

© 1996-2001 Agilent Technologies. All rights reserved.

**Toolsets: MPEG Protocol
Analysis (Agilent Technologies
B4645B)**

Using the MPEG Protocol Analysis Toolset



The MPEG Protocol Analysis Toolset adds MPEG protocol-based triggering and display capabilities to the logic analyzer. This toolset helps you look at MPEG data streams traveling across parallel buses.

The MPEG Protocol Analysis Toolset consists of:

- The "Find MPEG Packet" trigger function which lets you set up logic analyzer triggers in terms of MPEG protocols and packet data.
- The MPEG Tool which lets you display captured data in terms of the MPEG protocol being used.

You can choose from a number of pre-defined protocols (including Transport Stream, Program Stream, and PES Packet) when setting up triggers and displaying captured data. And, you can define your own custom or proprietary protocols by modifying protocol definitions.

NOTE:

Because the MPEG Protocol Analysis Toolset requires VisiTrigger capabilities, it can only be used with the Agilent Technologies 16715/16/17/18/19/50/51/52A logic analyzer modules.

"Installation and Licensing" on page 8

- "To check if the MPEG Protocol Analysis Toolset is installed" on page 8
- "To install the MPEG Protocol Analysis Toolset" on page 9
- "To license the MPEG Protocol Analysis 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 set up MPEG packet triggers" on page 16
- "To define and edit buses" on page 18 ("Bus Editor Dialog" on page 20)

- “To define and edit packet events” on page 21 (“Event Editor Dialog” on page 23)
- “To run the measurement” on page 25

“Displaying the Data” on page 26

- “To connect the MPEG tool (in the workspace)” on page 26
- “To set up the MPEG tool” on page 27
- “To select the output data columns” on page 29
- “To filter data rows” on page 33

“Modifying Protocol Definitions” on page 36

- “To choose a protocol” on page 36
- “To modify a protocol definition” on page 39
- “To reload the protocol definitions” on page 40
- “To reset the protocol definitions” on page 40

Reference

- “Protocol Definition Syntax” on page 44
- “Messages” on page 56
- “Limitations” on page 58

See Also

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

Glossary (see page 59)

Contents

Using the MPEG Protocol Analysis Toolset

1 Installation Guide

- Installation and Licensing 8
- To check if the MPEG Protocol Analysis Toolset is installed 8
- To install the MPEG Protocol Analysis Toolset 9
- To license the MPEG Protocol Analysis 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 13
 - Step 3. Label the logic analyzer channels 14
- Capturing the Data 16
 - To set up MPEG packet triggers 16
 - To define and edit buses 18
 - To define and edit packet events 21
 - To run the measurement 25
- Displaying the Data 26
 - To connect the MPEG tool (in the workspace) 26
 - To set up the MPEG tool 27
 - To select the output data columns 29
 - To filter data rows 33
- Modifying Protocol Definitions 36
 - To choose a protocol 36
 - To modify a protocol definition 39
 - To reload the protocol definitions 40
 - To reset the protocol definitions 40

Contents

3 Reference

| | |
|---|----|
| Protocol Definition Syntax | 44 |
| Protocol Block | 46 |
| Physical Layer Definition | 46 |
| Header Block | 47 |
| Field Definition | 47 |
| Numeric Values | 51 |
| IF Conditional Block | 51 |
| UNTIL Loop Block | 52 |
| STUFF_UNTIL Loop Block | 53 |
| Formulas | 54 |
| Messages | 56 |
| CRC code field is not byte aligned | 56 |
| CRC Start field is not byte aligned | 56 |
| Field longer than 32 bits cannot be used in a formula | 57 |
| Field longer than 32 bits cannot be used with SyntaxCheck | 57 |
| Size of header in packet too large | 57 |
| Start data width is not equal to the first field width | 57 |
| Limitations | 58 |

Glossary

Index

Installation Guide

Installation and Licensing

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

If the MPEG Protocol Analysis Toolset is not already installed, you need to install it.

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

- “To check if the MPEG Protocol Analysis Toolset is installed” on page 8
- “To install the MPEG Protocol Analysis Toolset” on page 9
- “To license the MPEG Protocol Analysis Toolset” on page 10

See Also

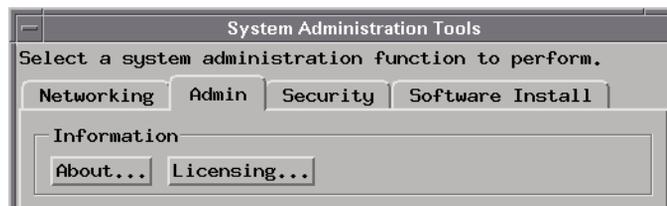
Licensing Policy for the Logic Analysis System and Tool Sets (see the *Agilent Technologies 16700A/B-Series Logic Analysis System* help volume)

To check if the MPEG Protocol Analysis 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.

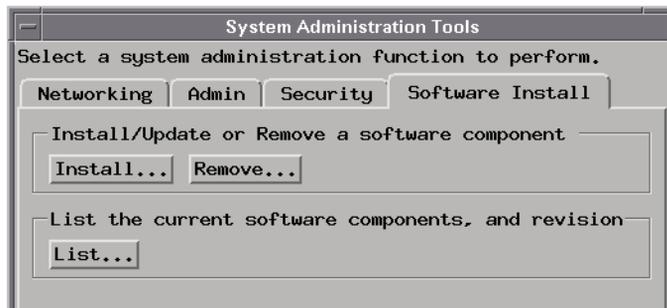


3. In the Tool Sets tab of the Licensing Dialog, look for the "B4645B - MPEG Protocol Analysis" product.
 - If it is not listed, go to "To install the MPEG Protocol Analysis Toolset" on page 9.
 - If it is listed, go to "To license the MPEG Protocol Analysis Toolset" on page 10.

To install the MPEG Protocol Analysis 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 MPEG Protocol Analysis Toolset; then, select the Install button.

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

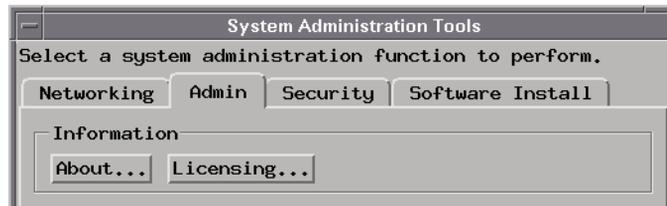
8. Select the Close button to close the Software Install window.
9. Select the Close button to close the System Administration Tools window.
10. Go to "To license the MPEG Protocol Analysis Toolset" on page 10.

See Also

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

To license the MPEG Protocol Analysis Toolset

1. To obtain a password, contact the password center listed on the Entitlement Certificate you received after purchasing the MPEG Protocol Analysis 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 "B4645B - MPEG Protocol Analysis" 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 MPEG Protocol Analysis Toolset.

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 MPEG Protocol Analysis Toolset, you need to connect logic analyzer probe channels to:

- An MPEG stream data bus.
- Control signals that identify the start of packet, valid data, and the end of packet (if these signals exist).

If there is no start of packet signal, the protocol definitions file should define a unique data value that identifies a start of packet (see “Physical Layer Definition” on page 46).

NOTE:

We strongly recommend either: a start of packet signal, or a unique start of packet data value in the protocol definitions file. Otherwise, proper decoding cannot be guaranteed.

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

- A clock signal that identifies when the data bus and control signals are 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)

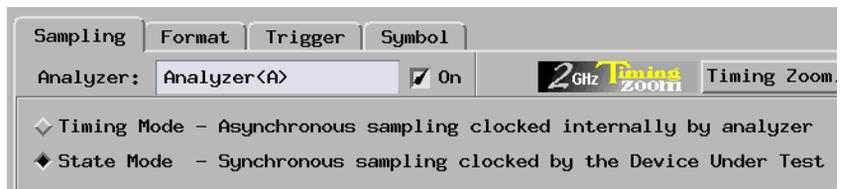
Step 2. Select the state sampling mode

When using the MPEG Protocol Analysis 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.

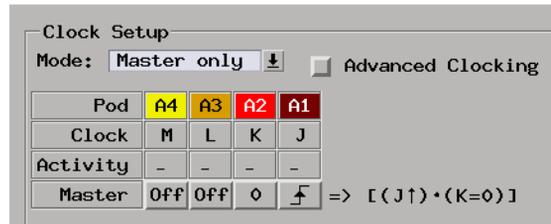


3. Specify the state analyzer's sampling clock input.

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:



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 MPEG Tool requires you to set one DATA label for the MPEG data bus channels.

