Help Volume

Analysis: Serial ATA Tool (Agilent Technologies N4219A)

Using the Serial ATA Tool

The Serial ATA Tool is used with the Agilent Technologies N4219A Serial ATA analysis probe. Together, they give a logic analyzer protocol-based triggering, display, and error detection capabilities.

You can insert the analysis probe into a Serial ATA link and capture the data passing through the link.

To use the analysis probe, connect the N4218A interposer to the analysis probe and connect the analysis probe to the logic analyzer, as described in the printed *Installation Guide*. Next, load a configuration file and calibrate the analysis probe. At this point, insert the interposer into a Serial ATA link.

Use the logic analyzer's "Find Serial ATA Packet" trigger function to specify when to capture data, and use the Serial ATA Tool to decode and display the captured packet information.

NOTE: Because the Serial ATA Tool requires VisiTrigger capabilities, it can only be used with the Agilent Technologies 16715/16/17/18/19/40/41/42/50/51/52/53/54/55/56A logic analyzer modules.

"Configuring the Logic Analyzer" on page 8

- "To load the configuration file" on page 8
- "Configuring the analysis probe" on page 13
- "Calibrating the Analysis Probe" on page 14

"Capturing the Data" on page 16

- "To set up packet triggers (8B Only)" on page 16
- "To trigger in 10B mode" on page 18
- "To define and edit packet events" on page 19 ("Event Editor Dialog" on page 20)
- "To run the measurement" on page 21

“Displaying the Data” on page 22

- “To set up the Serial ATA Tool” on page 22
- “To select the output data columns” on page 26
- “To color data rows” on page 24
- “To select other columns to display” on page 28
- “To select which fields to display” on page 29

“Modifying Protocol Definitions” on page 35

- “To access the protocol definitions” on page 35
- “To modify a protocol definition” on page 36
- “To reload the protocol definitions” on page 37
- “To reset the protocol definitions” on page 38

Reference

- “Labels Mapped to Logic Analyzer Channels” on page 10
- “Protocol Definition Syntax” on page 42

Concepts

- “Basics of 8B/10B Encoding” on page 48

See Also

The *Agilent Technologies N4219A Analysis Probe for Serial ATA Installation Guide* for information on connecting the analysis probe to a Serial ATA link and to the logic analyzer probe cables.

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

Glossary (see page 51)

Contents

Using the Serial ATA Tool

1 Task Guide

2 Reference

3 Concepts

Glossary

Index

1

Task Guide

Configuring the Logic Analyzer

Load a configuration file to configure the logic analyzer.

- "To load the configuration file" on page 8

After the configuration file is loaded, set up the analysis probe.

- "Configuring the analysis probe" on page 13

To load the configuration file

1. In the System window, select the Load Configuration ... command from the File menu.
2. In the File Manager dialog:
   a. Browse to the /logic/configs/hp/SerialATA directory.
   b. Select the appropriate configuration file:

Name	Description
N4219_Basic.___	1 Card, 2 Pods per Direction
N4219_Extended.___	2 Cards, 4 Pods per Direction

"Basic" configurations split a single logic analyzer card into two machines, with two bytes of data stored in each state.

"Extended" configurations use two logic analyzer cards, with four bytes of data stored in each state. This allows you to capture data at twice the rate as a basic configuration.

   c. Select the Load button.

Loading the configuration file:

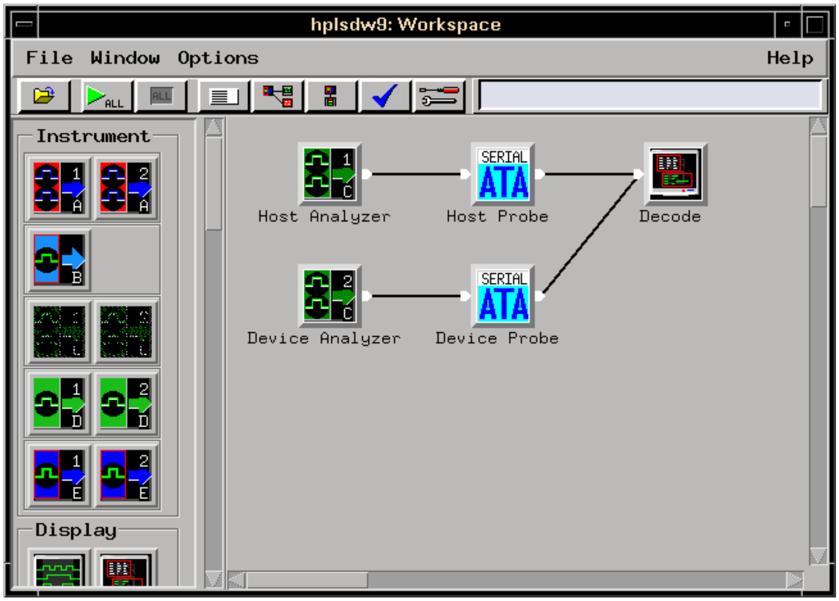
- Sets up the workspace (for more information, see "Workspace Setup" on page 9).
- Selects the state (synchronous) sampling mode (for more information, see "Logic Analyzer Sampling Mode" on page 10).

- Labels (maps) buses and signals coming from the analysis probe to logic analyzer channels (for more information, see “Labels Mapped to Logic Analyzer Channels” on page 10).
- Sets the logic analyzer pod threshold voltages for TTL logic levels.

NOTE: You must set up the workspace and analyzer labels by loading a configuration file. Do not attempt to change the labels using the analyzer's Format tab. Do not attempt to set up the workspace by dragging and dropping a Serial ATA Tool onto an analyzer.

Workspace Setup

When you load the configuration file, it places Instrument, Analysis, and Display tools in the Workspace window.

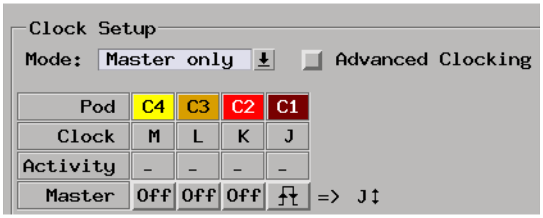


When you want to capture data in both directions (Host and Device) at the same time, you must use separate logic analyzer "machines" for each direction because the clock signals coming from each direction on the analysis probe are different.

The Serial ATA Tool will immediately try to decode the captured trace data. If no trace data has been captured, run a measurement.

Logic Analyzer Sampling Mode

When you load the configuration file, it selects the state (synchronous) sampling mode and specifies the rising edge and falling edge of the J clock input, with no qualifiers, as the analyzer's sampling clock.



Clock Setup

Mode: Master only

□ Advanced Clocking

Pod	C4	C3	C2	C1	
Clock	M	L	K	J	
Activity	–	–	–	–	
Master	Off	Off	Off	↕	=> J↕

Labels Mapped to Logic Analyzer Channels

When you load the configuration file, it maps labels to logic analyzer channels.

NOTE: Do not delete or change any of the labels defined by the configuration file. The Serial ATA Tool will not attempt any analysis if any of the labels are missing.

The labels are customized for the direction of traffic that is being analyzed. This is done to so that you may view both directions of traffic in a single Listing window. The two directions are referred to as "Host" (H_ prefix, from host to device) and "Device" (D_ prefix, from device to host).

Labels are comprised of these mnemonics:

H_	From host to device
D_	From device to host
10BWord	The probe will generate 10B data, even when it is in 8B mode, before it is framed and also if it encounters an invalid 10B code. In Extended Mode configurations there are two 10BWord labels; this is necessary since a single label may contain only 32 bits and there are a total of 40 bits in the 10B labels for a given state. In Basic Mode configurations, there is only one 10BWord label; the two 10BWords that

comprise a 10BDword are presented in as two lines in the decoded listing.

10BByte These labels break out the component bytes of the 10BWord labels.

8BWord The 8B translation of the 10B symbols that were received. When the probe is in 8B mode, these 8B symbols are de-scrambled.

8BByte These labels break out the component bytes of the 8BWord label.

K/D 1 indicates K (special) character, 0 indicates D (data) character. The K/D0, K/D1, K/D2, and K/D3 labels indicate the type of character for Byte0 through Byte3. The K/D label combines these into one binary value, with the MSB associated with Byte0. The K/D bits can be used for searching for activity in the Listing window.

8B/10B Indicates whether the data in this byte is 8B or 10B. A value of 1 in this label indicates 8B data.

