Help Volume

Toolsets: RapidIO (Agilent Technologies N4215A)

Using the RapidIO Toolset

RAPID IO TOOL

The RapidIO Toolset adds protocol-based display and error detection capabilities to the logic analyzer. This toolset helps you look at data traveling across 8/16 LP-LVDS buses.

The RapidIO Toolset looks at data captured synchronously on the transmit or receive data path (which includes a clock signal, an 8- or 16-bit data bus, and a packet framing control signal), and it decodes the packet information.

The RapidIO Toolset understands the RapidIO protocol. You can modify the protocol definitions if necessary.

NOTE: Because the 8/16 LP-LVDS bus is a high-speed interface that uses LVDS (Low Voltage Differential Signaling), the RapidIO Toolset is used with the Agilent Technologies 16760A logic analyzer and the E5379A differential probe.

“Installation and Licensing” on page 8

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

“Probing the Device Under Test” on page 12

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

“Capturing the Data” on page 15

- “To trigger using 800 Mb/s state mode trigger functions” on page 15
- “To trigger with another instrument” on page 19
- “To run the measurement” on page 21

"Displaying the Data" on page 22

- "To connect the RapidIO Tool (in the workspace)" on page 22
- "To set up the RapidIO Tool" on page 23
- "To select the output data columns" on page 25
- "To filter data rows" on page 27

"Modifying Protocol Definitions" on page 30

- "To access the protocol definitions" on page 30
- "To modify a protocol definition" on page 32
- "To reload the protocol definitions" on page 33
- "To reset the protocol definitions" on page 33

Reference

- "Protocol Definition Syntax" on page 36

See Also

The *Agilent Technologies E5378A, E5379A, and E5380A Probes for the 16760A Logic Analyzer User's Guide* for information on designing 100-pin Samtec connectors into the device under test.

Agilent Technologies 16760A Logic Analyzer Online Help (see the *Agilent Technologies 16760A 1500 Mb/s State/800 MHz Timing Logic Analyzer* help volume)

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

Glossary (see page 45)

Contents

Using the RapidIO Toolset

1 Installation Guide

2 Task Guide

3 Reference

1

Installation Guide

Installation and Licensing

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

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

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

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

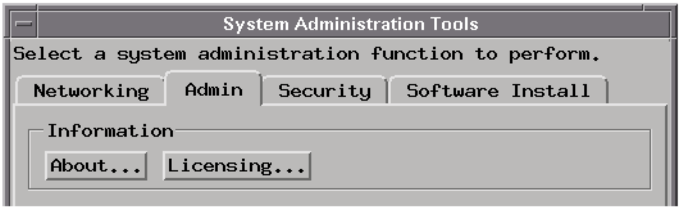
See Also

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 RapidIO 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.

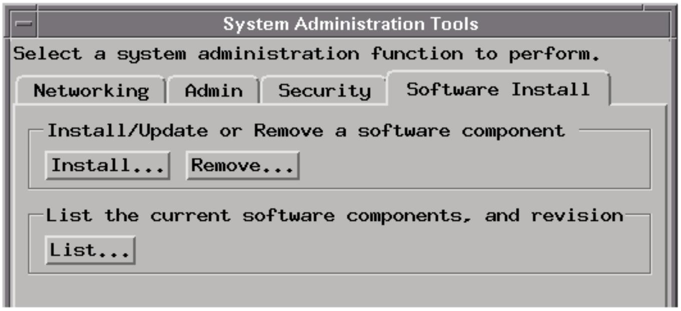
3. In the Tool Sets tab of the Licensing Dialog, look for the "N4215A - RapidIO Analysis" product.

- If it is not listed, go to “To install the RapidIO Toolset” on page 9.
- If it is listed, go to “To license the RapidIO Toolset” on page 10.

To install the RapidIO 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 RapidIO 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.

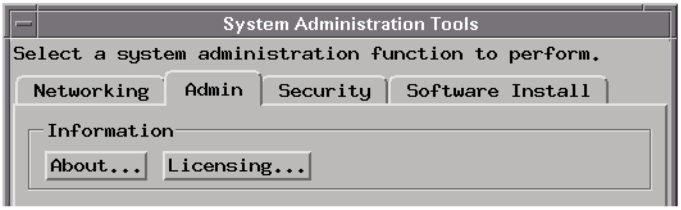
10. Go to “To license the RapidIO Toolset” on page 10.

See Also

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

To license the RapidIO Toolset

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



System Administration Tools

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 "N4215A - RapidIO 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 RapidIO Toolset.

2

Task Guide

Probing the Device Under Test

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

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

Step 1. Connect the logic analyzer probes

To use the RapidIO Toolset, you need to connect logic analyzer probe channels to the following signals in the transmit or receive data path of an 8/16 LP-LVDS bus:

- The data path's 8- or 16-bit data bus.
- The data path's 1-bit packet framing control signal.

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

- The data path's clock signal (which identifies when the data bus is valid and should be sampled by the logic analyzer).

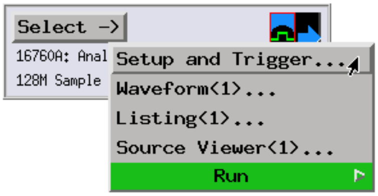
Because the 8/16 LP-LVDS bus is a high-speed interface that uses LVDS (Low Voltage Differential Signaling), you need to use the Agilent Technologies 16760A logic analyzer and the E5379A differential probe.

Next Step (see page 12)

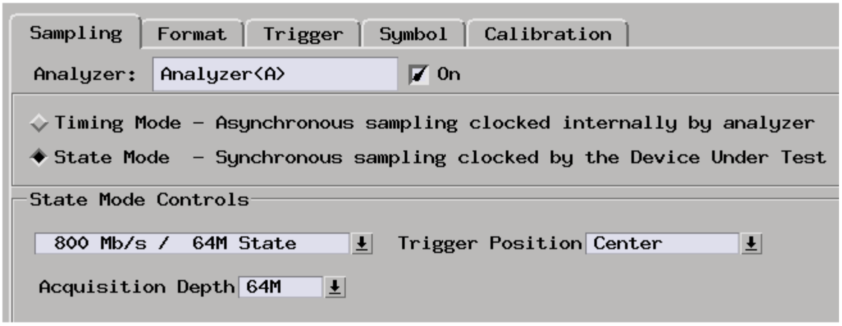
Step 2. Select the state sampling mode

When using the RapidIO 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

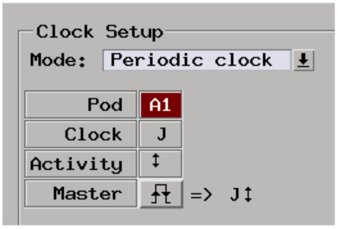
◆ State Mode – Synchronous sampling clocked by the Device Under Test