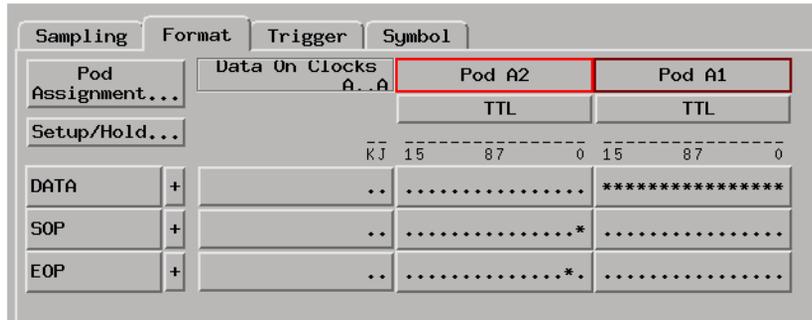
If the protocol definitions file's "Physical Layer Definition" on page 46 doesn't define the start packet width and value, you must set up a 1-bit label for the start of packet signal.

If there are signals for data valid and end of packet, set up 1-bit labels for those signals as well.

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 DATA label and 1-bit labels for the start of packet, data valid, and end of packet signals.
2. For the DATA label, specify which logic analyzer pod channels are probing the data bus.

- For the 1-bit labels, specify which logic analyzer pod channels are probing the start of packet, data valid, and end of packet signals.



In the Bus Editor dialog, you will specify which labels contain the start of packet, data valid, and end of packet signals.

See Also "Capturing the Data" on page 16

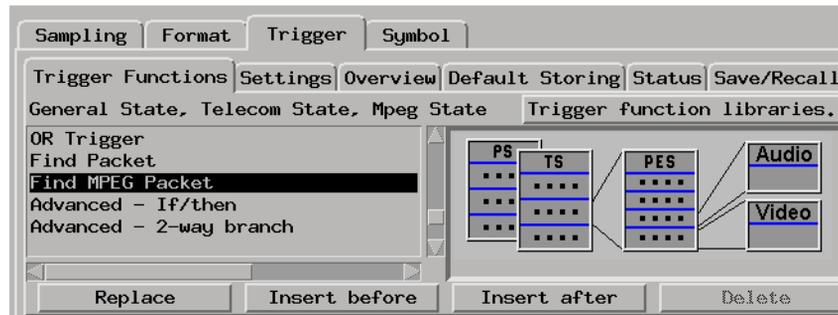
Capturing the Data

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

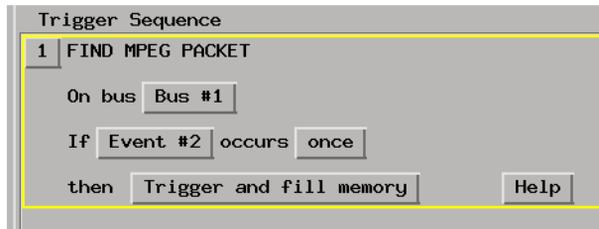
- “To set up MPEG packet triggers” on page 16
- “To define and edit packet events” on page 21 (“Event Editor Dialog” on page 23)
- “To run the measurement” on page 25

To set up MPEG packet triggers

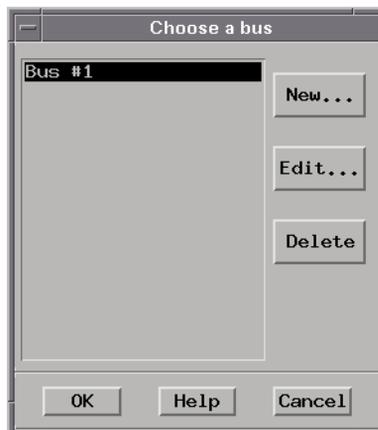
1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
2. Select the "Find MPEG Packet" trigger function and either replace the current trigger sequence level or insert a new level.



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

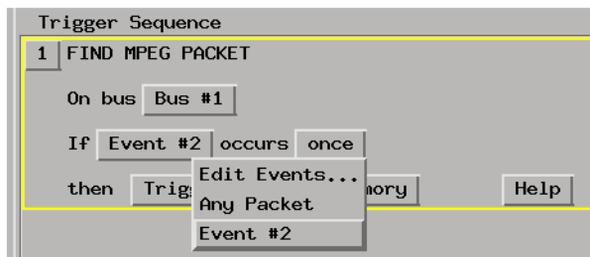


4. In the Choose a Bus dialog, select the bus definition you want to use and select the OK button.



If you need to define or edit a bus, see “To define and edit buses” on page 18.

5. Specify the packet event to find.



NOTE:

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

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

See Also

“Modifying Protocol Definitions” on page 36

“To define and edit buses” on page 18

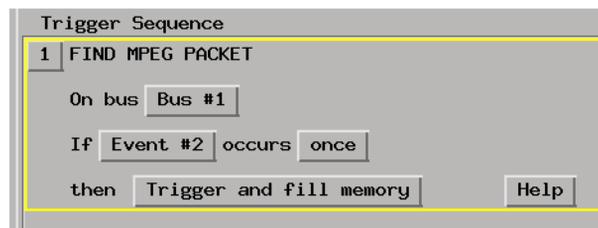
“To define and edit packet events” on page 21

“To run the measurement” on page 25

“Displaying the Data” on page 26

To define and edit buses

1. In the "Find MPEG Packet" trigger function (or in the MPEG Tool's Setup tab), select the bus button.



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

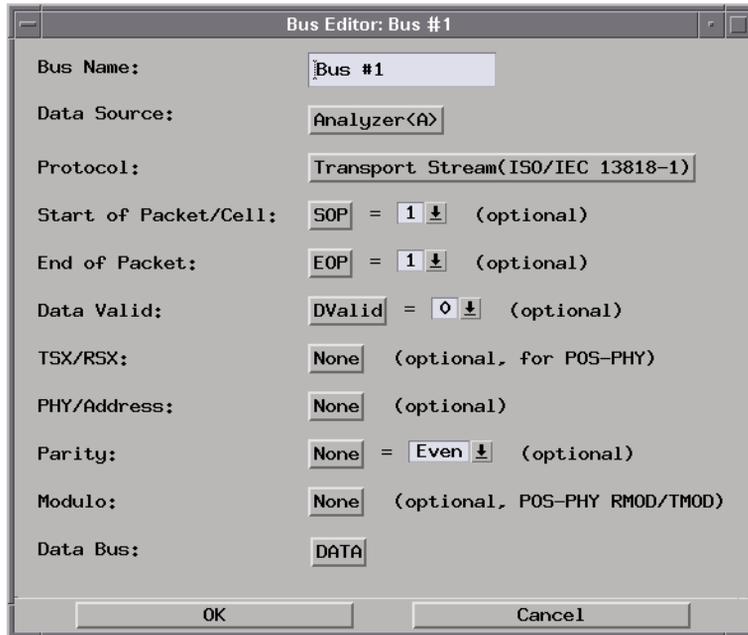


3. In the Bus Editor dialog, specify the characteristics of the bus.
4. When you are done editing the bus, select the OK button in the Bus Editor dialog and select the OK button in the Choose a Bus dialog.

See Also

“Bus Editor Dialog” on page 20

Bus Editor Dialog



NOTE:

Because the Bus Editor dialog is also used by the DataComm Analysis Toolset, it has items that are not related to the MPEG Protocol Analysis Toolset.