RD The symbol disparity of the current symbol using the following encoding:

- 00 - symbol has negative disparity
- 01 - symbol has positive disparity
- 10 - symbol has even disparity
- 11 - symbol has disparity error

This is not the running disparity.

ATAPI Indicates when a data packet contains an ATAPI command. A regular ATA command containing the Packet command indicates that the next packet will be a data packet containing an ATAPI command. Each packet is terminated with an EOF primitive. EOF primitives are not part of the ATA or ATAPI grammars and therefore are never involved in packet triggers. Because of this EOF characteristic, the ATAPI signal transitions only upon detection of EOF. The algorithm is:

```
Detect the Packet command.
Detect EOF, set ATAPI to 1
Detect EOF, set ATAPI to 0
```

BusStatus The 3-bit BusStatus label encodes the type of activity that occurred in this state. The status values are:

- 0 Squelch (inactive) the differential signal has diminished to zero.
- 1 Active (non-squelch) the differential signal is non-zero.
- 2 COMRESET/COMINIT has been detected
- 3 COMWAKE has been detected
- 4 Framed - normal data transmission state
- 5 Reserved
- 6 Reserved

The other labels (columns) you see in the listing display are generated by the Serial ATA Tool. Generated labels include:

Phase (Basic mode only.) This binary label is set to 0 for bytes 0 and 1 of a DWord, and is set to 1 for bytes 2 and 3.

InFIS This binary label is set to 1 when the curent state is in an FIS (a Frame Information Structure). This label changes from 0 to 1 on the state after an SOF primitive and back to 0 when an EOF is received. This is useful for searching for the beginning and end of an FIS when in 10B mode, or in 8B mode when primitives are not filtered by the analysis probe.

ReFrame This binary label is set to 1 on the state that causes the probe to achieve character synchronization. For all other states, this label is 0. In 10B mode, there should be at most one '1' in the trace. In 8B mode, if the Serial ATA link is slipping bits, '1' may occur several times in the trace. Each '1' indicates that there was a bit or byte slip between that state and the previous synchronization state.

BadCRC This binary label is 1 if a CRC error is detected for this state.

@

Configuring the analysis probe

Before you run a measurement, you need to tell the analysis probe what kind of link you are probing.

- "To set up the Serial ATA Tool" on page 22

It is also best to set up the decoding options before running a measurement. This will avoid delays caused by reprocessing the captured data.

- "To select which fields to display" on page 29

Calibrating the Analysis Probe

The analysis probe must be calibrated before use.

There must be no switching activity on the Serial ATA link during calibration.

The analysis probe calibrates itself when:

- The analysis probe is turned on.
- The Calibrate Probe button in the Serial ATA tool is selected.

Calibration takes about 45 seconds. During calibration, the Synchronization LEDs will be steady red, and the Amplitude LEDs will flicker.

To prevent switching activity during calibration

Two ways to prevent switching activity are:

1. Turn off the equipment under test, or
2. Unplug the interposer from the Serial ATA link.

To determine whether calibration was successful

1. If calibration fails, the Amplitude and Synchronization LEDs will all continuously flash red then green.
2. If calibration takes place before the configuration file is loaded, and the calibration fails, a "Probe not calibrated" message will be displayed when you load the configuration file.
3. If you use the Calibrate Probe button in the Serial ATA tool, and the calibration fails, a "Probe not calibrated" message will be displayed.
4. If you have connected a serial or LAN cable to the analysis probe, the results of the calibration will be displayed in your terminal emulator window. See "Verifying Analysis Probe Performance" on page 48.

NOTE: There must be no switching activity on the Serial ATA link during calibration.

To recalibrate using the Serial ATA tool

1. In the Workspace window, open the Serial ATA tool display.
2. Select the Setup tab.

3. Select Calibrate Probe.

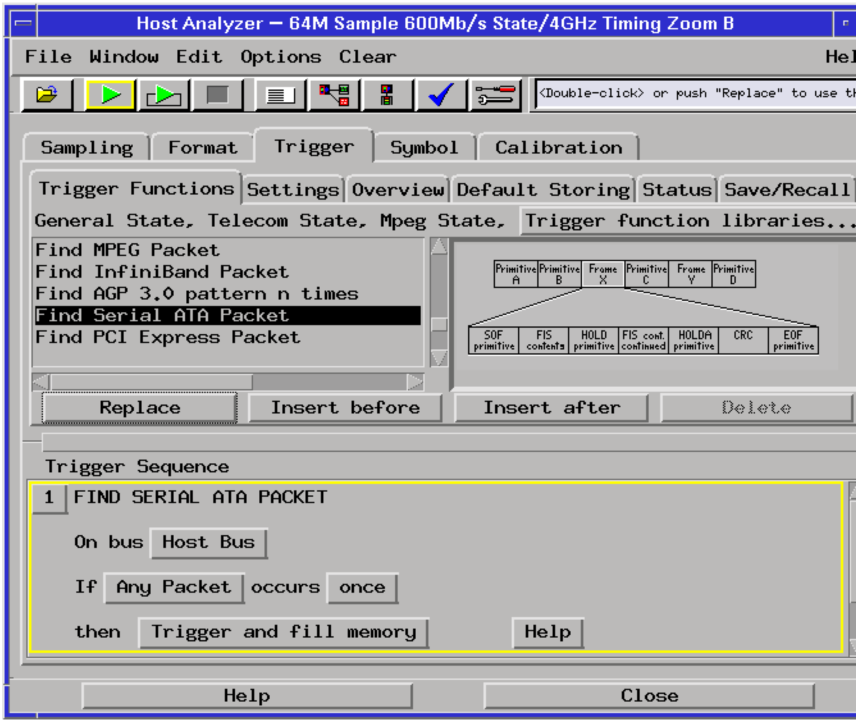
Capturing the Data

This section shows you how to set up packet triggers and run logic analyzer measurements.

- "To set up packet triggers (8B Only)" on page 16
- "To run the measurement" on page 21

To set up packet triggers (8B Only)

1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
2. Select the "Find Serial ATA Packet" trigger function and either replace the current trigger sequence level or insert a new level.



Find MPEG Packet
Find InfiniBand Packet
Find AGP 3.0 pattern n times
Find Serial ATA Packet
Find PCI Express Packet

Trigger Sequence

1 FIND SERIAL ATA PACKET

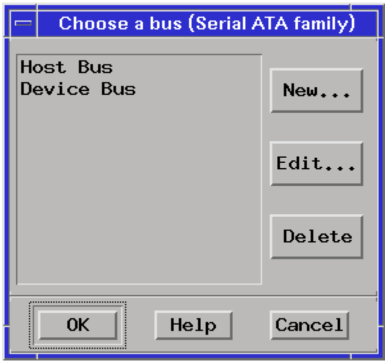
On bus Host Bus

If Any Packet occurs once

then Trigger and fill memory Help

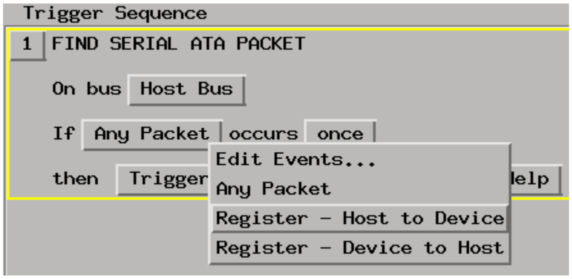
3. In the "Find Serial ATA Packet" trigger sequence level, select the bus button.

4. In the Choose a Bus dialog, select the bus definition you want to use and select the OK button.



Host Bus
Device Bus

5. Specify the packet event to find.



Trigger Sequence

1 FIND SERIAL ATA PACKET

On bus Host Bus

If Any Packet occurs once

then Trigger

NOTE: All packet fields start as "don't care" values. The more fields you enter specific values into, the more of the logic analyzer's internal triggering resources are used, and eventually you can run out of triggering resources.

So, enter the minimum number of fields necessary to get your trigger, and save trigger resources for other events.

See Also

"Modifying Protocol Definitions" on page 35

"To define and edit packet events" on page 19

"To run the measurement" on page 21

"Displaying the Data" on page 22

To trigger in 10B mode

The "Find Serial ATA Packet" trigger function does not work in 10B mode. Triggers have been stored to make it easy to trigger on the occurrence of a primitive in 10B mode.

1. Open the Setup window for either logic analyzer machine.
2. Select the Trigger tab.
3. Select the Save/Recall tab.
4. Select one of the 10B primitives.
5. If you wish, change the "occurs" count or the "then" action.