State Mode Controls

800 Mb/s / 64M State   Trigger Position Center

Acquisition Depth 64M

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



Clock Setup

Mode: Periodic clock

Pod	A1
Clock	J
Activity	↕
Master	=> J↕

Next Step (see page 13)

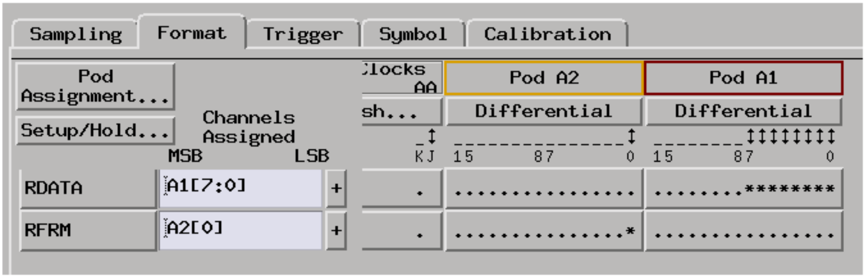
Step 3. Label the logic analyzer channels

The RapidIO Tool requires you to set up one label for the channels probing the 8- or 16-bit data bus and a second label for the channel probing the 1-bit packet framing control signal.

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 are two labels.

2. For each label, specify which logic analyzer pod channels are probing the signal(s) described by the label.



Channels Assigned MSB LSB		Clocks AA KJ	Pod A2 15 87 0	Pod A1 15 87 0
RDATA	A1[7:0]	.	....	........********
RFRM	A2[0]	.	...............*	....

See Also “Capturing the Data” on page 15

Capturing the Data

You can capture data using the 16760A logic analyzer's 800 Mb/s state mode trigger functions, or you can cross-trigger with another instrument and time-correlate the captured data.

Once triggering is set up, you run the measurement to capture data.

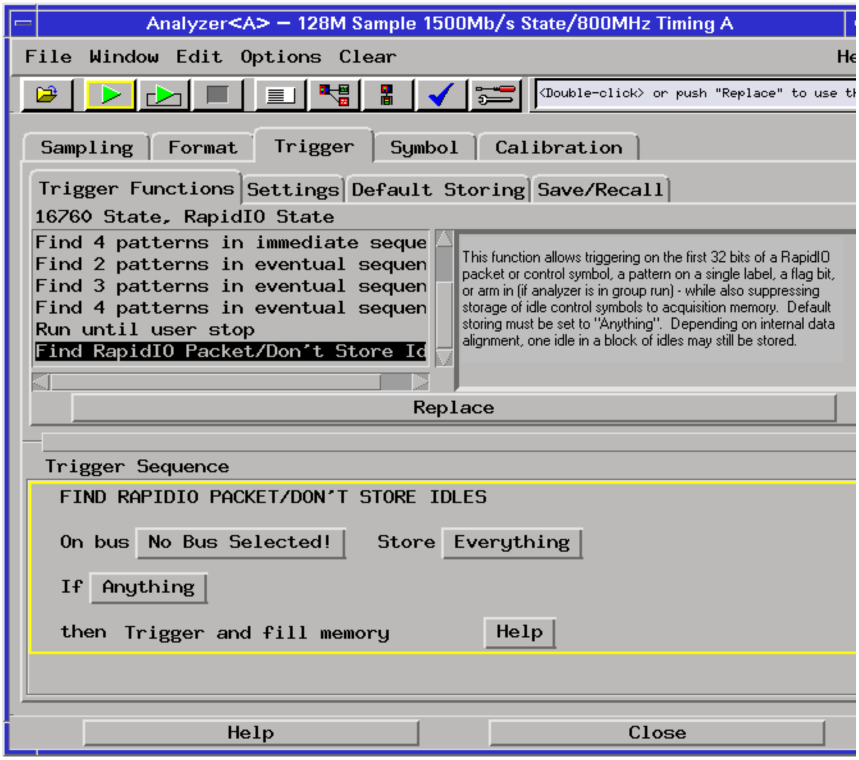
- “To trigger using 800 Mb/s state mode trigger functions” on page 15
- “To trigger with another instrument” on page 19
- “To run the measurement” on page 21

To trigger using 800 Mb/s state mode trigger functions

The 16760A logic analyzer's 800 Mb/s state mode trigger functions let you set up triggers on RapidIO packets or on up to 4 patterns that occur in consecutively captured states or up to 4 patterns that eventually occur in the captured states.

Using the "Find RapidIO Packet" trigger function

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



<Double-click> or push "Replace" to use th

16760 State, RapidIO State

Find 4 patterns in immediate seque
Find 2 patterns in eventual sequen
Find 3 patterns in eventual sequen
Find 4 patterns in eventual sequen
Run until user stop
Find RapidIO Packet/Don't Store Id

This function allows triggering on the first 32 bits of a RapidIO packet or control symbol, a pattern on a single label, a flag bit, or arm in (if analyzer is in group run) - while also suppressing storage of idle control symbols to acquisition memory. Default storing must be set to "Anything". Depending on internal data alignment, one idle in a block of idles may still be stored.

Trigger Sequence

FIND RAPIDIO PACKET/DON'T STORE IDLES

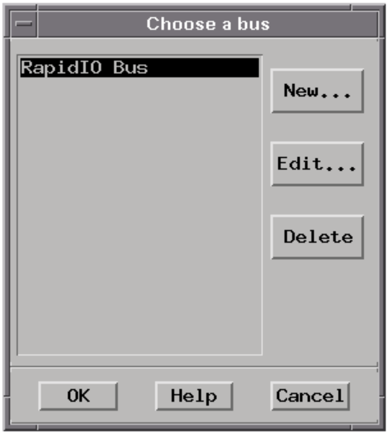
On bus No Bus Selected! Store Everything

If Anything

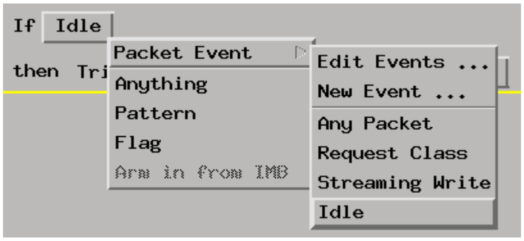
then Trigger and fill memory Help

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

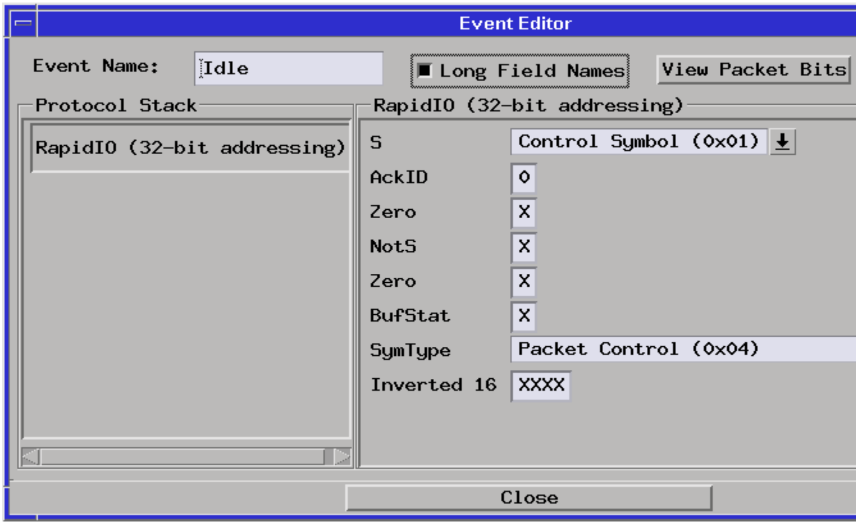
4. In the Choose a Bus dialog, select the bus definition you want to use and select the OK button.



