Timing/1GHz State Service Guide

The Format Tab

Format offers control of the internal resources of the logic analyzer. Format selections tell the logic analyzer:

- Whether it will use the full-channel or half-channel data capture mode.
- What is the maximum rate of the sampling clock during measurements.
- Which logic levels to use to interpret captured data.
- What signals are grouped together.
- "Setting the Acquisition Mode" on page 47
- "Defining Labels: Mapping Analyzer Channels to Your System" on page 49

See Also

"Working with Labels" on page 50

"Setting the Pod Threshold" on page 53

"Activity Indicators" on page 54

"Setting the State Clock" on page 48 (State Only)

"State Clock Sample Offset" on page 49 (State Only)

Setting the Acquisition Mode

The acquisition mode sets whether the analyzer uses an internal or external clock to determine when to get data from the target system. This setting also controls the channel width, memory depth, and maximum sampling rate.

In timing acquisition mode the analyzer stores measurement data at each sampling interval, using its own internal clock. In state acquisition mode, the analyzer stores measurement data at each clock edge of the target system.

To Set the Acquisition Mode

1. Select the sampling mode option button.
2. Choose the option you want.
   See the references below for more detail on the choices.

State Mode Full Channel 1 GHz

The synchronous state acquisition mode requires an external clock. Both pods are available on all configured cards. Memory depth is 64 Ksamples per channel.

Timing Mode Full Channel 2 GHz

Both pods are available on all cards. Memory depth is 64 Ksamples per channel. The logic analyzer samples at up to 2 GHz.

Timing Mode Half Channel 4 GHz

Both pods are restricted to bits 0 - 3. Memory depth is doubled to 128 K samples per channel. The logic analyzer samples at 4 GHz, and the speed cannot be adjusted.

When you switch from half-channel to full-channel mode, you are asked if you want to restore bits 4 - 7. A 'Yes' answer restores these bits in their previous setting.

See Also (State only)

"Setting the State Clock" on page 48

"State Clock Sample Offset" on page 49

Setting the State Clock

In the State Acquisition mode, the analyzer is controlled by the state clock channel of Pod 1 on the *master card*. You can set the logic analyzer to sample the data on either the rising or falling edge of the target system clock.

To Set the State Clock

1. Select the *Clock* option button.
2. Choose either *Rising* or *Falling*.

State Clock Sample Offset

Sample offset... adjusts the relative position of the clock edge with respect to the time the data is sampled.

To Change Sample Offset

1. Select the *Sample offset...* button.
2. Choose *Coarse - All pods*, *Fine - All pods*, or *Fine - Individual pods*. "Coarse" has a 200-ps granularity, and "fine" has a 50-ps granularity.
3. Adjust the offset. The arrow indicates where the data will be sampled relative to the clock edge.
   If the mode is *Fine - Individual pods*, there is an arrow for each pod.
4. Select the *Close* button.

NOTE: Changing the mode to *All pods* will erase any offset changes you have made to the individual pods.

NOTE: The *Fine* modes are limited to ±500 ps from the *Coarse* setting.

Defining Labels: Mapping Analyzer Channels to Your System

Labels group channels with a name that is relevant to your system under test. Both data channels and clock channels from more than one pod can be included in a single label, but labels with clock channels or channels from more than one pod cannot be included in *range terms*.

You can define 126 labels per analyzer. Each label can contain up to 32 channels per label.

To Define a Label

1. Select the label name button and choose *Rename*.
2. Enter a new name and select the *OK* button.

3. Assign bits (see page 50) to the label.
4. If necessary, insert more labels (see page 52) in the list.

See Also

"Working with Labels" on page 50

Working with Labels

Labels are used to group and identify logic analyzer channels so that you can easily remember what signals the various logic analyzer channels refer to.

- "Assigning Bits to a Label" on page 50
- "Reordering the Bits in a Label" on page 51
- "Inserting and Deleting Labels" on page 52
- "Turning Labels On and Off" on page 52
- "Label Polarity" on page 53

Assigning Bits to a Label

The bits in a label correspond to the physical logic analyzer probe channels. When you run the analyzer, data is gathered on all bits (channels) that are assigned to *labels*. Unassigned bits are inactive.

(*) (asterisk) indicates an assigned bit.

(.) (period) indicates an unassigned bit.

To Assign Bits

1. Select the bit assignment button to the right of the label name you want to define.
   Each bit assignment button corresponds to the data pod which is listed

above it.

2. Either choose one of the predefined groups, or choose *Individual*.
3. In *Individual*, select the bits to toggle them between an asterisk and a period.
   You can also hold the left mouse button and move the mouse to assign several bits at once.

NOTE: Labels can have a maximum of 32 channels assigned to them.

Bits assigned to a label are numbered from right to left. The least significant assigned bit on the far right is numbered 0. The next assigned bit to the left is numbered 1, and so on. Labels can contain bits that are not consecutive; however, bits are always numbered consecutively within a label. Above each column of bit assignment fields is a bit reference line that shows you the bit numbers and activity indicators.

See Also "Activity Indicators" on page 54

Reordering the Bits in a Label

The bit reorder feature allows the channel order, as it appears in the label, to be assigned independently of the physical order. This feature allows the probe tips for each channel to be physically connected where convenient. The *Reorder* function can then be used to logically rearrange the bits in a label.

To Reorder the Bits in a Label

1. Select the label name button that you want to reorder.
2. Choose *Reorder Bits*.
3. Set the bit order by using one of the following options:
   - To reorder the bits individually, for each channel, enter the number of the bit you want to map the channel to. You can also use the *Spin Buttons* to scroll through the list of bits.
   - To arrange the bits sequentially, select the button at the top of the dialog, then choose *Default Order*.

- To swap the high and low order bytes or words, select the button at the top of the dialog, then choose *Big-Endian/Little-Endian*.

NOTE: Reordered labels can not be used as *range terms* in triggers.

Inserting and Deleting Labels

To Insert Additional Labels

1. Select the *label* name button that you want to insert another label next to.
2. Choose *Insert before...* or *Insert after...*.

To Delete Labels

1. Select the label name button that you want to delete.
2. Choose *Delete*.

Turning Labels On and Off

You may want to turn off *labels* that you have created so that the label is not displayed. When a label is turned off, its name and bit assignments are preserved.

To Turn Off a Label

1. Select the button of the label that you want to turn off.
2. Choose *Label [ON]* to toggle it off.

To Turn On a Label

1. Select the button of the label that you want to turn on.
2. Choose *Label [OFF]* to toggle it on.

To Display a Label that was Off

1. Turn on the label.
2. At the bottom of the window, select the *Apply* button.
   The label's data appears in the display windows.

Label Polarity

The analyzer uses the label polarity to identify patterns and edges when *triggering* on and storing data. The default polarity for all labels is positive (+). Setting the polarity to negative (-) inverts the logic for all bits in a label.

To change the label polarity, select the polarity field, which toggles between positive (+) and negative (-).

NOTE: It is rare that negative logic would be used on a circuit. If the negative logic parameter is applied to a positive logic circuit, unpredictable results will be obtained.

Setting the Pod Threshold

The pod threshold is a voltage level which the data must cross before the analyzer recognizes it as a change in logic levels. You can specify a threshold level for each pod. The level specified for the master pod that includes the clock is also used for the clock threshold.

To Set the Threshold

1. Under the *Format* tab, select the threshold option button.
   The threshold option button is located just below the pod name.
2. Choose one of the threshold options described below.
3. Repeat steps 1 and 2 for the other pods.

NOTE: The clock threshold level is the same as the level assigned in the Pod Threshold field.

TTL

The threshold level is +1.5 volts.

ECL

The threshold level is -1.3 volts.

USER

When USER is selected, the threshold level is selectable from -5.0 volts to +5.0 volts.

Activity Indicators

Activity indicators are located above the column of bit assignment fields. When the logic analyzer is properly connected to an active target system, you see dashes and arrows in the Activity Indicator displays for each channel. An active channel looks like .

A dash at the top of the activity indicator display indicates that the signal connected to that channel is at a logic high. A dash at the bottom indicates that the signal is at a logic low. An arrow indicates that the signal is transitioning.

You can use these indicators to check whether there is proper probe connection: bits that are stuck low may not be properly connected. You can also verify that the signals in your target system are active: bits that are stuck high or low may not be crossing the threshold voltage.

Activity indicators are not displayed during an *acquisition*.

The Trigger Tab

The *Trigger* tab is used to set up a sequence of steps that determine when the analyzer *triggers*. At the top is a control area where you specify how much data to store before and after the trigger. In the trigger sequence area, you specify the conditions that you want the analyzer to trigger on. In the resource *terms* area at the bottom of the window, you define variables for use in the trigger sequence specification.

- "Setting Up a Trigger" on page 55
- "Adding and Deleting Sequence Steps" on page 57
- "Editing Sequence Steps" on page 58
- "Setting Up Loops and Jumps in the Trigger Sequence" on page 58
- "Saving and Recalling Trigger Sequences" on page 59
- "Clearing Part or All of the Trigger" on page 60

See Also

"Overview of the Trigger Sequence" on page 61

"Predefined Trigger Macros" on page 62

"Working with User-Defined Macros" on page 68

"Defining Resource Terms" on page 71

"Trigger Position Control" on page 76

"Oversampling: the Samples/Clock Control" on page 77

"Sample Period (Timing Only)" on page 77

"Arming Control" on page 78

Setting Up a Trigger

When setting up a *trigger sequence*, you typically trigger first on a simple pattern or edge. From that point, you execute an iterative

process of adding or fine-tuning sequence steps until the analyzer consistently triggers at the desired point.

