Help Volume

Toolsets: Gbit Ethernet (Agilent Technologies N4212A)

Using the Gbit Ethernet Toolset

GbE

The Gbit Ethernet Toolset adds protocol-based display and error detection capabilities to the logic analyzer. This toolset helps you look at data traveling across Gbit Ethernet TBI (Ten Bit Interface) buses.

The Gbit Ethernet 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 Gbit Ethernet Toolset understands a number of pre-defined protocols (including most of the Internet Protocol stack). You can modify these protocol definitions if necessary.

NOTE: The Gbit Ethernet Toolset will work with any Agilent Technologies logic analyzer whose state analysis speeds support the speed of the Gbit Ethernet TBI bus.

"Installation and Licensing" on page 8

- "To check if the Gbit Ethernet Toolset is installed" on page 8
- "To install the Gbit Ethernet Toolset" on page 9
- "To license the Gbit Ethernet 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 13
- "Step 3. Label the logic analyzer channels" on page 14

"Capturing the Data" on page 16

- "To trigger on special 8B/10B codes" on page 16
- "To trigger with a second analyzer" on page 18
- "To trigger with another instrument" on page 18
- "To run the measurement" on page 20

See Also

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

Glossary (see page 47)

Contents

Using the Gbit Ethernet Toolset

1 Installation Guide

2 Task Guide

3 Reference

4 Concepts

Glossary

Index

1

Installation Guide

Installation and Licensing

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

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

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

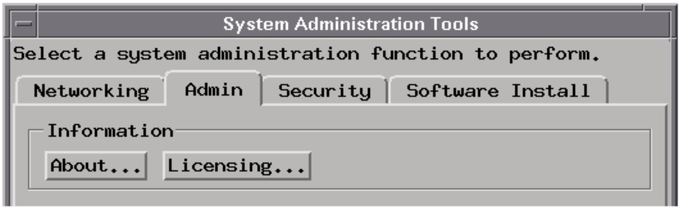
- “To check if the Gbit Ethernet Toolset is installed” on page 8
- “To install the Gbit Ethernet Toolset” on page 9
- “To license the Gbit Ethernet Toolset” on page 10

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 Gbit Ethernet 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.



System Administration Tools

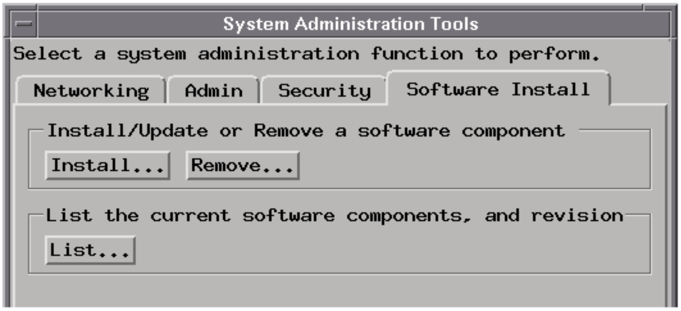
Select a system administration function to perform.

3. In the Tool Sets tab of the Licensing Dialog, look for the "N4212A - Gbit Ethernet Toolset" product.
   - If it is not listed, go to “To install the Gbit Ethernet Toolset” on page 9.
   - If it is listed, go to “To license the Gbit Ethernet Toolset” on page 10.

To install the Gbit Ethernet 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....



System Administration Tools

Select a system administration function to perform.

Install/Update or Remove a software component

List the current software components, and revision

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 Gbit Ethernet Toolset; then, select the Install button.

   The dialog will display "Progress: completed successfully" when the installation is complete.

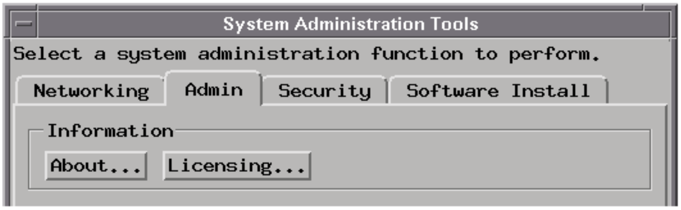
8. Select the Close button to close the Software Install window.
9. Select the Close button to close the System Administration Tools window.
10. Go to “To license the Gbit Ethernet Toolset” on page 10.

See Also

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

To license the Gbit Ethernet Toolset

1. To obtain a password, contact the password center listed on the Entitlement Certificate you received after purchasing the Gbit Ethernet 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.



Select a system administration function to perform.

4. In the Tool Sets tab of the Licensing Dialog, enter the password into the field for the "N4212A - Gbit Ethernet 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 Gbit Ethernet 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 13
- "Step 3. Label the logic analyzer channels" on page 14

Step 1. Connect the logic analyzer probes

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

- A Gbit Ethernet TBI (Ten Bit Interface) data bus.

  The Gbit Ethernet 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.

You can also connect other signals to the remaining logic analyzer CLK input channels and use them to "qualify" the main sampling clock signal. This can be useful, for example, when packet transfers are interrupted.

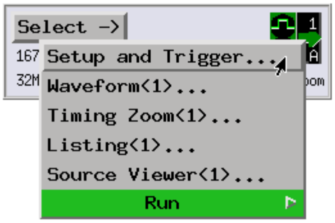
When setting up the logic analyzer's state sampling mode, you can specify that one CLK input signal be used to enable or disable the sampling signal on another CLK input. Disabling the sampling clock during interrupts will allow the logic analyzer to find a packet event that has been interrupted.

Next Step (see page 13)

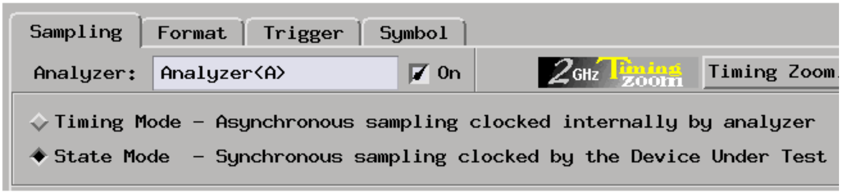
Step 2. Select the state sampling mode

When using the Gbit Ethernet 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.



