Help Volume

Toolsets: InfiniBand (Agilent Technologies N4206A)

The InfiniBand Toolset adds protocol-based display and error detection capabilities to the logic analyzer. This toolset helps you look at InfiniBand data streams traveling across 10-bit parallel data buses going to, or coming from, serializer/deserializer chips.

The InfiniBand Toolset decodes 8B/10B encoded data on the 10-bit data bus. It does not work with 10-bit parallel buses whose data has already been decoded in hardware.

The InfiniBand Toolset understands Data Packet and Link Packet protocols. You can modify these protocol definitions if necessary.

NOTE: The InfiniBand Toolset will work with any Agilent Technologies logic analyzer whose state analysis speeds support the the speed of the 10-bit data bus.

"Installation and Licensing" on page 8

- "To check if the InfiniBand Toolset is installed" on page 8
- "To install the InfiniBand Toolset" on page 9
- "To license the InfiniBand Toolset" on page 10

"Probing the Device Under Test" on page 12

- "Step 1. Connect the logic analyzer probes" on page 12
- "Step 2. Select the state sampling mode" on page 12
- "Step 3. Label the logic analyzer channels" on page 13

"Capturing the Data" on page 15

- "To trigger on special 8B/10B codes" on page 15
- "To trigger with a second analyzer" on page 17
- "To trigger with another instrument" on page 17
- "To run the measurement" on page 19

"Displaying the Data" on page 20

- "To connect the InfiniBand Tool (in the workspace)" on page 20
- "To set up the InfiniBand Tool" on page 21
- "To select the output data columns" on page 22
- "To filter data rows" on page 26

"Modifying Protocol Definitions" on page 30

- "To modify a protocol definition" on page 30
- "To reload the protocol definitions" on page 31
- "To reset the protocol definitions" on page 32

Reference

- "Protocol Definition Syntax" on page 36

Concepts

- "Passively Probing an InfiniBand System" on page 42
- "Basics of 8B/10B Encoding" on page 54

See Also

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

Glossary (see page 57)

Contents

Using the InfiniBand Toolset

1 Installation Guide

Installation and Licensing 8
To check if the InfiniBand Toolset is installed 8
To install the InfiniBand Toolset 9
To license the InfiniBand Toolset 10

2 Task Guide

Probing the Device Under Test 12
Step 1. Connect the logic analyzer probes 12
Step 2. Select the state sampling mode 12
Step 3. Label the logic analyzer channels 13

Capturing the Data 15
To trigger on special 8B/10B codes 15
To trigger with a second analyzer 17
To trigger with another instrument 17
To run the measurement 19

Displaying the Data 20
To connect the InfiniBand Tool (in the workspace) 20
To set up the InfiniBand Tool 21
To select the output data columns 22
To filter data rows 26

Modifying Protocol Definitions 30
To modify a protocol definition 30
To reload the protocol definitions 31
To reset the protocol definitions 32

Contents

3 Reference

Protocol Definition Syntax 36
Protocol Block 36
Physical Layer Definition 37
Header Block 37
Field Definition 38
Numeric Values 40

4 Concepts

Passively Probing an InfiniBand System 42
InfiniBand Logic Analysis Support 42
Termination Networks 45
Logic Analyzer Connectors 51
Appendix 53

Basics of 8B/10B Encoding 54

Glossary

Index

1

Installation Guide

Installation and Licensing

The InfiniBand Toolset software may already be installed with logic analysis system software (if they are ordered at the same time).

If the InfiniBand Toolset is not already installed, you need to install it.

Once the InfiniBand Toolset is installed, you must obtain a license to use it beyond the demo period.

- "To check if the InfiniBand Toolset is installed" on page 8
- "To install the InfiniBand Toolset" on page 9
- "To license the InfiniBand Toolset" on page 10

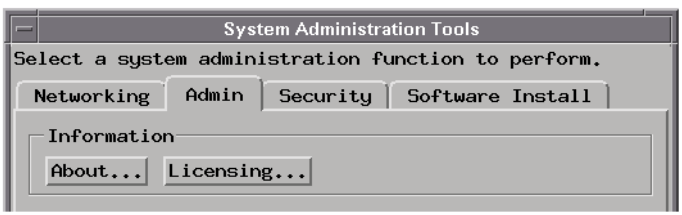
See Also

Licensing Policy for the Logic Analysis System and Tool Sets (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

To check if the InfiniBand Toolset is installed

1. In the main logic analysis system window, select the System Admin button  .

2. In the Admin tab of the System Administration Tools dialog, select the Licensing... button.



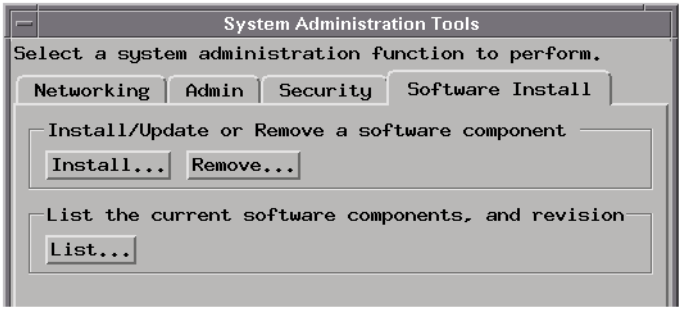
3. In the Tool Sets tab of the Licensing Dialog, look for the "N4206A - InfiniBand Toolset" product.

- If it is not listed, go to "To install the InfiniBand Toolset" on page 9.
- If it is listed, go to "To license the InfiniBand Toolset" on page 10.

To install the InfiniBand Toolset

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

1. Turn on the CD-ROM drive first; then, turn on the logic analysis system.
2. Insert the CD-ROM in the drive.
3. In the main logic analysis system window, select the System Admin button  .
4. In the Software Install tab of the System Administration Tools dialog, select Install....



5. Change the media type to "CD-ROM" if necessary, and select Apply.
6. Select the "AUXILIARY-SW" selection to open it.
7. Select the InfiniBand Toolset; then, select the Install button.

   The dialog will display "Progress: completed successfully" when the installation is complete.
8. Select the Close button to close the Software Install window.
9. Select the Close button to close the System Administration Tools window.

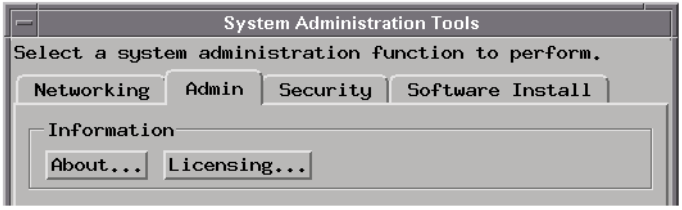
10. Go to “To license the InfiniBand Toolset” on page 10.

See Also

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

To license the InfiniBand Toolset

1. To obtain a password, contact the password center listed on the Entitlement Certificate you received after purchasing the InfiniBand Toolset.
2. In the main logic analysis system window, select the System Admin button  .
3. In the Admin tab of the System Administration Tools dialog, select the Licensing... button.



4. In the Tool Sets tab of the Licensing Dialog, enter the password into the field for the "N4206A - InfiniBand Toolset" product.
5. Select OK to close the Licensing Dialog.
6. Select Close to close the System Administration Tools window.
7. Restart the session.

You are now ready to use the InfiniBand Toolset.

2

Task Guide

Probing the Device Under Test

To probe the device under test (also known as a target system):

- "Step 1. Connect the logic analyzer probes" on page 12
- "Step 2. Select the state sampling mode" on page 12
- "Step 3. Label the logic analyzer channels" on page 13

See Also

"Passively Probing an InfiniBand System" on page 42

Step 1. Connect the logic analyzer probes

To use the InfiniBand Toolset, you need to connect logic analyzer probe channels to:

- A 10-bit parallel data bus (going to, or coming from, an InfiniBand serializer/deserializer chip).

  The InfiniBand Toolset decodes 8B/10B encoded data, so it won't work with 10-bit parallel buses whose data has already been decoded in hardware.

You also need to connect a logic analyzer CLK input channel to:

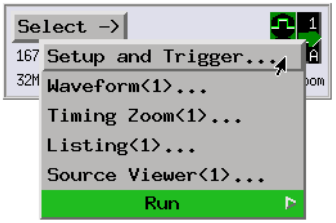
- A clock signal that identifies when the data bus is valid and should be sampled by the logic analyzer.