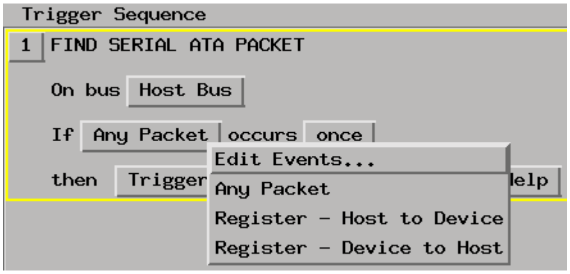
You can also use the other trigger functions to build a trigger which looks for particular patterns of raw 10B data on the bus.

See Also

- “To set up packet triggers (8B Only)” on page 16

To define and edit packet events

1. In the "Find Serial ATA Packet" trigger function, select the packet button, and choose Edit Events....



Trigger Sequence

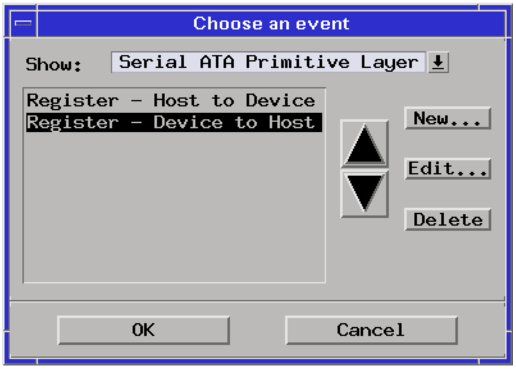
1 FIND SERIAL ATA PACKET

On bus Host Bus

If Any Packet occurs once

then Trigger

2. In the "Choose an event" dialog, either select New... to define a new event, or select an event and select the Edit... button to edit it.



Choose an event

Show: Serial ATA Primitive Layer

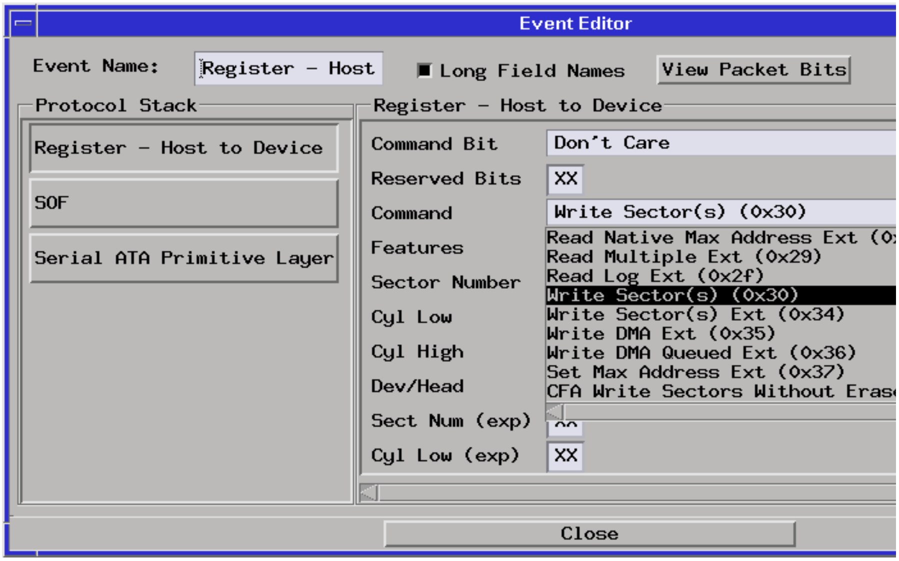
To delete an event definition, select the Delete button.

3. When you are done defining or editing an event, select the Close button in the Event Editor dialog and select the OK button in the "Choose an event" dialog.

See Also

“Event Editor Dialog” on page 20

Event Editor Dialog



Event Editor
■ Long Field Names
View Packet Bits
Protocol Stack
Register – Host to Device
SOF
Serial ATA Primitive Layer
Read Native Max Address Ext (0x
Read Multiple Ext (0x29)
Read Log Ext (0x2f)
Write Sector(s) (0x30)
Write Sector(s) Ext (0x34)
Write DMA Ext (0x35)
Write DMA Queued Ext (0x36)
Set Max Address Ext (0x37)
CFA Write Sectors Without Eras
Close

Event Name: Register – Host

Register – Host to Device

Command Bit	Don't Care
Reserved Bits	XX
Command	Write Sector(s) (0x30)
Features	
Sector Number	
Cyl Low	
Cyl High	
Dev/Head	
Sect Num (exp)	XX
Cyl Low (exp)	XX

Event Name Specifies the event definition name.

Long Field Names Specifies whether the full names of fields or the short 2- or 3-letter mnemonics for fields are displayed in the protocol column.

View Packet Bits Specifies whether the packet bits window is displayed. The Packet Bits window shows the packet event's 1s and 0s.

Protocol Stack This column shows the protocol stack layers used in the event definition. You can select a layer to edit the fields associated with that layer.

(Protocol) This column displays the fields of the selected protocol stack layer and lets you enter the field values that define the event.

See Also

"To define and edit packet events" on page 19

To run the measurement

- Select the Run button ▶ or the Group Run button ▶GROUP to start the measurement.

See Also

"Displaying the Data" on page 22

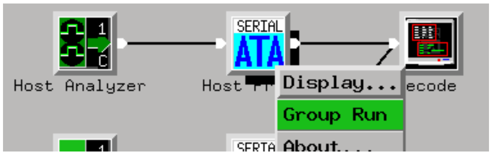
Displaying the Data

This section describes the options available when displaying captured Serial ATA data.

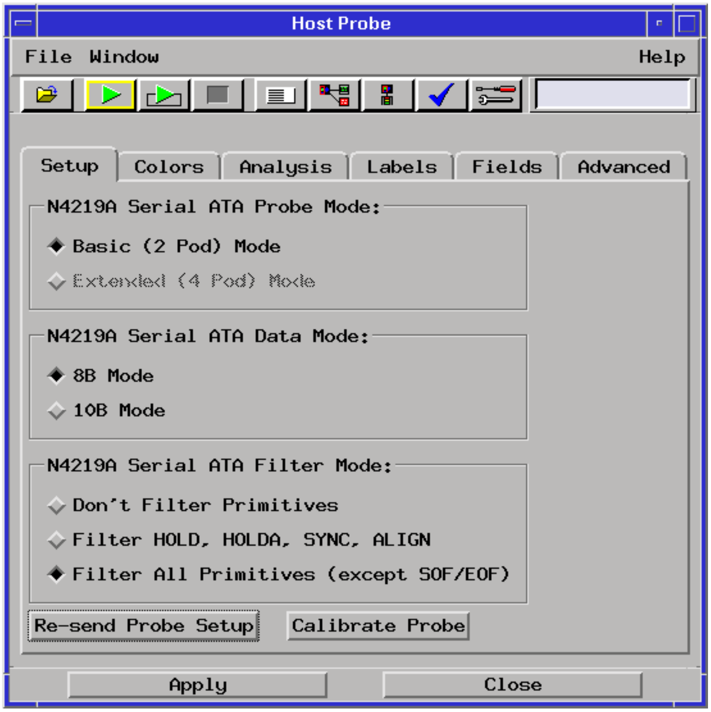
- "To set up the Serial ATA Tool" on page 22
- "To select the output data columns" on page 26
- "To color data rows" on page 24
- "To select other columns to display" on page 28
- "To select which fields to display" on page 29

To set up the Serial ATA Tool

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Setup tab, select the option which matches the link you are probing.



N4219A Serial ATA Probe Mode:

◆ Basic (2 Pod) Mode
◇ Extended (4 Pod) Mode

N4219A Serial ATA Data Mode:

◆ 8B Mode
◇ 10B Mode

N4219A Serial ATA Filter Mode:

◇ Don't Filter Primitives
◇ Filter HOLD, HOLDA, SYNC, ALIGN
◆ Filter All Primitives (except SOF/EOF)

Basic/Extended Probe Mode This option is set by loading the appropriate configuration file. (This is because the logic analyzer needs to output different labels for the two modes.)

8B/10B Data Mode In 8B mode, 8B data is presented to the analyzer when the data is properly framed. Note that 8B data cannot be generated for invalid 10B codes. In 10B mode, only 10B codes are generated.