To Set Up a Trigger

1. Define resource terms. (see page 71)
2. Select the first sequence level button; choose *Edit*. (see page 56)
3. Select the *Select new macro* button, and select the most appropriate macro. (see page 56)
4. Edit the sequence step. (see page 58)
5. If necessary, add/edit additional sequence steps. (see page 57)

See Also

"Predefined Trigger Macros" on page 62

"Working with User-Defined Macros" on page 68

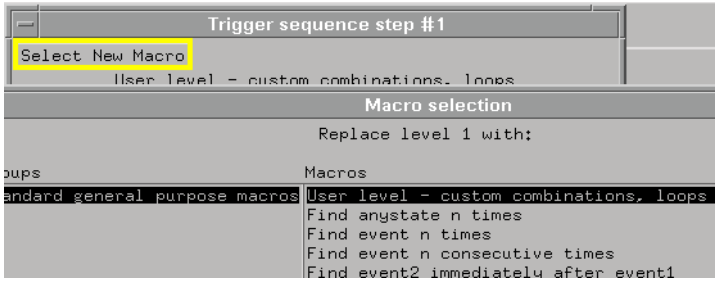
"Overview of the Trigger Sequence" on page 61

Selecting the Edit of a Sequence Step

To modify a Trigger Sequence Step, select the number button of the level that needs to be modified, and choose *Edit*.

Selecting a Macro to Match Trigger Conditions

Review the list of macros and select the one that matches the desired trigger conditions. In most cases one of the predefined macros will satisfy trigger conditions. If none of the predefined macros match choose the *User level* macro.



Selecting a Macro from the List

See Also

"Predefined Trigger Macros" on page 62

Adding and Deleting Sequence Steps

To Add Sequence Steps

1. Select the number button of the level that you want to insert other levels around.
2. Choose *Insert Before* or *Insert After*.
3. Select a macro for the new step.

To Delete Sequence Steps

1. Select the number button of the level that you want to delete.
2. Choose *Delete*.

Trigger Sequence Editing Options

When you select a sequence level number button, you see a selection menu with choices that allow you to modify the sequence. Choose the option you want from the choices below.

- Edit

  Changes the contents of the sequence step. You can change the resource *terms* or other assignment fields, such as durations and occurrences.

- Copy

  Copies the currently selected step. When you copy a level, the new level contains the same macro or user-defined step as the original.

- Replace

  Replaces the currently selected level. When you replace a level, you pick a new macro or user-level, and it then replaces the old one.

- Delete

  Deletes the level that is currently selected.

- Insert Before / Insert After

  Inserts an additional step before or after the selected step.

Editing Sequence Steps

You can modify the contents of a step in the *trigger* sequence by *editing* it.

To Edit a Sequence Step:

1. Select the sequence level number button and choose *Edit...*.
2. To change the macro, select the *Select new macro* button and choose a different macro from the list.
3. Select a resource term name to choose a different resource term. You can also choose negations of *terms* and combinations of terms.
4. Set the values of the other fields, such as durations and occurrence counts.
5. Select the *OK* button to close the sequence step editing dialog.

If you are editing a user-defined step, refer to:

"Setting Pattern Durations" on page 69

"Using Occurrence Counters" on page 70

"Setting Up Loops and Jumps in the Trigger Sequence" on page 58

Setting Up Loops and Jumps in the Trigger Sequence

To set up loops and jumps in your *trigger sequence*, use *Branches*. Branches are available in most of the *Predefined Macros* and User-Defined (see page 68) sequence steps. You may need to break down the macros (see page 67) in order to set up branches.

To Set Up a Branch or Loop

1. Select the *Find* button and choose the term that you want to branch on. The analyzer will take the branch if this term is found.
2. Select the *go to* option button and choose the sequence level that you want to branch to.
3. If you want a secondary branch, select the *Else on* button and choose the term that you want to branch on.
4. Select the *go to* option button and choose the sequence step that you want to branch to.

The analyzer starts at the first trigger sequence level. It evaluates the data coming from the *target system* and when the term of either the *find* or *else on* branch becomes true, follows that branch to the next specified level (or triggers). If you are using oversampling, remember that oversampled states are not evaluated.

Timers and occurrence counters are reset when a branch is taken, even if it loops back to the same level.

If both branches become true at the same time, the primary (*find*) branch is taken. Once the analyzer triggers, it stops evaluating data and simply fills its memory. When the acquisition memory is full, the logic analyzer stops and any analysis or display tools connected to it process the *data set*.

Saving and Recalling Trigger Sequences

You can save a *trigger* setup independently of configuration files within a session. Recalling the stored trigger sequence specification can change the trigger arming, memory depth, and trigger position as well as the trigger sequence and term definitions. The trigger sequence specification will not change the acquisition mode (full channel vs. half channel).

The Agilent Technologies 16517A/18A logic analyzer can hold up to 10 trigger sequence specifications for State or Timing mode, for a total of 20 specifications. When you exit your Agilent Technologies 16700A/B-

series *session*, the trigger sequences are cleared. They can be saved across sessions or be shared among multiple modules as part of a configuration file, however.

To Save a Trigger Sequence

1. Enter a name for the trigger sequence in the Title field just above the trigger sequence area.
2. Select the *Save* button.
3. Choose a memory location to store the trigger sequence in.

To Recall a Trigger Sequence

1. Select the *Recall* button.
2. Choose the trigger sequence that you want.

Clearing Part or All of the Trigger

To Clear Part or All of the Trigger:

1. Select *Clear* from the *menu bar*.
2. Choose the option you want from the choices described below:
   - All

     Clears sequence steps, resource *terms*, and resource term names back to their default values.
   - Sequence Levels

     Resets the *trigger* sequence to a default sequence.
   - Resource Terms

     Resets all resource term assignment fields, including *labels* and values back to their default values.
   - Resource Term Names

     Resets all the resource term names to their default values.

- Store/Recall Memories

  Deletes all saved trigger sequence specifications.

Overview of the Trigger Sequence

The *trigger* sequence is a sequence of steps that control the path that the analyzer takes to find the trigger point. The path taken resembles a flow chart, with each step in the sequence being an opportunity to direct the analyzer's selection process in the desired direction. You can edit the overall trigger sequence by adding or deleting sequence steps (see page 57).

Each step in the sequence is either a predefined macro (see page 62) or a user-defined step (see page 68). Both the predefined macros and user-defined steps contain variables that you define. The variables, called resource terms (see page 71) represent edges and bit patterns in the data.

When you run the analyzer, it searches for a match between the resource term values and the measurement data. When a match is found, the sequencing continues to the next step, loops back (see page 58) to a previous step, or jumps ahead (see page 58) to another step. Eventually, a path of "true" steps leads to the trigger point.

Each macro uses one or more of the analyzer's internal sequence levels (see page 61). Each user-defined step uses one internal sequence level.

See Also

"Setting Up a Trigger" on page 55

How the Internal Sequence Levels Are Used

The analyzer has four internal sequence levels that it uses to make up the *trigger sequence*. The levels are used differently, depending on whether you use predefined trigger macros or user-defined steps to construct your trigger sequence.

When you use user-defined steps (see page 68), all of the internal sequence levels are available. Each user-defined sequence level corresponds to one internal level.

When you use predefined trigger macros (see page 62), more than one of the internal sequence levels may be required for a single macro. The exact number of internal levels required per macro is shown in the macro library list. Even though some trigger macros use multiple sequence levels, macros are easier to use, and they are the most efficient way to construct a trigger sequence.

Predefined Trigger Macros

Trigger macros provide a simple way to set up the analyzer to *trigger* on common events and conditions. A library of macros is available for both the *state* and *timing* acquisition modes.

NOTE: Each macro requires at least one internal sequence level (see page 61), and in some cases, may require multiple levels. The number of levels used by each macro is described in the macro library.

Timing Trigger Macro Library

"Timing User Mode" on page 63
"Basic Timing Macros" on page 63
"Pattern/Edge Combinations" on page 64
"Setup and Hold Violations" on page 64
"Time Violations" on page 65

State Trigger Macro Library

"State User Mode" on page 65
"Basic State Macros" on page 66
"Sequence-Dependent Macros" on page 66
"Time Violations" on page 67

See Also

"Setting Up a Trigger" on page 55
"Defining Resource Terms" on page 71
"Breaking Down and Restoring Macros" on page 67

Timing User Mode

(user-defined macro)

The User level allows you to create a custom *trigger sequence* using a *trigger term*, comparison function, and a jump or loop. This macro uses one internal sequence level.

See Also

Working with User-Defined Macros (see page 68) for more information on the user-defined mode.

Basic Timing Macros

The following basic macros are found in the *Macro selection* window when the analyzer is in the *Timing* mode. Each macro uses one internal sequence level.

- Find anystate n times.

  This macro becomes true with the nth state it sees. It uses one internal sequence level.

- Find pattern present/absent for > duration.

  This macro becomes true when it finds a pattern you have designated that has been present or absent for greater than or equal to the set duration. It uses one internal sequence level.

- Find pattern present/absent for < duration.

  This macro becomes true when it finds a pattern you have designated that has been present or absent for less than the set duration. It uses one internal sequence level.

- Find edge.

  This macro becomes true when the edge you have designated is seen. It uses one internal sequence level.

- Find Nth occurrence of pattern present/absent for > duration

  This macro becomes true when it finds the Nth occurrence of a pattern that has been present or absent for greater than or equal to the set duration. Each valid pattern occurrence is only counted once, even though the same occurrence may be valid for multiple sample clocks. It uses one internal sequence level.