Analyzer: Analyzer<A> ☑ On

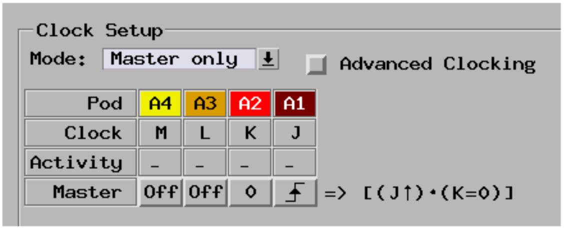
◇ Timing Mode – Asynchronous sampling clocked internally by analyzer

◆ State Mode – Synchronous sampling clocked by the Device Under Test

3. Specify the state analyzer's sampling clock input.

   If you've connected an additional signal from the device under test to another of the logic analyzer's CLK inputs for the purpose of "qualifying" the main sampling clock signal, include that signal in the clock setup.

   For example, suppose the main communication data bus clock signal has been connected to the pod 1 CLK input (J) and a signal that is low when not in a packet transfer interrupt has been connected the pod 2 CLK input (K). In this case, you would use a clock setup like:



Clock Setup

Mode: Master only

□ Advanced Clocking

Pod	A4	A3	A2	A1	
Clock	M	L	K	J	
Activity	–	–	–	–	
Master	Off	Off	0	↑	=> [(J↑)•(K=0)]

When there is a packet transfer interrupt, the signal connected to the pod 2 CLK input (K) is high, and there is no sampling clock.

Because no data values are sampled by the logic analyzer during the interrupt, the logic analyzer can find the rest of a packet after the interrupt.

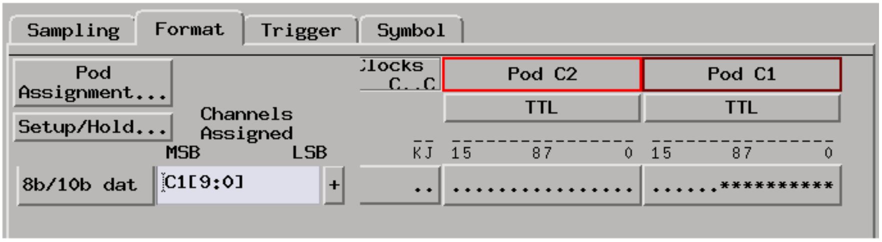
Next Step (see page 14)

Step 3. Label the logic analyzer channels

The Gbit Ethernet Tool requires you to set one 10-bit data label for the Gbit Ethernet TBI 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.



	MSB LSB		Clocks C..C	Pod C2	Pod C1
				TTL	TTL
			K J	15 8 7 0	15 8 7 0
8b/10b dat	C1[9:0]	+	..	....	......**********

See Also “Capturing the Data” on page 16

Capturing the Data

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

Otherwise, trying to set up triggers on 8B/10B encoded data is hard (and would quickly use up logic analyzer trigger resources). 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 16
- "To trigger with a second analyzer" on page 18
- "To trigger with another instrument" on page 18
- "To run the measurement" on page 20

To trigger on special 8B/10B codes

Because special 8B/10B codes are assigned to things like "start of packet", "end of 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.

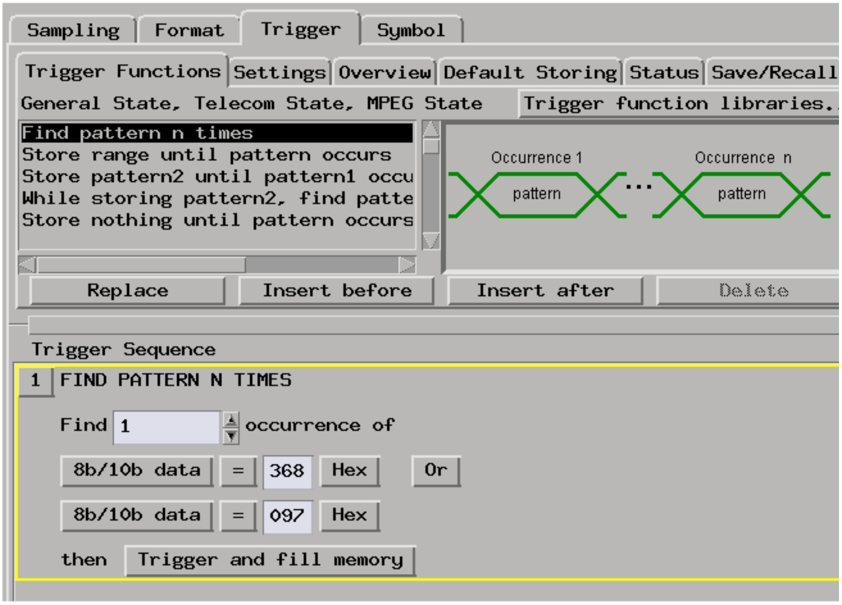
1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.

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 packet", K27.7, are 368 Hex and 097 Hex.


Sampling
Format
Trigger
Symbol
Trigger Functions
Settings
Overview
Default Storing
Status
Save/Recall
General State, Telecom State, MPEG State
Trigger function libraries..
Find pattern n times
Store range until pattern occurs
Store pattern2 until pattern1 occu
While storing pattern2, find patte
Store nothing until pattern occurs
Occurrence 1
Occurrence n
pattern
pattern
Replace
Insert before
Insert after
Delete