Filter Mode Choose which primitives should be filtered by the analysis probe hardware. The filter mode affects data in both directions. The filter mode only applies to 8B data.

- Don't Filter Primitives: Capture all primitives. This mode could be useful during initial turn-on of Serial ATA silicon.
- Filter HOLD, HOLDA, SYNC, ALIGN: The SYNC primitive is used to keep the link up (busy) while there is no other traffic. It's pretty easy

to fill a trace buffer with SYNC primitives. The HOLD, HOLDA and ALIGN primitives can occur within a FIS and could break a trigger setup. This mode could be useful when you are interested in triggering on packets and seeing primitives that occur only between packets.

- Filter All Primitives (except SOF/EOF): The SOF/EOF primitives frame a FIS and are required for packet decode. This mode could be useful when debugging code that is sending/receiving commands between host and device. Use this mode to capture the maximum possible number of packets.

1. No further action is needed to change the probe mode. Select Re-Send Probe Setup if an message appears stating that there is a setup error.

Remember that the probe setup affects all of the data coming from the analysis probe, so you cannot have separate settings for the Host and Device data.

See Also

"To select the output data columns" on page 26

"To color data rows" on page 24

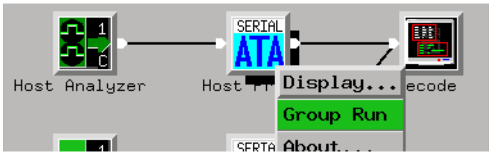
"To select other columns to display" on page 28

"To select which fields to display" on page 29

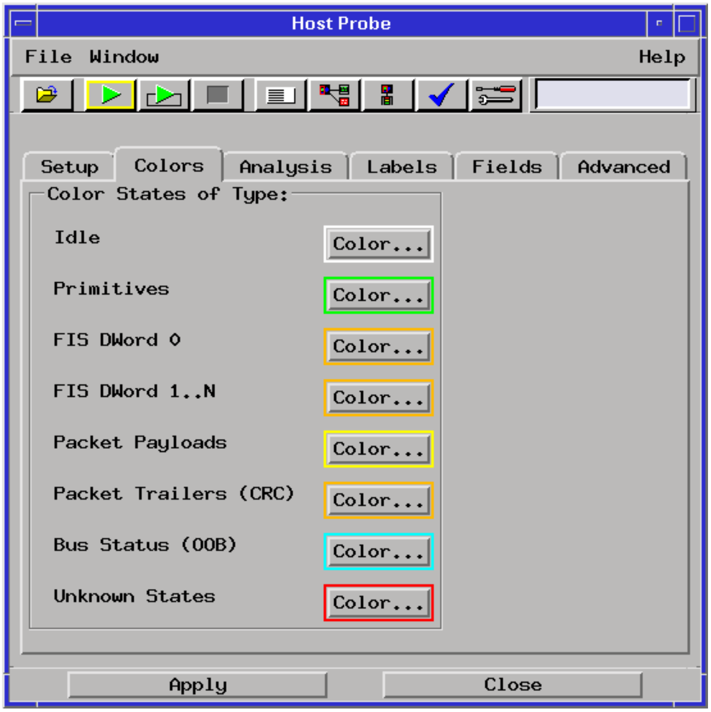
"To load the configuration file" on page 8

To color data rows

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Colors tab, select the color for each type of state:



3. Select Apply to apply the new colors to the decoded listing.

4. Select Close to close the Serial ATA Tool.

The Frame Information Structure is represented as "FIS DWord 0" and "FIS DWord 1..N". FIS DWord 0 contains the FIS Type, Command and Features.

To modify the shades of color available, select Edit Colors... from any of the Color... dialogs.

See Also

“To set up the Serial ATA Tool” on page 22

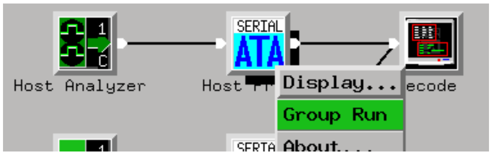
“To select the output data columns” on page 26

“To select other columns to display” on page 28

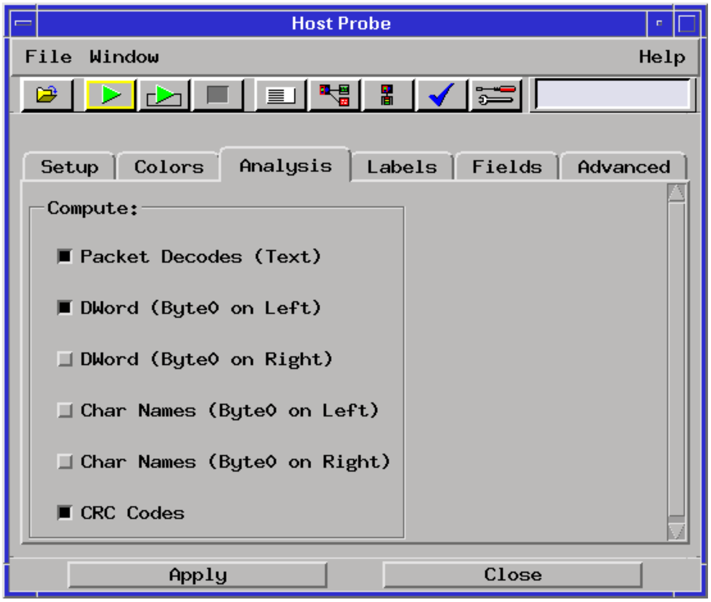
“To select which fields to display” on page 29

To select the output data columns

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Analysis tab, select the columns that should appear in the output data:



Compute:

- ■ Packet Decodes (Text)
- ■ DWord (Byte0 on Left)
- □ DWord (Byte0 on Right)
- □ Char Names (Byte0 on Left)
- □ Char Names (Byte0 on Right)
- ■ CRC Codes

Packet Decodes This text column contains a description of the decoded data. To adjust how much data is displayed in this column, select the "Fields" tab.

DWord These options add a column which displays data as 64-bit words. Choose whether you would like bytes to be displayed with Byte 0 on the right (as is common in Serial ATA documents) or with Byte 0 on the left (as used in the trigger macro and most other protocol specifications). In the Listing tool, the label name indicates whether Byte 0 is on the left or on the right.

These options are helpful in the Basic (2 Pod) mode. Since data is decelerated by 2 rather than by 4 in this mode, a DWord takes two states. The DWord labels place the two states on a single line for easier reading.

The "DWord (Byte0 on Left)" label under the Analysis tab is the text equivalent of the integer "DWord" label under the Labels tab.

Char Names These text columns display the data as 8B character names. Choose whether you would like bytes to be displayed with Byte 0 on the right (as is common in Serial ATA documents) or with Byte 0 on the left (as in most other protocol specifications). In the Listing tool, the label name indicates whether Byte 0 is on the left or on the right.

If invalid 10B codes are encountered, they are presented in this label as "inv".

CRC Codes This 1-bit column contains "1" when there is an error or "0" otherwise.

If a CRC code is bad, the "1" will be set on the same state as the first byte of the CRC code.

You can use this column to search for corrupt data by using the Search tab in the Listing tool.

1. Select Apply to output the selected data columns.
2. Select Close to output the selected columns and close the Serial ATA Tool.

Note that other output columns can be selected using the Labels tab.

See Also

"To set up the Serial ATA Tool" on page 22

"To color data rows" on page 24

"To select other columns to display" on page 28

"To select which fields to display" on page 29

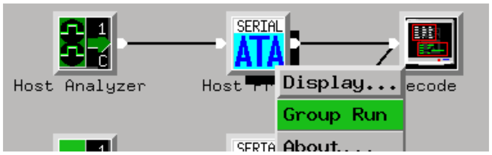
"Labels Mapped to Logic Analyzer Channels" on page 10

To select other columns to display

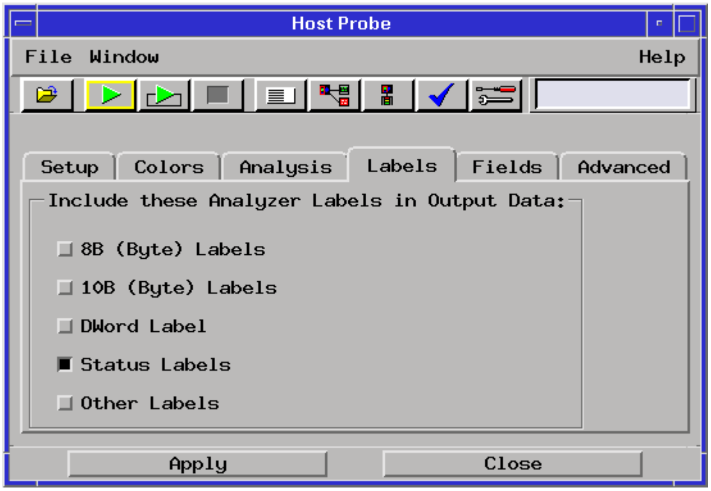
Choose which undecoded data from the analysis probe should be passed through the Serial ATA Tool and displayed in the listing window. You can suppress some or all of these labels (which appear as columns in the listing).