5. Specify the packet event to find.



6. To trigger only when fields of the packet have specific values, use the Event Editor.



Event Name: Idle

RapidIO (32-bit addressing)

S	Control Symbol (0x01)
AckID	0
Zero	X
NotS	X
Zero	X
BufStat	X
SymType	Packet Control (0x04)
Inverted 16	XXXX

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.

Using the "Find patterns" trigger functions

With these trigger functions, you can capture control symbols and other basic information in the 8/16 LP-LVDS data flow. Control symbols are 32-bits. With an 8-bit data bus, you need to capture 4 consecutive byte patterns. With a 16-bit data bus, you need to capture 2 consecutive byte patterns.

For example, to trigger on a packet-not-accepted acknowledgement control symbol when using an 8-bit data bus:

1. In the Trigger tab of the logic analyzer's setup window, select the Trigger Functions tab.
2. Select the "Find 4 patterns in immediate sequence" trigger function and replace the current trigger sequence level.
3. For each pattern in the sequence:

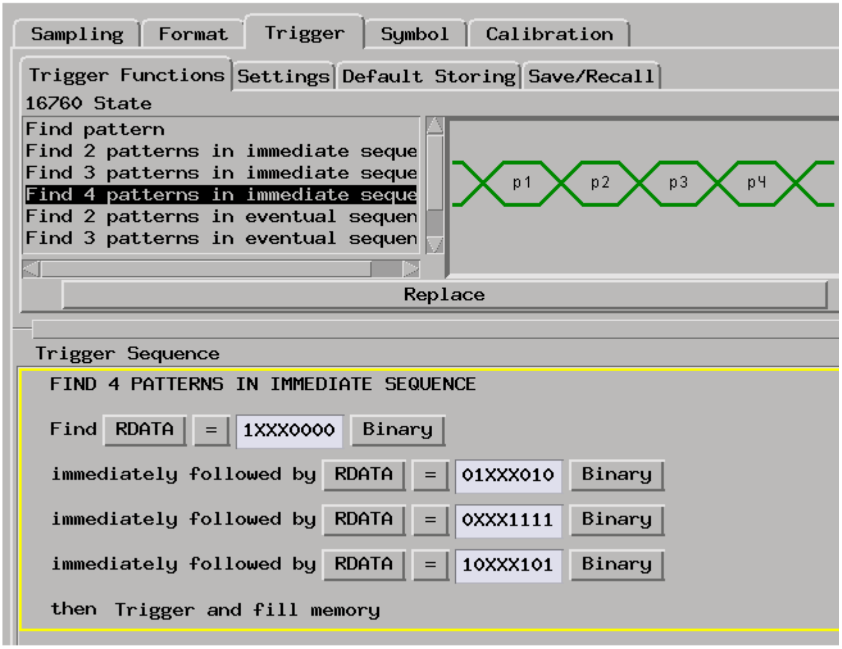
a. Select the data bus label.

b. Select the Binary number base.

c. Enter the control symbol format bit values.

To trigger on any packet-not-accepted control symbol, leave the "packet_ackID", "cause", and their complement bits as "don't cares".

To trigger on a specific packet-not-accepted control symbol, enter the "packet_ackID" and "cause" bit values.



16760 State

Find pattern
Find 2 patterns in immediate seque
Find 3 patterns in immediate seque
Find 4 patterns in immediate seque
Find 2 patterns in eventual sequen
Find 3 patterns in eventual sequen

Trigger Sequence

FIND 4 PATTERNS IN IMMEDIATE SEQUENCE

Find RDATA = 1XXX0000 Binary

immediately followed by RDATA = 01XXX010 Binary

immediately followed by RDATA = 0XXX1111 Binary

immediately followed by RDATA = 10XXX101 Binary

then Trigger and fill memory

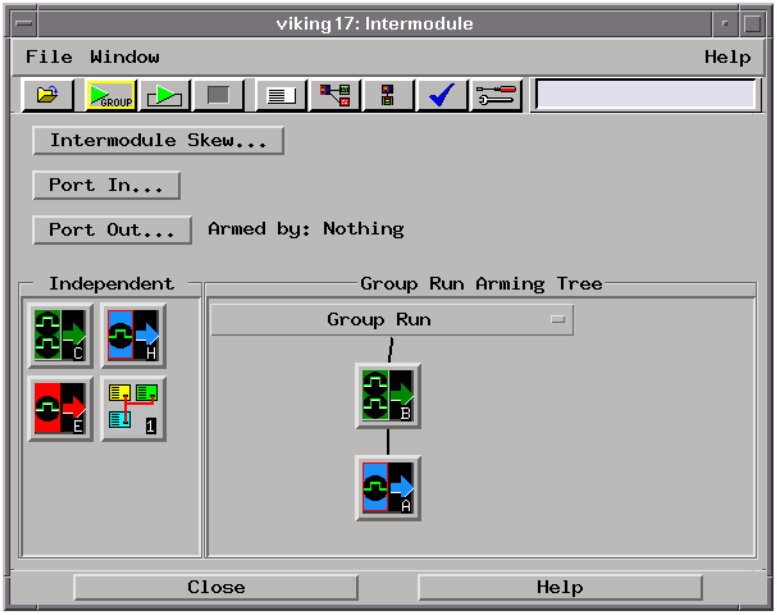
See Also "To run the measurement" on page 21

"Displaying the Data" on page 22

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:

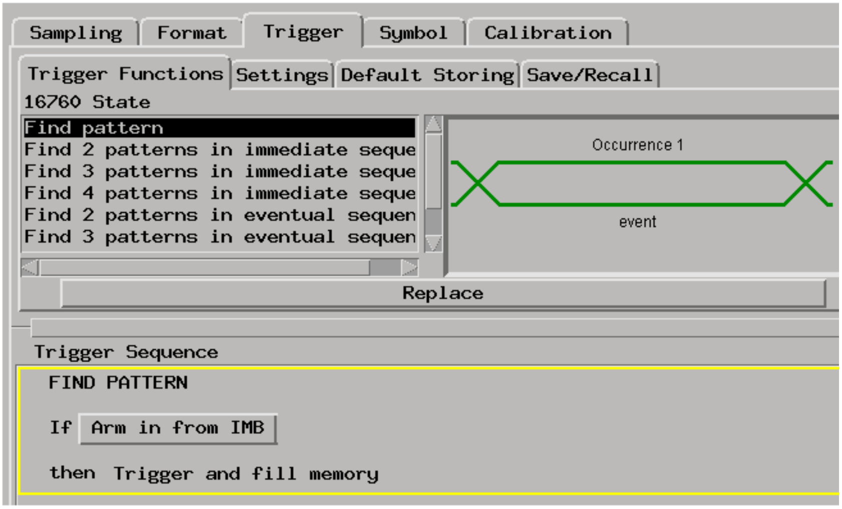
1. In the Intermodule window, set up the logic analyzer to be armed by the other instrument.