Next Step (see page 12)

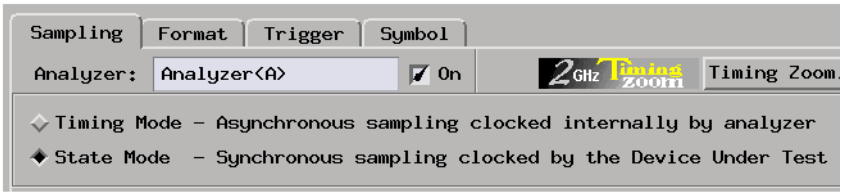
Step 2. Select the state sampling mode

When using the InfiniBand Toolset, the logic analyzer must be set up in the state (synchronous) sampling mode.

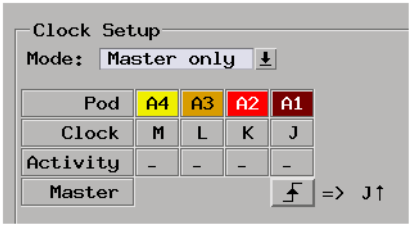
1. Choose the Setup... command.



2. In the Sampling tab, select the state mode.



3. Select the state speed configuration necessary for the speed of the 10-bit data bus.
4. Specify the state analyzer's sampling clock input.



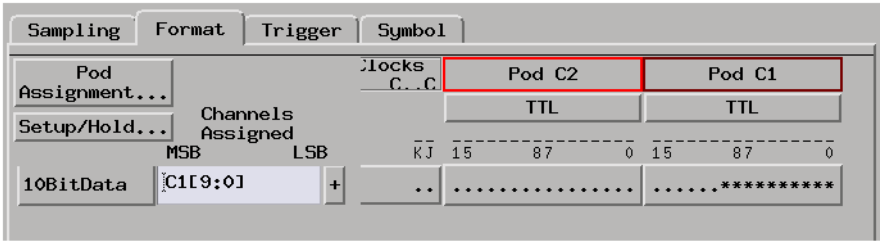
Next Step (see page 13)

Step 3. Label the logic analyzer channels

The InfiniBand Tool requires you to set one 10-bit data label for the InfiniBand data bus channels.

To label the logic analyzer channels:

1. In the Format tab of the logic analyzer's setup window, insert or rename labels so that there is a 10-bit data label.
2. For the 10-bit data label, specify which logic analyzer pod channels are probing the data bus.



See Also "Capturing the Data" on page 15

Capturing the Data

You can set up simple triggers on 8B/10B control codes (the "K" special characters), like K27.7 for "start of Data Packet", by knowing the 10-bit encodings for these characters.

Because logic analyzer triggering resources are limited when configured for high-speed state mode sampling, it's difficult to set up more complex triggers. However, you can also:

1. Use the second logic analyzer or another instrument to set up a trigger on some event in the device under test.
2. Set up the logic analyzer that probes the 10-bit data bus to capture data when the second analyzer or other instrument's trigger occurs.
3. Run the measurement.
4. Correlate the captured data in the Listing display tool.

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

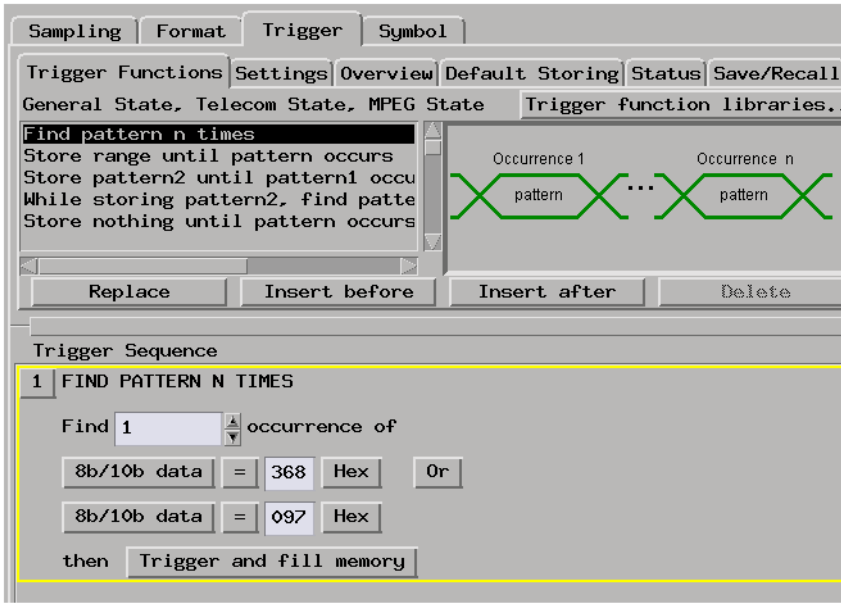
- "To trigger on special 8B/10B codes" on page 15
- "To trigger with a second analyzer" on page 17
- "To trigger with another instrument" on page 17
- "To run the measurement" on page 19

To trigger on special 8B/10B codes

Because special 8B/10B codes are assigned to things like "start of Data Packet", "start of Link Packet", etc., you can set up simple triggers by triggering on the special code values.

Remember that there are two possible codes for a "K" character, the one for positive running disparity and the one for neutral or negative running disparity, so you will have to set up to trigger on either code.

1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
2. Select the "Find pattern n times" trigger function and either replace the current trigger sequence level or insert a new level.
3. Select the 10-bit data bus label; then, insert another occurrence of the same label.
4. Select "Or" between the two labels.
5. Enter the special code values for positive and negative parity. For example, the special code values for "start of Data Packet", K27.7, are 368 Hex and 097 Hex. The special code values for "start of Link Packet", K28.2, are 0f5 Hex and 30a Hex.



See Also

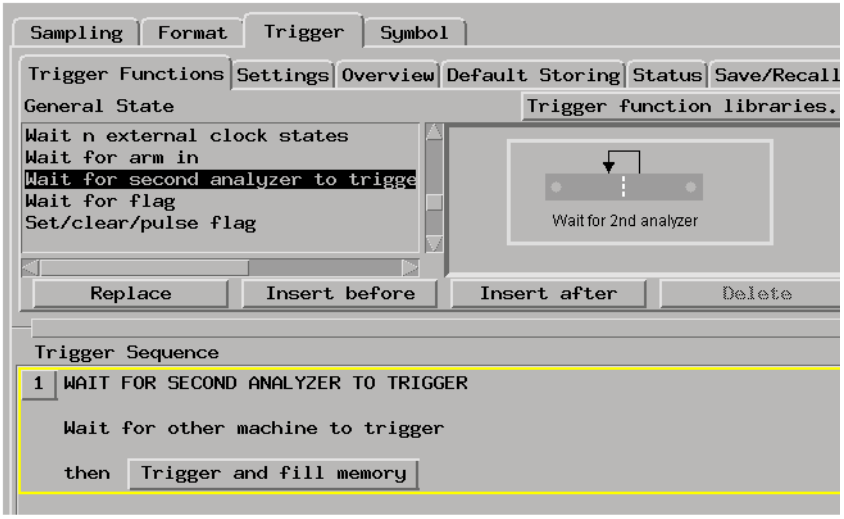
"To run the measurement" on page 19

"Displaying the Data" on page 20

To trigger with a second analyzer

If the second analyzer has been set up to probe other signals in the device under test and trigger on an event of interest:

1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
2. Select the "Wait for second analyzer to trigger" trigger function and either replace the current trigger sequence level or insert a new level.