- Find nth occurrence of an edge.

  This macro becomes true when it finds the designated occurrence of an edge you have selected. Note that the 500-MHz trigger sequencer may not count edges that occur closer than 2 ns. This macro uses one internal sequence level.

Pattern/Edge Combinations

The following macros are found in the *Macro selection* window when the analyzer is in the *Timing* mode. These predefined macros use a pattern, edge, or a combination of both as the *trigger* element. These macros use either one or two internal sequence levels.

- Find edge within a valid pattern.

  This macro becomes true when a selected edge type is seen within the time window defined by a pattern you have designated. It uses one internal sequence level.

- Find pattern occurring too soon after edge.

  This macro becomes true when a pattern you have designated is seen occurring within a set duration after a selected edge type is seen. It uses two internal sequence levels.

- Find pattern occurring too late after edge.

  This macro becomes true when one edge type you have selected occurs, and for a designated period after that first edge is seen, a pattern is not seen. It uses two internal sequence levels.

Setup and Hold Violations

- Find setup or hold violation

  This macro becomes true when there is a change on a selected group of channels which violates the specified setup or hold time criteria with respect to a selected edge. It uses two internal sequence levels.

- Find setup or hold violation clocked by an edge within a valid pattern

  This macro becomes true when either a designated setup or a hold criterion is violated on a selected group of channels which are being clocked by a selected edge which appears within a designated pattern. It

uses two internal sequence levels.

- Find setup or hold violation clocked by a pattern/pulse

  This macro becomes true when there is a change on a selected group of channels which violates the specified setup or hold time criteria with respect to a selected pattern. It uses three internal sequence levels.

Time Violations

The following macros are found in the *Macro selection* window when the analyzer is in the *Timing* mode. These predefined macros are specifically tailored to *trigger* on events occurring out of a predefined time range. These macros use either one or two internal sequence levels.

- Find 2 edges too close together.

  This macro becomes true when a second selected edge is seen occurring within a period you have designated after the occurrence of a first selected edge. It uses two internal sequence levels.

- Find 2 edges too far apart.

  This macro becomes true when a second selected edge occurs beyond a period you have designated after the first selected edge. It uses two internal sequence levels.

- Find width violations on a pattern/pulse.

  This macro becomes true when the width of a pattern violates minimum and maximum width settings you have designated. It uses one internal sequence level.

- Wait t sec.

  This macro becomes true after a period you have designated has expired. It uses one internal sequence level.

State User Mode

(User-defined macro)

The User level allows you to create a custom *trigger sequence* using a *trigger* term, comparison function, and a jump or loop. This macro

uses one internal sequence level.

See Also

Working with User-Defined Macros (see page 68) for more information on the user-defined mode.

Basic State Macros

The following basic macros are found in the *Macro selection* window when the analyzer is in the *State* mode. Each macro uses one internal sequence level.

- Find anystate n times.

  This macro becomes true with the nth state it sees. It uses one internal sequence level.

- Find pattern n times.

  This macro becomes true when it sees an event you have designated occurring a designated number of times. The events may occur consecutively, but do not have to. It uses one internal sequence level.

- Find pattern n consecutive times.

  This macro becomes true when it sees an event you have designated occurring a designated number of consecutive times. It uses one internal sequence level.

- Find pattern2 immediately after pattern1.

  This macro becomes true when the first event you have designated is seen immediately followed by a second designated event. It uses two internal sequence levels.

Sequence-Dependent Macros

The following basic macros are found in the *Macro selection* window when the analyzer is in the *State* mode. These macros each *trigger* on a particular sequence of events.

- Find patt2 n times after patt1, before patt3 occurs.

  This macro becomes true when it first finds a designated patt1, followed by a selected number of occurrences of a designated patt2. In addition, if a designated patt3 is seen anytime while the sequence is not yet true, the

sequence starts over. If patt2's nth occurrence is coincident with patt3, the sequence starts over. It uses two internal sequence levels.

- Find too few states between pattern1 and pattern2.

  This macro becomes true when a designated pattern1 is seen, followed by a designated pattern2, and with fewer than a selected number of states occurring between the two events. It uses two internal sequence levels.

- Find too many states between pattern1 and pattern2.

  This macro becomes true when a designated pattern1 is seen, followed by more than a selected number of states, before a designated pattern2. It uses two internal sequence levels.

Time Violations

The following macros are found in the *Macro selection* window when the analyzer is in the *State* mode. These predefined macros are specifically tailored to *trigger* on events occurring out of a predefined time range. These macros use either one or two internal sequence levels.

- Find pattern2 occurring too soon after pattern1.

  This macro becomes true when a designated pattern1 is seen, followed by a designated pattern2, and with less than a selected period occurring between the two events. It uses two internal sequence levels.

- Find pattern2 occurring too late after pattern1.

  This macro becomes true when a designated pattern1 is seen, followed by at least a selected period, before a designated pattern2 occurs. It uses two internal sequence levels.

- Wait n external clock states.

  This macro becomes true after a number of user clock states you have designated have occurred. It uses one internal sequence level.

Breaking Down and Restoring Macros

When you break down a macro, you gain access to all the resource assignment fields and branching options. You can change these fields to change the *trigger* structure. You might need to do this to create a

custom *trigger sequence* or to create loops and jumps.

To Break Down Macros

1. Select *Modify* in the Trigger window *menu bar*.
2. Choose *Break down macros*.

The contents of broken down macros are displayed in the long form used in a user-defined sequence step. If the macro uses two of the analyzer's internal sequence levels, (see page 61) both levels are separated out and displayed in the trigger sequence area of the Trigger window.

To Restore Macros

1. Select *Modify* in the Trigger window menu bar.
2. Choose *Restore macros*.

Use *Restore macros* to restore all macros to their original structure. Note that when the macros are restored, all changes are lost and any branching that is part of the original structure is restored.

See Also

Working with User-Defined Macros (see page 68) for information on working with macros that are broken down.

Working with User-Defined Macros

NOTE:

Before you begin to set up user-defined sequence steps, note that in most cases one of the predefined trigger macros (see page 62) will work.

You might need to set up a user-defined sequence step to accommodate a condition not covered by the macros, or if you need to set up loops and jumps in the sequence. Each user-defined sequence step has a "fill-in-the-blanks" type statement. You use resource *terms* to fill in the statement with the appropriate values.

To Set Up a User-Defined Macro

1. Select a sequence level number button; choose *Edit*.

2. Select the *Select New Macro* button; select the *User level* macro.
3. Select either *Occurrence Counter* or *Timer*.
4. If you selected *Occurrence Counter*, assign it to the *Find* or *else on* branch.
   The occurrence counter can only be used in one branch at a time.
5. Select a resource term for the Find branch.
6. Set the occurrence field or timer.
7. Select the goto *level* button; choose a sequence level or *Trigger*.
8. Optional - select a resource term for the else on branch.
9. Select the *goto level* button; choose a sequence level or *Trigger*.
10. If you used any pattern terms, set the duration filter.

In the Agilent Technologies 16517A/18A logic analyzer, it is possible to have more than one trigger condition. The logic analyzer does not do any check on the trigger sequence; before running, make sure that the sequence will eventually trigger.

For more information on the functions available in a user-defined step, refer to:

- “Defining Resource Terms” on page 71
- “Setting Pattern Durations” on page 69
- “Using Occurrence Counters” on page 70
- “Using Timer Terms” on page 73
- “Limits on the Number of Simultaneous Terms” on page 74
- “Setting Up Loops and Jumps in the Trigger Sequence” on page 58

Setting Pattern Durations

When a bit pattern is found during a *trigger sequence*, you can influence when the term actually becomes "true" by assigning a time duration. The Set Pattern Duration statement is available in all sequence levels, but you may have to break down (see page 67) any macros.

Present/Absent field

This field determines whether the resource term becomes true after being present or absent for the specified time interval.

</> Field

This field allows you to choose whether the pattern duration is longer or shorter than the specified time. When < is selected, the resource is true for one clock cycle after a pattern is present/absent for less than the duration value.

When > is selected, additional control over pattern duration becomes available. This control allows you to pick what happens after the pattern has been present/absent for more than the duration value.

Until exit, Upon exit, Upon entry Field

This part of the Set Pattern Duration statement dictates how long the resource is true after the pattern duration is met. The Until Exit selection keeps the resource true from the time the duration is met, until the end of the pattern. The Upon Entry selection makes the resource true for only one clock cycle after the duration is met. The Upon Exit selection makes the statement true for one clock cycle after the end of the pattern.

Using Occurrence Counters

The occurrence counter is assigned to either the "Find" branch or the "else on" branch. Whatever positive number you assign to the counter, the pattern must be seen that number of times before the term becomes true.

Note that in order to count occurrences of a pattern with a > duration, the *Upon Entry* or *Upon Exit* selections should be used. In the *Until Exit* mode, the occurrence counter is clocked at a 2 ns rate as long as the resource is true, in effect acting like a duration counter.

If the "else on" branch becomes true before all specified occurrences of the primary branch, the secondary "else on" branch is taken.

If you choose to use the occurrence counter in a sequence level, the timer is not available in that level.

Defining Resource Terms

Resource terms are variables that you can use in defining a trigger sequence. The terms available include bit patterns, edges, counters, and timers.