Armed by: Nothing

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

3. In the existing trigger function, select the label button; then, choose *Replace Event* and *Arm in from IMB*.



Sampling | Format | Trigger | Symbol | Calibration

Trigger Functions | Settings | Default Storing | Save/Recall

16760 State

```
Find pattern
Find 2 patterns in immediate seque
Find 3 patterns in immediate seque
Find 4 patterns in immediate seque
Find 2 patterns in eventual sequen
Find 3 patterns in eventual sequen
```

Replace

Trigger Sequence

```
FIND PATTERN

If  Arm in from IMB

then  Trigger and fill memory
```

See Also

"To run the measurement" on page 21

"Displaying the Data" on page 22

To run the measurement

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

See Also

"Displaying the Data" on page 22

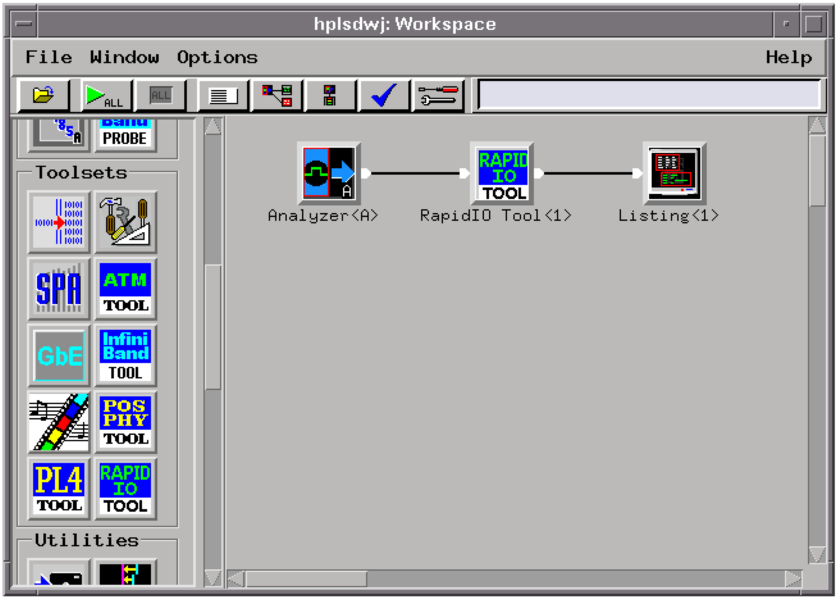
Displaying the Data

This section describes the options available when displaying data captured on the 8/16 LP-LVDS data path.

- "To connect the RapidIO Tool (in the workspace)" on page 22
- "To set up the RapidIO Tool" on page 23
- "To select the output data columns" on page 25
- "To filter data rows" on page 27

To connect the RapidIO Tool (in the workspace)

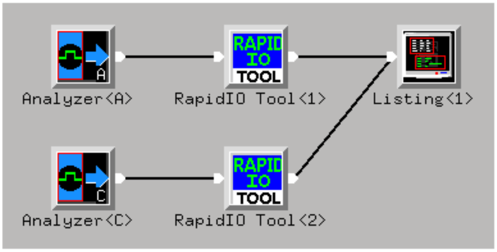
1. Drag the RapidIO Tool icon, and drop it between the logic analyzer instrument icon and the Listing display tool icon.



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



See Also

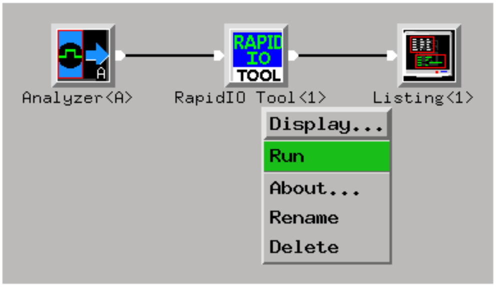
"To set up the RapidIO Tool" on page 23

"To select the output data columns" on page 25

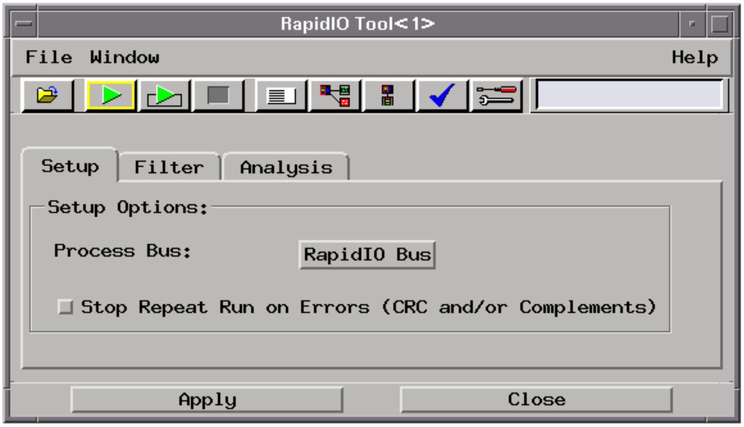
"To filter data rows" on page 27

To set up the RapidIO Tool

1. Display the RapidIO Tool.



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



Setup Options:

Process Bus: RapidIO Bus

□ Stop Repeat Run on Errors (CRC and/or Complements)

Process Bus Selects the defined bus to decode.

Stop Repeat Run on Errors (CRC and/or Complements) Specifies whether CRC or complement errors will stop repetitive run measurements.

NOTE: When you have selected the "stop repetitive run on errors" option, all of the captured data in an individual run must be decoded before the next run can occur. If the acquisition memory depth is large, decoding the entire trace can take a long time, and uncaptured errors can occur during the decode time. Therefore, we recommend that you limit the acquisition memory depth to less than 1M states when using this option and repetitive runs.