Trigger Sequence

1 FIND PATTERN N TIMES

Find 1 occurrence of

8b/10b data = 368 Hex Or

8b/10b data = 097 Hex

then Trigger and fill memory

See Also

"To run the measurement" on page 20

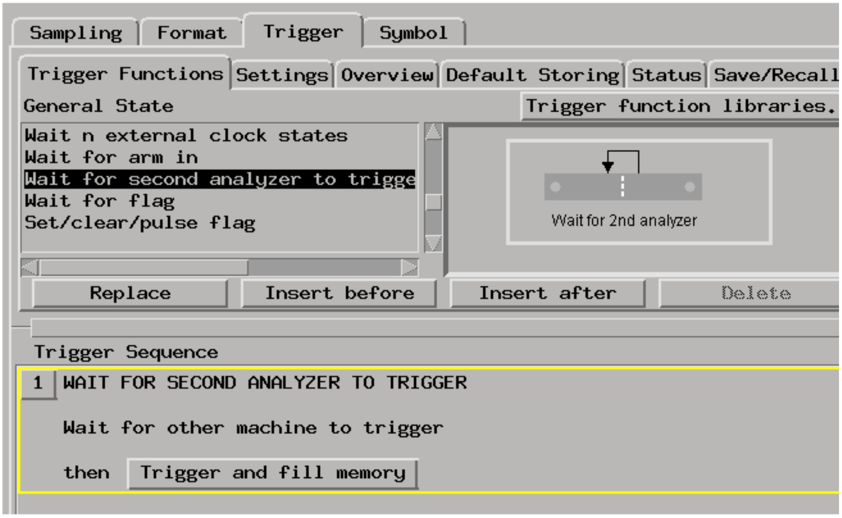
"Displaying the Data" on page 21

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.



```
Wait n external clock states
Wait for arm in
Wait for second analyzer to trigge
Wait for flag
Set/clear/pulse flag
```

```
1  WAIT FOR SECOND ANALYZER TO TRIGGER

   Wait for other machine to trigger

   then  Trigger and fill memory
```

See Also

“To run the measurement” on page 20

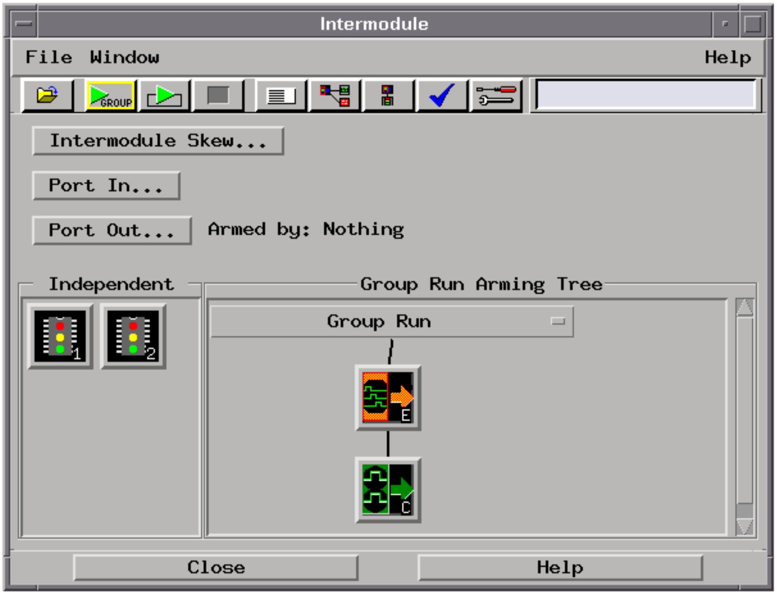
“Displaying the Data” on page 21

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

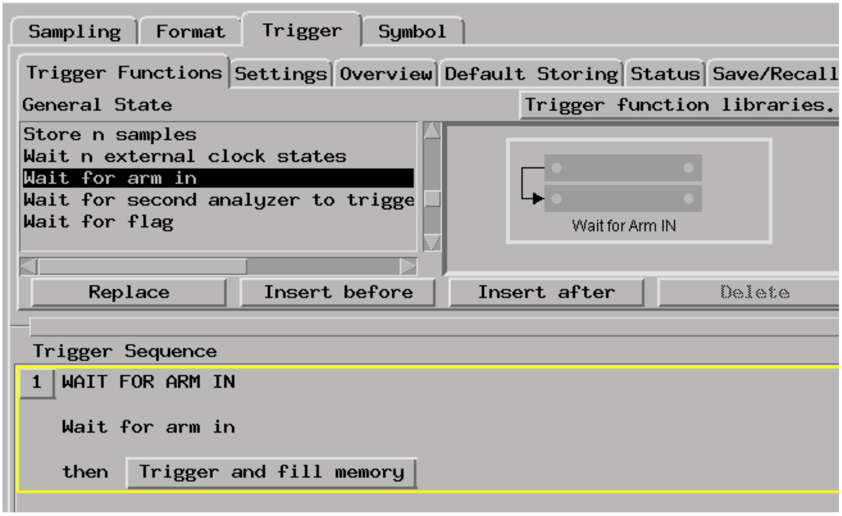
other instrument.



Armed by: Nothing

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.



```
1  WAIT FOR ARM IN

   Wait for arm in

   then  Trigger and fill memory
```

See Also “To run the measurement” on page 20

“Displaying the Data” on page 21

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 21

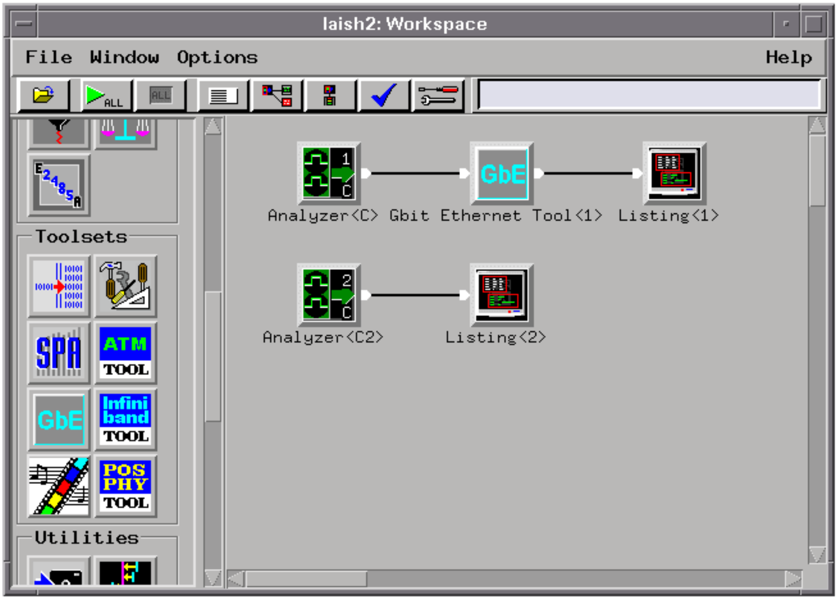
Displaying the Data

This section describes the options available when displaying data captured on the Gbit Ethernet TBI data bus.