It is best to display only the labels which you expect to use. By suppressing the labels which you don't need, it will be easier to find the labels which contain decoded data. The listing display will also scroll more quickly. If you discover that you need to see more labels, you can always come back to the Labels tab and select the labels, without re-running the measurement.

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Labels tab, select the labels (columns) that should appear in the output data:



Include these Analyzer Labels in Output Data:

□ 8B (Byte) Labels
□ 10B (Byte) Labels
□ DWord Label
■ Status Labels
□ Other Labels

8B Labels Show the data in 8B form, when possible.

10B Labels Show the data in 10B form.

DWord Labels Show the data as undecoded 32-bit words. Byte0 is shown on the left. The default base of these labels is hexadecimal, but it can be helpful to change the base of these labels to match the data. For example, if you expect a payload to contain ASCII data, set the label to ASCII to see the text in the payload.

Status Labels Show K/D, BusStatus, ATAPI, 8B/10B, ReFrame, InFIS, and Phase labels. These labels have one bit corresponding to each symbol.

Other Labels Show other, less frequently used labels. Most of these labels contain the same information as other labels, decomposed into smaller pieces. For example, the K/D bits for all of the symbols in a state are normally shown as one label. When you select Other Labels, separate K/D labels corresponding to each symbol are displayed. Examine the Format tab of the logic analyzer to see how these additional labels relate to other labels.

See “Labels Mapped to Logic Analyzer Channels” on page 10 for more information on specific labels.

1. Select Apply to output the selected data columns.
2. Select Close to output the selected columns and close the Serial ATA Tool.

See Also

“To set up the Serial ATA Tool” on page 22

“To color data rows” on page 24

“To select the output data columns” on page 26

“To select which fields to display” on page 29

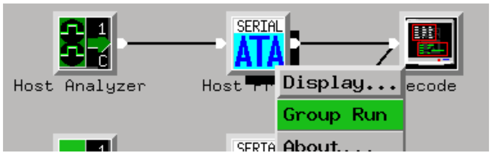
“Labels Mapped to Logic Analyzer Channels” on page 10

To select which fields to display

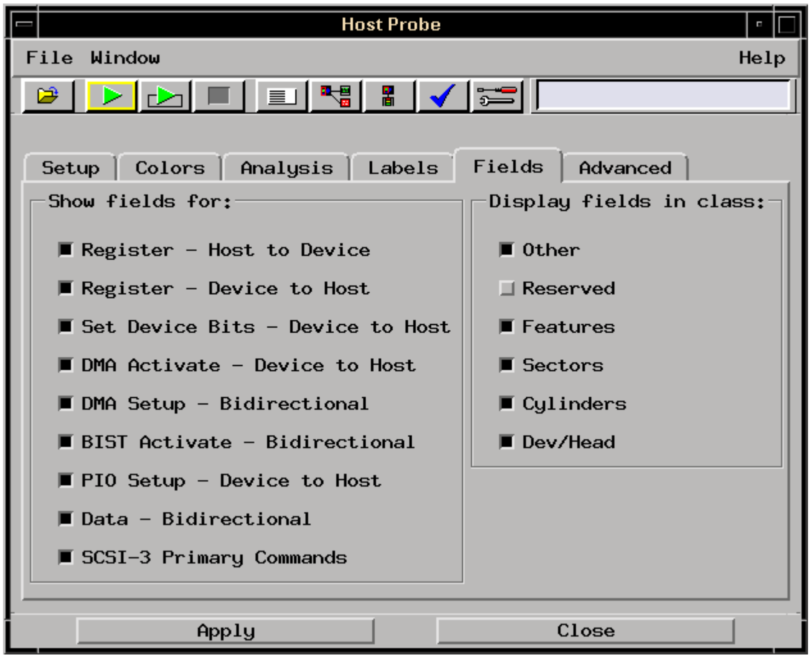
There are two ways to control whether or not to display the fields within a packet. First, you can control whether or not to decode the fields in link or data packets. If you want finer control, you can specify exactly which fields should be displayed.

To control decoding of fields by packet type

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Fields tab, select the types of packets which you would like decoded in the "Packet Decode" column of the output data:



Show fields for:

☑ Register - Host to Device
☑ Register - Device to Host
☑ Set Device Bits - Device to Host
☑ DMA Activate - Device to Host
☑ DMA Setup - Bidirectional
☑ BIST Activate - Bidirectional
☑ PIO Setup - Device to Host
☑ Data - Bidirectional
☑ SCSI-3 Primary Commands

Display fields in class:

☑ Other
☐ Reserved
☑ Features
☑ Sectors
☑ Cylinders
☑ Dev/Head

For packet types which are not selected, a one-line description of the packet will be displayed. For packet types which are selected, the fields in the packet will be decoded and displayed, subject to the selections made under the Fields tab and under the Advanced tab.

3. Select what kinds of information should be displayed for each packet:

Other All fields that are not in one of the remaining classes.

Reserved All fields that are reserved for future use.

Features All fields that contain the word "Features".

Sectors All fields that contain the word "Sectors".

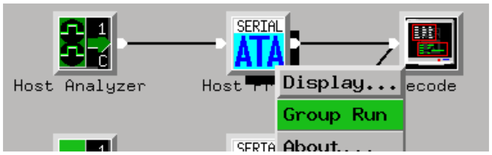
Cylinders All fields that contain the word "Cylinders".

Dev/Head All fields that contain the word "Dev/Head".

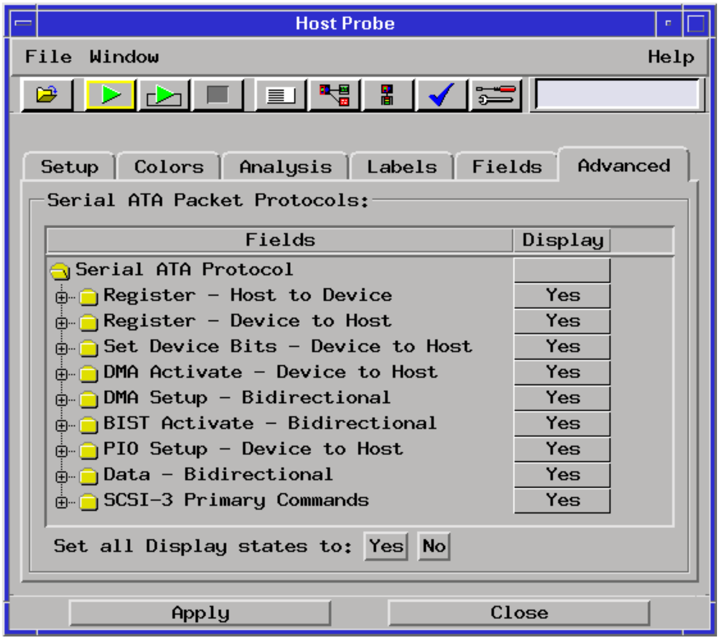
1. Select Apply to output data with the fields you selected. If a listing has already been acquired, the Serial ATA Tool will re-process the data.
2. Select Close to output data with the fields you selected and close the Serial ATA Tool. If a listing has already been acquired, the tool will re-process the data.

To control display of individual fields

1. Display the Serial ATA Tool.



2. In the Serial ATA Tool's Advanced tab, select the types of fields which you would like displayed in the "Packet Decode" column of the output data:



Serial ATA Packet Protocols:

Fields	Display
Serial ATA Protocol	
⊞ Register - Host to Device	Yes
⊞ Register - Device to Host	Yes
⊞ Set Device Bits - Device to Host	Yes
⊞ DMA Activate - Device to Host	Yes
⊞ DMA Setup - Bidirectional	Yes
⊞ BIST Activate - Bidirectional	Yes
⊞ PIO Setup - Device to Host	Yes
⊞ Data - Bidirectional	Yes
⊞ SCSI-3 Primary Commands	Yes

3. Select "+" to expand the fields for a FIS type.

4. Select Yes to display a field, or No to suppress it.

5. Select Apply to output data with the fields you selected. If a listing has already been acquired, the Serial ATA Tool will re-process the data.