- | | |
|----------------------------------|---|
| Bus Name | Specifies the bus definition name. |
| Data Source | Specifies which logic analyzer module captures data on the bus. |
| Protocol | Selects the protocol being used on the bus. |
| Start of Packet/ Cell | Selects the 1-bit label for the signal and value that indicates a start of packet. If no label is selected, the "Any Packet" event option in the Find MPEG Packet trigger becomes "Any Packet w/o SOP" to show that the bus was defined without a start of packet label. |
| End of Packet | Selects the 1-bit label for the signal and value that indicates |

| | |
|--------------------|--|
| | an end of packet. |
| Data Valid | Selects the 1-bit label for the signal and value that indicates valid data is on the bus. |
| TSX/RSX | Not used by the MPEG Protocol Analysis Toolset. |
| PHY/Address | Not used by the MPEG Protocol Analysis Toolset. |
| Parity | Not used by the MPEG Protocol Analysis Toolset. |
| Modulo | Not used by the MPEG Protocol Analysis Toolset. |
| Data Bus | Selects the label for the data bus being probed. Because most MPEG buses are 8 bits or less (well within than the 32-bit limit for a label), the Insert Before... and Insert After... options for selecting multiple data labels are not supported by the MPEG Protocol Analysis Toolset. |

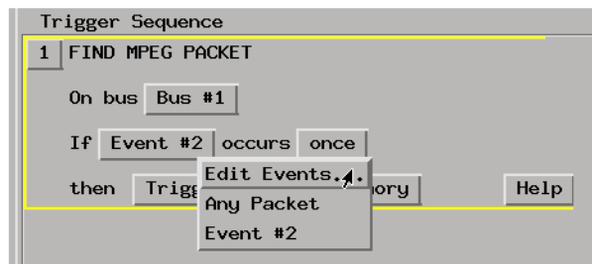
See Also

“To define and edit buses” on page 18

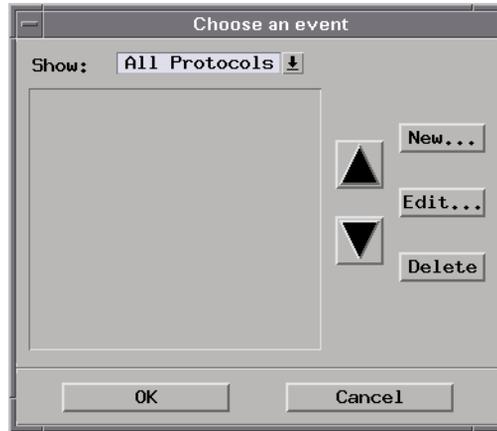
“To choose a protocol” on page 36

To define and edit packet events

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



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



To delete an event definition, select the Delete button.

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

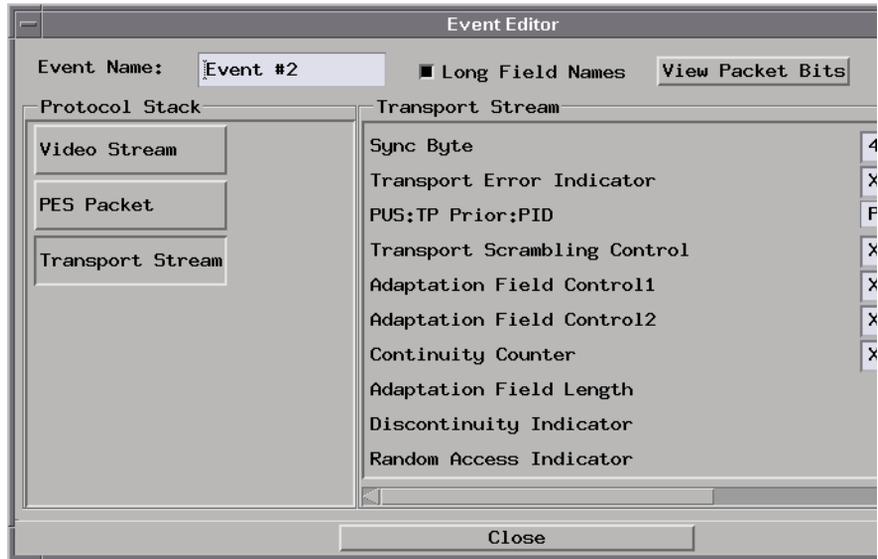
NOTE:

The "Any Packet w/o SOP" event appears when the selected bus does not define a label for the start of packet signal. The "Any Packet w/o SOP" event cannot be edited or deleted.

See Also

“Event Editor Dialog” on page 23

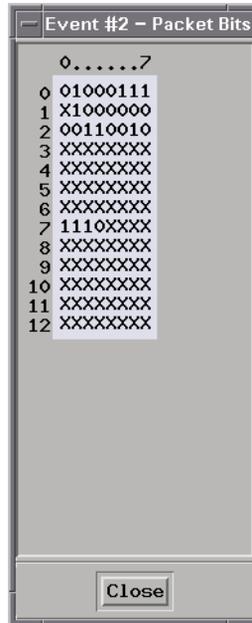
Event Editor Dialog



Event Name Specifies the event definition name.

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

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



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

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

NOTE:

Some fields in a protocol are only valid when previous fields contain particular values (as specified in the protocol definitions file); because of this, some of the boxes for entering protocol field values only appear when particular values are entered into previous fields.

See Also

“To define and edit packet events” on page 21

To run the measurement

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

See Also

“Displaying the Data” on page 26

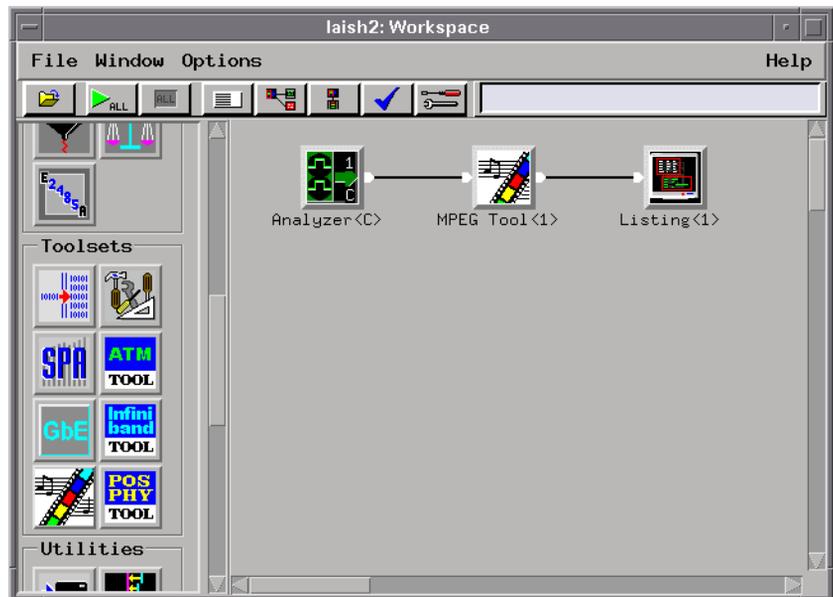
Displaying the Data

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

- “To connect the MPEG tool (in the workspace)” on page 26
 - “To set up the MPEG tool” on page 27
 - “To select the output data columns” on page 29
 - “To filter data rows” on page 33
-

To connect the MPEG tool (in the workspace)

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



The MPEG tool will immediately try to decode the captured trace data.

If no trace data has been captured, run a measurement.

See Also

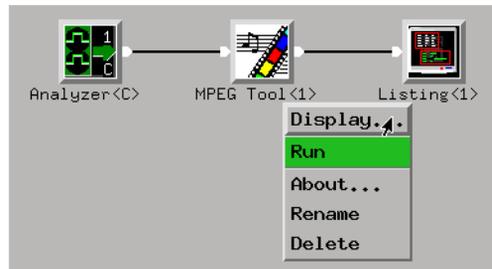
“To set up the MPEG tool” on page 27

“To select the output data columns” on page 29

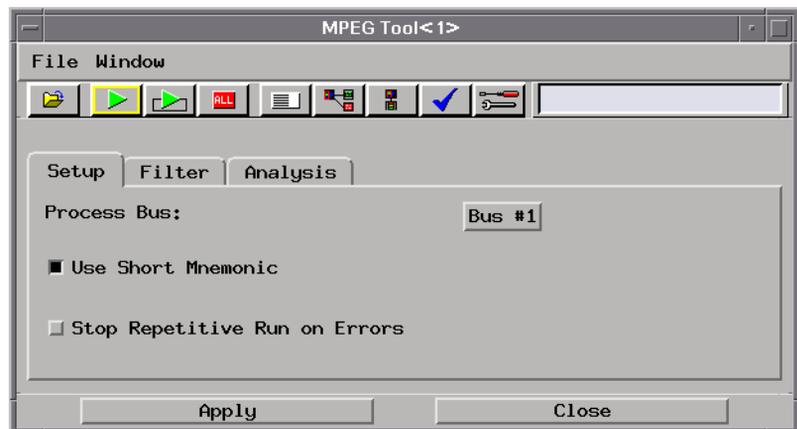
“To filter data rows” on page 33

To set up the MPEG tool

1. Display the MPEG tool.



2. In the MPEG tool's Setup tab, make the appropriate selections for the setup options:



Chapter 2: Task Guide

Displaying the Data

Process Bus Specifies the bus definition to use when processing the captured data.

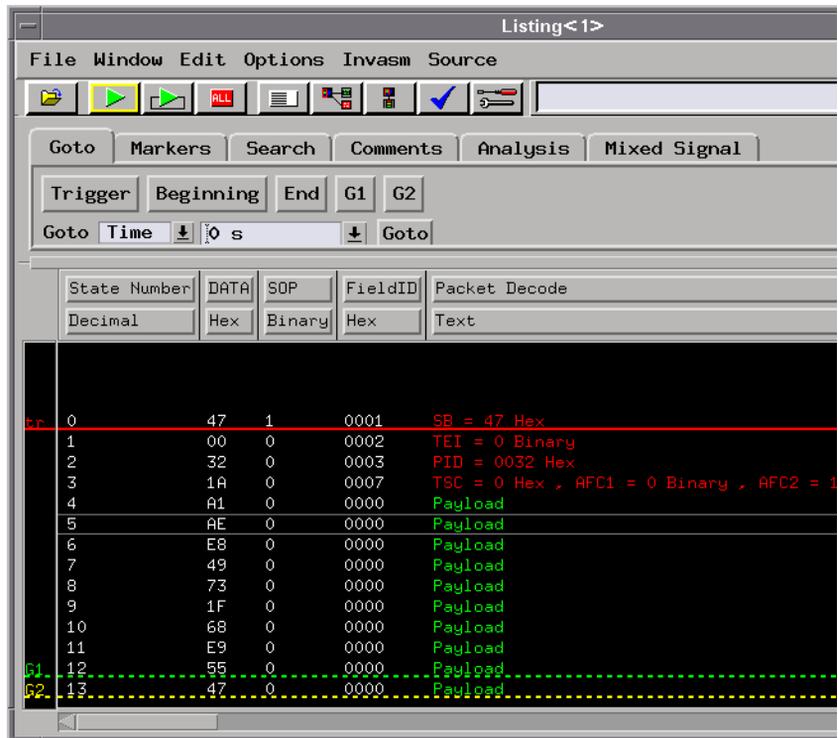
Use Short Mnemonic Specifies whether the decoded data is output with full text descriptions or short mnemonics. When short mnemonics are used, the decoded data is limited to one line.

Stop Repetitive Run on Errors Specifies whether errors will stop repetitive run measurements. Note that an error column must appear in the output data in order for an error of that type to be detected.

NOTE:

When using repetitive runs, all of the captured data in an individual run must be decoded (in order to check for errors) before the next run can occur. If your logic analyzer has deep memory, decoding the entire trace can take a long time. Therefore, we recommend that you limit the trace memory depth to less than 1M states when using repetitive runs.

3. Select Apply to change the setup.



4. Select Close to change the setup and close the MPEG tool.

See Also

“To select the output data columns” on page 29

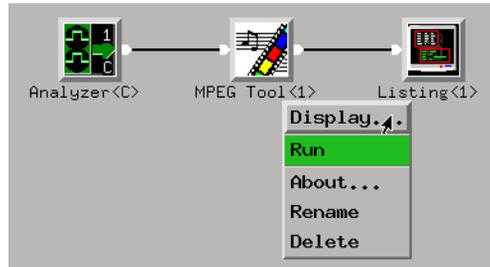
“To filter data rows” on page 33

To select the output data columns

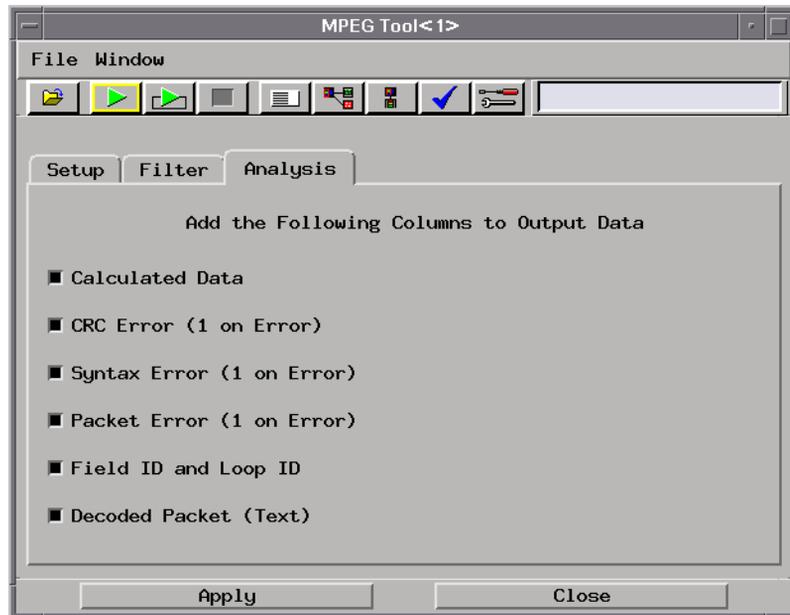
1. Display the MPEG tool.

Chapter 2: Task Guide

Displaying the Data



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



Calculated Data This 32-bit column contains the result of a formula calculation or "0" if there is no formula description for the field in the protocol file. If there are multiple fields in the captured data, only the formula result for the last field will be displayed.

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

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

Packet Error This 1-bit column contains "1" whenever there is an inconsistency between a packet length description in the protocol definitions file and the captured start of packet, valid data, or end of packet.

Field ID and Loop ID

This 16-bit column contains two 8-bit chunks of data. The lower 8-bits is the ID of the field as defined in the protocol definition file. The upper 8-bits contains the count of the field if the field is defined in a loop description (that is, an UNTIL block or a STUFF_UNTIL block) in the protocol file. This data is for post-processing by the Tool Development Kit.

Decoded Packet This text column contains a description of the decoded data. Up to 512 characters per state can be displayed.

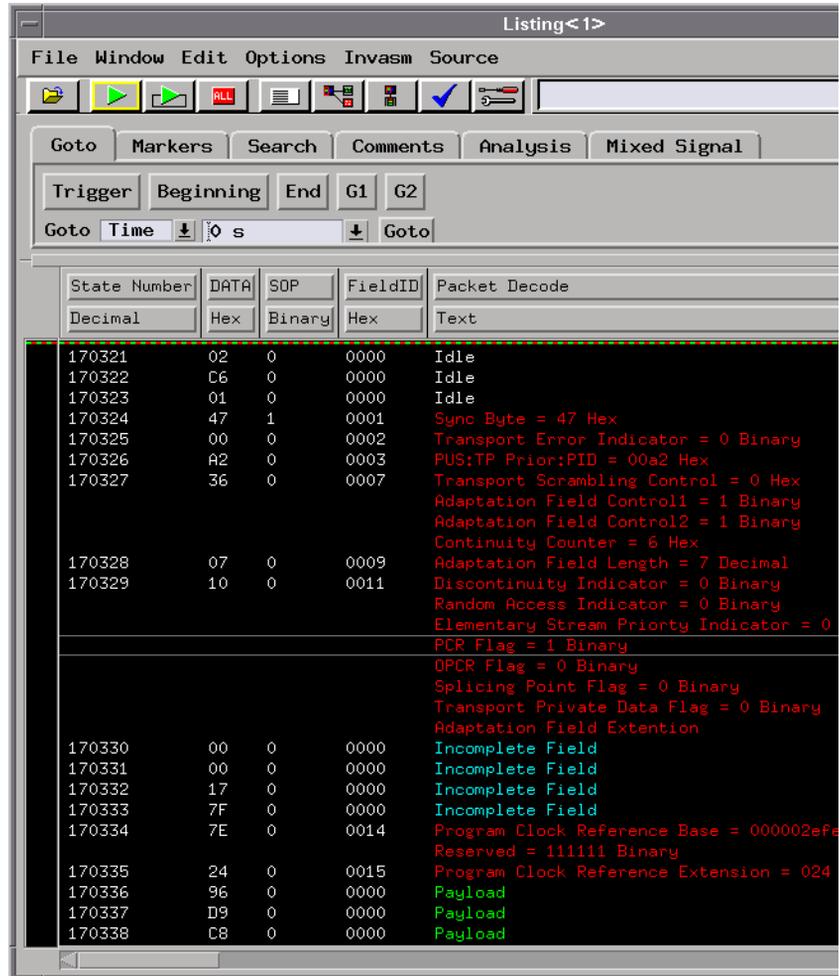
NOTE:

An error column must appear in the output in order for a repetitive run to be stopped on that type of error. Errors are always displayed in the decoded packet text column.

3. Select Apply to output the selected data columns.

Chapter 2: Task Guide

Displaying the Data



4. Select Close to close the MPEG tool.

See Also

“To set up the MPEG tool” on page 27

“To filter data rows” on page 33

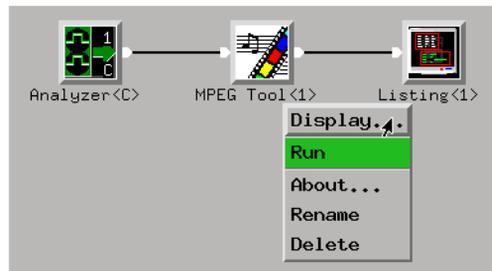
“Formulas” on page 54

“CRCEnd” on page 51

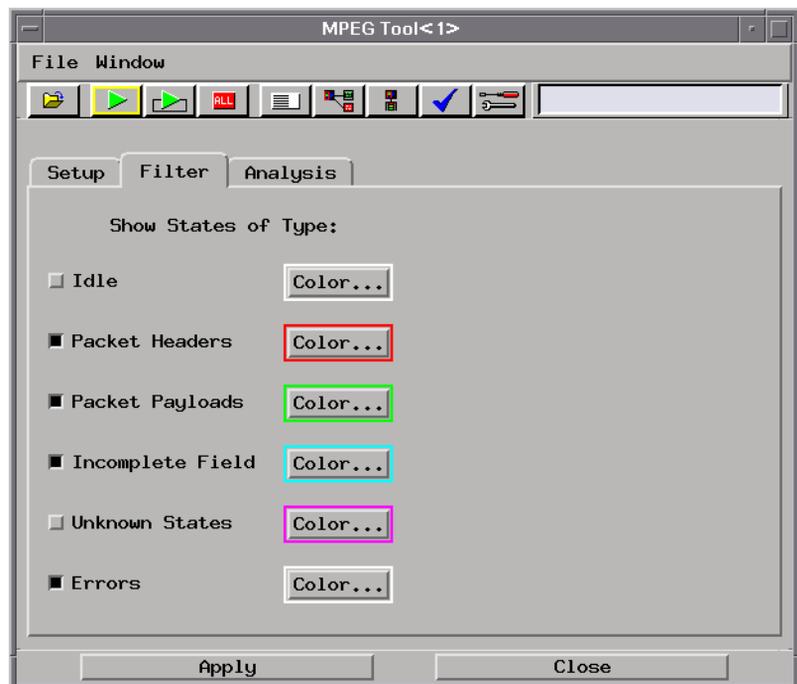
“SyntaxCheck” on page 50

To filter data rows

1. Display the MPEG tool.



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



Chapter 2: Task Guide

Displaying the Data

Idle Idle states.

Packet Headers Packet header states.

Packet Payloads Packet payload states.

Incomplete

Field These states contain only part of a defined field. For example, a 16-bit field captured on an 8-bit bus will take 2 states to complete.

Unknown States These states can occur when there is no end of packet signal, no "valid" signal, or no defined packet length in the protocol definition file; in these cases, the MPEG Tool cannot determine whether the states are Payload states or Idle states.

Errors These occur when there is a contradiction between bus signals and the packet description in the protocol definitions file:

- Illegal EOP. The end of packet signal occurs during Idle states.
- Illegal VALID. The data valid signal occurs without a start of packet signal or packet start data.
- Illegal SOP. The start of packet signal occurs before the previous packet is finished.
- Packet Size and VALID are inconsistent. The valid data signal occurs after the packet size length.
- Packet length expired within a header. The packet length end occurs before header decoding is finished.

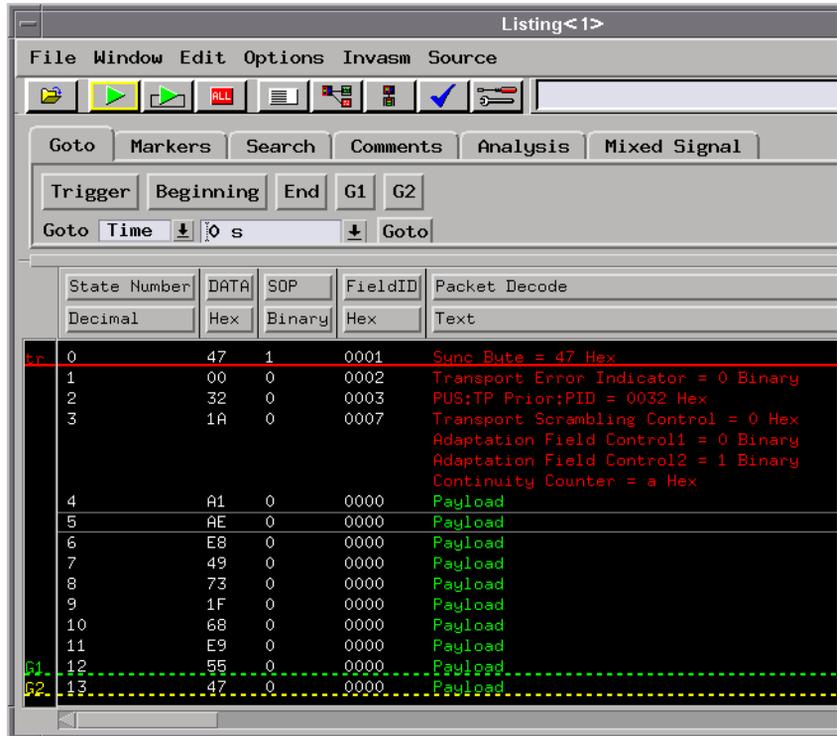
NOTE:

When a start of packet is found before the previous packet ends, decoding re-starts, and the previous packet decode is corrupted.

When an end of packet is found before the previous packet has ended (or before its defined length), packet decoding stops. This happens even while decoding packet headers.

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 MPEG tool.

See Also

“To set up the MPEG tool” on page 27

“To select the output data columns” on page 29

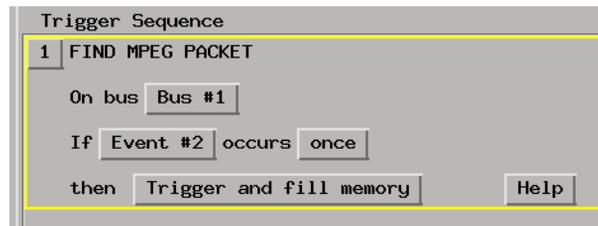
Modifying Protocol Definitions

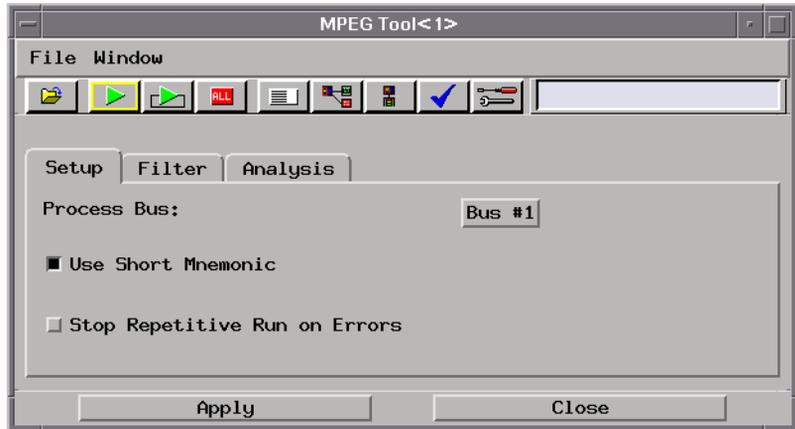
This section shows you how to modify the protocol definitions used with the MPEG Protocol Analysis Toolset.

- “To choose a protocol” on page 36
 - “To modify a protocol definition” on page 39
 - “To reload the protocol definitions” on page 40
 - “To reset the protocol definitions” on page 40
-

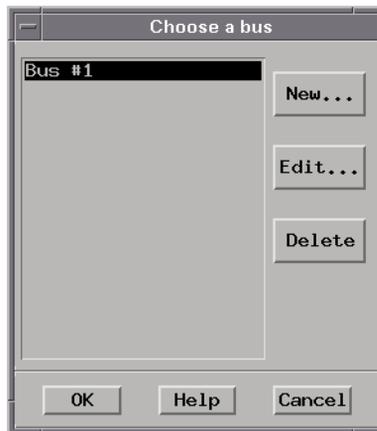
To choose a protocol

- In the "Find MPEG Packet" trigger function, or in the MPEG Tool's Setup tab, select the bus button.



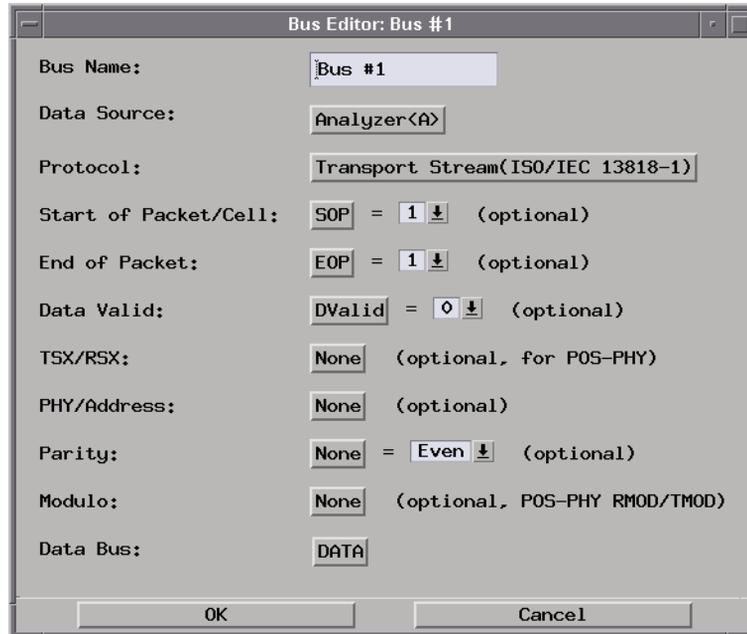


- In the Choose a Bus dialog, either select New... to define a new bus or select the bus you want to edit from the list and select the Edit... button.

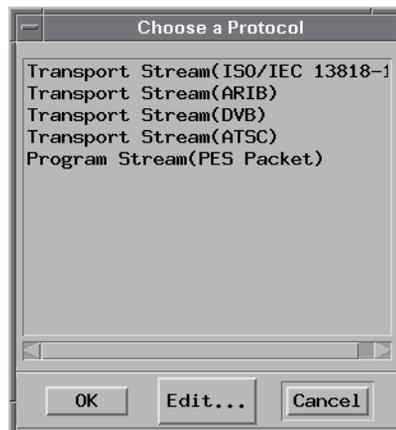


- In the Bus Editor dialog, select the Protocol button.

Chapter 2: Task Guide
Modifying Protocol Definitions



- In the Choose a Protocol dialog, select the protocol from the list.



See Also

“To modify a protocol definition” on page 39

“To reload the protocol definitions” on page 40

“To reset the protocol definitions” on page 40

To modify a protocol definition

1. Open the Choose a Protocol dialog (see “To choose a protocol” on page 36), select the Edit... button, and choose either GUI Text Editor, 'vi' Editor, or Edit Protocols Offline.



2. When you are done editing protocol definitions, choose the Reload Protocols command to tell the MPEG Protocol Analysis Toolset that changes have been made (see “To reload the protocol definitions” on page 40).

About the Protocol Definitions File

The protocol definitions file is:

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

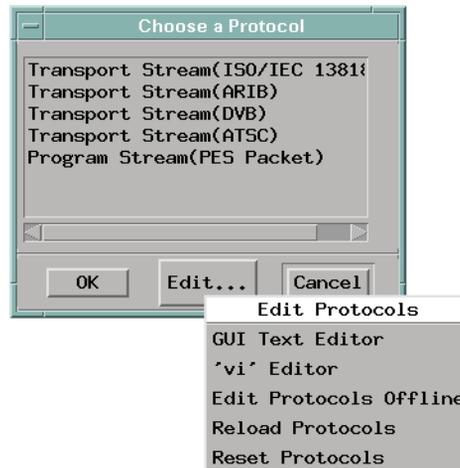
See Also

“Protocol Definition Syntax” on page 44

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 MPEG Protocol Analysis Toolset that changes have been made.

1. Open the Choose a Protocol dialog (see “To choose a protocol” on page 36), 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. Open the Choose a Protocol dialog (see “To choose a protocol” on page 36), select the Edit... button, and choose Reset Protocols.



Modifying Protocol Definitions

Reference

Protocol Definition Syntax

This section describes the protocol definition syntax.

Protocol definitions are used by: the Event Editor to let you set up MPEG packet triggers, and by the MPEG Tool to decode captured data.

- “Protocol Block” on page 46
- “Physical Layer Definition” on page 46
- “Header Block” on page 47
- “Field Definition” on page 47
 - “Data” on page 48
 - “LengthData” on page 48
 - “Symbols” on page 49
 - “ProtocolIndicator” on page 49
 - “SyntaxCheck” on page 50
 - “CRCStart” on page 51
 - “CRCEnd” on page 51
- “Numeric Values” on page 51
- “IF Conditional Block” on page 51
- “UNTIL Loop Block” on page 52
- “STUFF_UNTIL Loop Block” on page 53
- “Formulas” on page 54

Example

Here is the definition for the Transport Stream protocol:

```
Protocol
{
  Name "Transport Stream"

  PhyLayer 1 8 #h47 1504

  Header
  {
    SB "Sync Byte"                8 Hex    SyntaxCheck #h47
```

```

TEI "Transport Error Indicator" 1 Binary Data
PID "PUS:TP Prior:PID" 15 Hex ProtocolIndicator
{
  "Program Association Table" #h4000
  "Conditional Access Table" #h4001
  "PES Packet" #h4032
  "PES Packet" #h4033
  "PES Packet" #h4034
  "PES Packet" #h40a0
  "PES Packet" #h40a1
  "PES Packet" #h40a2
  "Program Map Table" #h4400
  "Program Map Table" #h4401
  "Program Map Table" #h4402
}
TSC "Transport Scrambling Control" 2 Hex Data
AFC1 "Adaptation Field Control1" 1 Binary Data
AFC2 "Adaptation Field Control2" 1 Binary Data
CC "Continuity Counter" 4 Hex Data
IF ( AFC1, #h1, == )
{
  AFL "Adaptation Field Length" 8 Decimal LengthData
  DCI "Discontinuity Indicator" 1 Binary Data
  RAI "Random Access Indicator" 1 Binary Data
  ESPI "Elementary Stream Priority Indicator" 1 Binary Data
  PCRF "PCR Flag" 1 Binary Data
  OPCRF "OPCR Flag" 1 Binary Data
  SPF "Splicing Point Flag" 1 Binary Data
  TPDF "Transport Private Data Flag" 1 Binary Data
  AFEF "Adaptation Field Extension Flag" 1 Binary Data
  IF ( PCRF, #h1, == )
  {
    PCRB "Program Clock Reference Base" 33 Hex Data
    RSV "Reserved" 6 Binary Data
    PCRE "Program Clock Reference Extension" 9 Hex Data
  }
  IF ( OPCRF, #h1, == )
  {
    OPCRB "Original Program Clock Reference Base" 33 Hex Data
    RSV "Reserved" 6 Binary Data
    OPCRE "Original Program Clock Reference Extension" 9 Hex Data
  }
  IF ( SPF, #h1, == )
  {
    SCD "Splice Countdown" 8 Hex Data
  }
  IF ( TPDF, #h1, == )
  {
    TPDL "Transport Private Data Length" 8 Decimal LengthData
    UNTIL ( TPDL )
    {
      PDB "Private Data byte" 8 Hex Data
    }
  }
  IF ( AFEF, #h1, == )
  {
    AFEL "Adaptation Field Extension Length" 8 Decimal LengthData
    LTWF "LTW Flag" 1 Binary Data
    PRF "Piecewise Rate Flag" 1 Binary Data
    SSF "Seamless Splice Flag" 1 Binary Data
    RSV "Reserved" 5 Binary Data
    IF ( LTWF, #h1, == )
    {
      LTWVF "LTW Valid Flag" 1 Binary Data
      LTWO "LTW Offset" 15 Hex Data
    }
    IF ( PRF, #h1, == )
    {
      RSV "Reserved" 2 Binary Data
      PR "Piecewise Rate" 22 Hex Data
    }
  }
  STUFF_UNTIL ( AFEL )
}

```

```
        {
          RSV "Reserved" 8 Hex Data
        }
      }
    STUFF_UNTIL ( AFL )
    {
      SB "Stuffing Byte" 8 Hex Data
    }
  }
}
```

See Also “To define and edit packet events” on page 21
 “Displaying the Data” on page 26

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 "Transport Stream"
}
```

See Also “Physical Layer Definition” on page 46
 “Header Block” on page 47

Physical Layer Definition

Syntax

```
PhyLayer 1 [bitlength value] [bitlength]
```

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

The "PhyLayer" 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 MPEG data bus. You won't expect to see Audio Stream packets without some layers below them (for example, Audio Stream over PES Packet over Transport Stream); therefore, you are given the option of specifying which protocols can be MAC layers.

The first optional bitlength and value are for specifying the length and value of a start packet. If no start of packet signal is described in the Bus Editor, the start of packet length and value must be described in the protocol definition file.

The second optional bitlength is for the total packet length (including the Header and Payload).

Example

```
PhyLayer 1 8 #h47 1504
```

Event Editor Notes

If a start packet is described in the protocol definition file, its value will appear in the event editor but cannot be modified.

See Also

“To define and edit buses” on page 18

“Step 3. Label the logic analyzer channels” on page 14

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 47

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 48
- “LengthData” on page 48
- “Symbols” on page 49
- “ProtocolIndicator” on page 49
- “SyntaxCheck” on page 50
- “CRCStart” on page 51
- “CRCEnd” on page 51

Example

```
SB "Sync Byte" 8 Hex SyntaxCheck #h47
```

This is the first field in the Transport Stream header, the Sync Byte. It is 8 bits long. It should be displayed as a Hex value (for example, 47 Hex), and it's a "SyntaxCheck" field.

Data

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

See Also

“Field Definition” on page 47

LengthData

LengthData indicates that the field length will be used for a "STUFF_UNTIL" later. If there is no "STUFF_UNTIL" that uses this

field length, this field type is the same as Data.

See Also

“Field Definition” on page 47

“STUFF_UNTIL Loop Block” on page 53

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.

As an example, here is the "Precedence" field from an IP (Internet Protocol) definition:

```
TMC "Trick Mode Control" 3 Binary Symbols
{
  "Fast forward" #b000
  "Slow motion" #b001
  "Freeze frame" #b010
  "Fast reverse" #b011
  "Slow reverse" #b100
  "Reserved" #b101
  "Reserved" #b110
  "Reserved" #b111
}
```

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 47

“Numeric Values” on page 51

ProtocolIndicator

The ProtocolIndicator field type is how a protocol references the next layer of the protocol stack. In Transport Stream, this is the "PUS:TP Prior:PID" field. In PES Packet, it's the "Stream ID" field.

The syntax for this type is exactly the same as for “Symbols” on page 49, except that each symbol's name is the name of another protocol, which must also be defined in the protocol file.

For example, here is the "PUS:TP Prior:PID" field of the Transport Stream protocol:

Chapter 3: Reference

Protocol Definition Syntax

```
PID "PUS:TP Prior:PID" 15 Hex ProtocolIndicator
{
  "Program Association Table" #h4000
  "Conditional Access Table" #h4001
  "PES Packet" #h4032
  "PES Packet" #h4033
  "PES Packet" #h4034
  "PES Packet" #h40a0
  "PES Packet" #h40a1
  "PES Packet" #h40a2
  "Program Map Table" #h4400
  "Program Map Table" #h4401
  "Program Map Table" #h4402
}
```

A value of 0x4000 in the "PUS:TP Prior:PID" field indicates Program Association Table, which is also defined in the file.

NOTE:

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

In a ProtocolIndicator block, don't cares ("x" or "X") can be used in Hex, Octal, and Binary numeric values:

```
SID "Stream ID" 8 Hex ProtocolIndicator
{
  "Program Stream Map" #hbc
  "Program Stream Directory" #hff
  "Audio Stream" #b110xxxxx
  "Video Stream" #hex
}
```

When using don't cares this way, you must ensure the numeric values are mutually exclusive; otherwise, the result of the definition is unpredictable. (The syntax check of the protocol file will not result in an error.)

See Also

“Field Definition” on page 47

“Symbols” on page 49

“Numeric Values” on page 51

SyntaxCheck

The SyntaxCheck field type is followed by a value, and it indicates that the field needs to be compared to the value. If the field value is different than the specified value, a "1" is output in the Syntax Error column.

See Also

“Field Definition” on page 47

“To select the output data columns” on page 29

CRCStart

CRCStart indicates that a CRC code calculation should start from this field.

See Also

“Field Definition” on page 47

CRCEnd

CRCEnd indicates that a CRC calculation was finished in the previous field and that this field includes a CRC code. The MPEG Tool calculates the CRC code and compares it to the value in this field; if the values are different, a "1" is output in the CRC Error column.

See Also

“Field Definition” on page 47

“To select the output data columns” on page 29

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'.

IF Conditional Block

Syntax

```
IF ('Formula')  
{  
  field definitions  
}
```

If the formula result is TRUE, the field definitions in the block are valid. If the formula result is FALSE, the definitions are skipped.

IF blocks can be nested.

IF-ELSE is not supported. When using consecutive IF blocks to mimic IF-ELSE, you must make sure the formula results are mutually exclusive.

Example

```
IF ( PCRF, #h1, == )
{
  PCRB  "Program Clock Reference Base"    33  Hex    Data
  RSV   "Reserved"                       6   Binary Data
  PCRE  "Program Clock Reference Extension" 9   Hex    Data
}
```

Event Editor Notes

If data entered in the Event Editor results in the formula being TRUE, the fields in the IF block are shown and may be edited; otherwise, the fields are not shown.

See Also

“Formulas” on page 54

“Field Definition” on page 47

UNTIL Loop Block

Syntax

```
UNTIL ('Formula')
{
  field definitions
}
```

The entire contents of the block is repeated until the total field size is equal or greater than the value of the formula.

For example, if the UNTIL block is 46 bits and the result of the formula is 10 bytes, the block will be repeated twice, resulting in a 92-bit definition.

Example

```
UNTIL ( HL, #h6, -)
{
  SID   "Stream ID"                8  Hex    Data
  FXD   "11"                       2  Binary Data
  PSBBS "P-STD Buffer Bound Scale"  1  Binary Data
  PSBSB "P-STD Buffer Size Bound" 13  Binary Data
}
```

Event Editor Notes

If data entered in the Event Editor yields a formula result, the fields in the UNTIL block are shown one time but cannot be edited.

Decode Notes

When the MPEG Tool decodes using UNTIL blocks, decoding doesn't stop if the packet length is greater than the defined bit length. This is because decoding the field is a higher priority than checking the packet length.

See Also

"Formulas" on page 54

STUFF_UNTIL Loop Block

Syntax

```
STUFF_UNTIL ('Short Mnemonic of field with type LengthData')
{
  field definitions
}
```

For fields that have a specified length (defined using the LengthData field type), but which also can have optional sub-fields with varying lengths, the STUFF_UNTIL block defines the stuffing-fields that are inserted at the end of the field until it reaches its specified length.

The short mnemonic used must be that of a field whose type is LengthData; otherwise, the result of the STUFF_UNTIL block is unpredictable. (The syntax check of the protocol file will not return an error.)

Example

```
IF ( AFEF, #h1, == )
{
  AFEL "Adaptation Field Extension Length" 8 Decimal LengthData
  LTWF "LTW Flag" 1 Binary Data
  PRF "Piecewise Rate Flag" 1 Binary Data
  SSF "Seamless Splice Flag" 1 Binary Data
  RSV "Reserved" 5 Binary Data
  IF ( LTWF, #h1, == )
  {
    LTWVF "LTW Valid Flag" 1 Binary Data
    LTWO "LTW Offset" 15 Hex Data
  }
  IF ( PRF, #h1, == )
  {
    RSV "Reserved" 2 Binary Data
    PR "Piecewise Rate" 22 Hex Data
  }
  STUFF_UNTIL ( AFEL )
  {
    RSV "Reserved" 8 Hex Data
  }
}
```

Event Editor Notes

If a LengthData value is entered in the Event Editor, the fields in the STUFF_UNTIL block are shown one time but cannot be edited.

Decode Notes

When the MPEG Tool decodes using STUFF_UNTIL blocks, decoding doesn't stop if the packet length is greater than the defined bit length. This is because decoding the field is a higher priority than checking the packet length.

See Also

“LengthData” on page 48

Formulas

Syntax

```
[ 'Short Mnemonic' | 'value' | operator ] , ...
```

Formulas must be in Reverse Polish Notation, and each term must be separated by a comma (,).

Formulas can be used in IF blocks, UNTIL blocks, or in regular field definitions, like:

```
'Mnem' "Full" 'Width(bits)' 'Format' 'Type' ('Formula') [ 'Format' ]  
IF ('Formula')  
UNTIL ('Formula')
```

When a formula is used in a regular field definition:

- The result of the formula is displayed in the "FieldData" output column.
- You have the option of giving the formula result its own number format.

Operators

| | |
|----|--|
| + | Add. |
| - | Subtract. |
| * | Multiply. |
| / | Divide. |
| & | Bitwise AND. |
| | Bitwise OR. |
| && | Logical AND. |
| | Logical OR. |
| == | Equal (comparison results in TRUE or FALSE). |
| != | Not equal (comparison results in TRUE or FALSE). |

Example

```
IF ( SPF, #h1, == )  
{  
  SCD "Splice Countdown" 8 Hex Data  
}
```

See Also

“IF Conditional Block” on page 51

“UNTIL Loop Block” on page 52

“Field Definition” on page 47

“To select the output data columns” on page 29

Messages

This section describes the messages that can occur when using the MPEG Protocol Analysis Toolset.

- “CRC code field is not byte aligned” on page 56
 - “CRC Start field is not byte aligned” on page 56
 - “Field longer than 32 bits cannot be used in a formula” on page 57
 - “Field longer than 32 bits cannot be used with SyntaxCheck” on page 57
 - “Size of header in packet too large” on page 57
 - “Start data width is not equal to the first field width” on page 57
-

CRC code field is not byte aligned

This error occurs during decode.

The CRC End field must be byte aligned.

See Also

“CRCEnd” on page 51

CRC Start field is not byte aligned

This error occurs during decode.

The CRC Start field must be byte aligned.

See Also

“CRCStart” on page 51

Field longer than 32 bits cannot be used in a formula

This message occurs when the protocol is loaded.

Fields which are longer than 32 bits cannot be used in a formula.

See Also “Formulas” on page 54

Field longer than 32 bits cannot be used with SyntaxCheck

This message occurs when the protocol is loaded.

Fields which are longer than 32 bits cannot be used in a syntax check.

See Also “SyntaxCheck” on page 50

Size of header in packet too large

This error can occur during decode.

The header length in a packet must be shorter than 8K bytes (65536 bits).

Start data width is not equal to the first field width

The packet start data length defined in the Physical Layer Definition must correspond to the first field's width.

See Also “Physical Layer Definition” on page 46

Limitations

This section describes limitations of the MPEG Protocol Analysis Toolset.

When setting up MPEG packet triggers:

- If the Adaptation Field Length data is not set and field data is set in the next protocol, the logic analyzer may not trigger correctly.

When displaying captured data:

- In the ATSC protocol, if there are two descriptors defined and the descriptor just before CRC_32 exists, the decoding around this descriptor is incorrect.

Glossary

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.

Glossary

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.

Glossary

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

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

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

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

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

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

edge 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

Glossary

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

Glossary

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 Interface 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

Glossary

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

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

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

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

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

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

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

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

pod See *pod pair*

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

preprocessor See *analysis probe*.

primary branch The primary

Glossary

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 message.

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*.

Glossary

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

Glossary

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.

A

Any Packet, 20
Any Packet w/o SOP, 20, 21
ASCII text editor, 39, 40
auxiliary software, 9

B

Binary field format, 47
binary numbers in protocol
 definitions, 51
Bus Editor dialog, 20
bus name, bus editor, 20
bus to process, 27
buses, defining/editing, 18
buses, multiple, 14

C

calculated data, 29
capturing the data, 16
CD-ROM drive, 9
channels, labeling logic analyzer,
 14
check if toolset installed, 8
Choose a Protocol dialog, 36
choosing a protocol, 36
CLK input channel, logic analyzer,
 12
color for state types, 33
communication data bus, 12
connecting logic analyzer probes,
 12
connecting the MPEG tool, 26
control signals, 12
CRC error, 29
CRCEnd field type, 51
CRCStart field type, 51

D

data bus, bus editor, 20
Data field type, 48
DATA label required, 14

data source, bus editor, 20
data valid, 12, 14, 33
data valid, bus editor, 20
data, capturing, 16
data, displaying, 26
Decimal field format, 47
decimal numbers in protocol
 definitions, 51
decoded packet, 29
deep trace memory and repetitive
 runs, 27
defining buses, 18
defining packet events, 21
demo period, 8
device under test, probing, 12
displaying the data, 26

E

editing buses, 18
editing packet events, 21
editing protocol definitions, 39
editing protocol definitions with vi
 editor, 39
end of packet, 12, 14, 33
end of packet, bus editor, 20
entitlement certificate, 10
error, CRC, 29
error, syntax, 29
errors, 33
errors, stop repetitive run on, 27
Event Editor dialog, 23
event name, event editor, 23

F

field definition, 47
field format, 47
field ID, 29
field type, 47
FieldData output column, 54
filter data rows, 33
Find MPEG Packet trigger
 function, 2, 16, 36

Format tab, 14
formulas, 54

G

Group Run button, 25

H

Header block, 47
Hex field format, 47
hex numbers in protocol
 definitions, 51

I

ID, field, 29
ID, loop, 29
idle states, 33
IF conditional block, 51
incomplete field, 33
installation, 8, 9
installed, check if toolset, 8
internal triggering resources, logic
 analyzer, 16
interrupted packet transfers, 12,
 13

J

J sampling clock input, 13

K

K sampling clock input, 13

L

labeling logic analyzer channels, 14
Length Data field type, 48
licensing, 8, 10
limitations, 58
logic analyzer channels, labeling,
 14
logic analyzer CLK input channel,
 12

logic analyzer internal triggering
resources, 16
logic analyzer modules supported,
2
logic analyzer probes, connecting,
12
long field names, event editor, 23
loop ID, 29

M

MAC layer, 46
measurement, running, 25
memory depth (trace) and
repetitive runs, 27
messages, 56
mnemonic, short, 27
modifying protocol definitions, 36,
39
MPEG packet triggers, 16
MPEG Protocol Analysis Toolset,
using, 2
MPEG tool, connecting, 26
MPEG tool, setup, 27
multiple communications buses, 14

N

networked computer, 40
numeric values in protocol
definitions, 51

O

Octal field format, 47
octal numbers in protocol
definitions, 51
output data columns, selecting, 29

P

packet bits display, event editor, 23
packet error, 29
packet events, 16
packet events, defining/editing, 21

packet headers, 33
packet payloads, 33
packet start value, 16
packet transfer interrupts, 12, 13
packet triggering, 2
packet, decoded, 29
password center, 10
PES Packet protocol, 2
physical layer, 36, 46
physical layer protocol, 16
pre-defined protocols, 2
probing the device under test, 12
process bus, 27
Program Stream protocol, 2
protocol definition syntax, 44
protocol definitions file, 39, 40
protocol definitions, editing with
the "vi" editor, 39
protocol definitions, modifying, 2,
36, 39
protocol definitions, reloading, 40
protocol definitions, reset, 40
protocol field values, event editor,
23
Protocol keyword, 46
protocol name, 46
protocol stack, event editor, 23
protocol, bus editor, 20
protocol, choosing, 36
protocol-based triggering, 2
ProtocolIndicator field type, 49

Q

qualifying the sampling clock
signal, 12, 13

R

reloading protocol definitions, 40
repetitive run on errors, stop, 27
repetitive runs and deep trace
memory, 27
reset protocol definitions, 40

resources, triggering, 16
Run button, 25
running a measurement, 25
running measurements, 26

S

sampling clock signal, 12, 13
sampling mode, selecting, 13
setup MPEG tool, 27
short mnemonic, 27
start of packet, 12, 14, 33
start of packet/cell, bus editor, 20
start packet length, 46
start packet value, 46
state sampling mode, selecting, 13
stop repetitive run on errors, 27
STUFF_UNTIL loop block, 53
Symbols field type, 49
synchronous sampling mode,
selecting, 13
syntax error, 29
syntax, protocol definition, 44
SyntaxCheck field type, 50
system administration tools, 8, 9,
10

T

target system, probing, 12
total packet length, 46
trace memory depth and repetitive
runs, 27
Transport Stream protocol, 2
triggering resources, logic analyzer
internal, 16
triggering, packet, 2
triggers, MPEG packet, 16

U

unknown states, 33
UNTIL loop block, 52

V

valid data, 12, 14

vi editor, 39, 40

view packet bits, event editor, 23

VisiTrigger capabilities, 2

W

workspace, connecting the MPEG
tool, 26