To Define a Resource Term

1. Select the appropriate tab to bring the group of terms forward.
2. Optional - Select the term name and enter a new name.
3. Optional - Select the label button to the right of the term name and choose *Replace*.
4. Select the *label* that you want to use.
5. Optional - add more labels (see page 75) to the term.
6. Optional - Set the numeric base.
7. Select the term value field to the right of the label name.
8. For pattern terms, enter the term value. For edge terms, assign edges to appropriate bits.

See Also

"Using Bit Pattern Terms" on page 71

"Using Edge Terms (Timing Only)" on page 72

"Using Combinations of Terms" on page 73

"Using Timer Terms" on page 73

"Using Occurrence Counters" on page 70

"Adding and Deleting Labels for Terms" on page 75

"Numeric Base" on page 75

"Limits on the Number of Simultaneous Terms" on page 74

Using Bit Pattern Terms

Bit pattern resource *terms* can be set to match a numeric value or bit pattern on a group of data channels such as a bus. In order for a pattern

to be found by the analyzer, the input data must match all bits of the pattern that are not defined as *Don't Cares (X)*. Bit patterns can also be used in a negated form.

To Define a Pattern Term

1. Select the *label* name to the right of the term name and choose the label that you want to use.
2. If necessary, add more labels (see page 75) to the term.
3. Select the term value field, to the right of the label name.
4. Enter the pattern for each label.
   Depending on the base setting, such as hex or octal, some characters will not be accepted. Don't cares are indicated by an *X*.



Right-click on any of the bit pattern value fields to quickly assign the pattern term to a preset value. *Clear (=X)* sets the value to all *X* (don't cares). *Set (=1)* sets the value to all 1s. *Reset (=0)* sets the value to all 0s.

Using Edge Terms (Timing Only)

Edge *terms* are only available in the *timing* acquisition mode. You can set an edge term to match transitions on one or more channels. When you specify an edge on more than one channel, the analyzer logically *ORs* the edges together. When the analyzer sees a transition that matches any of the ones specified in the edge term, the term becomes true.

The following edge choices are available for each bit:

Rising edge (↑)

Falling edge (↓)

Either rising or falling

To Define an Edge Term

1. Select the label name button and choose the *label* that you want to use.
2. Select the edge value button, to the right of the label name.
3. Set the edge for each channel.
   Don't cares are indicated by a (.).

NOTE: After you close the edge term assignment dialog, you may see *$* indicators in the term value field. This symbol indicates that the value can't be displayed in the selected base. Choose *Binary* base to see the actual assignments.

Using Timer Terms

Timer options are selected from within sequence levels. The timer term is evaluated as either true or false. A true timer term lets the sequence evaluation continue within the current statement. Because the timer works exclusively for each level it is assigned to, it will not affect any other level or the trigger. However, each sequence level can have its own timer with different values assigned. The timer term options are used in the following ways:

Timer expired

The timer term becomes true when its assigned time expires.

Timer not expired

The timer term is true while the assigned time has not expired. Once expired, the timer term becomes false. This is like negating the timer.

The timer is started as the sequence level is entered. If you return to the same sequence level again, the timer will be restarted. If you choose to use the timer in a sequence level, the occurrence counter is not available in that level.

Using Combinations of Terms

Resource *terms* can be used in combinations. Combinations use the logical AND and OR functions to combine predefined resource terms.

To Set Up a Combination

1. Edit the sequence step (see page 58) that you want a combination term to appear in.
2. Select the resource term assignment field that you want to make into a combination.
3. Choose *Combo...* from the menu of resource terms.
4. In the Combination dialog, select the *On/Off/Not* option button for each term that you want to use and choose *On* or *Not*.
   Not selects the logical negation of the term.
5. Select the logical operator option buttons and choose a logic operation.

NOTE: The combination of terms is now inserted into the term assignment field. If the term is too long to fit, the display is truncated.

Limits on the Number of Simultaneous Terms

The analyzer has four internal resources that it uses in trigger sequences. The four resources can be allocated across the four bit patterns and the two edges that are available in timing mode. Since there is a limit of four resources, you can not use all four bit patterns and both edges in a trigger sequence at the same time. It is possible to have all six resource term names assigned to different pattern and edge values; however, you will only be allowed to insert up to four of them in the trigger sequence.

Once you assign the four resources with both a resource term name and a value, the resources can be used either by themselves or in combination with each other.

The actual resource terms available for any particular assignment field varies depending on the acquisition mode, whether the timer or occurrence counter is used, and how many resource terms have already been used in other levels.

Pattern Durations

Pattern duration is assigned per sequence level. This means that you can assign a different duration to the same term name in a different

sequence level. However, each time you use that same term with different duration values, you use up another resource. In other words, using one resource term four times with different pattern durations is equivalent to using all four resource terms once with the same duration.

Timer/Counters

There is one timer or counter available per sequence level. If you choose to use a timer in one sequence step, the occurrence counter is not available in that same step. On a per-sequence level basis, the timer/counter is either a timer or a counter, but not both.

Adding and Deleting Labels for Terms

Labels are defined in Format, after which they are available throughout the other analyzer areas and attached display tools.

When you use more than one label to define a trigger *term*, the term conditions must be true on both labels for the term to be true.

To Add a Label to a Resource Term

1. Select the label name button.
2. Choose *Insert*.
3. Choose the label that you want to add to the resource term.

To Delete a Label from a Resource Term

1. Select the label name button.
2. Choose *Delete*.

Numeric Base

All *labels* have a numeric base field next to them. The base choices are Binary, Octal, Decimal, Hex, ASCII, Symbol and Twos Complement.

To Change the Numeric Base

1. Select the base option button.

2. Choose the base that you want.

NOTE: If the numeric base is changed in one window, the base in other windows may not change accordingly. For example, the base assigned to symbols is unique, as is the base assigned in the Listing window.

Trigger Position Control

The Trigger Position control determines how much data is stored before and after the *trigger* occurs for all subsequent *acquisitions*. The point where the trigger occurred is placed at a specified position relative to the data in memory.

When a Run is started, the analyzer will not look for a trigger until at least the proper percentage of pretrigger data has been stored. After a trigger has been detected, the specified percentage of posttrigger data is stored before the analyzer halts.

The trigger position choices are Start, Center, End, User Defined, or Delay.

- Start

  The trigger position is set at the starting point of available memory. This process results in maximum posttrigger data and minimum pretrigger data. Note that there will be a small amount of pretrigger data stored.

- Center

  The trigger position is set at the center point of available memory. This results in half pretrigger data and half posttrigger data.

- End

  The trigger position is set at the end point of available memory. This results in maximum pretrigger data and minimum posttrigger data. Note that there will be a small amount of posttrigger data stored.

- User Defined

  When the trigger position is set to User Defined, a trigger position slider appears. Use this slider to set the trigger position any where between 1%

and 99%. As the slider is adjusted, the *% poststore* indicator shows the amount of data that will be stored after the trigger point.

- Delay

  This option delays the start of data storage until some time after the trigger. The range of the delay time is affected by the sample period, but the absolute minimum is 0 seconds, and the absolute maximum is 64 seconds.

Sample Period (Timing Only)

Sample Period is used to set the time period between data samples. Every time a new sample is taken, the analyzer will see updated measurement data. The choices available for sample period depend on the *acquisition* mode, which is set in the Format window.

In *full channel* mode, the Sample Period is configurable. A list of choices appear when the button is selected. In *half channel* mode, the analyzer will automatically run at a 250 ps sample period, and is not adjustable.

Oversampling: the Samples/Clock Control

(state only)

The Samples/Clock control sets the number of extra sample points (oversampling) between the edges of the external state clock. All oversampled points are evenly distributed within the period of the external state clock. Oversampling is set in powers of two, up to a maximum of 32, or a sample rate not exceeding 2 GSa/s. The equation below shows how the maximum sample rate is limited:

(external clock rate) x (oversampled setting) = 2 GHz

The external clock rate is checked at each run. If at any time the rate of oversampling exceeds a 2 GHz rate, a lower Samples/Clock setting is automatically chosen. If you notice some of the higher oversampling rate choices are not available, this means those choices would have

exceeded the 2 GHz sampling rate when multiplied by the external clock rate. The interface will not make those choices available to you until the external clock rate decreases.

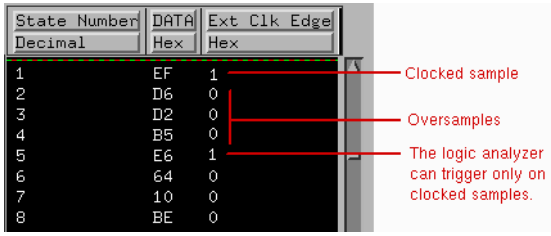
Viewing Oversampled Data

Once Samples/Clock is set to a value greater than 1, an additional label called *Ext Clk Edge* becomes available in the display tools. A 1 (one, or logic high) on label *Ext Clk Edge* indicates states that were clocked by the external state clock signal. A 0 (zero, or logic low) indicates states that were oversampled.

NOTE:

The analyzer *WILL NOT TRIGGER ON POINTS OF OVERSAMPLING*. The only points where the analyzer will trigger is on external clock transitions. If triggering is needed on these oversampled points, and the state clock is <500 MHz, convert from state mode to timing mode where all triggering is at 500 MHz.

The figure below shows data at the external clock transitions, data at the points of oversampling, and the data points where the analyzer could trigger.



Arming Control

An instrument must be *armed* before it can look for its *trigger*. When you *Run* an instrument, it is armed immediately. When you are using more than one instrument, you can use one instrument to arm another. The logic analyzer can be armed by the *Run* button, a *Group Run* through the *Intermodule Bus* (IMB), or by an external signal through the *SMB Port* connector on the back panel.