1. Select Apply to change the setup.
2. Select Close to change the setup and close the RapidIO Tool.

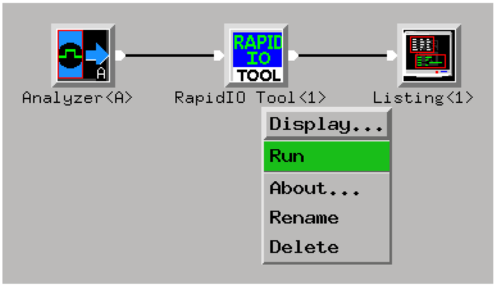
See Also

"To select the output data columns" on page 25

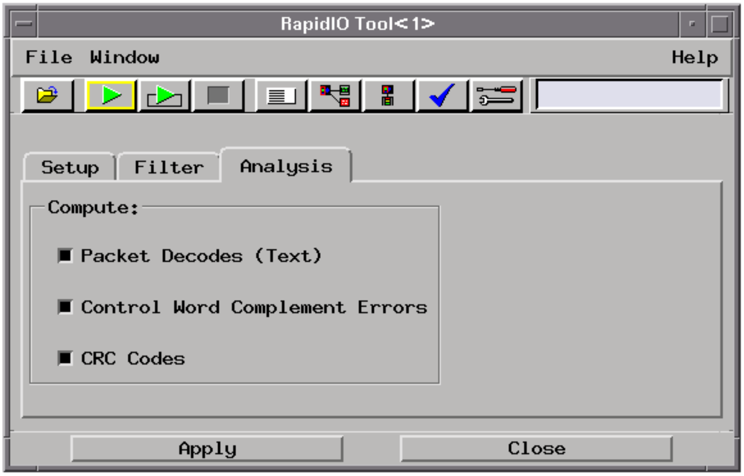
"To filter data rows" on page 27

To select the output data columns

1. Display the RapidIO Tool.



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



Compute:

■ Packet Decodes (Text)

■ Control Word Complement Errors

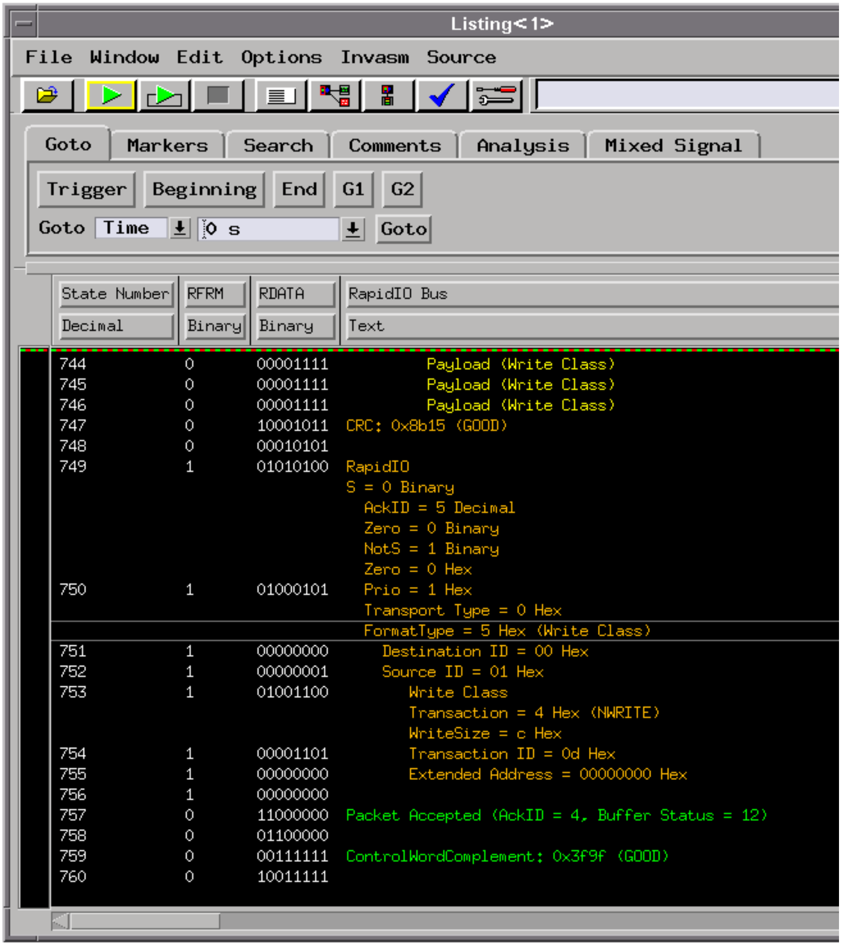
■ CRC Codes

Packet Decodes This text column contains a description of the decoded data including protocol header fields, protocol layers, payload, CRC, and control words.

Control Word Complement Errors The 1-bit "BadCntl" column contains "1" whenever control word complement errors are discovered; otherwise, it contains "0".

CRC Codes The 1-bit "BadCRC" column contains "1" whenever a packet has a checksum that does not match the data; otherwise, it contains "0".

1. Select Apply to output the selected data columns.



```
State Number  RFRM    RDATA     RapidIO Bus
Decimal       Binary  Binary    Text
744           0       00001111           Payload (Write Class)
745           0       00001111           Payload (Write Class)
746           0       00001111           Payload (Write Class)
747           0       10001011  CRC: 0x8b15 (GOOD)
748           0       00010101
749           1       01010100  RapidIO
                                S = 0 Binary
                                  AckID = 5 Decimal
                                  Zero = 0 Binary
                                  NotS = 1 Binary
                                  Zero = 0 Hex
750           1       01000101    Prio = 1 Hex
                                  Transport Type = 0 Hex
                                  FormatType = 5 Hex (Write Class)
751           1       00000000      Destination ID = 00 Hex
752           1       00000001      Source ID = 01 Hex
753           1       01001100         Write Class
                                       Transaction = 4 Hex (NWRITE)
                                       WriteSize = c Hex
754           1       00001101         Transaction ID = 0d Hex
755           1       00000000         Extended Address = 00000000 Hex
756           1       00000000
757           0       11000000  Packet Accepted (AckID = 4, Buffer Status = 12)
758           0       01100000
759           0       00111111  ControlWordComplement: 0x3f9f (GOOD)
760           0       10011111
```

2. Select Close to output the selected columns and close the RapidIO Tool.

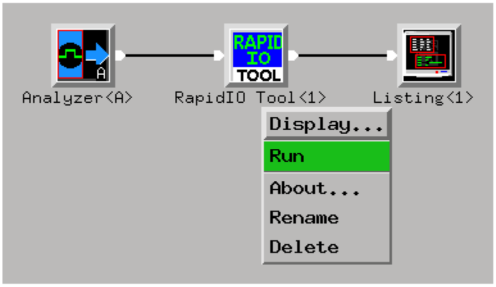
See Also

"To set up the RapidIO Tool" on page 23

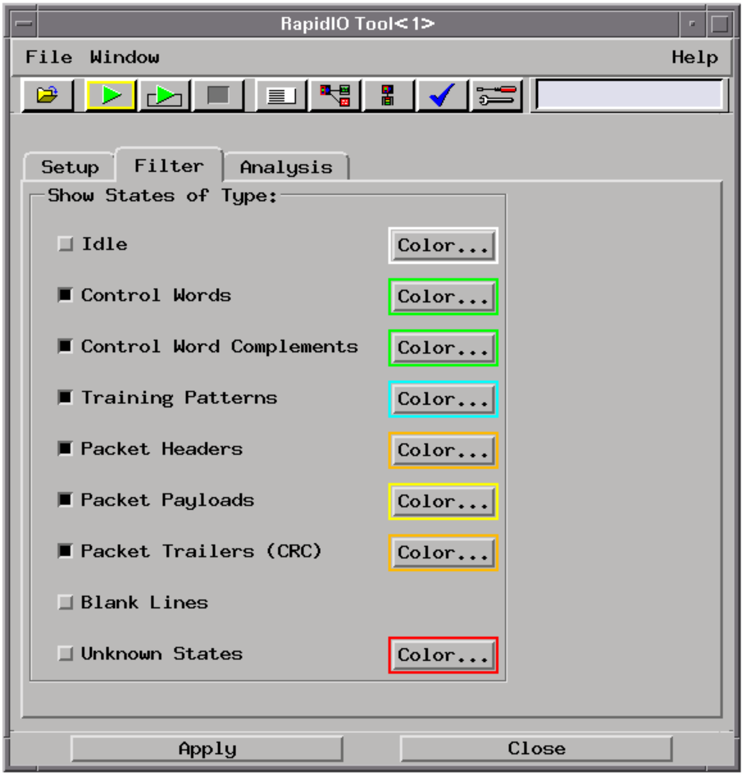
"To filter data rows" on page 27

To filter data rows

1. Display the RapidIO Tool.



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



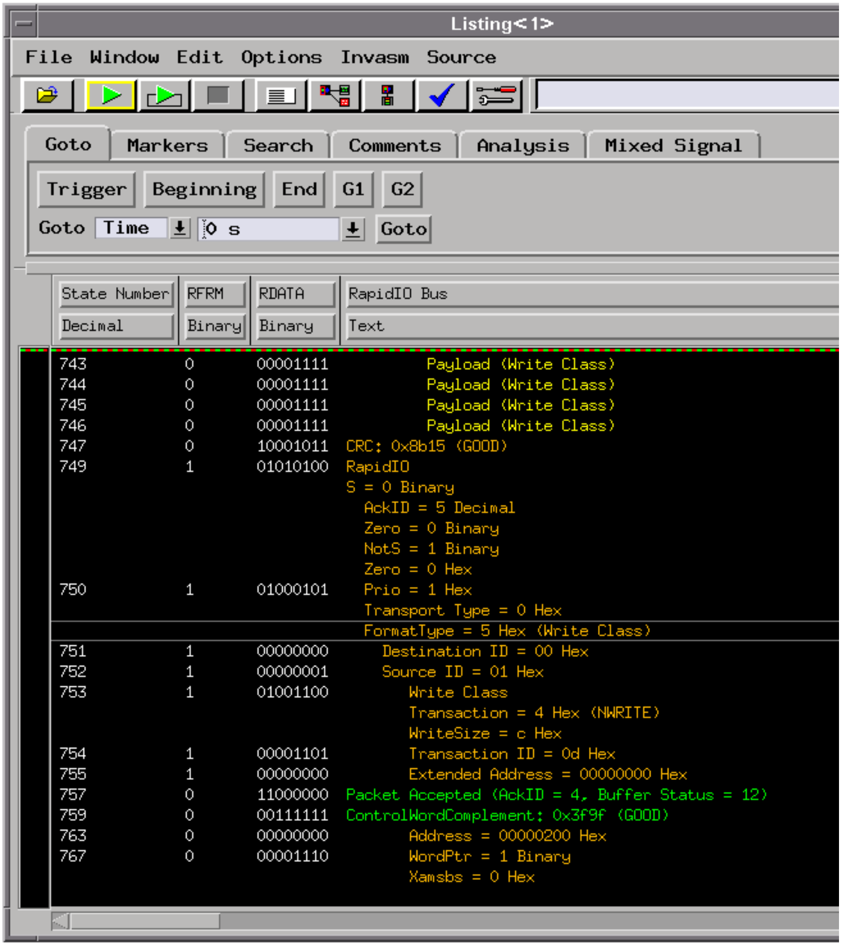
Show States of Type:

- □ Idle — Color...
- ■ Control Words — Color...
- ■ Control Word Complements — Color...
- ■ Training Patterns — Color...
- ■ Packet Headers — Color...
- ■ Packet Payloads — Color...
- ■ Packet Trailers (CRC) — Color...
- □ Blank Lines
- □ Unknown States — Color...

- Idle.
- Control Words.
- Control Word Complements.
- Training Patterns.
- Packet Headers.
- Packet Payloads.
- Packet Trailers (CRC).
- Blank Lines.
- Unknown States.

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

3. Select Apply to output the selected data rows.



```
State Number  RFRM    RDATA     RapidIO Bus
Decimal       Binary  Binary    Text
743           0       00001111           Payload (Write Class)
744           0       00001111           Payload (Write Class)
745           0       00001111           Payload (Write Class)
746           0       00001111           Payload (Write Class)
747           0       10001011  CRC: 0x8b15 (GOOD)
749           1       01010100  RapidIO
                                S = 0 Binary
                                  AckID = 5 Decimal
                                  Zero = 0 Binary
                                  NotS = 1 Binary
                                  Zero = 0 Hex
750           1       01000101    Prio = 1 Hex
                                  Transport Type = 0 Hex
                                  FormatType = 5 Hex (Write Class)
751           1       00000000      Destination ID = 00 Hex
752           1       00000001      Source ID = 01 Hex
753           1       01001100         Write Class
                                       Transaction = 4 Hex (NWRITE)
                                       WriteSize = c Hex
754           1       00001101         Transaction ID = 0d Hex
755           1       00000000         Extended Address = 00000000 Hex
757           0       11000000  Packet Accepted (AckID = 4, Buffer Status = 12)
759           0       00111111  ControlWordComplement: 0x3f9f (GOOD)
763           0       00000000         Address = 00000200 Hex
767           0       00001110         WordPtr = 1 Binary
                                       Xamsbs = 0 Hex
```

4. Select Close to close the RapidIO Tool.

See Also

Filtering Limitation in “To connect the RapidIO Tool (in the workspace)” on page 22

“To set up the RapidIO Tool” on page 23

“To select the output data columns” on page 25

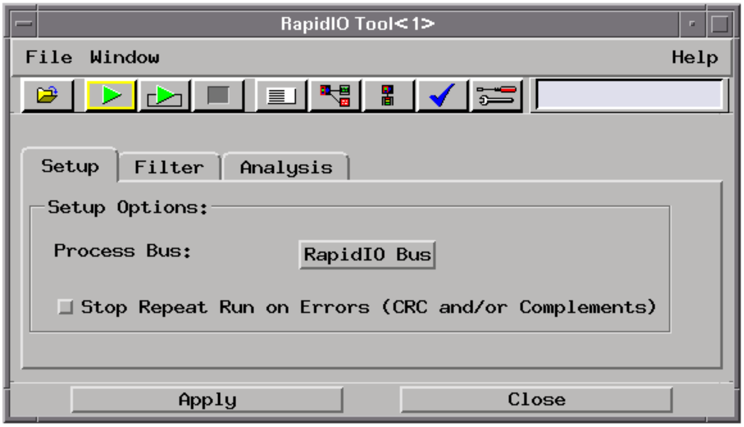
Modifying Protocol Definitions

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

- "To access the protocol definitions" on page 30
- "To modify a protocol definition" on page 32
- "To reload the protocol definitions" on page 33
- "To reset the protocol definitions" on page 33

To access the protocol definitions

- In the RapidIO Tool's Setup tab, select the Process Bus button.

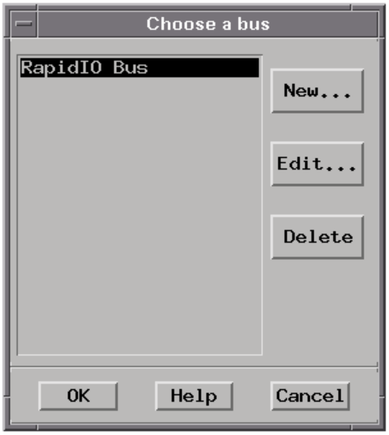


Setup Options:

Process Bus: RapidIO Bus

□ Stop Repeat Run on Errors (CRC and/or Complements)

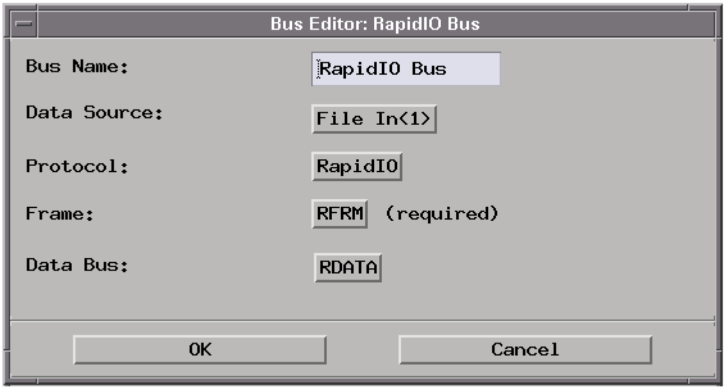
- In the Choose a bus dialog, select the RapidIO Bus; then, select the Edit... button.



Choose a bus

RapidIO Bus

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



Bus Editor: RapidIO Bus

Bus Name: RapidIO Bus

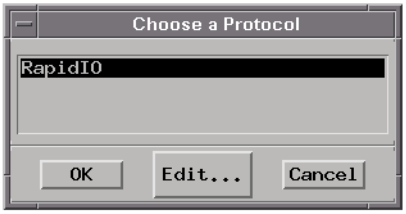
Data Source: File In(1)

Protocol: RapidIO

Frame: RFRM (required)

Data Bus: RDATA

- In the Choose a Protocol dialog, select the RapidIO protocol.



Choose a Protocol

RapidIO

See Also

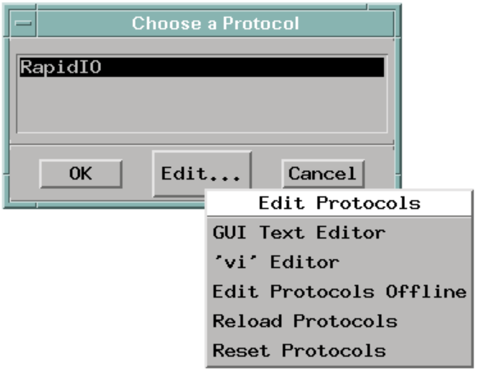
"To modify a protocol definition" on page 32

"To reload the protocol definitions" on page 33

"To reset the protocol definitions" on page 33

To modify a protocol definition

1. In the Choose a Protocol dialog (see "To access the protocol definitions" on page 30), 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 RapidIO Tool that changes have been made (see "To reload the protocol definitions" on page 33).

About the Protocol Definitions File

The protocol definitions file is:

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

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

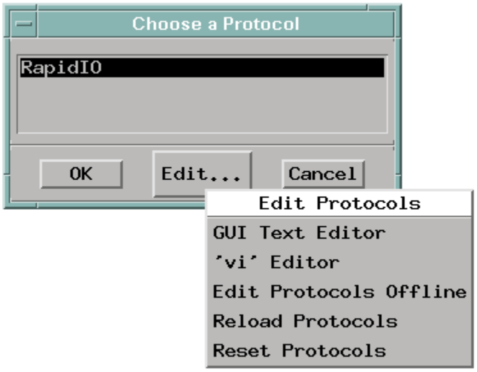
See Also

"Protocol Definition Syntax" on page 36

To reload the protocol definitions

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

1. In the Choose a Protocol dialog (see “To access the protocol definitions” on page 30), select the Edit... button, and choose Reload Protocols.

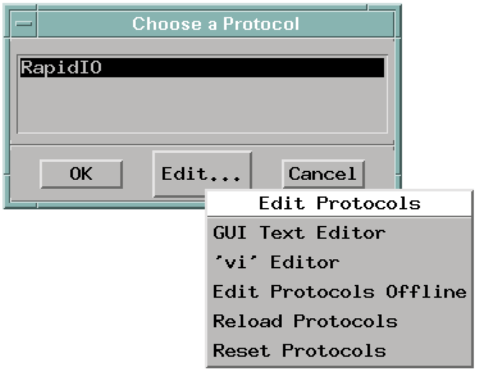


To reset the protocol definitions

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 Choose a Protocol dialog (see “To access the protocol definitions” on page 30), select the Edit... button, and choose Reset Protocols.


Choose a Protocol
RapidIO
OK
Edit...
Cancel
Edit Protocols
GUI Text Editor
'vi' Editor
Edit Protocols Offline
Reload Protocols
Reset Protocols

3

Reference

Protocol Definition Syntax

This section describes the protocol definition syntax.

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

- “Protocol Family” on page 38
- “Protocol Block” on page 39
- “Physical Layer Definition” on page 39
- “Header Block” on page 39
- “Field Definition” on page 40
    - “Data” on page 40
    - “Symbols” on page 41
    - “ProtocolIndicator” on page 41
- “Numeric Values” on page 42
- “IF Contitional Block” on page 42
- “Formulas” on page 43

Example

Here is the definition for the RapidIO protocol:

```
ProtocolFamily "RapidIO"

Protocol
{
  Name "RapidIO"
  PhyLayer 1

  Header
  {
    S "S" 1 Binary Data

    if ( S, 1, ==, BLOCKNAME "Control Symbol" )
    {
      if ( STY, 7, != )
      {
        if ( STY, 2, <= )
        {
          AID "AckID" 3 Decimal Data
        }
        if ( STY, 4, == )
        {
          SBT "SubType" 3 Decimal Symbols
          {
```

```
      "Idle" 0
      "Stomp" 1
      "EOP" 2
      "Restart-From-Retry" 3
      "Throttle" 4
      "Time-of-Day Sync" 5
    }
  }
  if ( STY, 5, == )
  {
    CMD "Command" 3 Decimal Symbols
    {
      "Send Training" 0
      "Reset" 3
      "Input-status" 4
    }
  }
  if ( STY, 6, == )
  {
    AIDST "Expected AckID" 3 Decimal Data
  }
  Z "Zero" 1 Binary Data
  NS "NotS" 1 Binary Data
  Z3 "Zero" 3 Hex Data
  if (STY, 6, ==)
  {
    LS "LinkStat" 4 Hex Symbols
    {
      "Uninitialized" 0
      "Initializing" 1
      "Error" 2
      "Stopped" 3
      "Retry-stopped" 4
      "Error-stopped" 5
      "OK, AckID 0" #h8
      "OK, AckID 1" #h9
      "OK, AckID 2" #ha
      "OK, AckID 3" #hb
      "OK, AckID 4" #hc
      "OK, AckID 5" #hd
      "OK, AckID 6" #he
      "OK, AckID 7" #hf
    }
  }

  if (STY, 0, ==, STY, 5, ==, || )
  {
    BS "BufStat" 4 Hex Data
  }
  if (STY, 2, ==)
  {
    ONE "One" 1 Binary Data
    CAU "Cause" 3 Hex Symbols
    {
      "Internal error" 0
      "Unexpected AckID" 1
      "Error on control symbol" 2
      "Input port stopped" 3
      "Bad CRC on packet" 4
      "S-bit parity error" 5
      "General error" 7
    }
  }

  if (STY, 4, ==)
  {
    CO "Contents" 4 Hex Data
  }
}
if (STY, 7, ==)
{
  TR "Training" 12 Binary Data
```

```
      }
      STY "SymType" 3 Hex Symbols
      {
        "Packet-Accepted" 0
        "Packet-Retry" 1
        "Packet-Not-Accepted" 2
        "Reserved" 3
        "Packet Control" 4
        "Link Request" 5
        "Link Response" 6
        "Start of Link Training" 7
      }
      if (STY, 7, !=)
      {
        I16 "Inverted 16" 16 Hex Data
      }
      if (STY, 7, ==)
      {
        TR1 "Training (1's)" 16 Hex Data
      }
    }

    if ( S, 0, ==, BLOCKNAME "Packet" )
    {
      AID "AckID" 3 Decimal Data
      Z   "Zero"  1 Binary Data
      NS  "NotS"  1 Binary Data
      Z2  "Zero"  2 Hex Data
      PR  "Prio"  2 Hex Data
      TT  "Transport Type" 2 Hex Data
      FT  "FormatType" 4 Hex ProtocolIndicator
      {
        "Request Class"        2
        "Write Class"          5
        "Streaming Write"      6
        "Maintenance"          8
        "Doorbell"            10
        "Message"             11
        "Response"            13
      }
      DST "Destination ID" 8 Hex Data
      SRC "Source ID" 8 Hex Data
    }
  }
  Summary STY FT
}
```

See Also

"Displaying the Data" on page 22

Protocol Family

Syntax

```
ProtocolFamily "name"
```

A protocol family statement associates following protocol definitions with a particular family of definitions. This allows the user interface to filter and group related protocols.

Example

```
ProtocolFamily "RapidIO"
```

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 "RapidIO"

}
```

See Also

"Physical Layer Definition" on page 39

"Header Block" on page 39

Physical Layer Definition

Syntax

```
PhyLayer 1
```

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

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 a RapidIO data bus.

Example

```
PhyLayer  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.

The choices for field type are:

- "Data" on page 40.
- "Symbols" on page 41.
- "ProtocolIndicator" on page 41.

Example

```
DST  "Destination ID"  8  Hex  Data
```

This is one of the fields in a packet header, the Destination ID. It is 8 bits long. It should be displayed as a hexadecimal value (for example, 00 Hex), 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 "SubType" field from the control symbol header definition:

```
SBT  "SubType"  3  Decimal  Symbols
{
  "Idle"                0
  "Stomp"               1
  "EOP"                 2
  "Restart-From-Retry"  3
  "Throttle"            4
  "Time-of-Day Sync"    5
}
```

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 "FormatType" field of a packet header:

```
FT  "FormatType"  4  Hex  ProtocolIndicator
{
  "Request Class"        2
  "Write Class"          5
  "Streaming Write"      6
  "Maintenance"          8
  "Doorbell"            10
  "Message"             11
  "Response"            13
}
```

A value of 11 in the FormatType field indicates Message, 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'.

IF Contitional 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 (STY, 7, !=)
{
  I16  "Inverted 16"  16  Hex  Data
}
if (STY, 7, ==)
{
  TR1  "Training (1's)"  16  Hex  Data
}
```

See Also

“Formulas” on page 43

“Field Definition” on page 40

Formulas

Syntax

```
[ 'Short Mnemonic' | 'value' | operator ] , ... , [ BLOCKNAME "name" ]
```

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

Formulas can be used in IF blocks, like:

```
IF ('Formula')
```

The BLOCKNAME option names the field definitions that follow when the formula result is TRUE.

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 (STY, 7, !=)
{
  I16  "Inverted 16"  16  Hex  Data
}
if (STY, 7, ==)
{
  TR1  "Training (1's)"  16  Hex  Data
}
```

See Also

“IF Contitional Block” on page 42

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 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*.

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

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

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

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

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

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

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

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

filter term A variable that you

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

Format The selections under the logic analyzer *Format* tab tell the logic analyzer what data you want to collect, such as which channels represent buses (labels) and what logic threshold your signals use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

pod See *pod pair*

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

preprocessor See *analysis probe*.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

trace See *acquisition*.

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

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

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

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

zooming In the oscilloscope or timing analyzer, to expand and contract the waveform along the time base by varying the value in the s/Div field. This action allows you to select specific portions of a particular waveform in acquisition memory that will be displayed on the screen. You can view any portion of the waveform record in acquisition memory.

Index

Publication Number: 5988-9065EN
January 1, 2003

Agilent Technologies