- “To connect the Gbit Ethernet Tool (in the workspace)” on page 21
- “To set up the Gbit Ethernet Tool” on page 22
- “To select the output data columns” on page 23
- “To filter data rows” on page 27

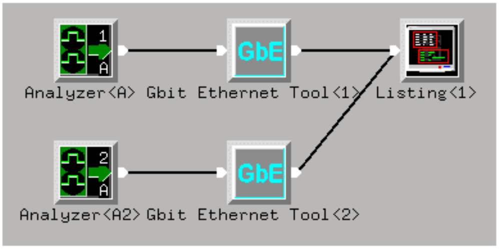
To connect the Gbit Ethernet Tool (in the workspace)

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



The Gbit Ethernet 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 Gbit Ethernet Tools go to a single Listing display, filtering does not work.

See Also

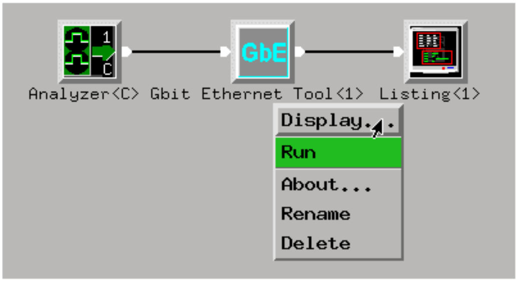
“To set up the Gbit Ethernet Tool” on page 22

“To select the output data columns” on page 23

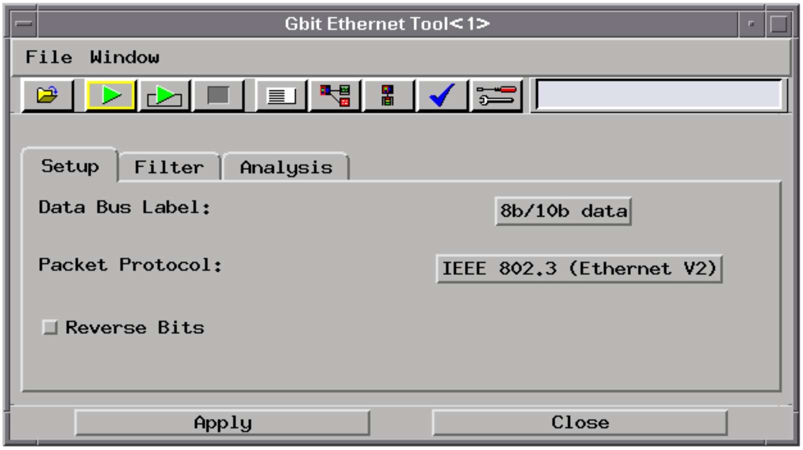
“To filter data rows” on page 27

To set up the Gbit Ethernet Tool

1. Display the Gbit Ethernet Tool.



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



Data Bus Label: 8b/10b data

Packet Protocol: IEEE 802.3 (Ethernet V2)

□ Reverse Bits

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

Packet Protocol Selects the protocol being used on the bus.

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 Gbit Ethernet Tool.

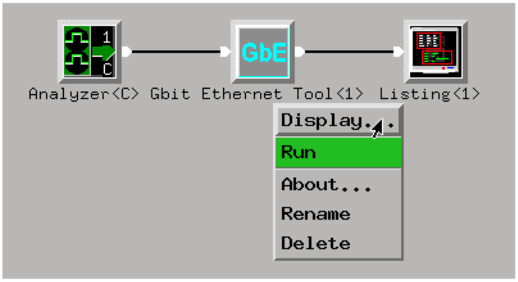
See Also

"To select the output data columns" on page 23

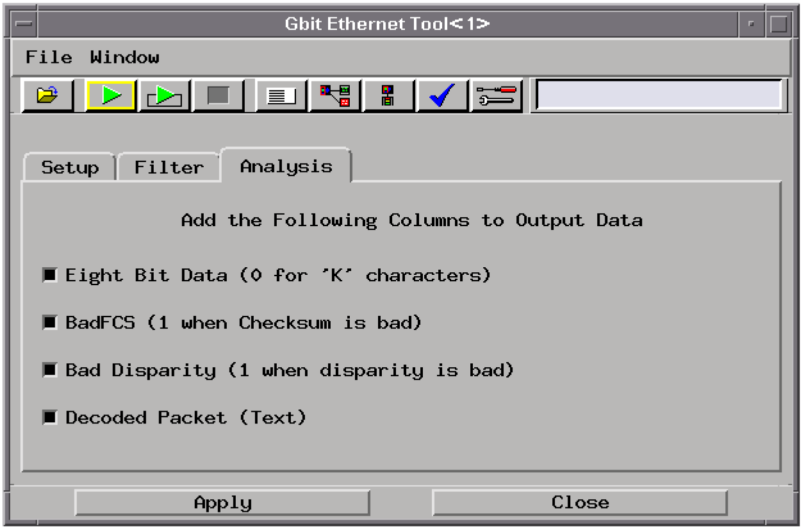
"To filter data rows" on page 27

To select the output data columns

1. Display the Gbit Ethernet Tool.



2. In the Gbit Ethernet Tool’s Analysis tab, select the columns that should appear in the output data:



Add the Following Columns to Output Data

■ Eight Bit Data (0 for ’K’ characters)

■ BadFCS (1 when Checksum is bad)

■ Bad Disparity (1 when disparity is bad)

■ Decoded Packet (Text)

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.

BadFCS This 1-bit column contains "1" whenever a packet has a checksum that does not match the data; otherwise, it contains "0".

When this data column is included in the output, you can use the System Performance Analysis (SPA) toolset or the

Distribution tool to get a count of the number of corrupt packets. The number of times BadFCS=1 divided by the total number of states in the data set can be used as a bit error rate computation.

Bad Disparity This selection can tell you if your data stream obeys the rules for Running Disparity, according to the 8B/10B encoding specifications found in the Fibre Channel specification.

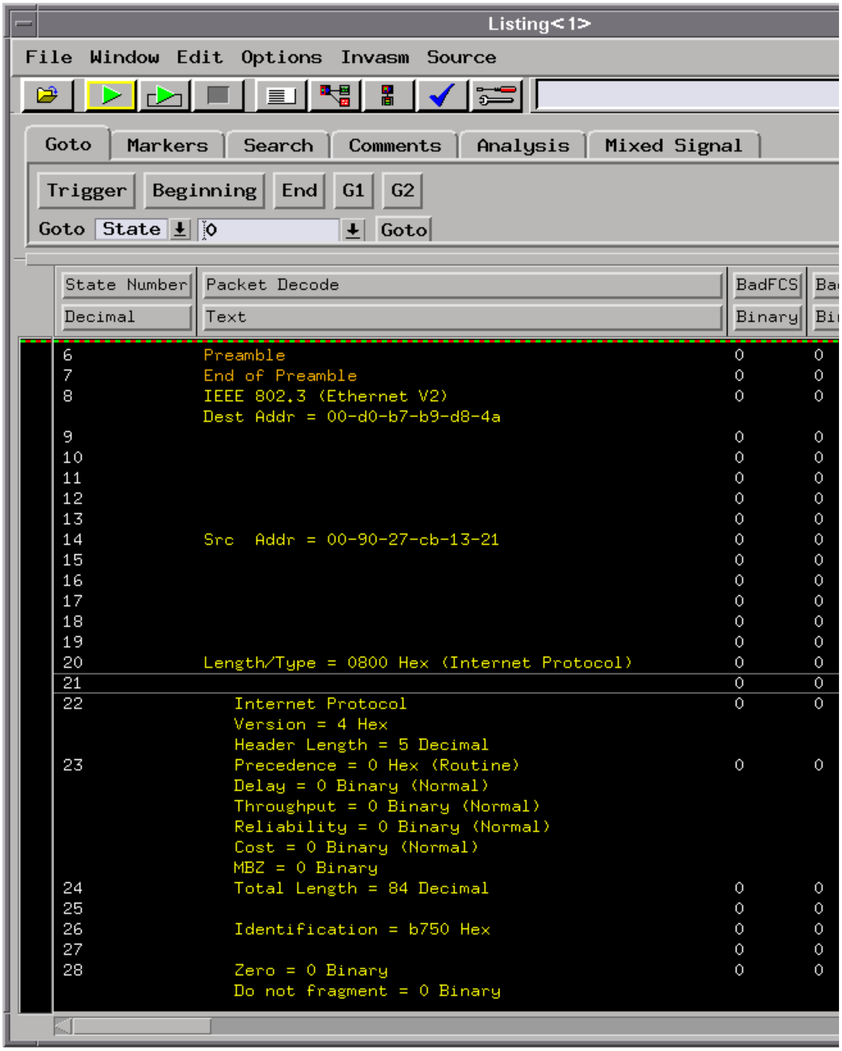
When selected, two columns are added to the output data:

- 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).
- BadDisparity. This 1-bit column contains "1" when the the Disparity column has an improper value, that is, when a negative disparity follows a negative or a positive follows a positive.

Each 10-bit value has an equal number of 1's and 0's (exactly 5 of each). This is to prevent low-frequency signal components from disturbing the phase-lock on receivers of the data stream. If a 10-bit value has more 1's near the end of the value, disparity is to be considered *positive*, and the next value sent should have either *neutral* or *negative* running disparity. Conversely, if the number has more 0's near the end of the value (and more 1's near the beginning), it has *negative* disparity and should be followed with a value with *neutral* or *positive* disparity.

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.



Listing<1>
File Window Edit Options Invasm Source
Goto Markers Search Comments Analysis Mixed Signal
Trigger Beginning End G1 G2
Goto State 0 Goto

```
State Number  Packet Decode                                        BadFCS  Ba
Decimal       Text                                                 Binary  Bi
6             Preamble                                             0       0
7             End of Preamble                                      0       0
8             IEEE 802.3 (Ethernet V2)                             0       0
              Dest Addr = 00-d0-b7-b9-d8-4a
9                                                                  0       0
10                                                                 0       0
11                                                                 0       0
12                                                                 0       0
13                                                                 0       0
14            Src  Addr = 00-90-27-cb-13-21                        0       0
15                                                                 0       0
16                                                                 0       0
17                                                                 0       0
18                                                                 0       0
19                                                                 0       0
20            Length/Type = 0800 Hex (Internet Protocol)           0       0
21                                                                 0       0
22               Internet Protocol                                 0       0
                 Version = 4 Hex
                 Header Length = 5 Decimal
23               Precedence = 0 Hex (Routine)                      0       0
                 Delay = 0 Binary (Normal)
                 Throughput = 0 Binary (Normal)
                 Reliability = 0 Binary (Normal)
                 Cost = 0 Binary (Normal)
                 MBZ = 0 Binary
24               Total Length = 84 Decimal                         0       0
25                                                                 0       0
26               Identification = b750 Hex                         0       0
27                                                                 0       0
28               Zero = 0 Binary                                   0       0
                 Do not fragment = 0 Binary
```

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

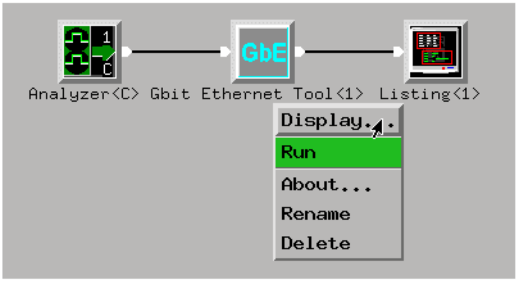
See Also

"To set up the Gbit Ethernet Tool" on page 22

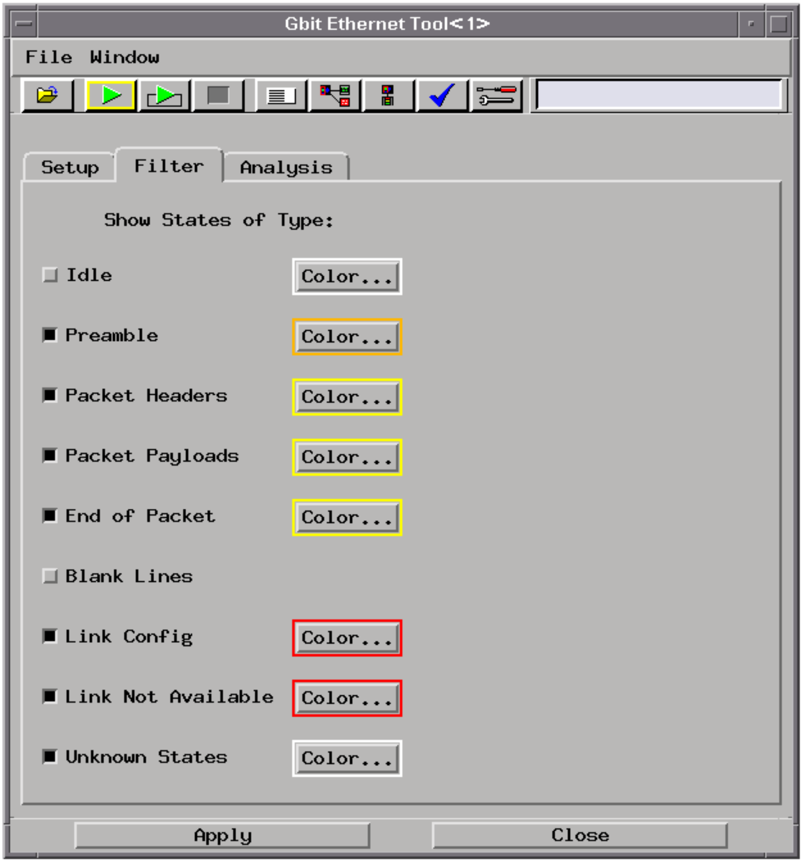
"To filter data rows" on page 27

To filter data rows

1. Display the Gbit Ethernet Tool.



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



Show States of Type:

☐ Idle

■ Preamble

■ Packet Headers

■ Packet Payloads

■ End of Packet

☐ Blank Lines

■ Link Config

■ Link Not Available

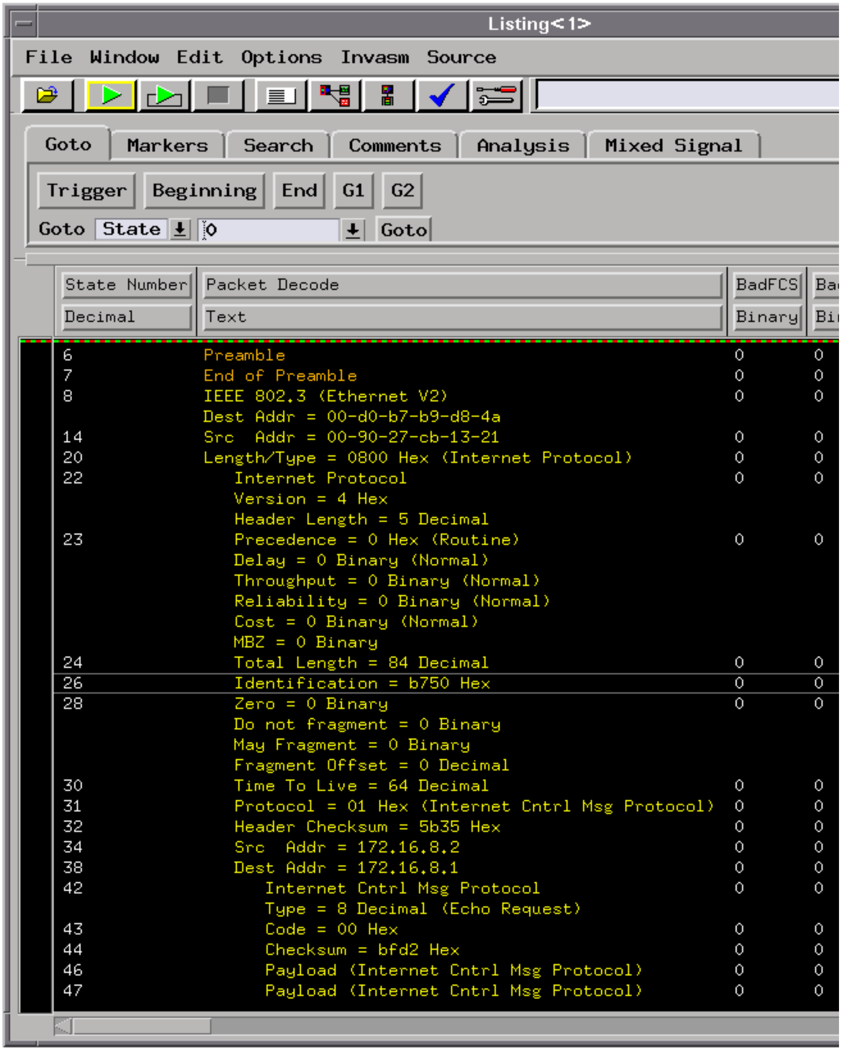
■ Unknown States

- Idle.
- Preamble.
- Packet Headers.
- Packet Payloads.
- End of Packet.
- Blank Lines.
- Link Config.
- Link Not Available.

- Unknown States.

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

3. Select Apply to output the selected data rows.



```
State Number  Packet Decode                                              BadFCS  Ba
Decimal       Text                                                       Binary  Bi
6             Preamble                                                   0       0
7             End of Preamble                                            0       0
8             IEEE 802.3 (Ethernet V2)                                   0       0
              Dest Addr = 00-d0-b7-b9-d8-4a
14            Src  Addr = 00-90-27-cb-13-21                              0       0
20            Length/Type = 0800 Hex (Internet Protocol)                 0       0
22               Internet Protocol                                       0       0
                 Version = 4 Hex
                 Header Length = 5 Decimal
23               Precedence = 0 Hex (Routine)                            0       0
                 Delay = 0 Binary (Normal)
                 Throughput = 0 Binary (Normal)
                 Reliability = 0 Binary (Normal)
                 Cost = 0 Binary (Normal)
                 MBZ = 0 Binary
24               Total Length = 84 Decimal                               0       0
26               Identification = b750 Hex                               0       0
28               Zero = 0 Binary                                         0       0
                 Do not fragment = 0 Binary
                 May Fragment = 0 Binary
                 Fragment Offset = 0 Decimal
30               Time To Live = 64 Decimal                               0       0
31               Protocol = 01 Hex (Internet Cntrl Msg Protocol)         0       0
32               Header Checksum = 5b35 Hex                              0       0
34               Src  Addr = 172.16.8.2                                  0       0
38               Dest Addr = 172.16.8.1                                  0       0
42                  Internet Cntrl Msg Protocol                          0       0
                    Type = 8 Decimal (Echo Request)
43                  Code = 00 Hex                                        0       0
44                  Checksum = bfd2 Hex                                  0       0
46                  Payload (Internet Cntrl Msg Protocol)                0       0
47                  Payload (Internet Cntrl Msg Protocol)                0       0
```

4. Select Close to close the Gbit Ethernet Tool.

See Also

Filtering Limitation in “To connect the Gbit Ethernet Tool (in the workspace)” on page 21

“To set up the Gbit Ethernet Tool” on page 22

“To select the output data columns” on page 23

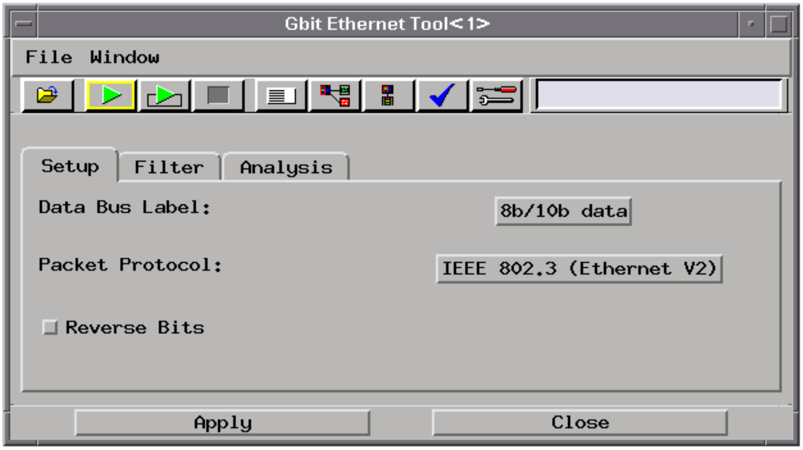
Modifying Protocol Definitions

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

- "To modify a protocol definition" on page 31
- "To reload the protocol definitions" on page 32
- "To reset the protocol definitions" on page 34

To modify a protocol definition

1. In the Gbit Ethernet Tool's Setup tab, select the packet protocol button.

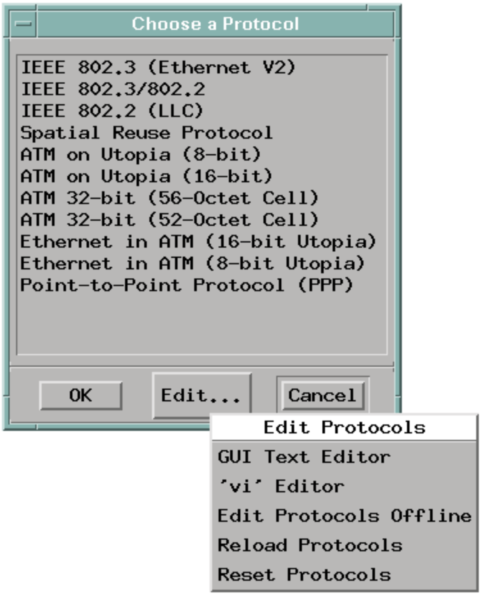


Data Bus Label: 8b/10b data

Packet Protocol: IEEE 802.3 (Ethernet V2)

□ Reverse Bits

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 Gbit Ethernet Tool that changes have been made (see “To reload the protocol definitions” on page 32).

About the Protocol Definitions File

The protocol definitions file is:

```
/logic/auxiliary/TeleCom/protocols/protocols.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.

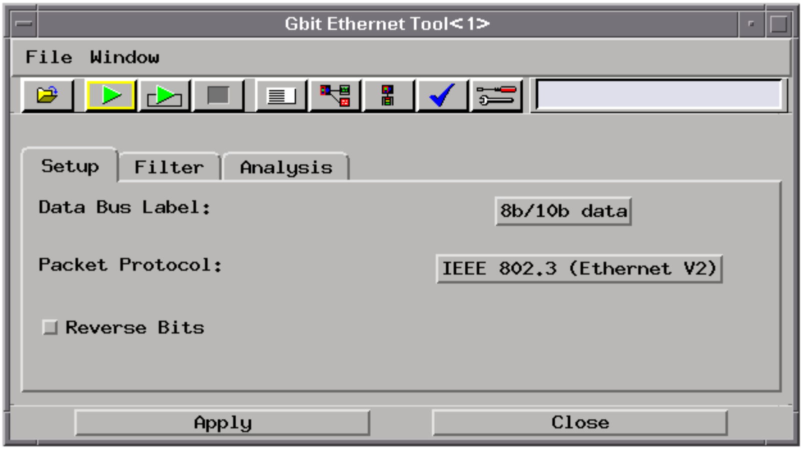
See Also

“Protocol Definition Syntax” on page 38

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 Gbit Ethernet Toolset that changes have been made.

1. In the Gbit Ethernet Tool's Setup tab, select the packet protocol button.

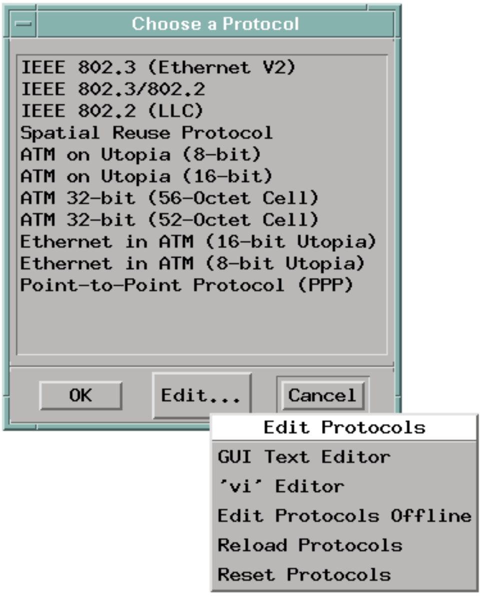


Data Bus Label: 8b/10b data

Packet Protocol: IEEE 802.3 (Ethernet V2)

□ Reverse Bits

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

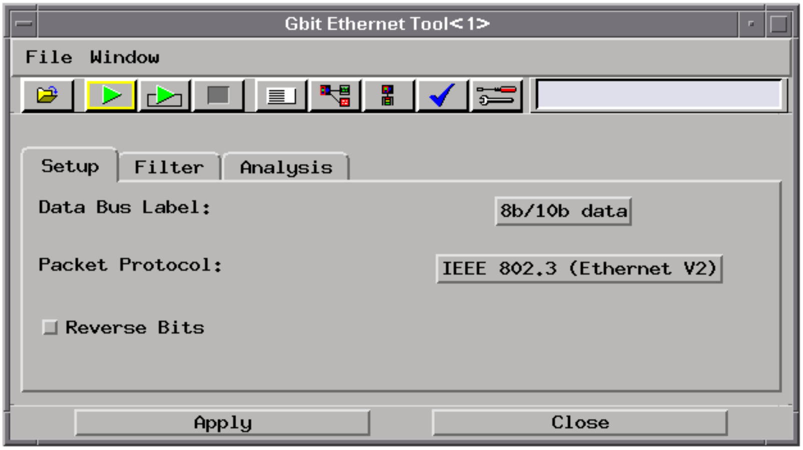


To reset the protocol definitions

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 Gbit Ethernet Tool's Setup tab, select the packet protocol button.

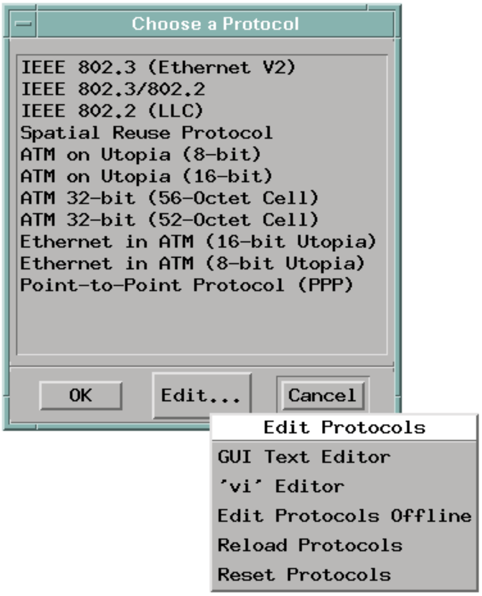


Data Bus Label: 8b/10b data

Packet Protocol: IEEE 802.3 (Ethernet V2)

□ Reverse Bits

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



IEEE 802.3 (Ethernet V2)
IEEE 802.3/802.2
IEEE 802.2 (LLC)
Spatial Reuse Protocol
ATM on Utopia (8-bit)
ATM on Utopia (16-bit)
ATM 32-bit (56-Octet Cell)
ATM 32-bit (52-Octet Cell)
Ethernet in ATM (16-bit Utopia)
Ethernet in ATM (8-bit Utopia)
Point-to-Point Protocol (PPP)

3

Reference

Protocol Definition Syntax

This section describes the protocol definition syntax.

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

- “Protocol Block” on page 39
- “Physical Layer Definition” on page 39
- “Header Block” on page 39
- “Field Definition” on page 40
    - “Data” on page 41
    - “Symbols” on page 41
    - “ProtocolIndicator” on page 41
- “Numeric Values” on page 42

Example

Here is the definition for the IEEE 802.3 (Ethernet V2) protocol:

```
Protocol
{
  Name "IEEE 802.3 (Ethernet V2)"
  PhysicalLayer 1

  Header
  {
    DA  "Dest Addr"    48  HardwareAddress  Data
    SA  "Src  Addr"    48  HardwareAddress  Data
    PT  "Length/Type"  16  Hex              ProtocolIndicator
    {
      "Internet Protocol"   #h0800
      "ARP Request"         #h0806
      "ARP Response"        #h0835
      "AppleTalk Datagram"  #h809B
      "SNA"                 #h80d5
      "Novell IPX"          #h8137
      "IPv6"                #h86dd
      "IPS"                 #h2007
    }
  }
}
```

See Also

“Displaying the Data” on page 21

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 "IEEE 802.3 (Ethernet V2)"

}
```

See Also

"Physical Layer Definition" on page 39

"Header Block" on page 39

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 Gbit Ethernet 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 40

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.
- DotNotation (IP address notation, only available on 32-bit fields).
- HardwareAddress (ethernet hardware address, only available on 48-bit fields).

The choices for field type are:

- "Data" on page 41.
- "Symbols" on page 41.
- "ProtocolIndicator" on page 41.

Example

```
DA  "Dest Addr"  48  HardwareAddress  Data
```

This is the first field in the Ethernet header, the Destination Address. It is 48 bits long. It should be displayed as a Hardware Address (for example, 11-22-33-44-55-66), 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 40

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 the 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 40

"Numeric Values" on page 42

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 41, 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 40

"Symbols" on page 41

"Numeric Values" on page 42

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

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
```

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 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