To Arm the Analyzer Using *Run* or the *SMB Port*:

1. Select *Arm* in the menu bar.
2. Choose the option that you want.

To Arm the Analyzer Using the Intermodule Bus

Refer to Overview - Multiple Instrument Configuration (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume).

Run/Group Run

Set the *Arm* control to *Run* if you are using just a single analyzer to make a measurement. When you select the Run/Stop button, you start the trigger sequence immediately.

SMB Port

Set the *Arm* control to *SMB Port* if you want to arm the analyzer with signal from an outside source. An arming signal can come from another module in the *frame*, such as an oscilloscope TRIG OUT signal. However, the only way the measurement can be time-correlated is if both the sending and receiving modules are configured as a Group Run in the Intermodule window. Generally, the SMB Port is used when you want an external trigger source to arm a module, or group of modules in the frame, or you want to take advantage of the faster arming path.

See Also

Overview - Multiple Instrument Configuration (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

Specifications and Characteristics

NOTE:

Definition of Terms

To understand the difference between specifications (see page 83) and characteristics (see page 83), and what gets a calibration procedure (see page 84) and what gets a function test (see page 84), refer to appropriate links within this note.

"Agilent Technologies 16517A/18A Logic Analyzer Specifications" on page 80

"Agilent Technologies 16517A/18A Logic Analyzer Characteristics" on page 81

Agilent Technologies 16517A/18A Logic Analyzer Specifications

The specifications are the performance standards against which the product is tested.

These specifications apply only to the Agilent Technologies 16517A/18A 4GHz Timing/1GHz State logic analyzer:

```
Minimum Input Voltage Swing:          500 mV peak to peak
Threshold Accuracy:                   +/-2% of input signal +/-50 mV
Minimum External Clock Period:        1 ns
Setup/Hold Time:
  Per Pod:**                          350 ps/350 ps
  Across Pods:                        750 ps/750 ps
                                      350 ps/350 ps, with manual deskew

* Specified for an input signal VH=-0.9 V, VL=-1.7 V, threshold=-1.3 V,
  slew rate=1 V/ns

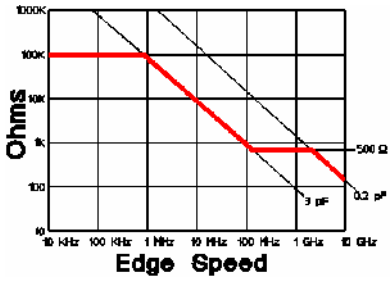
** For the frequency range of 62.5 MHz to 20 MHz, a duty cycle of
   40% to 60% is required.
```

Agilent Technologies 16517A/18A Logic Analyzer Characteristics

The characteristics are not specifications, but are included as additional information.

```
General information
  - Channel Counts:
    16517A                16
    with 1 16518A         32
    with 2 16518As        48
    with 3 16518As        64
    with 4 16518As        80
  - Memory Depth:
    Half Channel             131072 samples per channel
    Full Channel              65536 samples per channel

Probes
  - Input DC Resistance:          100 Kohm, +/- 2%
  - Input Capacitance:            0.2 pF and then through 500 ohms, 3 pF
  - Input Impedance:
      DC through 400 ns rise time, 100 Kohm, typical
      3.5 ns through 350 ps, 500 ohm, typical
```



Impedance Curve

```
  - Maximum Voltage:              +/- 40 V peak CAT I
  - Threshold Range:              +/- 5.0 V, adjustable in 10-mV increments
  - Threshold Setting:            Preset TTL, ECL, or User-Defined on a
                                  per-pod basis.
  - Input Dynamic Range:          +/- 5 V about the threshold
  - Minimum Input Overdrive:      250 mV or 30% of input, whichever is
                                  greater, above the pod threshold.

State Analysis
  - Maximum External Clock Speed:      1 GHz, requires a periodic clock
  - Minimum State Speed:               20 MSa/s, requires a periodic
                                       clock
  - State Clocks:                      One external clock is available
                                       on the master card.  No clocks
                                       are available on the expander
                                       card.  Clock edge is selectable
                                       as rising or falling.
```

```
  - State Clock Duty Cycle Range:
       1 GHz through 500 MHz:          45% - 55%, typical
       500 MHz through 250 MHz:        30% - 70%, typical
       250 MHz through 20 MHz:         20% - 80%, typical
  - Channel-to-Channel Skew:
       Per Pod:                        250 ps, typical
       Across Pods:                    1 ns, typical
                                       250 ps with manual deskew
  - Minimum Detectable Pulse Width:    900 ps
  - Oversampling:                      2x, 4x, 8x, 16x, and 32x, at a rate
                                       up to 2 GSa/s.
  - Maximum Delay after Triggering:
                               (2 to the 20th)x(sample period), or
                               16.78 ms at or below 16 ns sample
                               period.

Note: When oversampling, use oversampled period for period above.


Timing Analysis
  - Timing Modes:              Conventional timing
  - Timing Speed:
      Full Channel             15.3 KSa/s - 2 GSa/s
      Half Channel             4 GSa/s
  - Sample Period:
      Full Channel             500 ps minimum, 65.536 us maximum
      Half Channel             250 ps
  - Sample Period Accuracy:    0.005% of sample period
  - Minimum Detectable Pulse Width:
      4 GSa/s                  800 ps, typical
      2 GSa/s or less          1.1 ns, typical
  - Time Covered by Data:
      Minimum (2 or 4 GSa/s)   32.8 us
      Maximum (15.3 KSa/s)     4.3 s
  - Time Interval Accuracy:    +/-( sample period + channel-to-
                               channel skew + 0.005% of time
                               interval reading )
  - Channel-to-Channel Skew:   250 ps, typical
  - Maximum Delay after Triggering:
                               (2 to the 20th)x(sample period), or
                               16.78 ms at or below 16 ns sample
                               period.


Triggering
  - State Sequence Levels:           4 plus trigger point
  - Timing Sequence Levels:          4 plus trigger point
  - Maximum Sequencer Speed:         500 MHz
  - Maximum Occurrence Count Value:  16,777,216
  - Pattern Recognizers:             4. Each pattern recognizer is the
                                     AND combination of bit (0,1,X) patterns.
  - Pattern Width:                   16/32/48/64/80 channels.
  - Minimum Pattern Recognizer
    Pulse Width:                     2.25 ns
  - Edge Recognizers (Timing Only):  2. Recognize rising, falling, or
                                     either edge on any channel.  Edges are
                                     OR'd across all channels.
  - Edge Width:                      16/32/48/64/80 channels.
  - Edge Counting Frequency:         444 MHz
  - Edge Detection:                  Up to 1 GHz.
  - Greater Than Duration:           0 ns - 510 ns.  Accuracy is +/- 2.25 ns.
  - Less Than Duration:              4 ns - 510 ns.  Accuracy is +/- 2.25 ns.
  - Resource Terms per Trigger       4. Each edge recognizer and pattern
    Specification                    recognizer of a specified duration count
                                     as one resource term.
  - Branching:                       Each sequence level has two branching
                                     qualifiers.  When the qualifier is
                                     satisfied, the analyzer will branch to
                                     the specified sequence level.
  - Timer/Counter:                   1 timer or counter per sequence level.
                                     Restarted upon each entry to each level.
  - Timer/Counter Range:
```

```
      Timing mode                     0 s to 33 ms
      State mode, 500 MHz to 1 GHz    (ext. clock period)x(2 to the 23rd)
      State mode, less than 500 MHz   (ext. clock period)x(2 to the 24th)
  - Timer Resolution:
      Timing mode                     2 ns
      State mode, above 500 MHz       2x(ext. clock period)
      State mode, below 500 MHz       ext. clock period
  - Timer Accuracy:                   0.005% of timer value

Power Requirements
  All power supplies required for operating the logic analyzer are
  supplied through the backplane connector in the mainframe.

Operating Environment Characteristics
  - Indoor use only.
  - Temperature
    Instrument:                           0 to 55 degrees C (+32
                                          to 131 degrees F)
    Probe lead sets and cables:           0 to 65 degrees C (+32 to
                                          149 degrees F)
  - Humidity
    Instrument, probe lead sets,
    and cables:                           up to 95% relative humidity at
                                          40 degrees C (+104 degrees F)
  - Altitude                              To 4600 m (15,000 ft)
  - Vibration
    Operating:     Random vibration 5-500 Hz, 10 minutes per axis,
                   approximately 0.3 g rms
    Non-operating: Random vibration 5 to 500 Hz, 10 minutes per axis,
                   approximately 2.41 g rms; and swept sine resonant
                   search, 5 to 500 Hz, 0.75 g (0-peak), 5-minute
                   resonant dwell at 4 resonances per axis.
```

What is a Specification

A *Specification* is a numeric value, or range of values, that bounds the performance of a product parameter. The product warranty covers the performance of parameters described by specifications. Products shipped from the factory meet all specifications. Additionally, the products sent to Agilent Technologies Customer Service Centers for calibration and returned to the customer meet all specifications.

Specifications are verified by *Calibration Procedures*.

What is a Characteristic

Characteristics describe product performance that is useful in the application of the product, but that is not covered by the product warranty. Characteristics describe performance that is typical of the majority of a given product, but not subject to the same rigor associated with specifications.

Characteristics are verified by *Function Tests*.

What is a Calibration Procedure

Calibration procedures verify that products or systems operate within the specifications. Parameters covered by specifications have a corresponding calibration procedure. Calibration procedures include both performance tests and system verification procedure. Calibration procedures are traceable and must specify adequate calibration standards.