See Also

"To run the measurement" on page 19

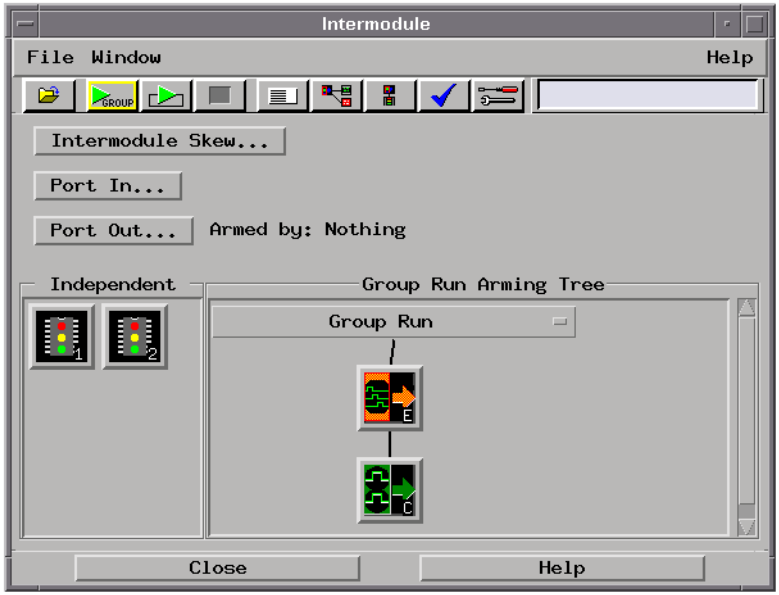
"Displaying the Data" on page 20

To trigger with another instrument

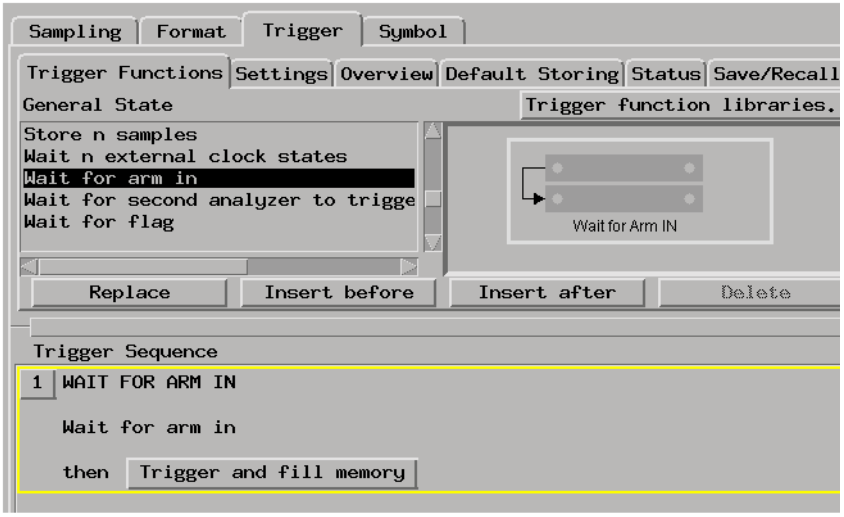
If another instrument has been set up to probe other signals in the device under test and trigger on an event of interest:

1. In the Intermodule window, set up the logic analyzer to be armed by the

other instrument.



2. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
3. Select the "Wait for arm in" trigger function and either replace the current trigger sequence level or insert a new level.



See Also

"To run the measurement" on page 19

"Displaying the Data" on page 20

To run the measurement

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

See Also

"Displaying the Data" on page 20

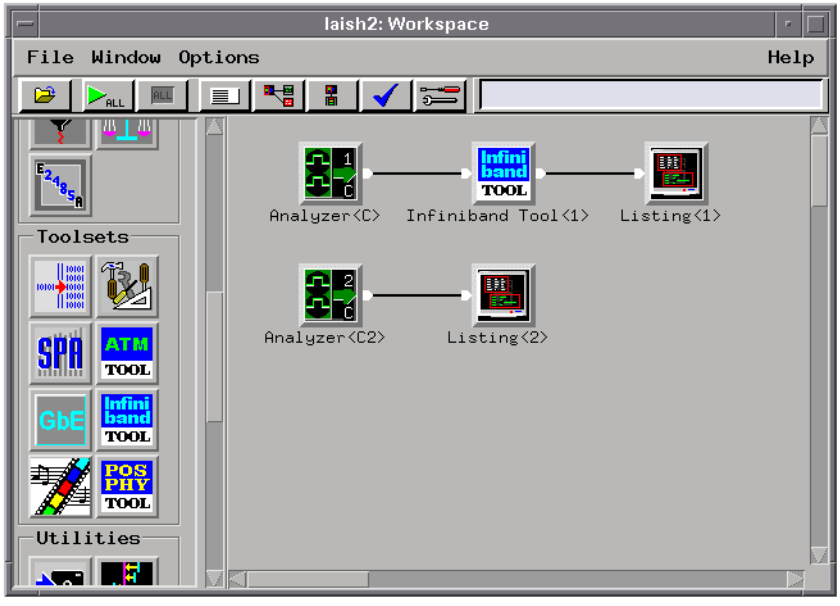
Displaying the Data

This section describes the options available when displaying data captured on the InfiniBand data bus.

- "To connect the InfiniBand Tool (in the workspace)" on page 20
- "To set up the InfiniBand Tool" on page 21
- "To select the output data columns" on page 22
- "To filter data rows" on page 26

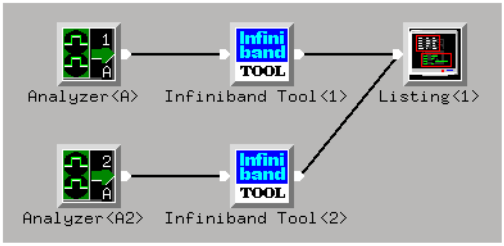
To connect the InfiniBand Tool (in the workspace)

1. Drag the InfiniBand Tool icon, and drop it between the logic analyzer instrument icon and the Listing display tool icon.



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

Filtering Limitation



NOTE: When the output of two or more InfiniBand Tools go to a single Listing display, filtering does not work.

See Also

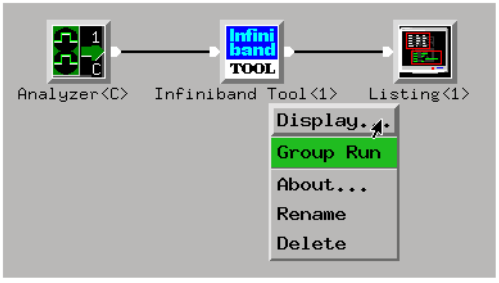
"To set up the InfiniBand Tool" on page 21

"To select the output data columns" on page 22

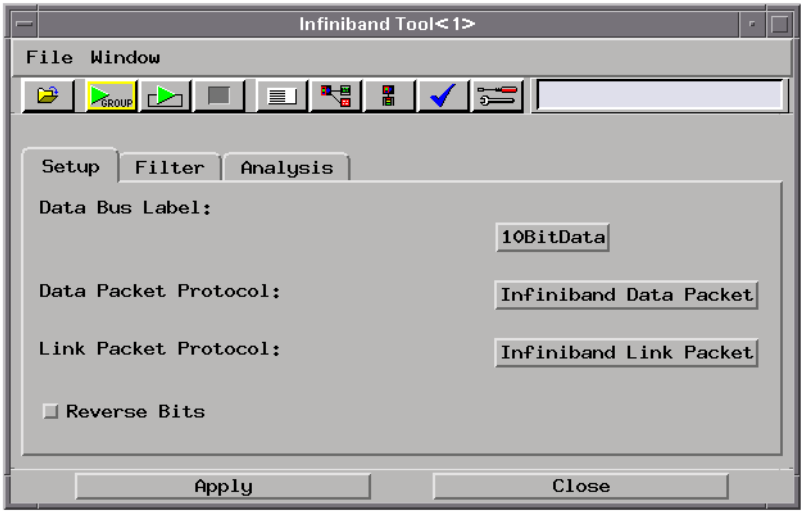
"To filter data rows" on page 26

To set up the InfiniBand Tool

1. Display the InfiniBand Tool.



2. In the InfiniBand Tool's Setup tab, select the appropriate options:



Data Bus Label Selects the 10-bit parallel data bus to decode.

Data Packet Protocol You should not need to modify this selection.

Link Packet Protocol You should not need to modify this selection.

Reverse Bits Specifies the ordering of the bits on the bus.

3. Select Apply to change the setup.
4. Select Close to change the setup and close the InfiniBand Tool.

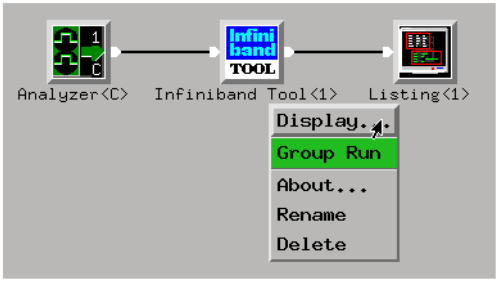
See Also

"To select the output data columns" on page 22

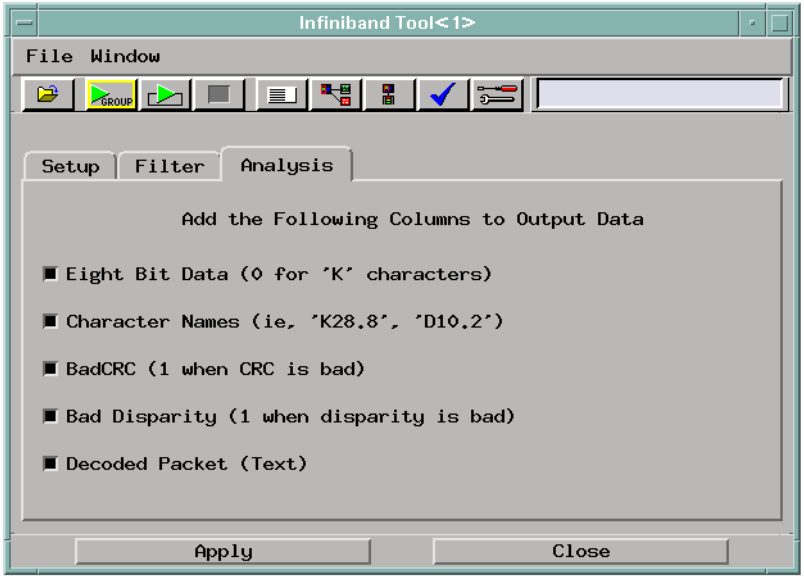
"To filter data rows" on page 26

To select the output data columns

1. Display the InfiniBand Tool.



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



Eight Bit Data The decoded bytes from the 10B codes. In the event of a special (or "K") character, the 8-bit value is set to zero because it is meaningless.