6. Select Close to output data with the fields you selected and close the Serial ATA Tool. If a listing has already been acquired, the tool will re-process the data.

Example

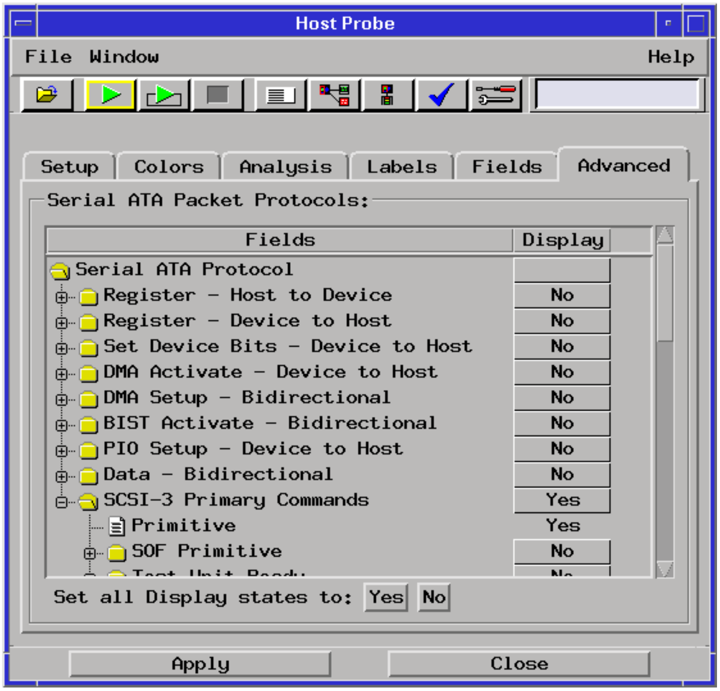
To display only SCSI-3 primitives:

1. Check that nothing in the Filter or Fields tabs is set to prevent display of SCSI-3 commands.

2. In the Serial ATA Tool's Advanced tab, select "Set all Display states to No".

3. Select the "+" in front of "SCSI-3 Primary Commands".

4. Select the "No" after "Primitive" to change it to "Yes".



Serial ATA Packet Protocols:

Fields	Display
Serial ATA Protocol	
Register - Host to Device	No
Register - Device to Host	No
Set Device Bits - Device to Host	No
DMA Activate - Device to Host	No
DMA Setup - Bidirectional	No
BIST Activate - Bidirectional	No
PIO Setup - Device to Host	No
Data - Bidirectional	No
SCSI-3 Primary Commands	Yes
Primitive	Yes
SOF Primitive	No
Test Unit Ready:	No

Set all Display states to: Yes No

5. Select Apply.

Helpful suggestions

- If you want to look at only a few fields, go to the Advanced tab and set all display states to No, then set just the fields of interest to Yes.
- The options selected in the Fields and Advanced tabs will take effect after the next run of the analyzer or after the "Apply" button is selected. If you want to change several options it will be faster to change all the options in all the tabs that you want to change and then select "Apply" once, rather than selecting "Apply" after each individual option is changed.
- If you have made many changes, consider saving the logic analysis system's configuration. The selections you made will be saved as part of the configuration.

  If you save a configuration, change the protocol file, then load the configuration, this is what will happen: The buttons will be matched by name, using the entire protocol path to the given field as its name. If a name in some part of the path is changed (for example, changing "SCSI-3 Primary Commands" to "Improved SCSI-3 Primary Commands"), then none of the buttons under that path will be restored. If new protocols or fields are added without changing the names of the existing protocols and fields, then the existing fields are restored and the new fields are added and set to Yes.

- To see a protocol or a field in the listing, it must be enabled in both the Fields and the Advanced tab.

See Also

"To set up the Serial ATA Tool" on page 22

"To color data rows" on page 24

"To select the output data columns" on page 26

"To select other columns to display" on page 28

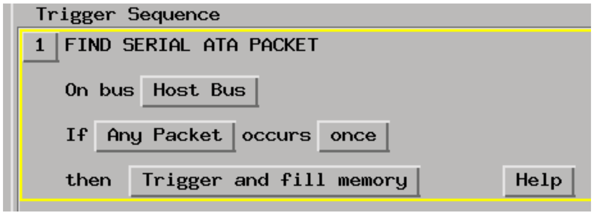
Modifying Protocol Definitions

This section shows you how to modify the protocol definitions used with the Serial ATA Tool.

- "To access the protocol definitions" on page 35
- "To modify a protocol definition" on page 36
- "To reload the protocol definitions" on page 37
- "To reset the protocol definitions" on page 38

To access the protocol definitions

1. In the "Find Serial ATA Packet" trigger function, select the bus button.



Trigger Sequence

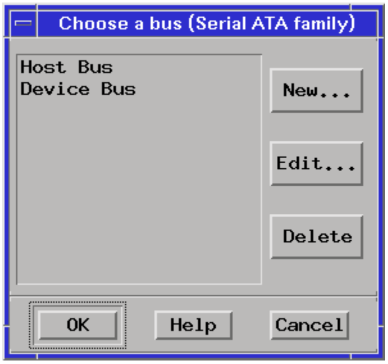
1 FIND SERIAL ATA PACKET

On bus Host Bus

If Any Packet occurs once

then Trigger and fill memory

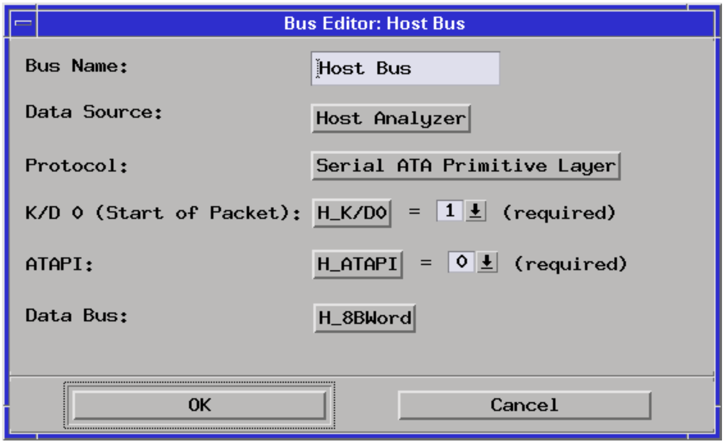
2. In the Choose a Bus dialog, either select New... to define a new bus or select the bus you want to edit from the list and select the Edit... button.



Choose a bus (Serial ATA family)

Host Bus
Device Bus

3. In the Bus Editor dialog, select the Protocol button.



Bus Name: Host Bus

Data Source: Host Analyzer

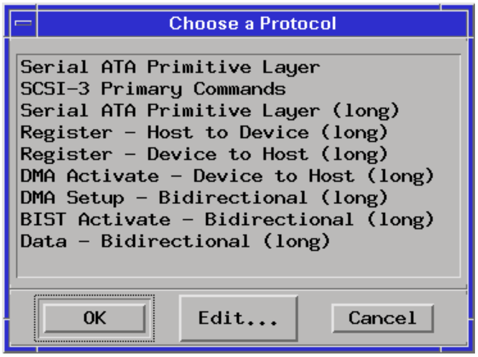
Protocol: Serial ATA Primitive Layer

K/D 0 (Start of Packet): H_K/D0 = 1 (required)

ATAPI: H_ATAPI = 0 (required)

Data Bus: H_8BWord

4. In the Choose a Protocol dialog, select the Serial ATA Primitive Layer.



Serial ATA Primitive Layer
SCSI-3 Primary Commands
Serial ATA Primitive Layer (long)
Register - Host to Device (long)
Register - Device to Host (long)
DMA Activate - Device to Host (long)
DMA Setup - Bidirectional (long)
BIST Activate - Bidirectional (long)
Data - Bidirectional (long)

See Also

"To modify a protocol definition" on page 36

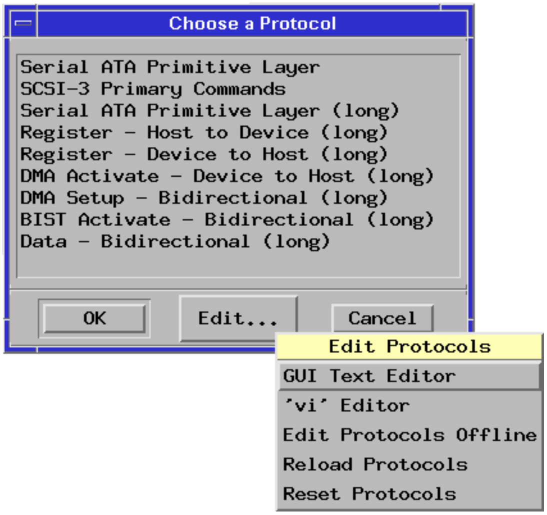
"To reload the protocol definitions" on page 37

"To reset the protocol definitions" on page 38

To modify a protocol definition

1. In the Choose a Protocol dialog (see "To access the protocol definitions" on page 35), select the Edit... button, and choose either GUI Text Editor, 'vi'

Editor, or Edit Protocols Offline.