Calibration procedures verify products meet the specifications by comparing measured parameters against a pass-fail limit. The pass-fail limit is the specification less any required guardband.

The term "calibration" refers to the process of measuring parameters and referencing the measurement to a calibration standard rather than the process of adjusting products for optimal performance, which is referred to as an "operational accuracy calibration".

What is a Function Test

Function tests are quick tests designed to verify basic operation of a product. Function tests include operator's checks and operation verification procedures. An operator's check is normally a fast test used to verify basic operation of a product. An operation verification procedure verifies some, but not all, specifications, and often at a lower confidence level than a calibration procedure.

Using Symbols

You can use symbol names in place of data values when:

- Setting up triggers
- Displaying captured data
- Searching for patterns in Listing displays
- Setting up pattern filters
- Setting up ranges in the System Performance Analyzer

Symbol names can be: variable names, procedure names, function names, source file line numbers, etc.

You can load symbol name definitions into the logic analyzer from a program's object file or from a general-purpose ASCII format symbol file, or you can define symbol names in the logic analyzer.

- "To load object file symbols" on page 86
- "To adjust symbol values for relocated code" on page 87
- "To create user-defined symbols" on page 88
- "To enter symbolic label values" on page 89
- "To create an ASCII symbol file" on page 90
- "To create a readers.ini file" on page 90

See Also

To go to a pattern in the Listing (see the *Listing Display Tool* help volume)

To modify the Source Viewer trace setup (see the *Listing Display Tool* help volume)

To define System Performance Analyzer state interval ranges (see the *System Performance Analyzer* help volume)

To load object file symbols

Object files are created by your compiler/linker or other software development tools.

1. Generate an object file with symbolic information using your software development tools.
2. If your language tools cannot generate object file formats that are supported by the logic analyzer, create an ASCII symbol file (see page 90).
3. Select the *Symbol* tab and then the *Object File* tab.
4. Select the label name you want to load object file symbols for.

   In most cases you will select the label representing the address bus of the processor you are analyzing.
5. Specify the directory to contain the symbol database file (*.ns*) in the field under, *Create Symbol File (.ns) in This Directory*. Select *Browse...* if you wish to find an existing directory name.
6. In the *Load This Object/Symbol File For Label* field, enter the object file name containing the symbols. Select *Browse...* to find the object file and select *Load* in the Browser dialog.

   If your logic analyzer is NFS mounted to a network, you can select object files from other servers.
7. If your program relocates code, see “To adjust symbol values for relocated code” on page 87.

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

To reload object file symbols

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

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

To delete object file symbol files

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

See Also

"Symbol File Formats" on page 98

To adjust symbol values for relocated code

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

To adjust symbol values for a single section of code

1. Select the *Symbol* tab and then the *Object File* tab.
2. In the *Object Files with Symbols Loaded For Label* list, select the file whose symbols you wish to relocate.
3. Select the *Relocate Sections...* button.
4. In the *Section Relocation* dialog, select the field you wish to edit in the section list.
5. Enter the new value for that field and press Enter on your keyboard.
6. Repeat steps 4 through 6 above for any other sections to be relocated.
7. Select *Close*.

To adjust all symbol values

1. Select the *Symbol* tab and then the *Object File* tab.
2. In the *Object Files with Symbols Loaded For Label* list, select the file whose symbols you wish to relocate.

3. Select the *Relocate Sections...* button.
4. Enter the desired offset in the *Offset all sections by* field. The offset is applied from the linked address or segment.
5. Select *Apply Offset*.
6. Select *Close*.

To create user-defined symbols

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

To replace user-defined symbols

1. Under the *Symbol* tab, select the *User Defined* tab.
2. Select the label you want to replace symbols for.
3. Select the symbol to replace.
4. At the bottom of the *User Defined* tab, modify the symbol name, numeric base, Pattern/Range type, and value, as desired.
5. Select the *Replace* button.
6. Repeat steps 3 through 5 to replace other symbols, if desired.

To delete user-defined symbols

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

To load user-defined symbols

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

1. In the menu bar of your analyzer window, select *File* and then *Load Configuration...*.
2. In the Load Configuration dialog, select the directory and filename to be loaded.
3. Select the target of the load operation.
4. Select *Load*.

   User-defined symbols that were resident in the logic analyzer when the configuration was saved are now loaded and ready to use.

To enter symbolic label values

When entering label values in the Pattern or Range subtabs of the Trigger tab:

1. Choose the *Symbols* or *Line #s* number base.
2. Select the *Absolute XXXX* button.
3. In the *Symbol Selector* dialog, select the symbol you want to use. All of your symbols for the current label, regardless of type, will be available in the dialog.

   - Use the Search Pattern (see page 97) field to filter the list of symbols

by name. You can use the Recall button to recall a desired Search Pattern.

- Use the Find Symbols of Type selections to filter the symbols by type.

4. Select the symbol you want to use from the list of *Matching Symbols*.
5. If you are using object file symbols, you may need to:
   - Set *Offset By* (see page 97) to compensate for microprocessor prefetches.
   - Set Align to x Byte (see page 98) to trigger on odd-byte boundaries.
6. Select the Beginning, End, or Range of the symbol.
7. Select the *OK* button.

   The name of your symbol now appears as the value of the label.
8. Select the *Cancel* button to exit the *Symbol Selector* dialog without selecting a symbol.

See Also "Symbols Selector Dialog" on page 96

To create an ASCII symbol file

General-purpose ASCII symbol files are created with text editing/processing tools.

See Also "General-Purpose ASCII (GPA) Symbol File Format" on page 99

To create a readers.ini file

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

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

Reader options

C++Demangle

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

C++DemOptions

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

MaxSymbolWidth

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

OutSectionSymbolValid

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

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

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

ReadElfSection

```
2= Process all globals from ELF section (Default)
   Get size information of local variables
1= Get size information of global and local variables
   Symbols for functions will not be read, and
   only supplemental information for those symbols in
   the Dwarf or stabs section will be read.
0= Do not read the Elf Section
```

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

StabsType

```
 StabsType=0  Reader will determine stabs type (Default)
 StabsType=1  Older style stabs
              (Older style stabs have individual symbol
              tables for each file that was linked into
              the target executable, the indexes of each
```

```
              symbol table restart at 0 for each file.)
 StabsType=2  Newer style stabs
              (New style stabs have a single symbol table
              where all symbols are merged into a large
              symbol array).
```

ReadOnlyTicoffPage

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

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

AppendTicoffPage

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

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

Examples

Example for Elf/Stabs

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

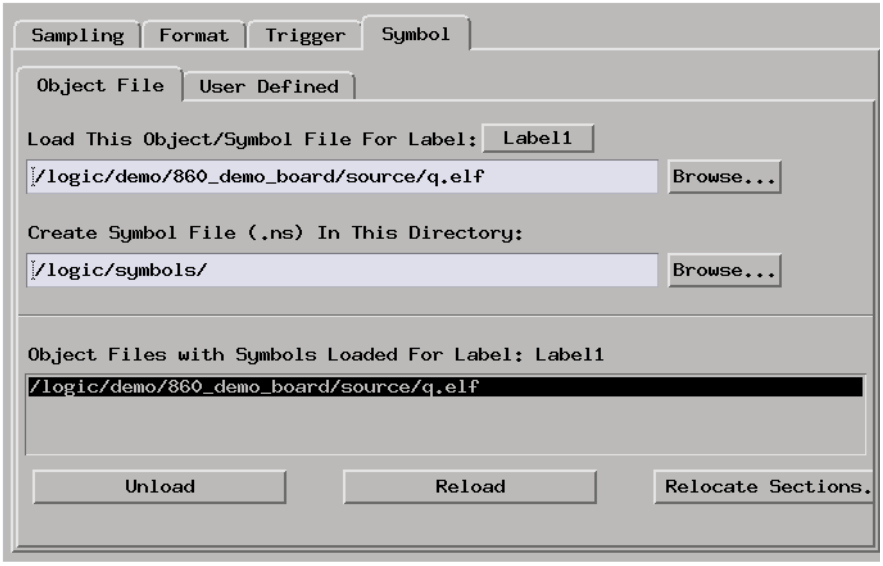
Example for Coff/Stabs (using Ticoff reader)

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

Example for Ticoff

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

The Symbols Tab



The Symbols tab lets you load symbol files or define your own symbols. Symbols are names for particular data values on a label.