Character Names This text column contains the character name (for example, "K28.8" or "D10.2").

BadCRC This 1-bit column contains "1" when there is an error or "0"

otherwise. The InfiniBand tool checks:

- Invariant (32-bit) CRC in Data Packets.
- Variant (16-bit) CRC in Data Packets.
- Flow Control (16-bit) CRC in Link Packets.

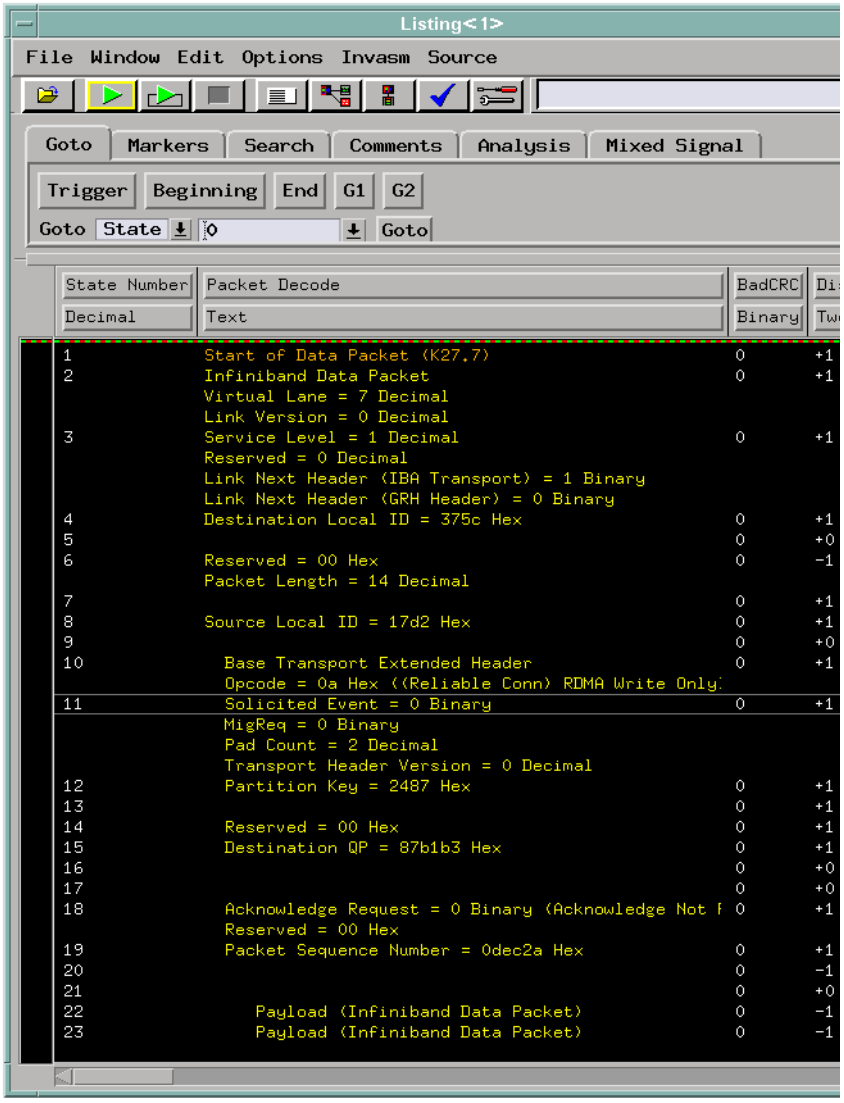
If a CRC code is bad, the "1" will be set on the same state as the first byte of the CRC code.

You can use this column to search for corrupt data by using the Search tab in the Listing tool. Search for BadCRD = 1.

Bad Disparity. This 2-bit signed integer value indicates the running disparity *at the end* of the current 10-bit value. To display this value as a signed integer, select the "Twos" (2's complement) format for the column in the Listing tool (instead of "Hex", for example).

Decoded Packet. This text column contains a description of the decoded data. Special codes (the "K" characters) and ordered sets are shown in a different color than packet headers and payloads.

3. Select Apply to output the selected data columns.



4. Select Close to output the selected columns and close the InfiniBand Tool.

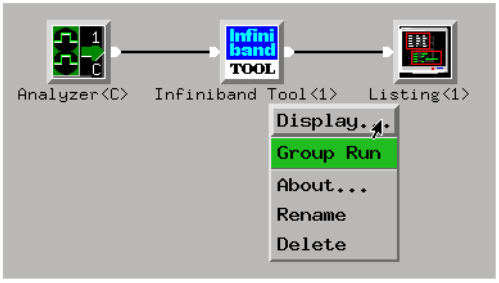
See Also

"To set up the InfiniBand Tool" on page 21

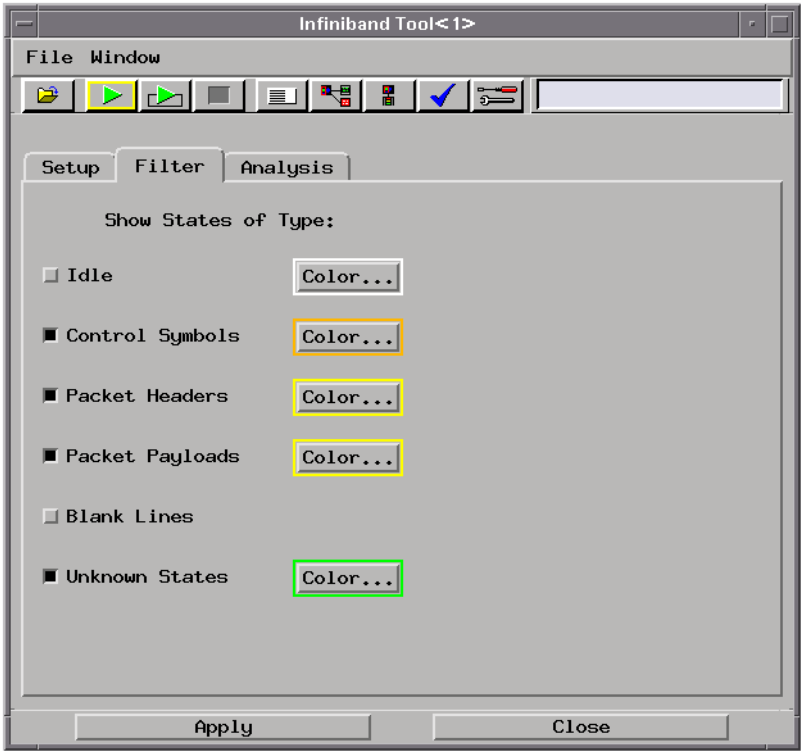
"To filter data rows" on page 26

To filter data rows

1. Display the InfiniBand Tool.



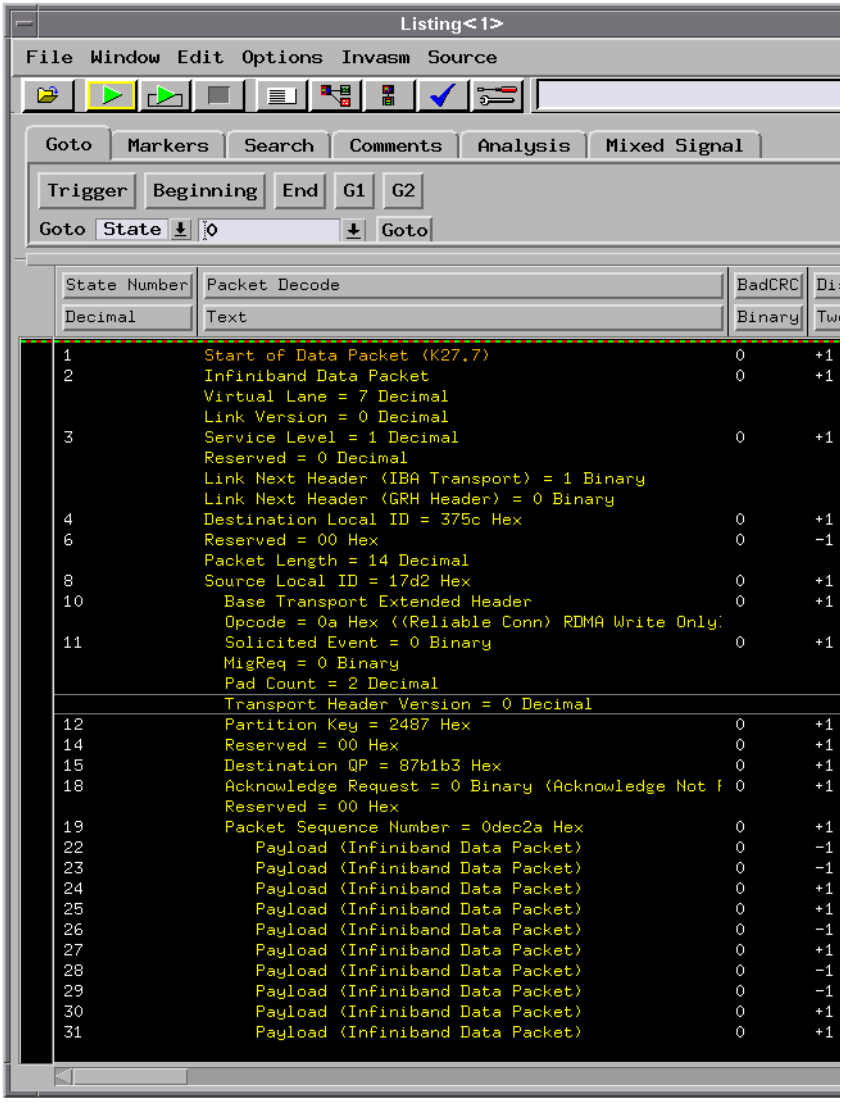
2. In the InfiniBand Tool's Filter tab, select the type of states that should be output as rows of data:



- Idle.
- Control Symbols.
- Packet Headers.
- Packet Payloads.
- Blank Lines.
- Unknown States.

To change the color of a state type, select the Color... button.

3. Select Apply to output the selected data rows.



4. Select Close to close the InfiniBand Tool.

See Also

Filtering Limitation in “To connect the InfiniBand Tool (in the workspace)” on page 20

“To set up the InfiniBand Tool” on page 21

"To select the output data columns" on page 22

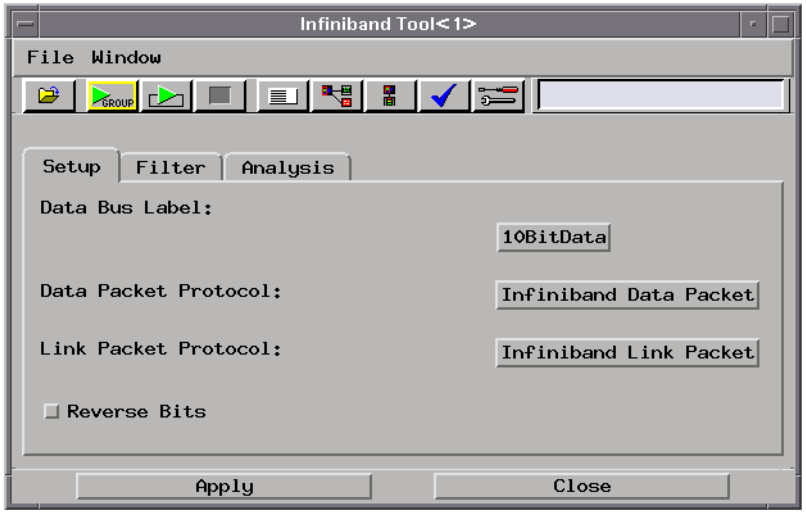
Modifying Protocol Definitions

This section shows you how to modify the protocol definitions used with the InfiniBand Toolset.

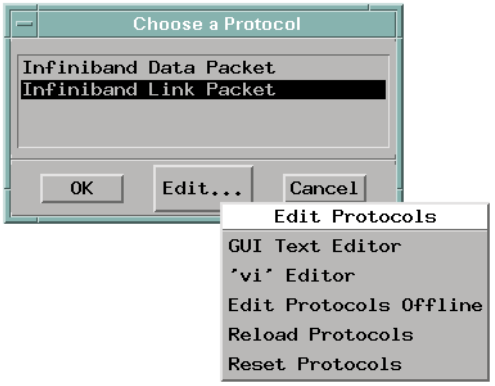
- "To modify a protocol definition" on page 30
- "To reload the protocol definitions" on page 31
- "To reset the protocol definitions" on page 32

To modify a protocol definition

1. In the InfiniBand Tool's Setup tab, select either the data packet protocol button or the link packet protocol button.



2. In the Choose a Protocol dialog, select the Edit... button, and choose either GUI Text Editor, 'vi' Editor, or Edit Protocols Offline.



3. When you are done editing protocol definitions, choose the Reload Protocols command to tell the InfiniBand Tool that changes have been made (see "To reload the protocol definitions" on page 31).

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.

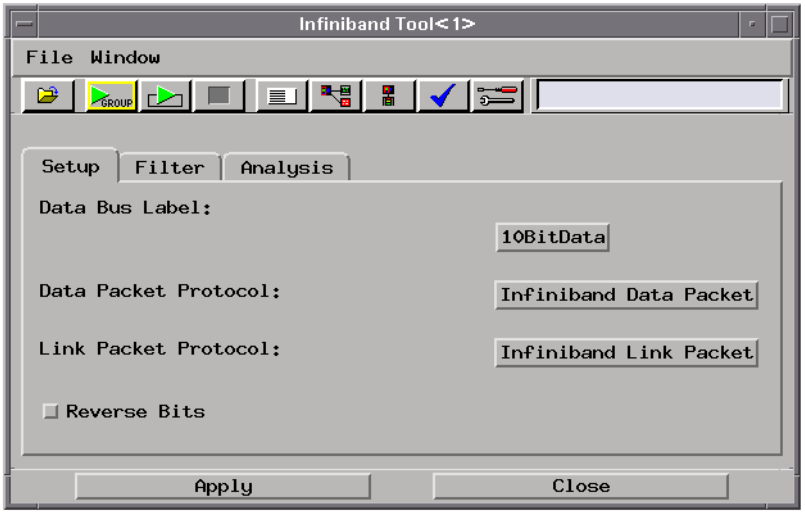
See Also

"Protocol Definition Syntax" on page 36

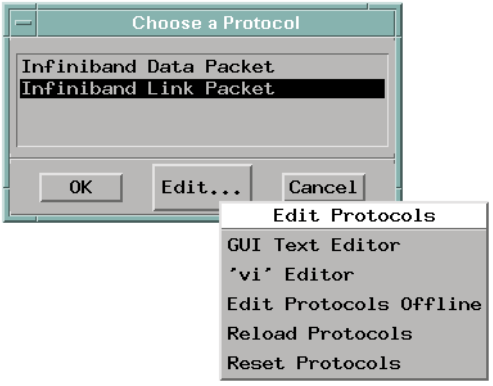
To reload the protocol definitions

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 InfiniBand Toolset that changes have been made.

1. In the InfiniBand Tool's Setup tab, select either the data packet protocol button or the link packet protocol button.



2. In the Choose a Protocol dialog, select the Edit... button, and choose Reload Protocols.

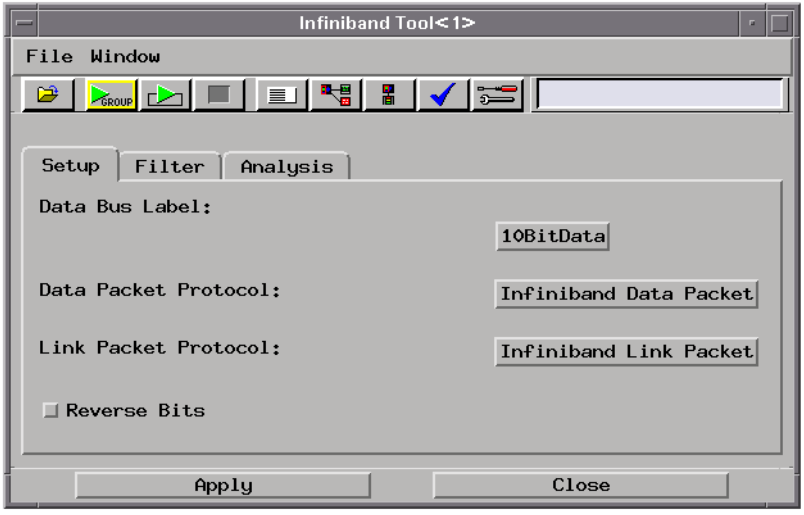


To reset the protocol definitions

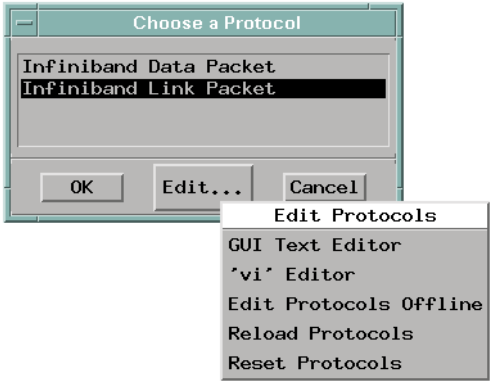
If you want to return to the factory default protocol definitions, you can use the Reset Protocols command.

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

1. In the InfiniBand Tool's Setup tab, select either the data packet protocol button or the link packet protocol button.



2. In the Choose a Protocol dialog, select the Edit... button, and choose Reset Protocols.



3

Reference

Protocol Definition Syntax

This section describes the protocol definition syntax.

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

- "Protocol Block" on page 36
- "Physical Layer Definition" on page 37
- "Header Block" on page 37
- "Field Definition" on page 38
  - "Data" on page 38
  - "Symbols" on page 38
  - "ProtocolIndicator" on page 39
- "Numeric Values" on page 40

Example

Here is the definition for the Infiniband Link Packet protocol:

```
Protocol
{
  Name "Infiniband Link Packet"
  PhysicalLayer 1
  Header
  {
    OP "Operand" 4 Hex Symbols
    {
      "IBA Flow Control (Normal)" #h0
      "IBA Flow Control (Init)" #h1
    }
    FCTBS "Flow Control Total Blocks Sent" 12 Decimal Data
    VL "Virtual Lane" 4 Decimal Data
    FCCL "Flow Control Credit Limit" 12 Decimal Data
  }
}
```

See Also

"Displaying the Data" on page 20

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 "Infiniband Link Packet"
}
```

See Also

"Physical Layer Definition" on page 37

"Header Block" on page 37

Physical Layer Definition

Syntax

```
PhysicalLayer 1
```

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

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 an InfiniBand data 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 38

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 38
- "Symbols" on page 38
- "ProtocolIndicator" on page 39

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 38

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 38

"Numeric Values" on page 40

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 38, 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 38

"Symbols" on page 38

"Numeric Values" on page 40

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'.

4

Concepts

Passively Probing an InfiniBand System

This application note describes the techniques required for passively probing a parallel InfiniBand channel with an Agilent Technologies 16700-series logic analysis system.

"InfiniBand Logic Analysis Support" on page 42

- "Terminology" on page 43
- "Generic Block Diagram" on page 43
- "Agilent Technologies HDMP-2634 SERDES ASICs" on page 44
- "Logic Analysis Features" on page 44

"Termination Networks" on page 45

- "Built-in Termination Networks" on page 46
- "Discrete Termination Networks" on page 46
- "Routing Techniques" on page 49

"Logic Analyzer Connectors" on page 51

- "Mictor Connectors" on page 51
- "Connector Placement Techniques" on page 51
- "Mictor Connector Pinout" on page 52

"Appendix" on page 53

- "Supported Logic Analyzer Modules for InfiniBand" on page 53
- "Probing Part Numbers and Ordering Information" on page 53
- "Supplemental Agilent Documentation" on page 53
- "Technical Support" on page 53

InfiniBand Logic Analysis Support

- "Terminology" on page 43
- "Generic Block Diagram" on page 43
- "Agilent Technologies HDMP-2634 SERDES ASICs" on page 44

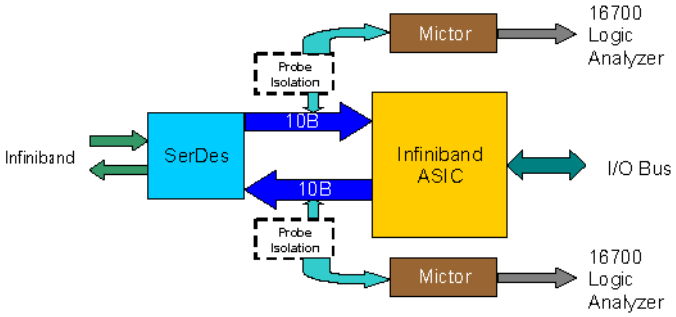
• "Logic Analysis Features" on page 44

Terminology

SerDes	Serializer / De-serializer ASIC (typically pronounced: Sir-Deez).
IBA	InfiniBand Architecture.
SUT	Signal Under Test.
SMT	Surface Mount.
PCB	Printed Circuit Board.
LVTTL	Low Voltage TTL Logic.
SSTL-2	Stub Series Terminated Logic (2.5 Volt).
Mictor	Amp Matched Impedance Connector (used with high-density logic analyzer cables).
8B/10B	InfiniBand data coding scheme.

Generic Block Diagram

Due to loading restrictions and logic analyzer performance, probing raw differential InfiniBand data with a logic analyzer is nearly impossible. The logic analyzer must probe the de-serialized data to passively measure traffic on Infiniband channels. For systems utilizing discrete SerDes chips, the parallel data is available to route to logic analyzer connectors. Mictor connectors, along with optional isolation circuitry to minimize logic analyzer loading on the parallel data, are connected to the 10-bit bus running between the SerDes and the Infiniband controller ASIC. A simplified block diagram of this connection scheme is shown:



Systems which contain the SerDes built into a custom ASIC will require an auxiliary bus to output the parallel data to the ASIC pins to be compatible with a logic analyzer. Without an external SerDes or auxiliary bus the 8B/10B traffic is not observable and cannot be probed by the logic analyzer.

Agilent Technologies HDMP-2634 SERDES ASICs

This app note supports probing and analysis of the Agilent Technologies HDMP-2634 SerDes.

The Agilent SerDes drives 8B/10B encoded data at 10-bits x 125 MHz (DDR) at an SSTL-2 logic level. The data is sampled by the logic analyzer on both the rising and falling edge of the clock for an effective data rate of 250 MHz. Data is displayed on the logic analyzer in raw 8B/10B encoded format, unless the InfiniBand Toolset is used to convert and decode the data.

Logic Analysis Features

When Mictor connectors are designed into into your InfiniBand HCA and TCA systems, you can use the 16700-series logic analysis system for full real-time acquisition of InfiniBand traffic along with concurrent transactions on other system buses such as PCI-X, CPU buses, PC266 DDR, and Rambus. Events on one bus can trigger measurements on all buses, and the events captured on each bus can be displayed in time-correlated views.

Triggering

When capturing 8B/10B Infiniband traffic, the logic analyzer will be

capturing data at 250 MHz. At this speed, the logic analyzer must be set to the 400 MHz state sampling mode (or the 333 MHz state sampling mode for the 16717), in which the number of available sequence levels is 6.

Generic Data Display

The logic analyzer contains a waveform viewer, and linear listing window to view sampled data. Both of these tools display the data in the raw format that was sampled by the logic analyzer.

InfiniBand Toolset

You can use the InfiniBand Toolset in the 16700-series logic analysis system to display decoded 8B/10B data and detect errors in the data stream.

Termination Networks

Termination networks are required for probing a digital signal. The termination network isolates the logic analyzer cabling and sensing circuitry from the signal under test and provides a high impedance, low capacitance load at the probe point.

Additional information about probing with Agilent Technologies logic analyzers can be found at www.agilent.com.

- "Built-in Termination Networks" on page 46
- "Discrete Termination Networks" on page 46
  - "Designing a Discrete RCR Termination Network" on page 47
  - "Equivalent Load" on page 48
  - "Extrapolating T1 and T2 Trace Lengths" on page 49
- "Routing Techniques" on page 49
  - "Series Terminated LVTTL" on page 49
  - "Load Terminated LVTTL" on page 50
  - "SSTL-2" on page 50
  - "SSTL-3" on page 50

Built-in Termination Networks

The first and simplest technique for probing the output of the SerDes ASIC is to use cabling from Agilent Technologies with the termination networks built into the cable tip (E5346A high-density cable).

Signals are simply routed on the printed-circuit board to Mictor connectors and terminated internally on the connection cable. Connector placement is critical to minimize the stub length to minimize reflections on signals with fast rise times. However, even the best routing cannot eliminate the stubs completely since the path to the connector and the cable tip is a stub.

The general rule of thumb is to keep the stub length less than 1/5 the rise time of the signal under test (SUT). Refer to the following equations for deciding whether to use built in termination networks.

```
lstub <= 1/5 * risetime / propspeed

lstub <= 1/5 * risetime / propspeed <= 1/5 * 1000ps / (150ps / in) <= 1.3 in.
(1ns risetime)

lstub <= 1/5 * risetime / propspeed <= 1/5 * 500ps / (150ps / in) <= 0.65 in.
(500ps risetime)
```

Whenever it is possible to route the signals within the previous constraints, use the built in termination networks. They are designed for the Agilent Technologies logic analysis systems and are compensated to provide flat frequency response between the signal under test and the logic analyzer input comparators.

The Agilent SerDes has a roughly 1-1.5 ns rise time, which require traces to be kept in between 1.3 - 2 inches in length.

Discrete Termination Networks

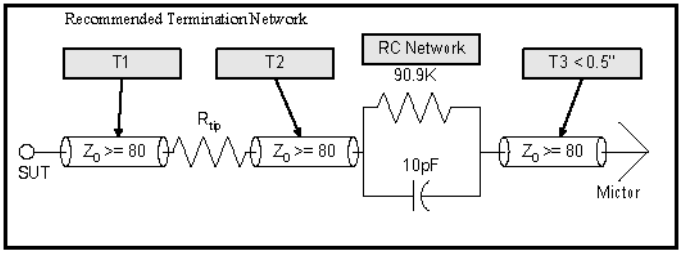
When routing constraints prohibit using the built-in termination networks, external components can be designed into the circuit to increase the length between the logic analyzer connector and the signal under test. However, this is only recommended under rare cases when routing issues prohibit the use of standard cabling. Discrete termination networks are complex to design and are beyond the scope of this document to treat in the general case. Two solutions are provided for signals with 500 ps and 1000 ps rise times.

The following sections describe the recommended termination network

and how to select components for a target system.

- "Designing a Discrete RCR Termination Network" on page 47
- "Equivalent Load" on page 48
- "Extrapolating T1 and T2 Trace Lengths" on page 49

Designing a Discrete RCR Termination Network. Refer to the following schematic diagram for designing a discrete termination network.



SUT Risetime	T1 Length	T2 Length	T3 Length	T1/T2/T3 Impedance	Rtip
>= 1ns	<= 1.4 in.	<= 1.4 in.	<= 0.5 in.	Z0 >= 80 ohms	196 ohms
>= 500ps	<= 0.7 in.	<= 0.7 in.	<= 0.5 in.	Z0 >= 80 ohms	215 ohms

The key parameter to consider when designing a discrete probing network is the rise time of the signal under test. Maximum trace lengths and tip resistor values are dependant upon the rise time.

NOTE: The previous values are derived from complex equations that are a function of the logic analyzer connector and cabling electrical characteristics, input network, standard component values, trace impedance and length, and parasitic capacitance and inductance . It is beyond the scope of this document to provide these equations for general discussion. If the maximum trace lengths specified here are not sufficient to probe the signal under test, please consult an Agilent Technologies technical specialist.

T1, T2, and T3 are all 80 ohms or higher printed-circuit board traces on the target system that connect the various components in the probing network. It is important to keep the trace impedance as high as possible. It is recommended that outer layers be used which minimize the capacitance and maximize the impedance and propagation speeds. In general, design the trace on an outer layer with as high impedance

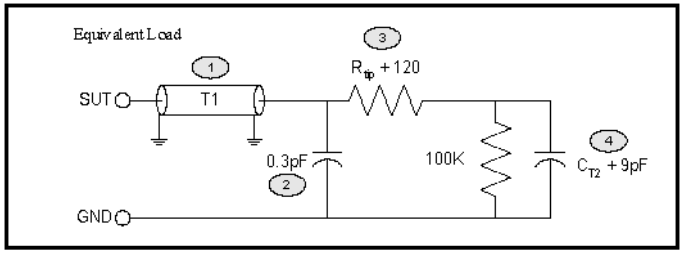
as possible without implementing out of the ordinary design techniques such as ground plane cutouts. Keep the trace lengths within the tolerances specified in the table.

The tip resistor Rtip must be placed within T1length of the signal under test (SUT). The trace length and tip resistor have been selected to provide maximum frequency response into the logic analyzer input network.

The RC network is connected to Rtip via a printed-circuit trace T2 which should not exceed T2length. Use precision 0402 or 0603 surface mount components for this network (1% R, 5% C).

The Mictor connector should be placed as close to the RC network as possible, not to exceed T3length = 0.5 inches in printed-circuit board trace length. The input impedance of the connector and logic analyzer cable is 120 ohms. In practice it is very difficult to achieve a matched impedance printed-circuit board trace, resulting in an inevitable mismatch. To keep the reflections from degrading the setup and hold margins, the maximum length of T3 has been specified in the previous table.

Equivalent Load. The discrete probing network and the logic analyzer input can be modeled as the following equivalent load.



1. T1 is a printed-circuit board trace transmission line specified in Section 2.2.1. Rtip is the sum of the tip resistor and the logic analyzer cable impedance.
2. The 0.3pF capacitance is the parasitic capacitance of the tip resistor pads.
3. The equivalent series resistance is the sum of the tip resistor Rtip and the 120 ohm (DC component) logic analyzer cable impedance.

4. The capacitive load is the sum of the logic analyzer capacitance plus the T2 printed-circuit board transmission line lumped capacitance. Assuming T2 has a length of 1.4 inches and a 2pF per inch capacitance, the equivalent capacitive load can be calculated by the following equation:

```
Cequiv = 9pF + T2pF/in * T2length = 9pF + 2pF/in * 1.4 in. = 11.8pF
```

Extrapolating T1 and T2 Trace Lengths. It is fairly straightforward to extrapolate the previous values for faster, slower, and intermediate rise times. For signals with rise times faster than 300 ps, a more detailed analysis must be performed. Refer to the following equations for calculating the optimal maximum trace length for the T1 and T2 printed-circuit board traces. Note that the following lengths are only approximations since the resistor values are selected to be compatible with industry standard components.

```
lmax =~ risetime * (0.7 in. / 500ps),  300ps < risetime <= 750ps,  Rtip = 215
ohms
                                       risetime > 750ps,           Rtip = 196 ohms
```

Routing Techniques

Regardless of whether discrete networks are used or not, the following describes some general guidelines for probing high-speed signals and buses. Low voltage TTL logic (LVTTL) and SSTL logic buses are described here.

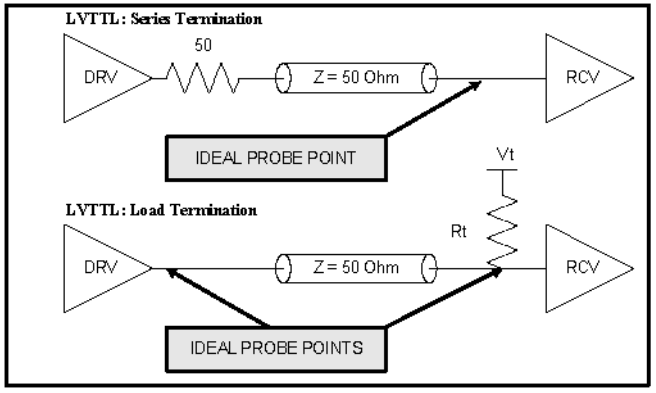
NOTE: The following schematics are meant to be a basic guide for probing placement, not a description of how to terminate transmission lines. Please refer to the vendors documentation for exact termination techniques.

- "Series Terminated LVTTL" on page 49
- "Load Terminated LVTTL" on page 50
- "SSTL-2" on page 50
- "SSTL-3" on page 50

Series Terminated LVTTL. Series termination resistors can be used to dampen reflections on traces exceeding the length equals 1/5 rise time rule. Signals propagating along a source series transmission line resolve at the load. Midpoints along the transmission line will contain a partial signal rise until the reflection returns and completes the

transition. If the intermediate step function voltage happens to be near the threshold voltage of the logic analyzer, metastability may occur and ultimately degrade setup and hold margins into the logic analyzer. Series terminated transmission lines should be probed at the load.

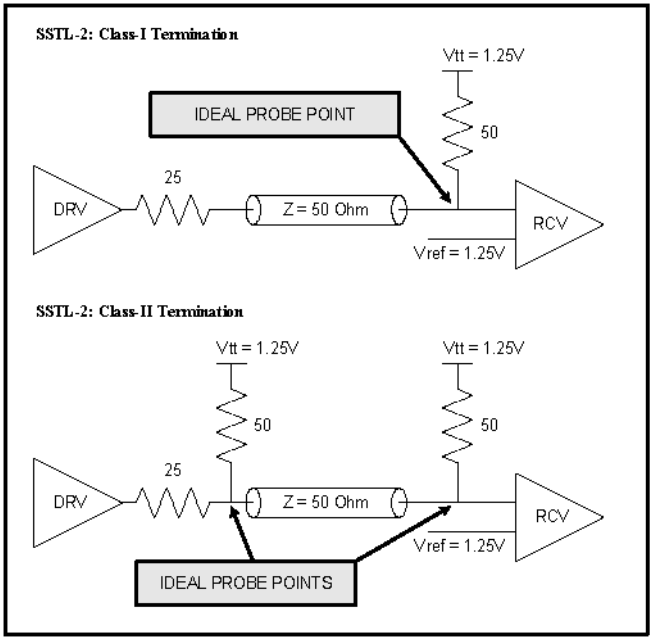
Load Terminated LVTTL. Uni-directional and bi-directional load terminated transmission lines can generally be probed at the driver, load, or anywhere in between as long as all signals are probed at the same position. It is important to keep the stub lengths less than 1/5 the rise time. Ideally, one endpoint should be selected for probing.



SSTL-2. The Agilent SerDes drives/receives signals in an SSTL-2 format. While SSTL-2 provides excellent high-speed signal fidelity during transmission across a printed-circuit board, issues arise that do not normally occur with single-ended LVTTL logic.

An SSTL-2 transmission line is described by the following diagram. Class-I termination specifies a 50 ohm resistor to Vtt at the receiver. Class-II termination requires a 50 ohm resistor to Vtt at both the source and receiver. A 25 ohm series resistor is typically designed into the driver. In general, it is safe to probe the signal at or very near the load termination resistors, which may be either end for Class-II buses.

SSTL-3. The technique for termination and probing 3.3V SSTL logic is the same except the bias voltage is higher and the logic analyzer threshold must be set to 1.5V instead of 1.25V.



Logic Analyzer Connectors

- "Mictor Connectors" on page 51
- "Connector Placement Techniques" on page 51
- "Mictor Connector Pinout" on page 52

Mictor Connectors

Mictor Connectors are used to connect the signal under test (SUT) to the logic analyzer. The connectors each provide for 32 signals plus two clocks. The Mictor connectors each have 38 surface mount pins plus 5 through-hole ground returns. Mechanical documentation and layout information can be found on the Agilent Technologies web site.

Connector Placement Techniques

In general, connectors should be placed as close to the probe point as

possible to minimize stubs. Refer to “Built-in Termination Networks” on page 46 for maximum recommended stub lengths. If discrete termination networks are used because of real estate concerns, connector placement typically becomes less critical since the maximum distance is effectively doubled. Traces from the termination networks should be matched in length so that all signals have equal propagation delays and setup and hold margins are preserved.

Mictor Connector Pinout

The following table describes an example pinout for the Mictor-38 connector. Note that Agilent uses a staggered pin numbering (1 / 2 alternating sides) while other vendors may recommends numbering in order on one side. Either way, as long as the signals are routed to the correct physical location, the numbering scheme does not matter.

Agilent requires pins 1-4 to be left unconnected. In addition, the state analyzer clocks MUST be connected to pins 5 and 6. The 5 center plated mounting holes MUST be connected to ground. All other signals are user definable. Bused signals should place the MSB on pin 7/8, descending downward to the LSB.

Refer to the SerDes documentation for pin numbers.

Mictor-38 Connector for the Agilent SerDes			
Odd Pin Side	Signal	Signal	Even Pin Side
1	NC	NC	2
3	NC	NC	4
5	CLKA	CLKB	6
7	DA[9]	DB[9]	8
9	DA[8]	DB[8]	10
11	DA[7]	DB[7]	12
13	DA[6]	DB[6]	14
15	DA[5]	DB[5]	16
17	DA[4]	DB[4]	18
19	DA[3]	DB[3]	20
21	DA[2]	DB[2]	22
23	DA[1]	DB[1]	24
25	DA[0]	DB[0]	26
27			28
29			30
31			32
33			34
35			36
37			38

Appendix

- "Supported Logic Analyzer Modules for InfiniBand" on page 53
- "Probing Part Numbers and Ordering Information" on page 53
- "Supplemental Agilent Documentation" on page 53
- "Technical Support" on page 53

Supported Logic Analyzer Modules for InfiniBand

Infiniband applications should use the 16717A logic analyzer family and higher. While the frequency of older logic analyzer cards may be sufficient, the eye-finder feature and better setup and hold window make the 16717A and higher necessary.

Agilent Model	State Speed	Channels	Memory Depth
16717A	333 MHz	68	2M State
16750/1/2A	400 MHz	68	4M / 16M / 32M State

Agilent Model	Description
16702B	Mainframe with display and touch-screen
16700B	Mainframe with no display
16701B	Expansion Frame

Probing Part Numbers and Ordering Information

Agilent Model	Description
E5346-60001	Set of 5 Mictor connectors for use with high-density cable.
E5346A	High-density logic analyzer cable with built-in termination networks.
E5351A	High-density logic analyzer cable with NO termination. Requires discrete termination.

Supplemental Agilent Documentation

Publication	Description
5968-4632A	Probing Solutions for Agilent Technologies Logic Analysis Systems

Technical Support

Agilent Technologies web site:

http://www.agilent.com/

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    11100 010
```

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
```

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 16600A-series or 16700A/B-series mainframes. One card fills one slot in the mainframe. A module may comprise a single card or multiple cards cabled together.

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 16600A-series 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, 16716A, and 16717A logic analyzers.

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

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.

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

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

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 avalaible is determined by the channel width of the instrument.

pod See *pod pair*

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

preprocessor See *analysis probe*.

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

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

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

Numerics

10-bit data bus, 12
10-bit data label required, 13
8B/10B encoding, 54

A

arm in, 17
ASCII text editor, 30, 31
auxiliary software, 9

B

bad CRC, 22
bad disparity, 22
Binary field format, 38
binary numbers in protocol definitions, 40
blank lines, 26
block coding, 54
bus to process, 21
buses, multiple, 13

C

capturing the data, 15
CD-ROM drive, 9
channels, labeling logic analyzer, 13
character names, 22, 54
check if toolset installed, 8
CLK input channel, logic analyzer, 12
color for state types, 26
connecting logic analyzer probes, 12
connecting the InfiniBand Tool, 20
control symbols, 26
CRC error, 22

D

D characters, 54
data bus label, 21
Data field type, 38
Data Packet protocol, 2, 21
data, capturing, 15
data, displaying, 20
data, eight bit, 22
Decimal field format, 38
decimal numbers in protocol definitions, 40
decoded packet, 22
demo period, 8
device under test, probing, 12
disparity, 54
disparity, bad, 22
displaying the data, 20

E

editing protocol definitions, 30
editing protocol definitions with vi editor, 30
eight bit data, 22
end of packet, 13
entitlement certificate, 10
error, CRC, 22

F

field definition, 38
field format, 38
field type, 38
filter data rows, 26
Find InfiniBand Packet trigger function, 2
Find pattern n times, 15
Format tab, 13

G

Group Run button, 19

H

Header block, 37
Hex field format, 38
hex numbers in protocol definitions, 40

I

idle states, 26
InfiniBand Tool, connecting, 20
InfiniBand Tool, setup, 21
InfiniBand Toolset, using, 2
installation, 8, 9
installed, check if toolset, 8
instrument, other, 17

J

J sampling clock input, 12

K

K characters, 15, 54

L

labeling logic analyzer channels, 13
licensing, 8, 10
Link Packet protocol, 2, 21
logic analyzer channels, labeling, 13
logic analyzer CLK input channel, 12
logic analyzer modules supported, 2
logic analyzer probes, connecting, 12

M

MAC layer, 37
measurement, running, 19
modifying protocol definitions, 30
multiple communications buses, 13

N

networked computer, 31
numeric values in protocol definitions, 40

Index

O

Octal field format, 38
octal numbers in protocol definitions, 40
other instrument, 17
output data columns, selecting, 22

P

packet data, 13
packet headers, 26
packet payloads, 26
packet, decoded, 22
password center, 10
physical layer, 37
pre-defined protocols, 2
probing application note, 42
probing the device under test, 12
protocol definition syntax, 36
protocol definitions file, 30, 31
protocol definitions, editing with the "vi" editor, 30
protocol definitions, modifying, 2, 30
protocol definitions, reloading, 31
protocol definitions, reset, 32
Protocol keyword, 36
protocol name, 36
protocol-based triggering, 2
ProtocolIndicator field type, 39

R

reloading protocol definitions, 31
reset protocol definitions, 32
reverse bits, 21
Run button, 19
running a measurement, 19, 20
running disparity, 54

S

sampling clock signal, 12
sampling mode, selecting, 12
second analyzer, 17
set up InfiniBand Tool, 21
special 8B/10B codes, trigger on, 15
special characters, 54
start of Data Packet, 15
start of Link Packet, 15
start of packet, 13
state sampling mode, selecting, 12
Symbols field type, 38
synchronous sampling mode, selecting, 12
syntax, protocol definition, 36
system administration tools, 8, 9, 10

T

target system, probing, 12
trigger on special 8B/10B codes, 15

U

unknown states, 26

V

valid packet data, 13
vi editor, 30, 31

W

wait for arm in, 17
Wait for second analyzer to trigger, 17
workspace, connecting the InfiniBand Tool, 20