Choose a Protocol

```
Serial ATA Primitive Layer
SCSI-3 Primary Commands
Serial ATA Primitive Layer (long)
Register - Host to Device (long)
Register - Device to Host (long)
DMA Activate - Device to Host (long)
DMA Setup - Bidirectional (long)
BIST Activate - Bidirectional (long)
Data - Bidirectional (long)
```

2. When you are done editing protocol definitions, choose the Reload Protocols command to tell the Serial ATA Tool that changes have been made (see “To reload the protocol definitions” on page 37).

About the Protocol Definitions File

The protocol definitions file is:

`/logic/auxiliary/TeleCom/protocols/infiniband.pro`

When the logic analysis system's file system is mounted by (or made available to) another computer on the network, you can use any ASCII text editor to modify the protocol definitions file. Don't forget to use the Reload Protocols command afterward.

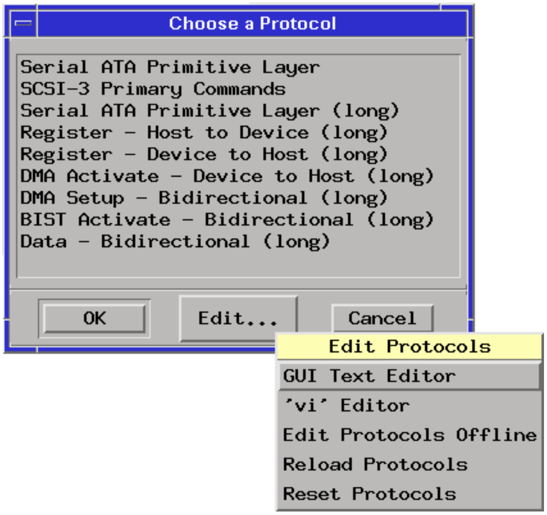
See Also

“Protocol Definition Syntax” on page 42

To reload the protocol definitions

After the protocol definitions file has been edited with "vi" (or some other ASCII text editor on a networked computer), you must use the Reload Protocols command to tell the Serial ATA Tool that changes have been made.

1. In the Choose a Protocol dialog (see “To access the protocol definitions” on page 35), select the Edit... button, and choose Reload Protocols.

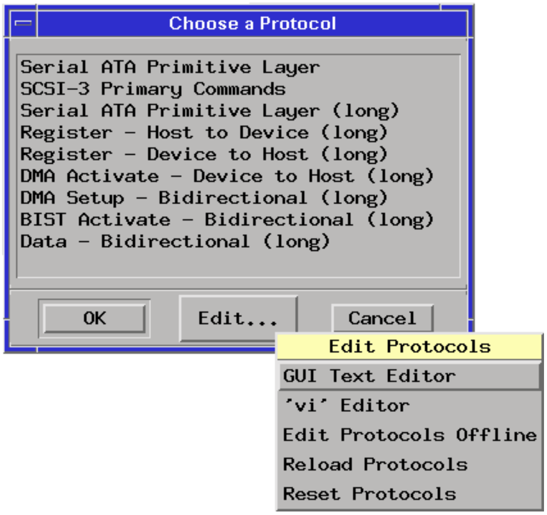

Choose a Protocol
Serial ATA Primitive Layer
SCSI-3 Primary Commands
Serial ATA Primitive Layer (long)
Register - Host to Device (long)
Register - Device to Host (long)
DMA Activate - Device to Host (long)
DMA Setup - Bidirectional (long)
BIST Activate - Bidirectional (long)
Data - Bidirectional (long)
OK
Edit...
Cancel
Edit Protocols
GUI Text Editor
'vi' Editor
Edit Protocols Offline
Reload Protocols
Reset Protocols

To reset the protocol definitions

If you want to return to the factory default protocol definitions, you can use the Reset Protocols command.

CAUTION: Resetting the protocol definitions will delete any new or modified protocol definitions.

1. In the Choose a Protocol dialog (see “To access the protocol definitions” on page 35), select the Edit... button, and choose Reset Protocols.



Serial ATA Primitive Layer
SCSI-3 Primary Commands
Serial ATA Primitive Layer (long)
Register - Host to Device (long)
Register - Device to Host (long)
DMA Activate - Device to Host (long)
DMA Setup - Bidirectional (long)
BIST Activate - Bidirectional (long)
Data - Bidirectional (long)

2

Reference

Protocol Definition Syntax

This section describes the protocol definition syntax.

Protocol definitions are used by the Serial ATA Tool to decode captured data.

- “Protocol Block” on page 42
- “Physical Layer Definition” on page 43
- “Header Block” on page 43
- “Field Definition” on page 44
  - “Data” on page 44
  - “Symbols” on page 44
  - “ProtocolIndicator” on page 45
- “Numeric Values” on page 46

Example

Here is the definition for the Serial ATA SOF protocol:

```
 Protocol
{
  Name "SOF"
  Header
  {
    FT "FIS Type" 8 Hex ProtocolIndicator
    {
      "Register - Host to Device"        #h27
      "Register - Device to Host"        #h34
      "Set Device Bits - Device to Host" #ha1
      "DMA Activate - Device to Host"    #h39
      "DMA Setup - Bidirectional"        #h41
      "BIST Activate - Bidirectional"    #h58
      "PIO Setup - Device to Host"       #h5f
      "Data - Bidirectional"             #h46
    }
  }
}
```

See Also

“Displaying the Data” on page 22

Protocol Block

Syntax

```
Protocol
```

```
{
  Name "protocol name"
}
```

A protocol definition starts with the keyword "Protocol"; then, the definition is enclosed in brackets "{ }".

The first line inside defines the name of the protocol (which must be enclosed in double quotes).

Example

```
Protocol
{
  Name "Serial ATA Primitive Layer"
}
```

See Also

"Physical Layer Definition" on page 43

"Header Block" on page 43

Physical Layer Definition

Syntax

```
PhysicalLayer 1
```

After the Name definition in a Protocol block is an optional physical layer definition.

The "PhysicalLayer" definition is only required if the protocol can be used as a MAC layer or Physical layer. In other words, it is used for protocols that are at the bottom of the protocol stack on a Serial ATA bus.

Example

```
PhysicalLayer  1
```

Header Block

After the Protocol block and possibly a physical layer definition is the list of fields in the header, enclosed in the "Header { }" block.

Example

```
Header
{
  field definitions
}
```

See Also

“Field Definition” on page 44

Field Definition

Syntax

```
'Mnemonic' "'Full Name'" 'Width(in bits)' 'Format' 'Type'
```

For each field defined in a Header block, there is a line with a short 2- or 3-letter Mnemonic, a full name (enclosed in double-quotes), the length of the field (in bits), a format specification, and a type indicator.

The choices for field format are:

- Binary
- Octal
- Hex
- Decimal

The choices for field type are:

- “Data” on page 44
- “Symbols” on page 44
- “ProtocolIndicator” on page 45

Example

```
VL  "Virtual Lane"  4  Decimal  Data
```

This is the third field in the Infiniband Link Packet header, the Virtual Lane. It is 4 bits long. It should be displayed as a Decimal value, and it's a "Data" field.

Data

Data says to display the numeric value of the field in the format specified.

See Also

“Field Definition” on page 44

Symbols

The Symbols field type says there is a table of text names for various values in the field. If a value matches one of these, the decoder will display the name of the value, instead of the numeric value. Otherwise, the field will be displayed as a numeric value, in the format specified.