Two kinds of symbols are available:

- Object File Symbols. These are symbols from your source code and symbols generated by your compiler.
- User-Defined Symbols. These are symbols you create.
- “Symbols Selector Dialog” on page 96
- “Symbol File Formats” on page 98
- “General-Purpose ASCII (GPA) Symbol File Format” on page 99

Multiple files

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

Object file versions

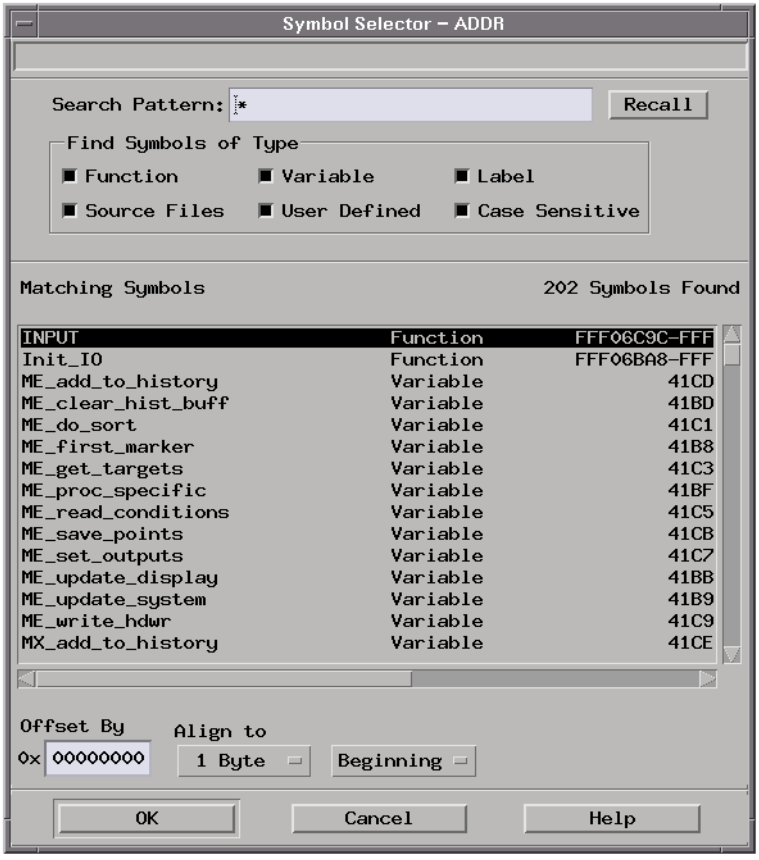
During the load process, a symbol database file with a *.ns* extension will be created by the system. One *.ns* database file will be created for each symbol file you load. Once the *.ns* file is created, the Symbol Utility will use this file as its working symbol database. The next time you need to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

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

See Also

"Using Symbols" on page 85

Symbols Selector Dialog



Search Pattern: Lets you enter partial symbol names and the asterisk wildcard character (*) to limit the symbols to choose from (see “Search Pattern” on page 97). Use the Recall button to select from previous search patterns.

Find Symbols of Type Lets you limit the types of symbols to choose from.

Matching Symbols Lists the symbols that match the search pattern. You choose a symbol from this list.

Offset By Lets you add an offset value to the starting point of a symbol. This can be useful when compensating for microprocessor prefetches (see "Offset By Option" on page 97).

Align to Lets you mask the lower order bits of a symbol's value. This can be useful for triggering on odd byte boundaries (see "Align to x Byte Option" on page 98).

Beginning/End/ Range When a symbol represents a range of addresses, you can choose the beginning address of the range, the end address of the range, or the whole range.

See Also "To enter symbolic label values" on page 89

Search Pattern

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

Asterisk wildcard (*) The asterisk wildcard represents "any characters." When you perform a search on the symbol database using just the asterisk, you will see a list of all symbols contained in the database. The asterisk can also be added to a search word to find all symbols that begin or end with the same letters. For example, to find all of the symbols that begin with the letters "st", select the Search Pattern field and enter "st*".

Offset By Option

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

Example An 80386 processor has a prefetch depth of 16 bytes. Assume functions

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

Align to x Byte Option

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

Example

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

Symbol File Formats

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

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

For ELF/DWARF1, ELF/stabs, and ELF/stabs/Mdebug files, C++ symbols are demangled so that they can be displayed in the original

C++ notation. To improve performance for these ELF symbol files, type information is not associated with variables. Hence, some variables (typically a few local static variables) may not have the proper size associated with them. They may show a size of 1 byte and not the correct size of 4 bytes or even more. All other information function ranges, line numbers, global variables and filenames will be accurate. These behaviors may be changed by creating a readers.ini (see page 90) file.

See Also

"To load object file symbols" on page 86

"To create an ASCII symbol file" on page 90

"To create a readers.ini file" on page 90

General-Purpose ASCII (GPA) Symbol File Format

General-purpose ASCII (GPA) format files are loaded into a logic analyzer just like other object files.

If your compiler does not produce object files in a supported format, or if you want to define symbols that are not included in the object file, you can create an ASCII format symbol file.

Typically, ASCII format symbol files are created using text processing tools that convert the symbol table information from a compiler or linker map output file.

Different types 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" on page 100 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 longer, the logic analyzer only uses the first 16 characters.

The address or address range must be a hexadecimal number. It must appear on the same line as the symbol name, and it must be separated from the symbol name by one or more blank spaces or tabs. Address ranges must be in the following format:

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

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

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

For more detailed descriptions of GPA file records and associated symbol definition syntax, refer to the following topics:

- "SECTIONS" on page 101
- "FUNCTIONS" on page 102
- "VARIABLES" on page 103
- "SOURCE LINES" on page 103
- "START ADDRESS" on page 104
- "Comments" on page 104

GPA Record Format Summary

Format

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

[FUNCTIONS]
func_name  start..end

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

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

```
[START ADDRESS]
address

#comment text
```

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

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

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

NOTE: 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.

Format

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

section_name A symbol representing the name of the section.

start The first address of the section, in hexadecimal.

end The last address of the section, in hexadecimal.

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

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

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

Example

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

FUNCTIONS

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

Format

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

func_name A symbol representing the function name.

start The first address of the function, in hexadecimal.

end The last address of the function, in hexadecimal.

Example

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

VARIABLES

You can specify symbols for variables using:

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

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

Format

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

var_name A symbol representing the variable name.

start The first address of the variable, in hexadecimal.

end The last address of the variable, in hexadecimal.

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

Example

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

SOURCE LINES

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

Format

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

file_name The name of a file.

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

address The address of the source line, in hexadecimal.

Example

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

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

START ADDRESS

Format

```
[START ADDRESS]
address
```

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

Example

```
[START ADDRESS]
00001000
```

Comments

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 s symbol entry.

Format

```
#comment text
```

Example

```
#This is a comment
```

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

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

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

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

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

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

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

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

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

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

click When using a mouse as the

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

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

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

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

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

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

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

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

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

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

debug mode See *monitor*.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Ethernet address See *link-level address*.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

master card In a module, the master card controls the data acquisition or output. The logic analysis system references the module by the slot in which the master card is plugged. For example, a 5-card Agilent Technologies 16555D would be referred to as *Slot C:*

machine because the master card is in slot C of the mainframe. The other cards of the module are called *expansion cards*.

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

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

module/probe interconnect cable

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

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

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

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

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

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

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

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

by the channel width of the instrument.

pod See *pod pair*

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

preprocessor See *analysis probe*.

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

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

processor probe See *emulation probe*.

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

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

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

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

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

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

measurement as part of its data acquisition cycle.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

trace See *acquisition*.

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

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

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

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

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

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

Index

Symbols

&, 69
&> duration trigger field, 69
*, bit assignment, 50
+, label polarity, 53
-, label polarity, 53
., bit unassignment, 50

Numerics

1 GHz full channel state mode, 47
16517/18A logic analyzer, testing, 46
16517-68701 master kit, 11
16517A logic analyzer, 2
16517A/18A characteristics, 80
16517A/18A specifications, 80
16518-68701 expander kit, 11
16518A expansion logic analyzer, 2
2 GHz full channel timing mode, 47
4 GHz half channel timing mode, 47

A

acquisition modes, state, 47
acquisition modes, timing, 47
acquisition modes, when to set, 18
acquisitions, interpreting, 30
acquisitions, processing, 30
activity indicator, in Format, 50
Align to x Byte option, 98
Align to x Byte option for symbols, 98
altitude characteristics, 81
analyzer, arming, 78
arming analyzer, 78
arming control, 78
arrows, activity indicators, 54
ASCII format symbols, 101, 102, 103, 104
ASCII symbol file, 103

B

bad data in measurement, 44
base, numeric, 75
basic state measurements, example, 26
basic timing measurements, example, 22
bit activity indicator, 50
bit assignment, 50
bit assignments, 52
bit order, changing, 51
bit pattern terms, 71
bit reference line, 50
bit significance, 50
bits, assigning, 50
bits, maximum per label, 50
bits, reordering, 51
BNC to SMB adapter, 11
break down macros, 67
browsing, 97
browsing the symbol database, 97
byte mode, numeric base, 75

C

cable activity indicators, 54
cable information, 13
calibration, operational accuracy, 16
capacitive loading, 45
center trigger position, 76
channel activity indicators, 54
channel counts, characteristic, 81
channels, assigning to label, 49
channels, maximum per label, 50
channel-to-channel skew, characteristic, 81
characteristic, 83
clear menu, 60
clear sequence levels, 60
clear sequence/resource/name, 60
clock channel, 13
clock duty cycle requirements, characteristic, 81
clock edge, adjusting, 49
clock threshold level note, 53
clock, defining, 48
clocks, when to set, 18
code, assigning address offsets, 87
COFF symbol reader options, 92
combination terms, 73
comments, 104
configuration files, loading, 34
configurations, compatibility across models, 34
configurations, storing, 34
connection methods, overview, 10
connections, 17
conventional timing acquisition modes, 47
count states timing macro, 63
custom state macros, 65
custom timing macros, 63
custom trigger macros, 59

D

dashes, activity indicators, 54
data channels are not connected properly message, 42
data channels, naming, 49
data may not be valid message, 40
data set, interpreting, 30
defaulting the trigger, 60
definition, 83
definition, calibration procedure, 84
definition, function test, 84
definition, operational accuracy calibration, 84
definition, specification, 83
delay trigger position, 76
depth of memory, characteristic, 81
deskewing, 16