Example

Here is the "Precedence" field from an IP (Internet Protocol) definition:

```
PR  "Precedence"  3  Hex  Symbols
{
  "Routine"               0
  "Priority"              1
  "Immediate"             2
  "Flash"                 3
  "Flash Override"        4
  "Internetwork Control"  6
  "Network Control"       7
}
```

Notice that the "Symbols" field type is followed by a symbol table description, enclosed in "{ }" brackets. For each symbol, there is a name (in double-quotes) and a numeric value.

See Also

“Field Definition” on page 44

“Numeric Values” on page 46

ProtocolIndicator

The ProtocolIndicator field type is how a protocol references the next layer of the protocol stack.

The syntax for this type is exactly the same as for “Symbols” on page 44, except that each symbol's name is the name of another protocol, which must also be defined in the protocol file.

Example

Here is the "Length/Type" field of the Ethernet protocol:

```
PT  "Length/Type"  16  Hex  ProtocolIndicator
{
  "Internet Protocol"            #h0800
  "ARP Request"                  #h0806
  "ARP Response"                 #h0835
  "AppleTalk Datagram Protocol"  #h809B
  "Novell IPX"                   #h8137
  "IPS"                          #h2007
}
```

A value of 0x0800 in the length/type field indicates Internet Protocol, which must also be defined in the protocol definitions file.

NOTE: While many protocols are already defined, not all possible values and protocols have been included in the protocol definition.

See Also

“Field Definition” on page 44

“Symbols” on page 44

“Numeric Values” on page 46

Numeric Values

Numeric values can be plain decimal numbers, or they can be hex, octal, or binary numbers in the following formats:

```
For Hex,   a number is specified like this:  #hff00
For Binary, like this:                       #b1111111100000000
For Octal,  like this:                       #q177400
```

The reason for these somewhat unusual formats is because the letter 'X' often means "Don't Care" in the logic analysis system, so 0xff could be misinterpreted as a value with a don't care digit. And, the 'q' in the Octal specifier is to avoid the similarity between the letter 'o' and the number '0'.

3

Concepts

Basics of 8B/10B Encoding

History

When communications technology started pushing transmission speeds, it started running into capacitance problems with copper cable. A copper wire is like a long string of capacitors connected in parallel. The longer the cable, the more capacitance. That capacitance takes low-frequency signals and charges up the "capacitors" in the wire, making it harder to lock into the high frequency communication data.

For example, if you're transmitting a large block of data, and the data is mostly 1's, it's like putting a DC voltage on the cable and charging up the capacitance in the wire. Then, when you suddenly send a burst of high-frequency change (say, alternating 1's and 0's, 1010101010....), the charge on the cable causes a sluggish response that causes the data to get lost.

To solve these problems, *block coding* was introduced. Block coding translates a block of data into a longer block of data that has more transitions between 1's and 0's. The 8B/10B block code adopted by Gigabit Ethernet and InfiniBand maps every byte (8 bits) into a 10-bit value that has 3-8 transitions and a balanced number of 1's and 0's. (The 8B/10B block code was designed by IBM in the mid-1980's and has been used in FibreChannel communication links between computers and mass storage devices.)

In addition to solving capacitace problems with copper wire, the 8B/10B encoding also has benefits with fiber optic communications. The high-frequency changes cause a laser's optical spectrum to broaden slightly, which results in reduced modal noise in multimode fiber links. The DC balance reduces heating effects in the laser and allows for simpler AC coupled receivers.

How it Works

Of the 1024 possible values in a 10-bit number, there are enough unique values that provide 3-8 transitions and a balanced number of 1's and 0's (usually five apiece) to represent the 256 possible values in an 8-bit number. With more transitions between 1's and 0's, you get the electrical and optical benefits that let you run longer cables and still have good signal integrity at high speeds.

The 10-bit codes for each 8-bit value are stored in a lookup table. Rather than using a mathematical formula to generate the 10-bit codes (or to decode them), a table of values is used, and the 8-bit or 10-bit value is used as an array index into the table.

Disparity

Not all of the 10-bit codes have five 1's and five 0's. Some have six 1's and four 0's, and some have four 1's and six 0's. To keep the number of 1's and 0's in the bit stream balanced, the 8B/10B coding keeps track of whether more 1's or 0's have been transmitted (*running disparity*), and it maps alternate codings for each 8-bit value which will keep the 1's and 0's in the bit stream balanced.

Disparity is calculated like this:

- The 10-bit number is broken into the first 6 bits, and the last 4 bits.
- If the first six bits have more 1's this is called a *positive running disparity*.
- If the first six bits have a positive disparity, the last 4 bits should have a *neutral* or *negative disparity*. Neutral is when there are the same number of 1's and 0's. When a value is neutral, the running disparity is kept from the last non-neutral value.

So really, there are two lookup tables: one for when the running disparity at the end of the last character was positive, and one for when the running disparity is negative.

Character Names

The 10-bit codes for 8-bit values are often referred to using character names that come from the first 5 bits of the 8-bit value separated from the last 3 bits. For example:

```
D28.2    010 11100
```

D28.2 represents the encoding for the binary value above, where 28 is the decimal representation of the first 5 bits, and 2 is decimal representation of the last 3 bits.

Special Characters

In addition to the 10-bit codes for the 256 8-bit values, there are a few extra 10-bit codes called *special characters*. Special characters are used for data delimiters like Start of Packet, End of Packet, Idle, and configuration messages.

These special characters are often described with character names, but they use a "K" character instead of a "D". The special characters are:

```
K28.0
K28.1
K28.2
K28.3
K28.4
K28.5
K28.6
K28.7
K23.7
K27.7
K29.7
K30.7
```

Glossary

8B/10B encoding A block coding scheme that maps 8-bit data values to 10-bit data values which have 3-8 transitions between 1's and 0's and a balanced number of 1's and 0's. A *running disparity* is calculated to keep track of the balance. One 10-bit value is used when there is *positive disparity*, and a different 10-bit value if there is *neutral* or *negative disparity*. The 8B/10B block code was designed by IBM in the mid-1980's and is used in FibreChannel, InfiniBand, and Gigabit Ethernet.

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.

cell The basic unit of transmission in an ATM network. It is a fixed-size *packet* of 53 bytes, made up of 5 header bytes and 48 payload bytes.

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.

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.

CRC (Cyclic Redundancy Check)

A common technique for detecting data transmission errors.

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 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.

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.

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*.

disparity A calculation of the balance of 1's and 0's in a transmitted data stream. Disparity is used in *8B/10B encoding* to determine which of two possible 10-bit values should be used to represent a particular 8-bit value.

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 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.

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*.

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.

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.

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.

link In the InfiniBand architecture, a link is a full duplex transmission path between any two network fabric elements, such as channel adapters or switches.

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.

MAC layer The Medium Access Control layer is one of two layers that make up the Data Link Layer of the *OSI Reference Model*. The MAC layer is responsible for moving data packets to and from one Network Intercafe Card (NIC) to another across a shared channel.

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 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*.

OSI Reference Model The Open System Interconnection Reference Model is an ISO standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station, proceeding to the bottom (physical) layer, over the channel to the next station, and back up the hierarchy. Logic analyzers typically capture data at the *physical layer* or *MAC layer*.

packet A piece of a message transmitted over a packet-switching network, switch fabric, or multiplexed with other packets (like in an MPEG-2 transport stream). A packet has a *header* which identifies the packet and a *payload* which contains the actual data. Packets are also sometimes called *cells*.

packetized data Data that has been broken down into smaller pieces for transmission over a packet-switching network or switch fabric, or for multiplexing with other data streams (like in an MPEG-2 transport stream).

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 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*.

physical layer The first layer of the *OSI Reference Model* which manages placing data on and taking data off the transmission medium. In reference to protocol definitions, physical layer describes a *protocol* that is used at the bottom of the *protocol stack* on a data bus.

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.

port In the InfiniBand architecture, a port is a location on a channel adapter or switch to which a *link* connects.

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.

protocol stack A set of protocol layers that work together. The *OSI Reference Model* that defines seven protocol layers is often called a stack, as is the set of TCP/IP protocols that define communications over the internet.

protocol An agreed-upon format for transmitting data between two devices. The protocol determines: the type of error checking, data compression, encoding, how sending devices indicate they have finished sending a message, and how receiving devices indicate they have received a messaage.

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.

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.

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

Publication Number: 5988-9085EN
January 1, 2003

Agilent Technologies