Index

E

edge terms, 72
edge trigger macro, 65
edge trigger terms, 72
edge within pattern macro, 64
edit sequence steps, 56
edit trigger sequence steps, 58
ELF symbol reader options, 91
ELF/DWARF file format, 98
ELF/stabs file format, 98
else on branch, 58
end trigger position, 76
environmental characteristics, 81
error message, sample rate exceeds trigger sequencer rate, 41
error message, trigger may not be included in acquired data, 41
error messages, cannot run, 39
error messages, clock is not connected properly, 42
error messages, clock out of spec, 40
error messages, data channels not connected, 42
error messages, invalid card configuration, 38
error messages, invalid skew values, 44
error messages, loss of sync, 40
error messages, maximum of 32 channels per label, 37
error messages, measurement halted, 39
error messages, memory failed, 38
error messages, oversampled states, 41
error messages, skew values are from a different instrument, 38
error messages, skew values do not match, 43
error messages, skew values from different instrument, 43
error messages, trigger rate exceeds sample rate, 41
error messages, trigger sequencer, 41
error messages, using default 16517/18 skew values, 37
error messages, waiting for trigger, 36
errors in data, 44
errors, error messages, 35
example, 97, 98
ext clk edge, 77

F

file versions, 86
files, 86
find early event state macros, 67
find edge macro, 63
find event state macros, 66
find late event state macros, 67
find n times state macros, 66
find pattern state macros, 66
find sequence state macros, 66
finding the symbol you want, 97
flowchart, 17
format tab, use, 18, 47
full channel, timing, 47
functions, 102

G

glossary of terms, 2
group bit assignment, 50

H

half channel, timing, 47
help, symbols, 94
help, trigger, 78
help, trigger sequence, 61
how to set up, 17
humidity characteristics, 81

I

IEEE-695 file format, 98
if branch, 58
in ASCII format, 101, 102, 103, 104
in symbol browser, 97
input capacitance, probe, characteristic, 81
input impedance, probe, characteristic, 81
input resistance, probe, characteristic, 81
internal sequence, 61
internal sequence levels, 61

L

label polarity, changing, 53
label values, symbolic, 89
labels, 19, 50
labels, activating, 52
labels, add after, 52
labels, add before, 52
labels, adding, 52
labels, assigning bits, 50
labels, defining, 49
labels, deleting, 52
labels, deleting from terms, 75
labels, inserting, 52
labels, inserting on terms, 75
labels, polarity, 53
labels, reordering bits, 51
labels, turning off, 52
labels, turning on, 52
line numbers, 103
load, reducing, 45
loading, 86
loading files including symbols, 86
loading object file symbols, 86
loading user-defined symbols, 89
logic analyzer hangs, 35

M

macros, user-defined, 58

main system help page, 2
masking off addresses of symbols, 98
maximum delay after triggering, characteristic, 81
maximum ground length recommendations, 10
maximum input voltage, 13, 14
maximum input voltage, characteristic, 81
measurement doesntrun', 35
measurement halted message, 39
measurements, overview of basic state, 26
measurements, overview of timing, 22
memory and trigger, 76
memory depth, 47
memory depth, characteristic, 81
messages, pod identified, 41
minimum detectable pulse width, characteristic, 81
minimum external clock period, specification, 80
minimum input overdrive, 14
minimum input voltage swing, specification, 80
minimum signal amplitude, 14

N

negative logic, note, 53
note, activity indicators, 54
note, maximum bits per label, 50
note, negative logic, 53
note, reorder bits, 51

O

object file symbol files, 86
object file symbols, 97
occurrence counter, characteristic, 81
occurrence counter, state trigger, 70
occurrence counters, when reset, 58
occurs field, 70
odd-numbered addresses, 98
odd-numbered addresses represented by symbols, 98
offset addresses, assigning, 87
Offset By option of the symbol browser, 97
OMF96 file format, 98
OMFx86 file format, 98
operating environment characteristics, 81
overdrive, definition, 15
oversample, 77
oversampling, characteristic, 81
oversampling, what is, 77
overview, measurement process, 17

P

pattern after edge macro, 64
pattern count, state trigger, 70
pattern duration macro, 63
pattern duration, characteristic, 81
pattern durations, setting, 69
Pattern field, 97
pattern recognizers, characteristic, 81
pattern, state trigger macros, 66
pattern/edge trigger macros, 64
performance verification, 16517/18A, 46
period, sample time, 77
pod id button, 13
pod thresholds, setting, 53
predefined macros, 61, 62
predefined trigger macros, 62
predefined trigger macros, displaying, 67
predefined trigger macros, modifying, 67
prefetch, triggering beyond, 97
probe leads, 10
probes, accessory kit, 11
probes, equivalent circuit, 13
probes, general-purpose, 10
probes, ground leads, 11
probes, recommendations, 10
probes, signal leads, 11
probing accessories, 11
probing individual lines, 10
probing options, 10
problems making measurements, 35
pulse width macro, 65

R

radix, numeric base, 75
rate, sample period, 77
readers.ini file, 90
recalling trigger sequences, 59
relocating sections of code, 87
reset bit pattern, 71
restore macros, 67
results, 97
roadmap, 17

S

sample offset field, 49
sample period, 77
sample period, characteristic, 81
sample rate, setting, 47
samples/clock control, 77
saving trigger sequences, 59
Search Pattern field, 97
searching the symbol database, 97
sections, 101
selecting macros, 56
self test, 16517A, 46
self test, 16518A, 46
sequence levels, characteristic, 81

sequencer, maximum levels, characteristic, 81
set up, trigger sequence, 58
setting threshold, 53
settings, saving, 34
setup/hold time, specification, 80
shortcut, bit assignment, 50
signal loading, 14
signals, requirements, 14
skew adjust tab, 16
skew, channel-to-channel, characteristic, 81
source line numbers, 103
speed, state/timing, characteristic, 81
Stabs symbol reader options, 92
start address, 104
start trigger position, 76
state channel width, 47
state clock pulse width, specification, 80
state clocks, characteristic, 81
state macros, 62
state macros, predefined, 65
state measurements, basic overview, 26
state modes, 47
state speed, characteristic, 81
state trigger macros, 66, 67
storing trigger setups, 59
symbol demangling, 90
symbol file formats, 98
symbol file versions, 86
symbol selector dialog, 96
symbolic label values, 89
symbols, 97
symbols, loading object file symbols, 86
symbols, loading user-defined, 89
symbols, outside defined sections, 91
symbols, types and use, 94
symbols, user-defined, 88
symbols, using, 85

T

tab, symbols, 94
target system, probing, 10
temperature characteristics, 81
terms, 19
then branch, 58
threshold accuracy, specification, 80
threshold logic levels, ECL, 53
threshold logic levels, TTL, 53
threshold range, probe, characteristic, 81
TI COFF file format, 98
time covered by data, characteristic, 81
time interval accuracy, characteristic, 81
time violation macros, 65
timer, characteristics, 81
timers, using, 73
timers, when reset, 58
timing analysis characteristics, 81
timing macros, 62
timing macros, basic predefined, 63
timing macros, predefined, 63
timing measurements, basic overview, 22
timing modes, 47
timing speed, characteristic, 81
timing, memory depth, 47
trigger bit patterns, 71
trigger characteristics, 81
trigger edge terms, 71
trigger edit, 56
trigger macros, 61
trigger macros, creating, 68
trigger macros, modifying, 68
trigger macros, pattern/edge, 64
trigger macros, predefined, 62
trigger macros, selecting, 56
trigger macros, state, 65, 66, 67
trigger macros, time violation, 65
trigger macros, timing, 63
trigger macros, user-defined, 68
trigger macros, using, 55
trigger pattern macros, 63
trigger pattern terms, 71
trigger position control, 76
trigger resource terms, 71
trigger resource terms, combination, 73
trigger sequence branches, 58
trigger sequence jumps, 58
trigger sequence levels, goto, 58
trigger sequence loops, 58
trigger sequence steps, 56
trigger sequence steps, copying, 57
trigger sequence steps, editing, 58
trigger sequence steps, replacing, 57
trigger sequence, adding steps, 57
trigger sequence, clearing, 60
trigger sequence, customizing, 68
trigger sequence, default, 60
trigger sequence, deleting, 57
trigger sequence, editing, 57
trigger sequence, explanation, 61
trigger sequence, inserting, 57
trigger sequence, specifying, 55
trigger sequencer rate exceeds sample rate note, 41
trigger sequencer will not see oversampled states, 41
trigger sequences, loading, 59
trigger sequences, saving, 59
trigger sequences, storing, 59
trigger set up, 58
trigger tab, reference, 55
trigger tab, use, 19
trigger terms, adding labels, 75
trigger terms, bit pattern, 71
trigger terms, combination, 73

trigger terms, deleting labels, 75
trigger terms, edge, 72
trigger terms, limits on, 74
trigger terms, timers, 73
trigger variables, 71
trigger, continue timer, 73
trigger, pattern durations, 69
trigger, pause timer, 73
trigger, poststore, 76
trigger, resetting, 60
trigger, setting up, 55
trigger, start timer, 73
trigger, stop timer, 73
trigger, substeps, 19
triggering on a symbol, 97, 98
triggering on a symbol beyond prefetch depth, 97
troubleshooting the logic analyzer, 35

U

unassigned bits, 50
user macros, trigger, 59
user threshold logic level, 53
user-defined macros, 58, 61, 68
user-defined symbols, 88
user-defined symbols, loading, 89
user-defined trigger, 68

V

variables, 103
versions, 86
versions of symbol files, 86
vibration characteristics, 81

W

wait state macros, 67
wait time macro, 65
warning messages, using default skew values, 43
warranty, what is covered, 83
wildcard characters, 97

Index

Publication Number: 5988-9020EN



Agilent Technologies

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP