Agilent Logic Analyzer Help 16

WHAT'S NEW 17

Version A.01.10 What's New 18

GETTING STARTED 19

Tutorial........ 20

Getting to know your logic analyzer........ 20

Logic analysis basics 20

Measurement overview........ 20

Time saving tasks 20

Measurement overview........ 22

Turning on the logic analyzer........ 22

Connecting to the target system 23

Credit card demo board 24

Setting up bus/signal names........ 25

Delete bus/signal names 25

Add new bus/signal name 25

Map signals into the analyzer........ 26

Setting the acquisition mode 27

Setting up a simple trigger 28

Open the tutorial configuration file 29

Load the configuration file 29

View the data 29

Using markers 31

To create a marker........ 31

To place a marker in data 31

Go To a marker in data 31

Zooming in on the data........ 32

Expand a bus........ 32

Change the scale 32

Time saving tasks 33

Loading and saving configuration files........ 33

Saving and recalling trigger setups........ 34

To recall a trigger setup 34

Quick marker measurements 35

Searching data 36

Toolbars, tooltips and mouse shortcuts 37

Toolbars........ 37

Mouse shortcuts........ 37

Tooltips........ 37

Measurement Examples 38

Making a timing analyzer measurement........ 39

Bus and signal setup 39

Acquisition mode setup 39
Trigger setup 39
Run the measurement 39
Making a state analyzer measurement 40
Bus and signal setup 40
Acquisition mode setup 40
Trigger setup 40
Run the measurement 40
To trigger on one of multiple edges or glitches 41

PROBING 42

General Purpose Probing 42
Adaptor to board connectors 43
Analysis Probes 44

SETTING UP THE LOGIC ANALYZER 45

To set thresholds 46
Assigning Bus/Signal Names to Logic Analyzer Probes 47
Buses/Signals Setup 47
Read only options 47
To add a new bus or signal 49
To delete a bus or signal 50
To delete an individual bus or signal 50
To delete all buses and signals 50
To rename a bus or signal 52
To assign channels 53
To set numeric base 54
To set polarity 56
To set setup/hold 57
To add user comments 59
To add a folder 60
To alias a bus/signal name 61
To sort bus/signal names 62
To edit symbols for a bus/signal 63
To add a symbol 63
To edit a symbol 63
To delete a symbol 63
To save symbols 64
To display symbols 65
Example 66
Selecting the Sampling Mode 67
Sampling Setup 67

State - Synchronous Sampling mode 67
Timing - Asynchronous Sampling mode 68
To set the acquisition mode 69
To select the timing analyzer 69
To select the state analyzer 69
To set acquisition depth 70
To set the trigger position 71
To select the state clock type 72
Master 72
Master/Slave 72
Demultiplex 73
To set the state clock qualifier 74
To set up advanced clocking 75
To set the sampling period 76
To set sampling options 77
Transitional timing 78
More on storing transitions 78
Minimum transitions stored 78
Maximum transitions stored 78
Transitional timing considerations 78
Data storage 78
Sequence level branching 79
Global counters 79
Storing Time Tags 79
Increasing Duration of Storage 79
Invalid Data 79
Trigger Position 79
CAPTURING DATA FROM THE DEVICE UNDER TEST 80
TRIGGERING 81
Set quick trigger with rectangle 82
To draw the rectangle 82
Specifying Simple Triggers 84
Simple Trigger 84
To set bus pattern triggers 86
Operators 86
To set signal trigger options 87
Options 87
To trigger on a glitch 87
Specifying Advanced Triggers 88
Advanced Trigger Dialog 88
To build a trigger sequence 89

Trigger steps 89
To set store qualification.......... 91
Default storage 91
Sequence step storage.......... 91
Storage interaction.......... 92
How to read event and action statements 94
To insert events and actions 95
To negate a function statement 97
To modify trigger step display 98
ANDing and ORing Event statements.......... 100
To configure a counter 101
To insert a counter event 101
To insert a counter action.......... 101
To configure a timer.......... 103
Model 103
Timing Acquisition Mode.......... 103
State Acquisition Mode.......... 103
To insert a timer check event 103
To insert a timer start action 104
External Triggering.......... 106
Trigger signal characteristics 106
To trigger other instruments - Trigger Out 106
To trigger analyzer from another instrument - Trigger In 106
To store a trigger 108
Most recently used trigger list 108
Stored favorite triggers list 108
To start/stop measurements 109
To run the analyzer in single mode 109
To run the analyzer in repetitive run mode 109
To stop the analyzer 109
Saving Captured Data (and Logic Analyzer Setups) 110
To save a configuration file 111
To export data to CSV format files 113
ANALYZING THE CAPTURED DATA 115
Loading Saved Data and Setups.......... 116
To open a configuration file.......... 117
To recall a recently used configuration file 119
To import 167xx fast binary data 120
Offline Analysis 122
General offline analysis considerations 122

Analyzing 16700-Series Logic Analyzer Data 123
More 16700 considerations 123
16700 Pod and Bit Association in Offline Analysis 125
Offline Analysis on 1680/90-Series Logic Analyzers 126
Offline Analysis on Personal Computers 127
Minimum PC Requirements 128
Offline File Formats 129
.ala and .xml Formats 129
Fast Binary Data Format 129
CSV Format 129
Analyzing Waveform Data 130
Waveform Display Window 130
Bus/Signal configuration 130
To reposition bus/signal names 131
Simple Trigger 131
Marker display bar 131
Waveform display area 131
To set waveform window properties 133
To set window properties 133
To set row properties 133
To set background color 133
To set font size 133
To set an overlay color 134
To show activity indicators 134
To show tool tip - values 134
To set markers - snap to edge 134
To set markers - place on edge 135
To select the Bus/Signal 135
To set data color 135
To set row height 135
To set numeric base 136
To show data values 136
To show soft glitch 136
To set bus options 136
Change delay 138
Change scale (time/division) 139
Change the scale 139
Draw a rectangle in the data 139
To overlay bus/signals 141
To group signals into a bus 142
To add or delete display windows 143
To add a new display windows 143
To delete display windows 143
Turning window tabs on/off 144

Tool Tips 145
Analyzing Listing Data 146
Listing Display Window 146
Column configuration 146
To reposition bus/signal names 147
Simple Trigger 147
Marker display bar 147
Listing display area 147
Column configuration 149
Insert Column Before 149
Insert Column After 149
Delete Column 149
Assign Channels 149
Rename 149
Symbols 149
Display 149
Properties 149
To set listing window properties 151
To set window properties 151
To set column properties 151
To set background color 151
To set font size 151
Column display options 151
To show center rectangle 152
To select the Bus/Signal name 152
To set data color 153
To set column width 153
To set column alignment 153
To set numeric base 153
To set marker relative 153
Comparing Captured Data to Reference Data 155
To copy data to the reference buffer 156
To find differences in the compared data 157
To compare only a range of samples 158
To offset the reference data 159
To run until a number of compare differences 160
To set the Compare display options 161
Searching the Captured Data 162
Searching 162
To search for a value 163
To perform complex pattern searches 164
To Go To a specific position in the acquisition 166
Favorite Find Pattern 167

To store a favorite pattern.. 167
To recall a favorite pattern 167
To delete a favorite pattern 168
Find Options 169
To use the Place function 169
To use the Place new marker on each occurrence function 169
Marking, and Measuring Between, Data Points 170
Working with Markers 170

MAKING MARKER MEASUREMENTS 171

To create new markers 172
Using abbreviated names 172
To place markers in data 173
Place marker at center screen 173
Place marker at mouse cursor 173
Go To Markers 174
To center the display about a marker 175
To delete a marker 176
To rename a marker 177
To send a marker to the back 178
To set marker properties 179
To rename the marker 179
Using abbreviated names 179
To change the marker color 179
To hide the marker 179
To lock marker in viewer 179
To change the marker type 180
To drag and drop markers in data 181
Reading off-screen markers 182
To create a new interval measurement 183
To create a new value at measurement 184
Displaying Names (Symbols) for Bus/Signal Values 185
Printing Captured Data 185
To print data and screens 185
To print data 185
To copy and print text 185
To copy and print a screen 185
To install a printer 187
To connect a LAN 188

SOLVING PROBLEMS 189

Error Messages 190

Error messages 190
Acquisition errors 190
Bus/Signal errors 190
File errors 191
Hardware errors 192
Help file errors 192
Import/Export and Translator errors 193
Naming errors 194
Tool errors 194
Trigger Errors 195
Warning Messages 197
Informational messages 198
Config Translator Application 199
To convert a configuration file 199
Information not converted from file 200
Self Tests 201
To access self test menu 201
Self test descriptions 201
Register Test 201
Memory Test 201
Comparator Test 201
Trigger Bus Test 202
Trigger Arm Test 202
Clock Paths Test 202
Memory Modes Test 202
Calibration Test 202
CONCEPTS 204
Logic analysis basics 204
When should I use an oscilloscope 205
When should I use a logic analyzer 206
What is a Logic Analyzer 207
What is a timing analyzer 207
What is a state analyzer 207
Clocking data in the timing analyzer 208
Sampling in the timing analyzer 209
Sampling accuracy 209
Triggering the timing analyzer 211
Pattern Trigger 211
Edge Trigger 211
Clocking data in the state analyzer 213
Sampling in the state analyzer 214
Triggering the state analyzer 215

Probing options ... 217
General Purpose Probing ... 217
Adaptor to board connectors ... 218
Analysis Probes ... 219
REFERENCE ... 220
Product Overview ... 221
Supplied Accessories ... 222
1680A/AD-series ... 222
1690A/AD-series ... 222
Optional Accessories ... 222
Documentation ... 222
Quick Start/Installation Guide ... 222
Online Help System ... 222
Agilent Technologies Web Sites ... 222
Corporation/Contact ... 222
Product Information ... 223
Documentation ... 223
Intrinsic Support ... 223
Table of 1680-series channel, memory, and speed ... 224
Model ... 224
Channel Count and Memory Depth ... 224
Table of 1690-series channel, memory, and speed ... 225
Model ... 225
Channel Width and Memory Depth ... 225
Front Panel Operation ... 226
Run/Stop ... 226
Item ... 226
Description ... 226
Save/Open Setup ... 226
Item ... 226
Description ... 226
General purpose knob ... 226
Alphanumeric Keypad ... 227
Item ... 227
Description ... 227
Shortcuts ... 227
Item ... 227
Description ... 227
Vertical ... 228
Item ... 228

Description 228
Horizontal 228
Item 228
Description 228
Marker 229
Item 229
Description 229
Keyboard Commands 230
Access Menus 230
File Operations 230
Edit Operations 230
Search Operations 231
View Operations 231
Run/Stop Operations 231
Window Operations 231
Help Operations 231
Miscellaneous 232
Menus 233
File Menu 234
Edit Menu 235
View Menu 237
Setup Menu 238
Tools Menu 239
Markers Menu 240
Run/Stop Menu 241
Window Menu 242
Help Menu 243
Toolbars 244
Standard Toolbar 245
Setup Toolbar 247
Markers Toolbar 248
Run/Stop Toolbar 249
Viewers Toolbar 250
To create a custom toolbar 251
To add tool icons 251
To remove tool icons 251
Markers Display Bar 252
Marker Measurement Display Bar 253
Windows 254

Compare Display Window 255
Dialogs 256
Analyzer Setup Dialog 257
System Options Dialog 258
To change how many triggers are saved 258
To change the length of the file history 258
To create data when offline 258
To set Trigger In and Trigger Out 258
To set message level 259
Recall Trigger Dialog 260
To recall a trigger 260
Properties Dialog 261
Symbol Selection 262
Trigger Functions 263
Timing mode trigger functions 263
Edge 263
Bus Pattern 263
Other 263
Advanced 263
State mode trigger functions 264
Patterns 264
Other 264
Advanced 264
Timing Mode 265
Timing mode trigger functions 265
Edge 265
Bus Pattern 265
Other 265
Advanced 265
Edge 267
To edit this function 267
"N" number of edges 268
To edit this function 268
Edge followed by edge 269
To edit this function 269
Edge followed by pattern 270
To edit this function 270
Edges too far apart 271
To edit this function 271
Pattern 272
To edit this function 272
Edge and Pattern 273
To edit this function 273
Pattern present for > "T" time 274

To edit this function ... 274
Pattern present for < "T" time ... 275
To edit this function ... 275
Pattern absent for > "T" time ... 276
To edit this function ... 276
Pattern absent for < "T" time ... 277
To edit this function ... 277
Pattern too late after edge ... 278
To edit this function ... 278
Pattern "AND" Pattern (timing) ... 279
To edit this function ... 279
Pattern "OR" Pattern (timing) ... 280
To edit this function ... 280
Find anything "N" times (timing) ... 281
To edit this function ... 281
Width violation on pattern or pulse ... 282
To edit this function ... 282
Wait "T" seconds ... 283
To edit this function ... 283
Wait for external arm (timing) ... 284
To edit this function ... 284
Run until user stop (timing) ... 285
To edit this function ... 285
Reset and start timer (timing) ... 286
To edit this function ... 286
Advanced If/Then (timing) ... 287
To edit this function ... 287
Advanced 2-Way Branch (timing) ... 288
To edit this function ... 288
Advanced 3-Way Branch (timing) ... 289
To edit this function ... 289
Advanced 4-Way Branch (timing) ... 290
To edit this function ... 290
State Mode ... 291
State mode trigger functions ... 291
Patterns ... 291
Other ... 291
Advanced ... 291
Pattern "N" times ... 292
To edit this function ... 292
"N" consecutive samples with Pattern1 ... 293
To edit this function ... 293
Pattern1 followed by Pattern2 ... 294
To edit this function ... 294
Pattern1 immediately followed by Pattern2 ... 295
To edit this function ... 295

Pattern1 followed by Pattern2 before Pattern3 296
To edit this function.. 296
Too few states between Pattern1 and Pattern2 297
To edit this function.. 297
Too many states between Pattern1 and Pattern2.. 298
To edit this function.. 298
Pattern2 occurring too soon after Pattern1 299
To edit this function.. 299
Pattern2 occurring too late after Pattern1.. 300
To edit this function.. 300
Find anything "N" times (state).. 301
To edit this function.. 301
Run until user stop (state)... 302
To edit this function.. 302
Reset and start timer (state) 303
To edit this function.. 303
Wait for external arm (state) 304
To edit this function.. 304
Wait "N" external clock states.. 305
To edit this function.. 305
Pattern "AND" Pattern (state) 306
To edit this function.. 306
Pattern "OR" Pattern (state) 307
To edit this function.. 307
Advanced If/Then (state) 308
To edit this function.. 308
Advanced 2-Way Branch (state).. 309
To edit this function.. 309
Advanced 3-Way Branch (state).. 310
To edit this function.. 310
Advanced 4-Way Branch (state).. 311
To edit this function.. 311
Specifications and Characteristics 312
1680/1690-Series Logic Analyzer Characteristics 313
General Information 313
Probes.. 313
State Analysis 313
Timing Analysis 313
Triggering... 314
Operating Environment Characteristics 314
1680/90A-Series Logic Analyzer Specifications 315
About the Probe Cable.. 316
Signal Requirements 317
Minimum Signal Amplitude 317
Signal Loading 317

Maximum Probe Input Voltage 317
Overdrive 317
What is a Specification? 319
What is a Calibration Procedure? 319
What is a Characteristic? 320
What is a Function Test? 320
Data Formats 321
ALA Format 321
General 321
Trigger specifications 321
Analyzer characteristics 322
Bus/Signal folders 322
Bus/Signal information (per bus/signal) 322
Marker information 322
Find value 323
Search event parameters 323
Symbols 323
Tools (e.g., inverse assemblers) 323
Filters 323
Listing display attributes 323
Waveform display attributes 324
CSV Format 325
XML Format 326
<Setup> Element 326
<Module> Element 326
<Tool> Element 327
<Data> Element 327
<Module> Element 328
<Tool> Element 328
Fast Binary Data Format 330
Data Organization 330
Object IDs 330
DataGroup 330
DataGroup Header 330
DataSet 331
Label Entry 332
Label Data 333
Label Data Attributes Bitset 334
IntegralData 335
IntegralArray<type size> 336
BitPackedData 336
BitBlockData 337
BitBlock 337
PagedBitBlock 337
PagedIntegralData<type size> 338

Strings 338
Label Extractor 339
Reorder 339
Vertical Header 340
Abscissa Data Type (x-axis information) 340
Abscissa Attributes bitset 341
Time Correlation Info 342
State Correlation Info 342

GLOSSARY 344

Agilent Logic Analyzer Help

The *Agilent Logic Analyzer* application is used with the 1680- and 1690-series logic analyzers. It can also be used by itself on a Windows NT or Windows 2000 computer for *offline analysis* of data captured on 1680-, 1690-, or 16700-series logic analyzers.

What's New

Getting Started

Probing the Device Under Test

Setting Up the Logic Analyzer

- Setting the Logic Analyzer Threshold Voltage
- Assigning Bus/Signal Names to Logic Analyzer Probes
- Selecting the Sampling Mode

Capturing Data from the Device Under Test

- Setting Up Quick Triggers
- Specifying Simple Triggers
- Specifying Advanced Triggers
- Triggering From, and Sending Triggers To, Other Instruments
- Storing and Recalling Triggers
- Starting Measurements
- Saving Captured Data (and Logic Analyzer Setups)

Analyzing the Captured Data

- Offline Analysis (after Loading Saved Data and Setups)
- Analyzing Waveform Data
- Analyzing Listing Data
- Comparing Captured Data to Reference Data
- Searching the Captured Data
- Marking, and Measuring Between, Data Points
- Displaying Names (Symbols) for Bus/Signal Values
- Printing Captured Data

Solving Problems

Concepts

Reference

Glossary

See Also

The *1680 Series Logic Analyzers Quick Start/Installation Guide*.

The *1690 Series Logic Analyzers Quick Start/Installation Guide*.

What's New

This release, version A.01.20, of the *Agilent Logic Analyzer* application:

- Lets you, in the timing (asynchronous) sampling mode, trigger on multiple glitches/edges.
- Lets you perform a simple compare of captured data sets.
- Saves/loads user-defined symbols to/from XML format logic analyzer configuration files.

  This means you can use text processing tools to re-format symbol information from software development tools, insert them into an XML format configuration file, and load them into the *Agilent Logic Analyzer* application (see XML Format).
- Contains an API and documentation to enable inverse assembler and analysis tool development.
- Extends its offline analysis capability to 16700-series logic analyzers.

See Also

Version A.01.10 What's New

Version A.01.10 What's New

Version A.01.10 of the *Agilent Logic Analyzer* application:

- Added a Japanese version of the online help.

Getting Started

Tutorial - Getting to know your logic analyzer
Measurement Examples

Tutorial

Getting to know your logic analyzer

The following tutorial is intended to give new users a quick overview of logic analyzer basics. In addition to learning the concepts of logic analysis, you will see some of the logic analyzer's more common features by going through a measurement overview. Finally, you are shown some easy time saving tasks that can quickly make you as productive as the more experienced user.

Logic analysis basics

- When should I use an oscilloscope
- When should I use a logic analyzer
- What is a logic analyzer
  - Timing analyzer
    - Clocking
    - Sampling
    - Triggering
  - State analyzer
    - Clocking
    - Sampling
    - Triggering
- Probing options

Measurement overview

The following overview does not require an active target system. However, in order to show features that work on data, you are asked to load a configuration file between steps 5 and 6 that contains data to finish the exercise.

- Turning on the logic analyzer
- Connecting to the target system
- Setting up bus/signal names
- Setting the acquisition mode
- Setting up a simple trigger
- Open the tutorial configuration file
- Using markers
- Zooming in on the data

Time saving tasks

- Loading and saving configuration files
- Saving and recalling trigger setups
- Quick marker measurements
- Searching data
- Toolbars and mouse shortcuts

See Also

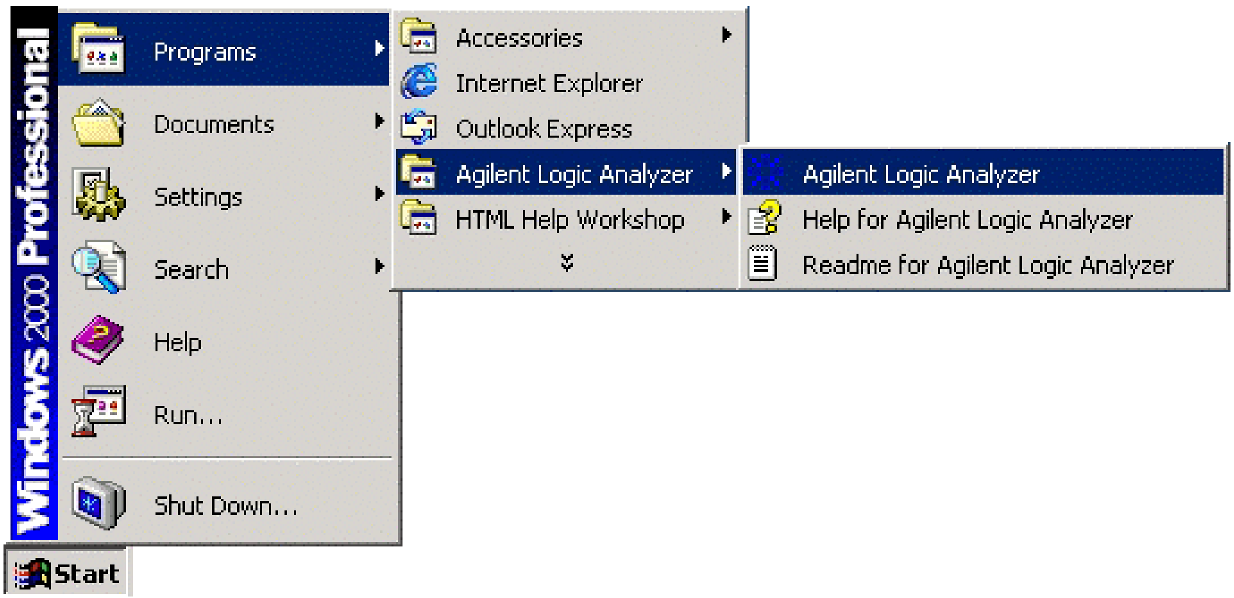
Product Overview

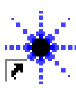
Measurement overview

Turning on the logic analyzer

Tutorial Home | Next Topic | Previous Topic

1. Plug in the power cable and press the front-panel On/Off button.
2. From the Windows Start bar, click Start>Programs>Agilent Logic Analyzer>Agilent Logic Analyzer.





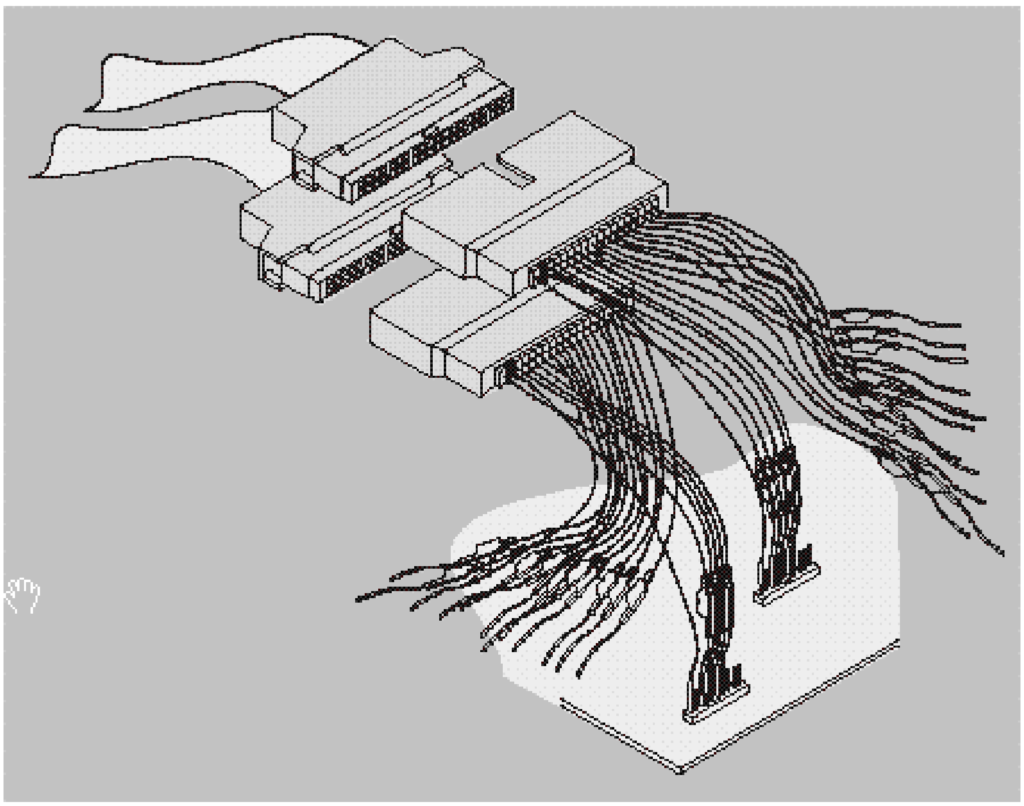
Optional: If you have a logic analyzer shortcut icon on screen, double-click the icon.

Connecting to the target system

Tutorial Home | Next Topic | Previous Topic

The standard probing that comes with the logic analyzer is the passive general purpose probe with sixteen channels per cable. Each channel is terminated at both ends with 100k ohm and 8 pF. With this type of probing, you can also disconnect the leads and plug the cable connector directly into a connector on the device under test.

> Note: In this tutorial, no probe connections are required. Later on in this tutorial, you are asked to load a configuration file containing data to simulate the results of a probed device under test. However, at this time, if you have the credit card demo board available, you can connect it and capture real data for this tutorial.



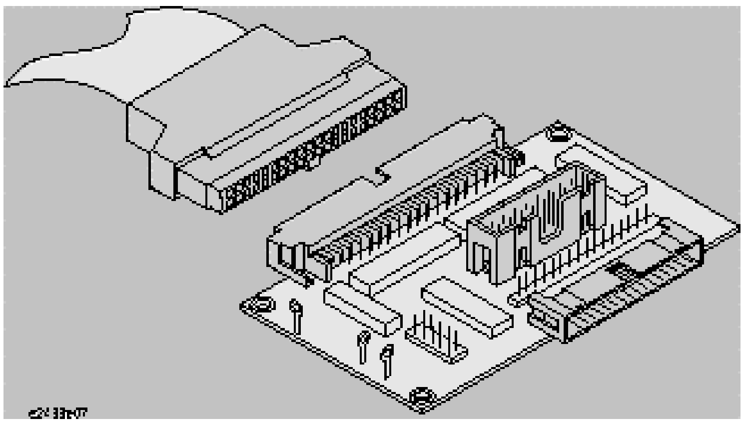
As the number of channel connections increase, other probing options may be more convenient.

Credit card demo board

Tutorial Home | Previous Topic

If you have a credit card demo board available, you can connect it as shown below and capture the same data used in this measurement overview. Other benefits are that you will see real activity indicators in the bus/signal setup dialog.

1. Connect the probe cable of Pod 1 into connector J1 of the demo board.



Setting up bus/signal names

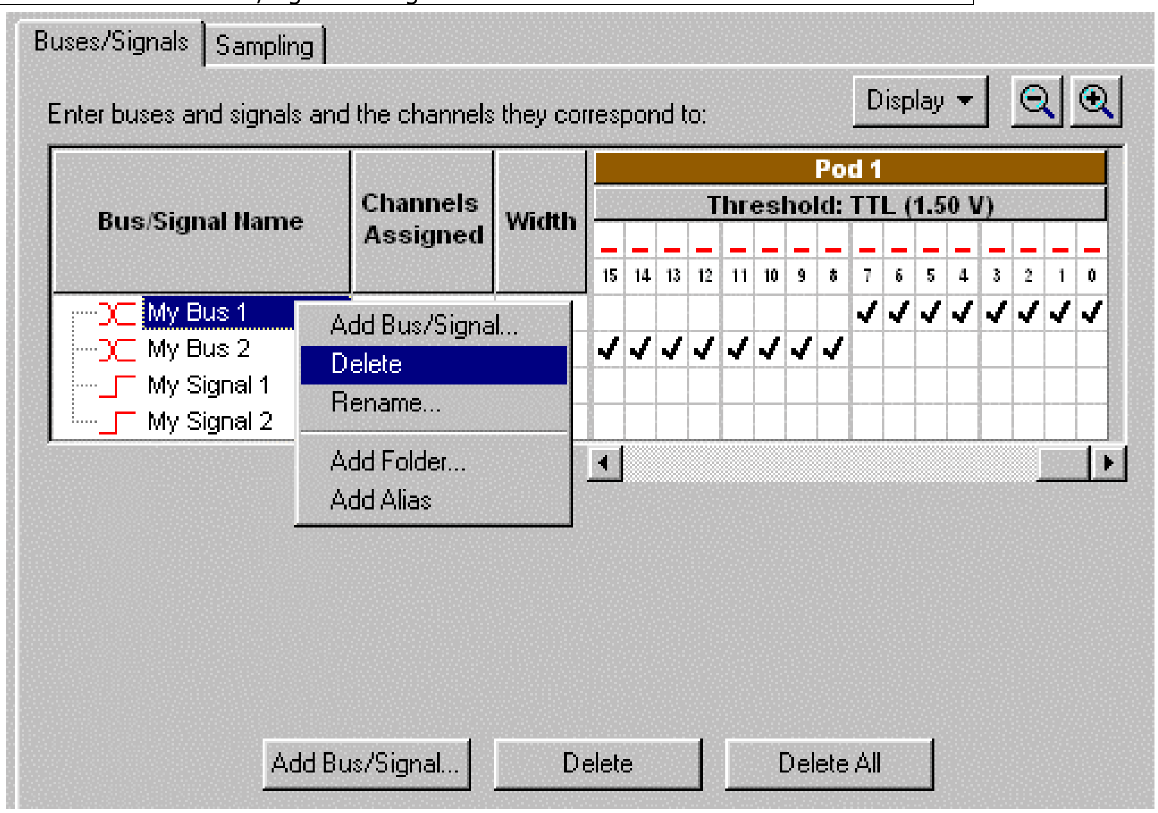
Tutorial Home Next Topic Previous Topic

By default, the analyzer has two buses (My Bus 1 & 2) and two signals (My Signal 1 & 2) configured in the interface. The following exercise cleans up the display defaults and re-configures the analyzer bus/signal setup for a new measurement.

Delete bus/signal names

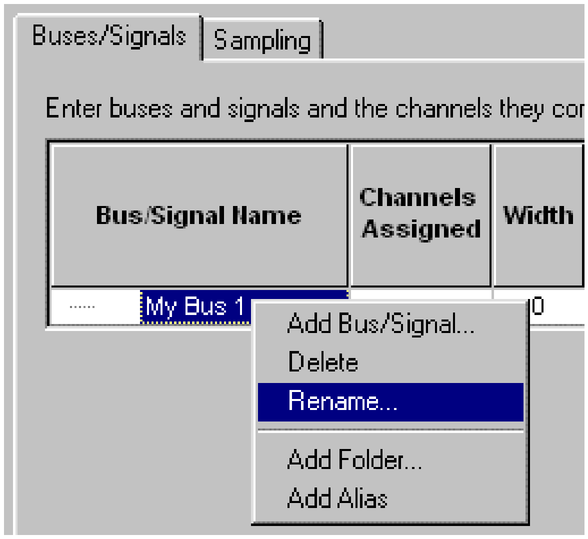
1. In the menu bar click Setup>Buses/Signals.
2. In the Analyzer Setup dialog that appears, right-click on My Bus 1, then select Delete. Repeat until all bus signal names are deleted. After the last bus/signal is deleted, "My Bus 1" appears again as a default name.

Tip: You can delete all bus/signal configurations at once with the Delete All button.



Add new bus/signal name

1. In the Analyzer Setup dialog, right-click on My Bus 1, then select Rename.
2. From the popup keypad that appears, type in the new name "counter".
3. Select Ok.



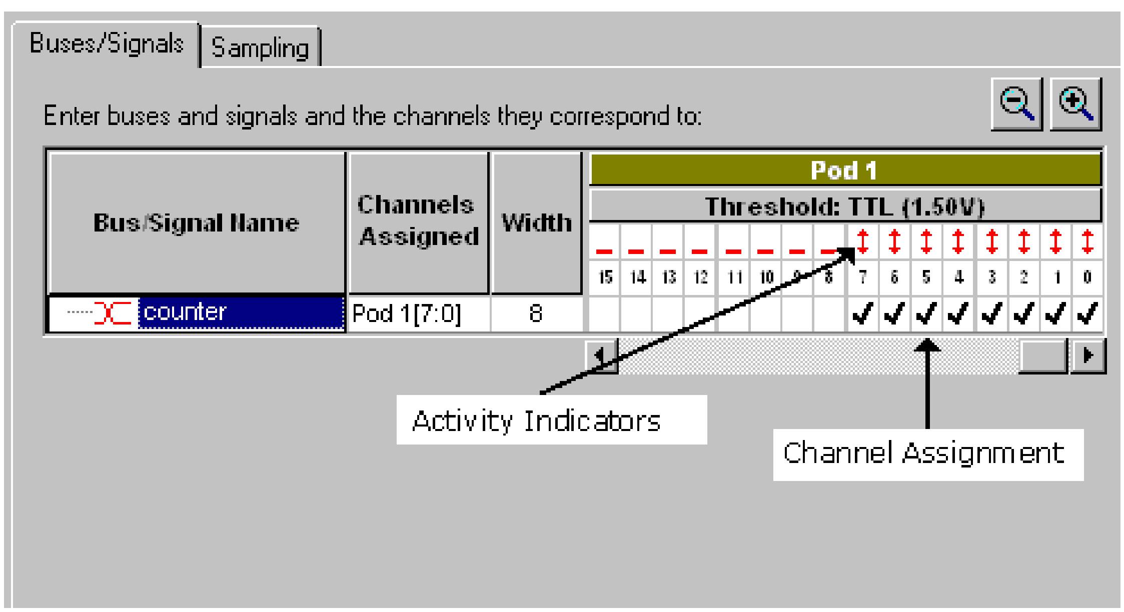
Map signals into the analyzer

The analyzer must be told which probed signals from the device under test are to be included in the measurement, and how you want them grouped in the analyzer. In this exercise, you assign channels 0 - 7 on Pod 1 under the name "counter". Notice that when more than one channel is assigned to "counter" it becomes a bus.

1. Check the activity indicators for verification of proper connection to the target system. You should see a transition arrow on all 8 channels.

Note: If you have the credit card demo board connected for this tutorial, you will see activity indicators as shown below. If you will be loading the demo configuration file (later in this tutorial) you will not see activity.

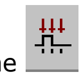
1. Click each channel assignment box under channels 0 - 7 on Pod 1. Notice that as you assign channels, the configuration information is updated for the bus/signal.
2. Click Ok.

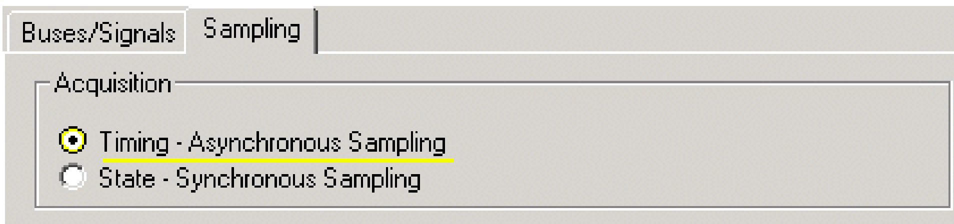


Setting the acquisition mode

Tutorial Home Next Topic Previous Topic

Under the Sampling tab of the Analyzer Setup dialog is where you set the analyzer to be either a timing or state analyzer. You also set either the timing options, such as memory depth or sampling period, or the state clocking options.

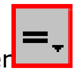
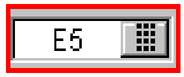
1. From the menu bar, click Setup>Timing/State (Sampling)... ., or click the  icon in the toolbar.
2. Select Timing - Asynchronous Sampling.
3. Click Ok.



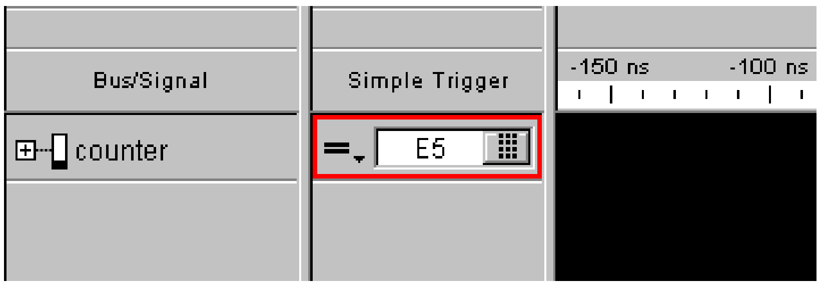
Setting up a simple trigger

Tutorial Home | Next Topic | Previous Topic

The Simple Trigger is a quick way to configure the analyzer to trigger on either a data pattern on a bus, or an attribute of a single signal such as a rising edge or a low logic level.

1. In the Simple Trigger, click on the pattern qualifier  and set it to Equal.
2. Click in the text entry field 

and enter the data pattern "E5".



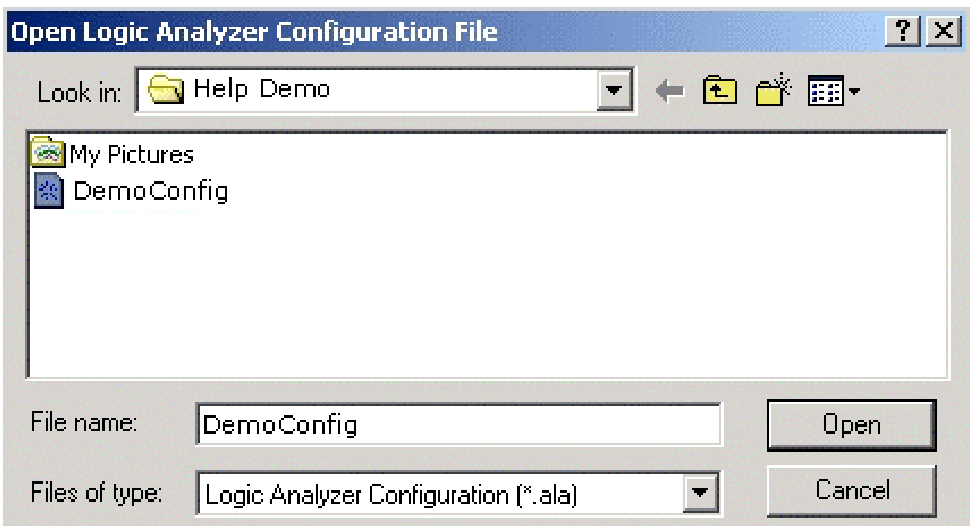
Open the tutorial configuration file

Tutorial Home | Next Topic | Previous Topic

At this point in a measurement, you would normally run the logic analyzer. However, because you are not connected to a device under test, you cannot capture real data. You will have to load a configuration file that contains this data.

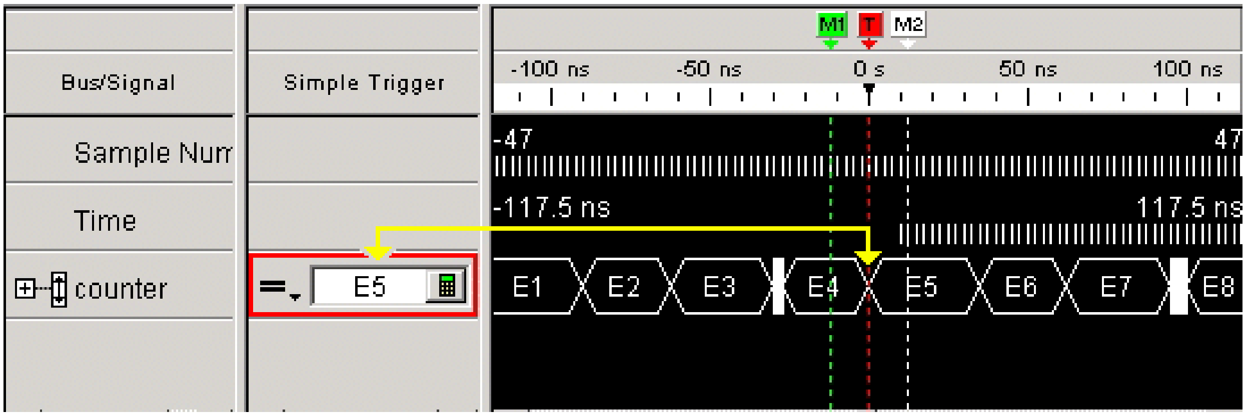
Load the configuration file

1. Select File>Open.
2. From the file manager dialog, select the file named DemoConfig.ala from the following directory: C:\Documents and Settings\All Users\Documents\Agilent Technologies\Logic Analyzer\Default Configs\Agilent\Help Demo\
3. Select Open.



View the data

Notice how the logic analyzer triggered on data pattern E5 and placed it in the center of the display. The red line shows that the trigger point is at the start of the data pattern E5.


M1
T
M2
Bus/Signal
Simple Trigger
-100 ns
-50 ns
0 s
50 ns
100 ns
Sample Num
-47
47
Time
-117.5 ns
117.5 ns
counter
E5
E1
E2
E3
E4
E5
E6
E7
E8

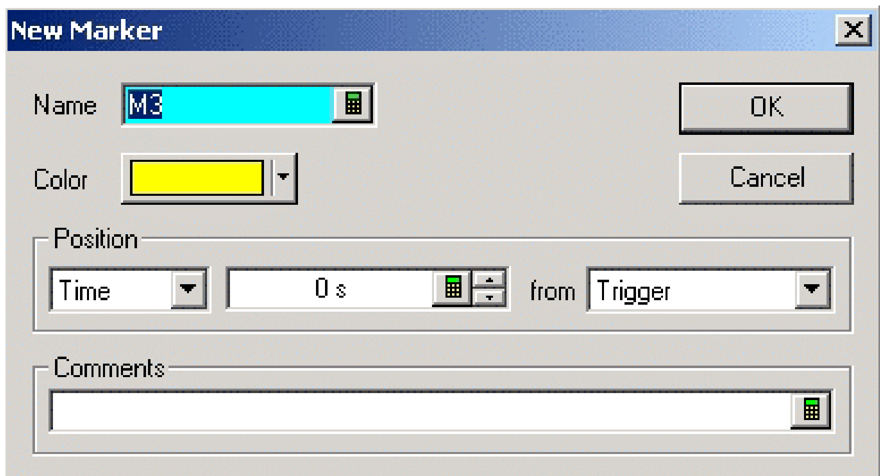
Using markers

Tutorial Home | Next Topic | Previous Topic

Markers are used for creating reference points in data. Once markers are placed in data, you can use them to quickly see what time, sample, or data value the marker is set on.

To create a marker

1. From the menu bar, click Markers>New.
2. From the New Marker dialog that appears, configure the new marker and if desired, specifically a position it in data. When you do not position the marker, by default it is placed at the trigger point.
3. Select Ok.



To place a marker in data

When you first create a new marker, you have the option to place it in data at a specific point in time or a specific sample number. The following exercise shows you other ways to position markers in data.

1. In the display, click on marker M3 (your new marker) and while holding the mouse button down, drag maker M3 to -100ns before trigger, then release. Notice that the marker position value changes as you move it.
2. From the menu bar, click Markers>Place On Screen, then select M1 and click Ok. Notice how M1 is placed at center screen at the red trigger line.
3. Point the mouse cursor at any desired point in data, then right-click and select Place Marker. From the Place Marker dialog that appears, choose the M2 marker. Notice that the marker is placed where the mouse was pointing.

Go To a marker in data

Once you have markers set in data, you can quickly find any of them as follows.

1. From the menu bar, click Markers>GoTo.
2. Select the marker you want to find, and click Ok.

Zooming in on the data

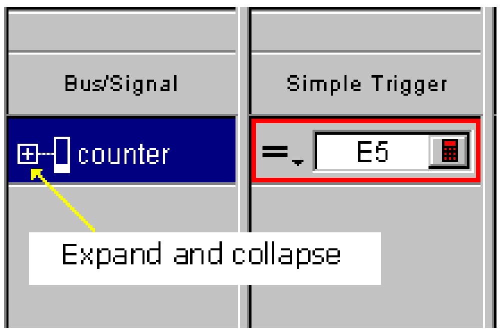
Tutorial Home | Next Topic | Previous Topic

Data from a timing analyzer is displayed similarly to oscilloscope data. Both an analyzer and scope display waveforms on a horizontal time axis. Therefore to zoom in or out on a waveform, change the Scale (time/division) of the time axis of the waveform.

Both state and timing analyzers can have multiple signals grouped together in a bus. To get a view of all signals, you can expand a bus into individual signals.

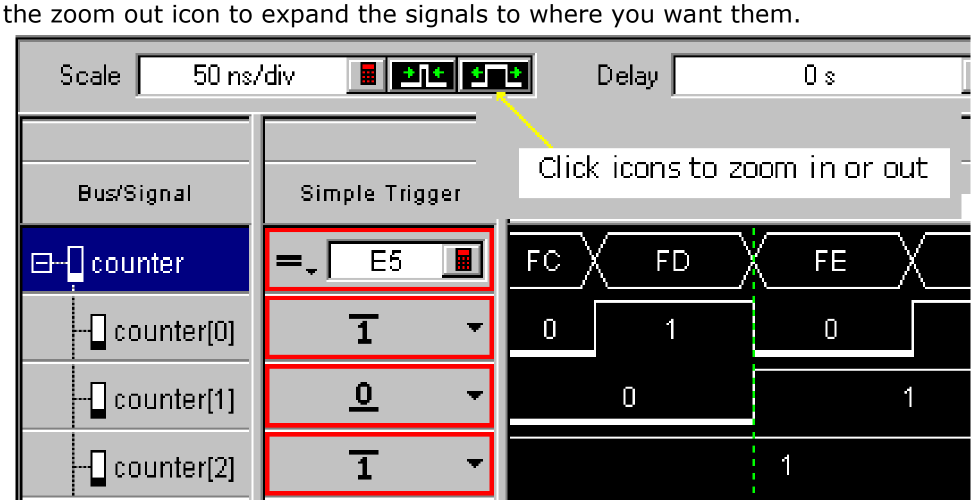
Expand a bus

Click the "+" symbol just to the left of the bus named "counter". The collection of signals under "counter" breaks out into individual signals named counter[0] - counter[7].



Change the scale

Click the zoom out icon to expand the signals to where you want them.



Time saving tasks

Loading and saving configuration files

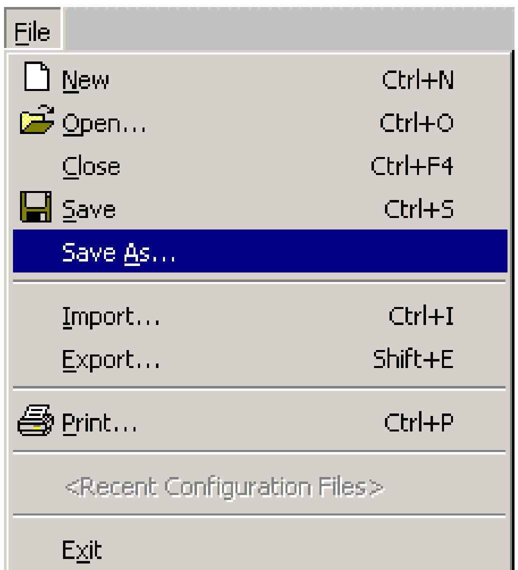
Tutorial Home | Next Topic | Previous Topic

Many times it is quicker to open an existing configuration file with a similar setup than to create a new configuration from scratch. You simply open a similar file, make the appropriate changes to the setup, then save the file as a new filename.

Note: When you rename an existing configuration file, you retain the saved trigger setups and "Find" search favorites from the first configuration file.

You already have learned how to open a configuration file. In the following exercise, you will save the "democonfig" file to a new name.

1. From the menu bar, click File>Save As... .
2. From the file manager dialog that appears, type in the new name "myconfig", then click Save.



Saving and recalling trigger setups

Tutorial Home | Next Topic | Previous Topic

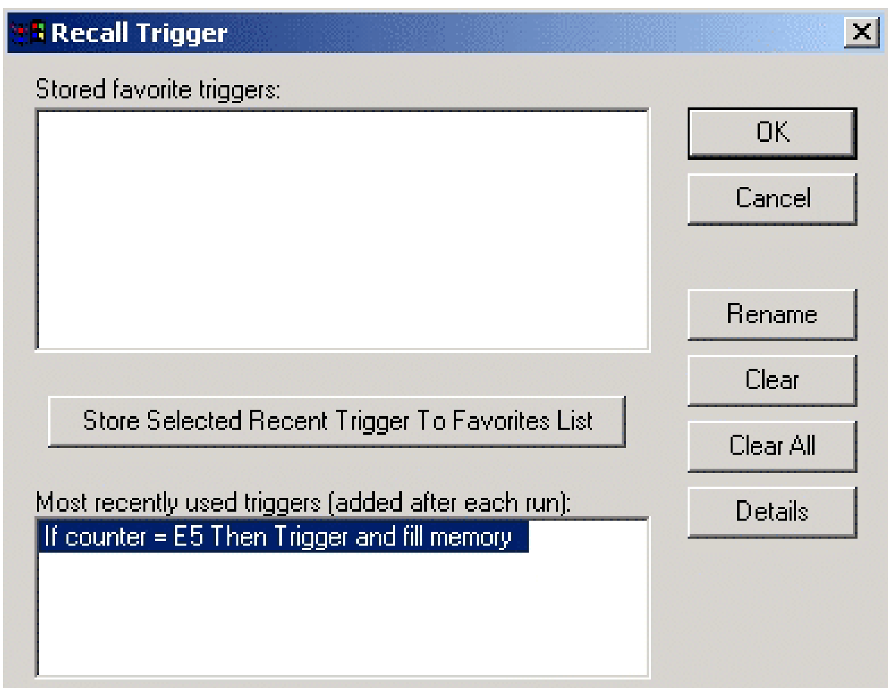
Each time you setup a new trigger and run the measurement, the trigger setup is stored in the analyzer. It is quicker to recall a trigger setup rather than re-configure the trigger setup each time.

Note: The analyzer must be run before the trigger setup is stored. Also, trigger setups are stored as part of the configuration file. If you load a new configuration file, the trigger setups will be overwritten by trigger setups stored with the new file.

To recall a trigger setup

1. From the menu bar, click Setup>Recall Trigger.
2. From the lower list, select the desired trigger setup, then click Ok.

Tip: When the list of most recently used triggers get long, you can store the most often used triggers in the upper favorites list.

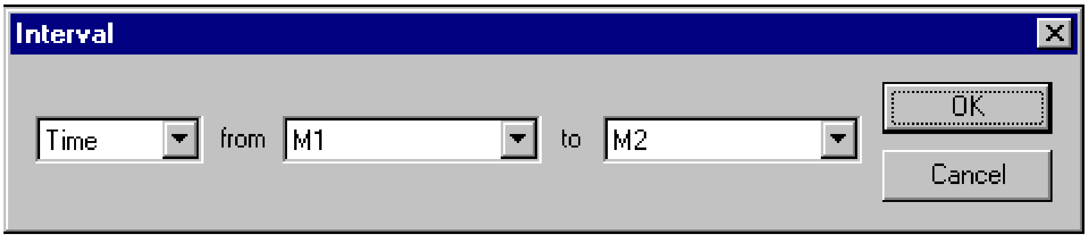


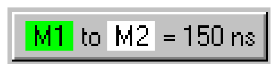
Quick marker measurements

Tutorial Home | Next Topic | Previous Topic

You can quickly read the time or number of samples between markers.

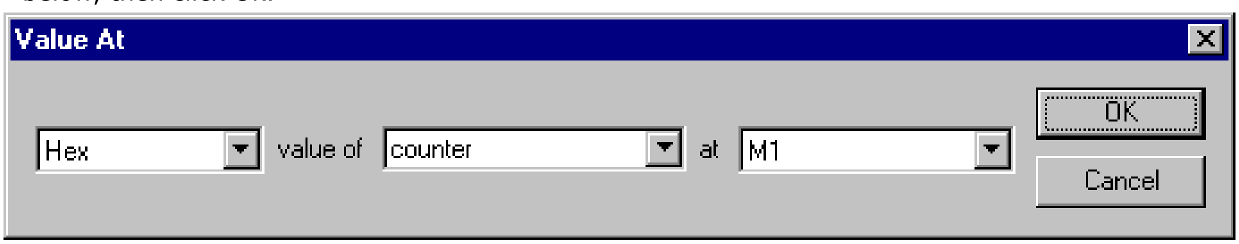
1. Click Markers>New Interval Measurement.
2. Configure the Interval dialog to display the Time from M1 to M2 as shown below, then click Ok.



The result of the interval measurement 

is displayed in the marker measurements display bar.

3. Click Markers>New Value At Measurement.
4. Configure the Value At dialog to display the Hex value of counter at M1 as shown below, then click Ok.



The result of the value at measurement 

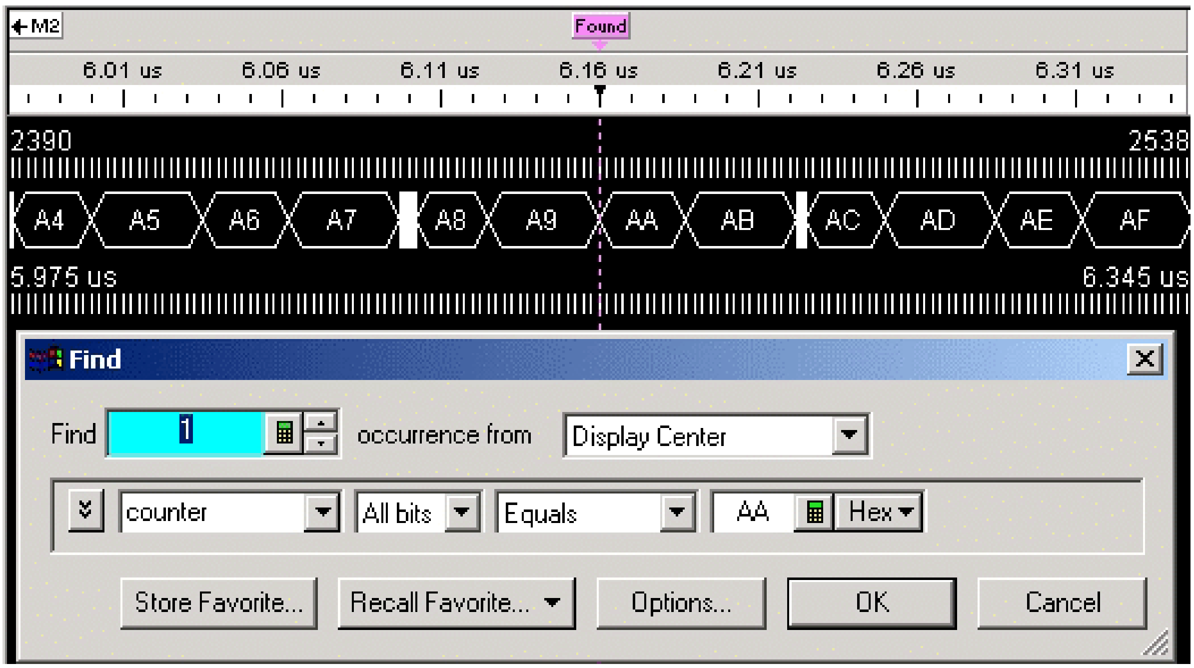
is displayed in the marker measurement display bar.

Searching data

Tutorial Home | Next Topic | Previous Topic

You can search for a data pattern on a bus, or a single signal. You can also choose when the search begins and ends. Finally, you can save the search criterion in a favorites list.

1. From the menu bar, click Edit>Find.
2. From the Find dialog that appears, configure the search criterion as shown below to find "AA".
3. Select Ok.



As you configure the Find dialog, try to think of it as constructing a sentence that reads left-to-right.

"Find the 1st occurrence from Display Center, on a Bus named counter, including All bits, a pattern that Equals AA".

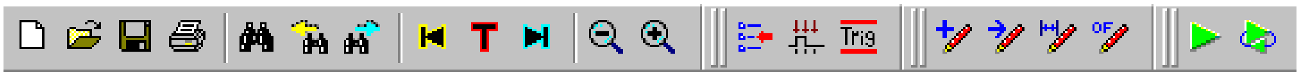
Toolbars, tooltips and mouse shortcuts

Tutorial Home | Previous Topic

Throughout this tutorial, the menu bar has been used to access features. There are two other ways to access features as well as other useful tips that can save you time.

Toolbars

Below the menu bar are groups of icons that represent shortcuts to many dialogs and features. For more information refer to Toolbars in the main help.

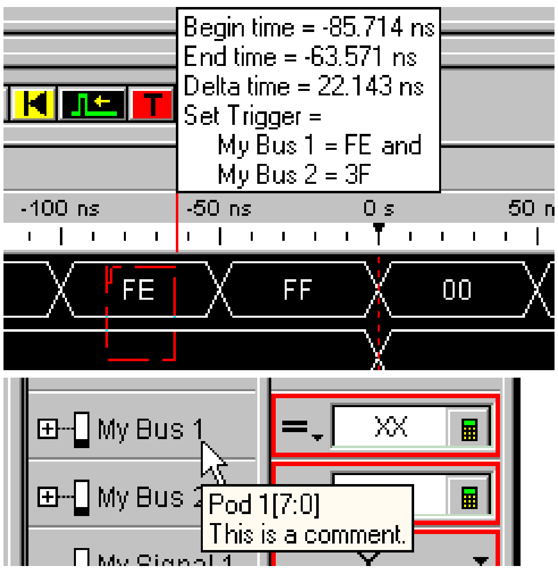


Mouse shortcuts

There are many mouse shortcuts available. To access them simply point the mouse over a screen element such as a marker, or screen area, then right-click the mouse. Mouse shortcuts are especially useful within the waveform and listing data display areas.

Tooltips

Tooltips are small information displays that appear during operations such as moving markers, setting a trigger with the mouse, or hovering the mouse over a bus/signal name. Use them as comments, or to monitor your progress or current positions.



Measurement Examples

The following measurement examples show you the typical order of steps to setup and run a measurement. As you go through the examples, you will encounter steps such as probing or triggering where alternative choices are available. In these steps, select the probing or trigger example that best fits your measurement.

Making a timing analyzer measurement

Making a state analyzer measurement

See Also

Tutorial - Getting to know your logic analyzer

Timing mode trigger functions

State mode trigger functions

Making a timing analyzer measurement

The following measurement example shows you the steps necessary to configure and run the logic analyzer for a typical timing analyzer measurement. As you go through the example, make the appropriate choices from the selection lists that best match the kind of configuration you need.

Tip: If you are new to logic analysis, refer to "Getting to know your logic analyzer" for a quick tutorial on logic analysis concepts and measurements.

1. Connect the Probing to the device under test.
2. Turn on the logic analyzer.

Bus and signal setup

3. In the menu bar, select Setup>Bus/Signal... .
4. From the Buses/Signals tab, assign bus/signal names to the probed signals on the target system. You do this by either renaming existing names, or deleting and creating new names.
5. From the Buses/Signals tab, assign channels under the appropriate pods for all probed buses/signals on the device under test.

Acquisition mode setup

6. In the Analyzer Setup dialog, select the Sampling tab.
7. From the Sampling tab, set the acquisition mode to Timing - Asynchronous Sampling.
8. Set the Sampling Options.
9. Set the Sampling Period.

Trigger setup

10. The trigger required to capture specific data depends on the measurement. However, the trigger is generally set in two ways.

- From within the data display, setup a Simple Trigger .
- From the Advanced Trigger dialog, setup a timing mode Advanced Trigger function.

Run the measurement

11. Run the measurement.

See Also

To set the trigger position

To set acquisition depth

Making a state analyzer measurement

The following measurement example shows you the steps necessary to configure and run the logic analyzer for a typical state analyzer measurement. As you go through the example, make the appropriate choices from the selection lists that best match the kind of configuration you need.

Tip: If you are new to logic analysis, refer to "Getting to know your logic analyzer" for a quick tutorial on logic analysis concepts and measurements.

1. Connect the Probing to the device under test.

   Note: Be sure that the clock signals of your device under test are connected to clock channels on the pods. Any unused clock channels can be used for additional data channels and will not feed into the state clock setup.

2. Turn on the logic analyzer.

Bus and signal setup

3. In the menu bar, select Setup>Bus/Signal... .
4. From the Buses/Signals tab, assign bus/signal names to the probed signals on the target system. You do this by either renaming existing names, or deleting and creating new names.
5. From the Buses/Signals tab, assign channels under the appropriate pods for all probed buses/signals on the device under test.

Acquisition mode setup

6. In the Analyzer Setup dialog, click the Sampling tab.
7. From the Sampling tab, set the acquisition mode to State - Synchronous Sampling.
8. Set the state clock type.
9. Set the state clock qualifiers.
10. If necessary, set the advanced state clocking.

Trigger setup

11. The trigger required to capture specific data depends on the measurement. However, the trigger is generally set in two ways.

- From within the data display, set up a Simple Trigger .
- From the Advanced Trigger dialog, set up an Advanced Trigger function.

Run the measurement

12. Run the measurement.

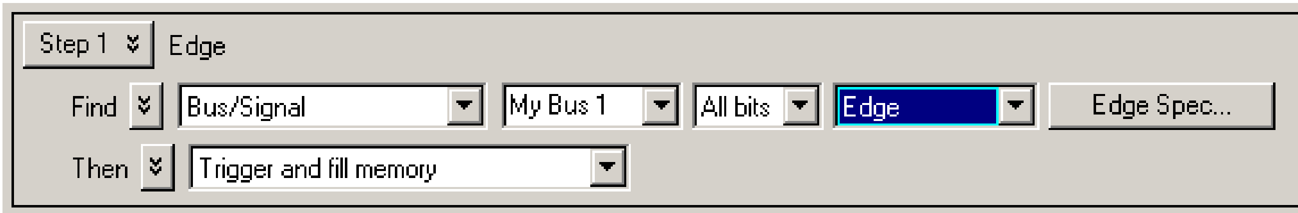
See Also

To set the trigger position

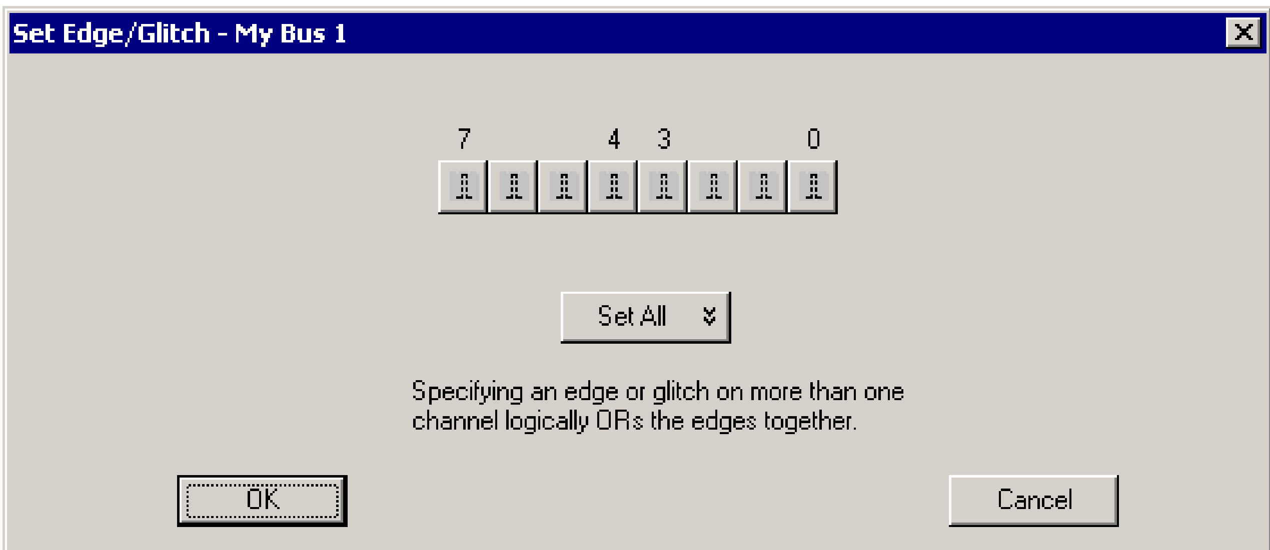
To set acquisition depth

To trigger on one of multiple edges or glitches

1. In the timing sampling mode, set up an Advanced Trigger.
2. Select the bus on which you're looking for one of multiple edges or glitches.
3. Select **All bits** in the bus.
4. Select **Edge**.



5. Click **Edge Spec...**.
6. In the Set Edge/Glitch dialog, specify edges or glitches you are looking for; use the**Set All** button to make a selection for all signals in the bus.



7. Click **OK** to close the Set Edge/Glitch dialog.
8. Click **OK** to close the Advanced Trigger dialog.

Note: Glitches are not drawn on the screen. You need an oscilloscope to further troubleshoot glitches and find out when they occur.

Probing

The following probing options are available:

- General Purpose Probing (Standard)
- Adapter to board connectors (Optional)
- Analysis probes (Optional)

So far we've talked about some of the differences between scopes, timing and state analyzers. Before we're ready to apply these new tools, we should talk about one more subject – the probing system.

A scope probe is designed to gain easy access to the target system while minimizing the signal distortion. Since we want to look at parametric information like voltage levels and rise times, it is important that the probe doesn't load the circuit under test significantly. A typical scope probe has 1 M ohm impedance shunted by 10 pF, depending on the bandwidth required.

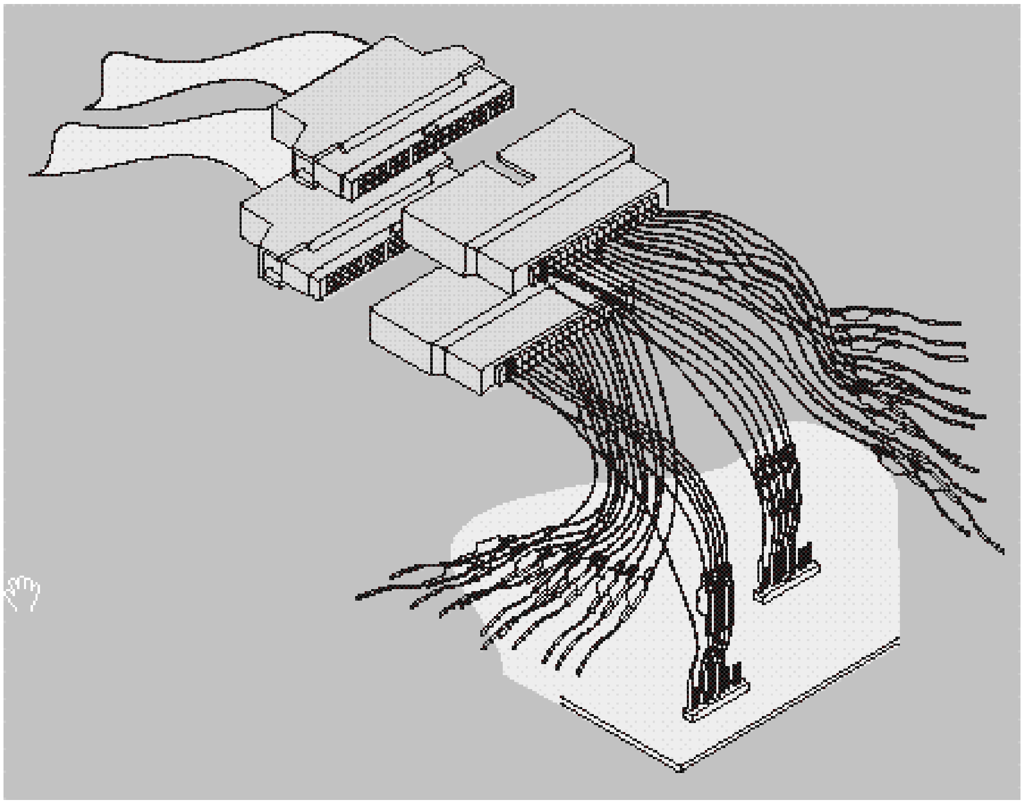
Logic analyzer probes are designed to allow connection of a high number of channels to the target system easily by trading off amplitude accuracy of the signal under test. Remember that a logic analyzer only distinguishes between two voltage levels!

Traditionally, logic analyzers used active probe pods, which had an integrated signal detection circuitry for eight channels integrated. From these pods, we could connect with leads to the circuit under test.

The typical impedance of a logic analyzer probe is in the area of 100 k ohm shunted by 8 pF at the input of the active pod. The connecting wires, however, add another 8 pF stray capacitance, giving a total of 16 pF per channel.

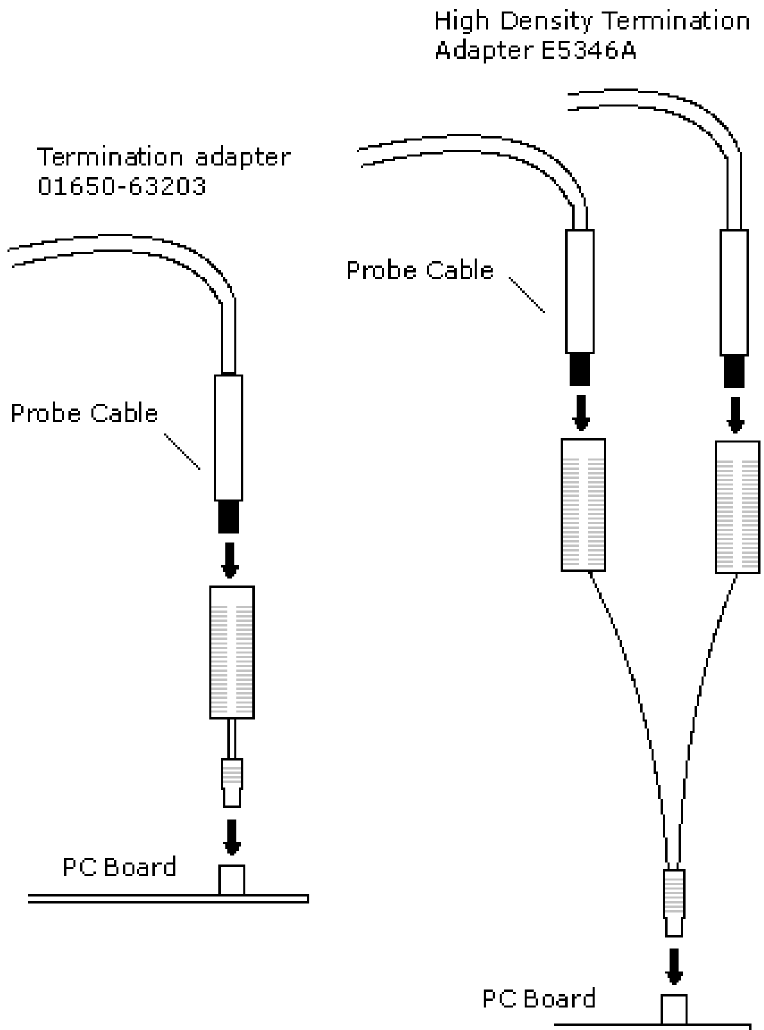
General Purpose Probing

Physical connections to digital systems must be reliable and convenient to deliver accurate data to the logic analyzer with minimum intrusion to the target system. The standard general purpose probing solution shown below is shipped with the logic analyzer. Each channel is terminated at both ends with 100k ohm and 8 pF.



The standard set plugs directly into any .1-inch grid with 0.026 to 0.033-inch diameter round pins or 0.025-inch square pins. All probe tips work with the Agilent Technologies 5059-4356 surface mount grabbers and the Agilent Technologies 5959-0288 through-hole grabbers.

Adaptor to board connectors



Both the 01650-63203 and the E5346A adapters include termination for the logic analyzer. The 01650-63203 termination adapter plugs into a 2 x 10 pin header with 0.1 inch spacing. The E5346A high-density adapter connects to an AMP "Mictor 38" connector.

Analysis Probes

Connecting a state analyzer to a microprocessor system requires some effort in terms of mechanical connection and clock selection. Remember, we have to clock the state analyzer whenever data or addresses on the bus are valid. With some microprocessors it might be necessary to use external circuitry to decode several signals to derive the clock for the state analyzer.

Analysis probes (formerly called preprocessors) are microprocessor-specific interfaces that make it easier to probe buses. Generally, analysis probes consist of a circuit board that attaches to the microprocessor (possibly through an adapter) and a configuration file. The configuration file sets up the logic analyzer's clocks, buses, and signals correctly, and may include an inverse assembler. The circuit board provides access to logical groups of pins through headers designed to connect directly to the logic analyzer.

Setting Up the Logic Analyzer

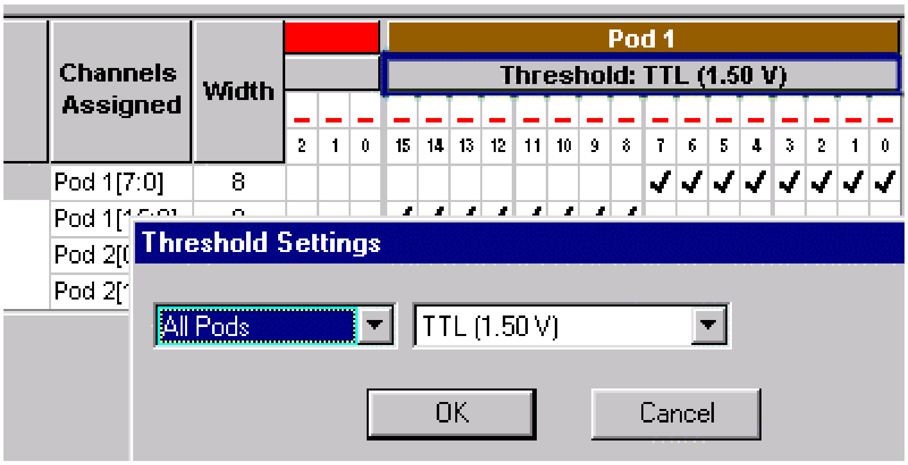
Setting the Logic Analyzer Threshold Voltage
Assigning Bus/Signal Names to Logic Analyzer Probes
Selecting the Sampling Mode

To set thresholds

It is very important that you specify a threshold voltage that matches what your target system is using. Incorrect threshold voltages result in incorrect data.

1. From the menu bar, select Setup>Bus/Signal.
2. Click any Threshold field. The Threshold fields are located under the Pod label.

   The Threshold Settings window appears.

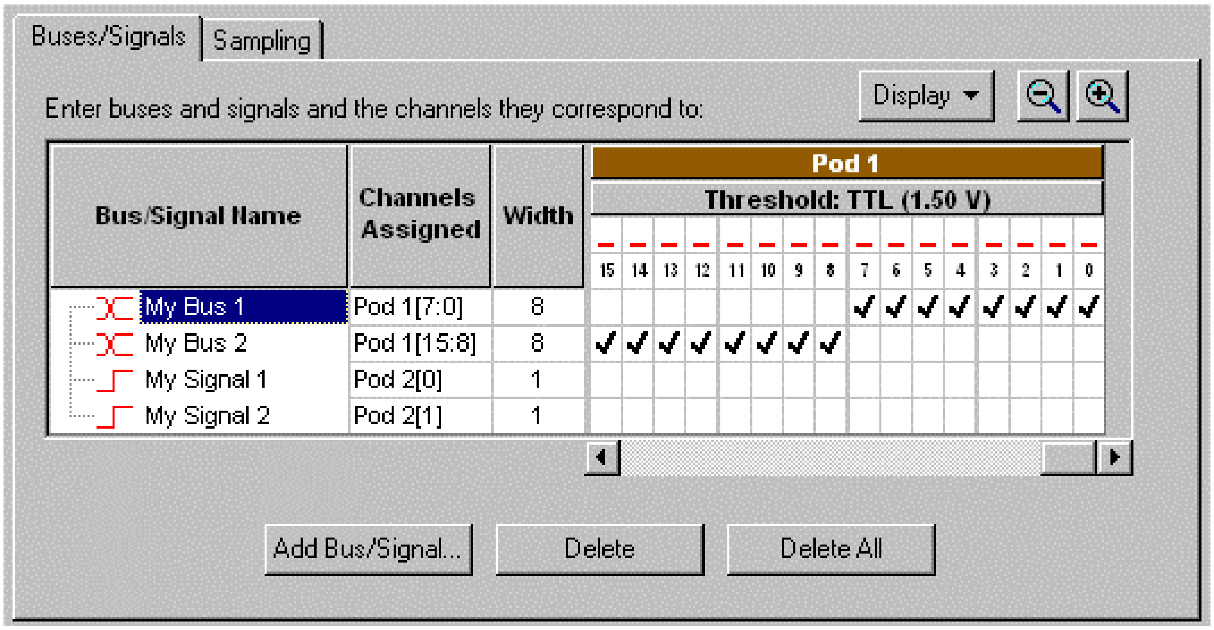


   a. Select All Pods or just a single pod.
   b. Specify the threshold level. Choices are

AGP (1.32 V)	CCT (1.50 V)	CMOS 5V (2.50 V)
ECL (-1.30 V)	GTL (0.80 V)	GTLPlus (1.00 V)
HSTL (0.75 V)	LVCMOS 1.5V (0.75V)	LVCMOS 1.8V (0.90 V)
LVCMOS 2.5 V (1.25 V)	LVCMOS 3.3 V (1.65 V)	LVPECL (2.00 V)
LVTTL (1.40 V)	PECL (3.70 V)	SSTL2 (1.25 V)
SSTL3 (1.50 V)	TTL (1.50 V)	User (-6.00 to 6.00 V)

Assigning Bus/Signal Names to Logic Analyzer Probes

Buses/Signals Setup



The Buses/Signals tab is accessed through the menu bar's Setup>Bus/Signal. The Buses/Signals tab is used to map (assign) bus and signal names in the interface to the pod and channel connections of the probes. You also use the Buses/Signal tab to set up thresholds, polarity, numeric base, and enter user comments. Through the Display... field, you can select what bus/signal setup information is displayed.

The following tasks are performed in the Buses/Signals setup tab.

To add a new bus or signal

To delete a bus or signal

To rename a bus or signal

To assign channels

To set thresholds

To set numeric base

To set setup/hold

To set polarity

To add user comments

To add a folder

To alias a bus/signal name

To sort bus/signal names

To edit symbols

Read only options

The following fields are read only and cannot be edited. The display of these items can be turned on/off under the Display... field.

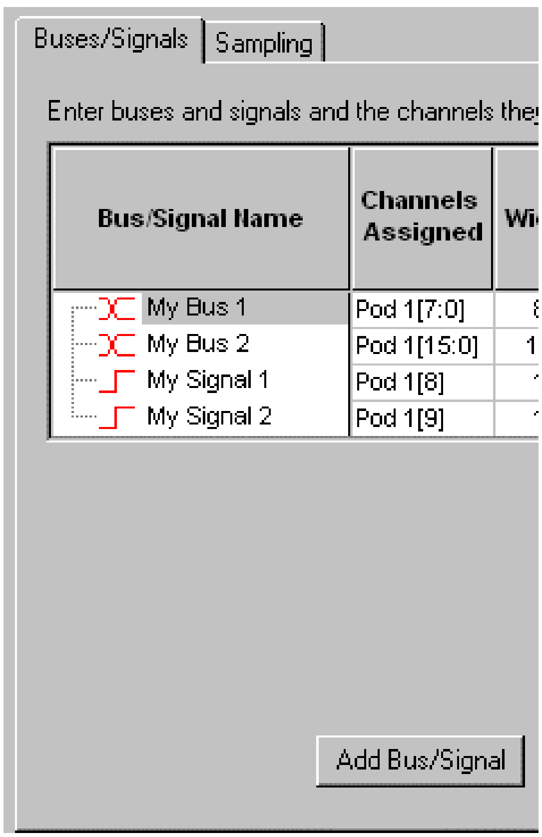
Channels Assigned	The Channels Assigned column displays the channel assignments in textual form.
Width	The Width column displays the number of assigned channels on each bus.
Activity	The Activity row displays the type of signal activity on each channel. • Low bar = A stable low level. • High bar = A stable high level. • Transition arrows = An active signal transition between low and high
Channel Numbers	The Channel Numbers row displays pod channel numbers

To add a new bus or signal

The add bus/signal feature allows you to add new buses and signals to the configuration. Once added to the configuration, the new bus/signal is automatically inserted into the data displays and also becomes available in any bus/signal insert function.

1. From the menu bar, select **Setup>Bus/Signal**.
2. Select **Add Bus/Signal** to add a new bus or signal.
3. The new bus/signal will appear with a system generated default name. Rename the new bus/signal if desired.

Note: Before a new bus/signal can be added to the configuration, at least one channel must be assigned to the bus/signal.



See Also

To delete a bus or signal
To rename a bus or signal
To assign channels
Buses/Signals Setup

To delete a bus or signal

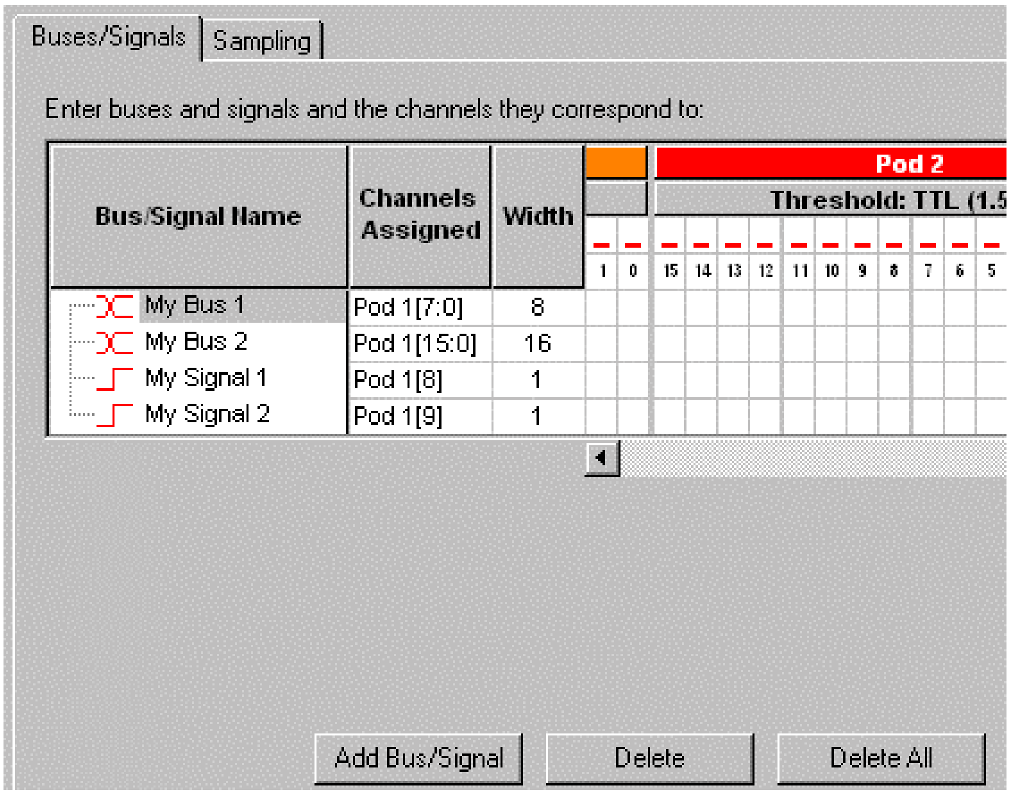
The delete bus or signal feature allows you to remove buses and signals individually or all at once. The delete bus or signal feature is accessed through the setup menu or the setup toolbar.

To delete an individual bus or signal

To delete all buses and signals

To delete an individual bus or signal

1. From the menu bar, select **Setup>Bus/Signal**.
2. Highlight the bus or signal you want to delete.
3. Click
   Delete.



To delete all buses and signals

1. From the menu bar, select **Setup>Bus/Signal**.
2. Click **Delete All**.

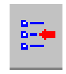
Note: Some tools "lock" buses and signals because they use the bus or signal to produce their own output. Delete and Delete All will not delete these locked buses and signals. A locked bus or signal has a grey icon to the left of the name instead of a red icon.

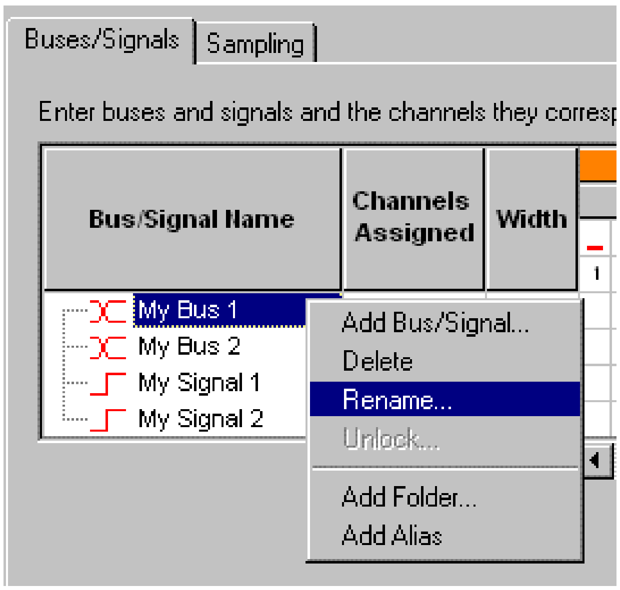
See Also

Buses/Signals Setup

To rename a bus or signal

The rename bus/signal feature allows you to change bus and signal names. All channel, pod, and clock assignments for the renamed bus/signal remain unchanged.

1. From the menu bar, select **Setup>Bus/Signal**, or click the  icon in the setup toolbar.
2. Right-click the bus or signal name and choose **Rename...**



3. Enter the new bus or signal name.
4. Select **OK**.

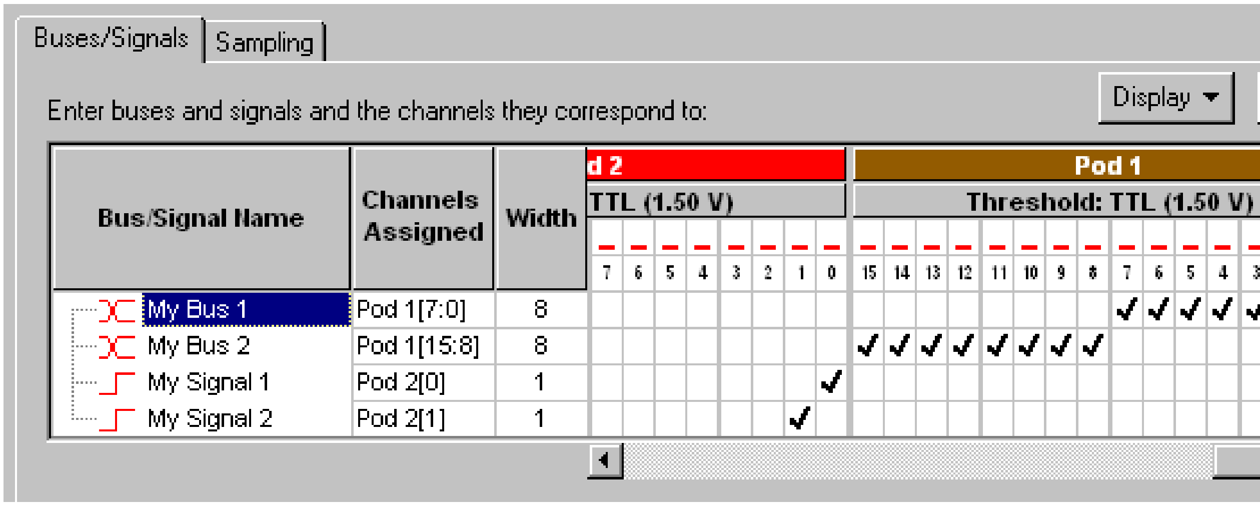
See Also

To add a new bus or signal
To delete a bus or signal
Buses/Signals Setup

To assign channels

To make the logic analyzer display match your system's design, assign the physical channels of the logic analyzer to bus and signal names.

1. From the menu bar, select Setup>Bus/Signal.
2. In the Buses/Signals tab, select squares in the grid to assign channels to bus and signal names. For each signal probed in your device under test, you should have a black check mark mapping the channel to a pod and to a signal name in the interface. Example: In the picture below, channels 0-7 (pod 1) are mapped to My Bus 1, channels 8-15 (pod 1) are mapped to My Bus 2, and channels 8 and 9 (pod 2) are mapped to My Signal 1 & 2 respectively.

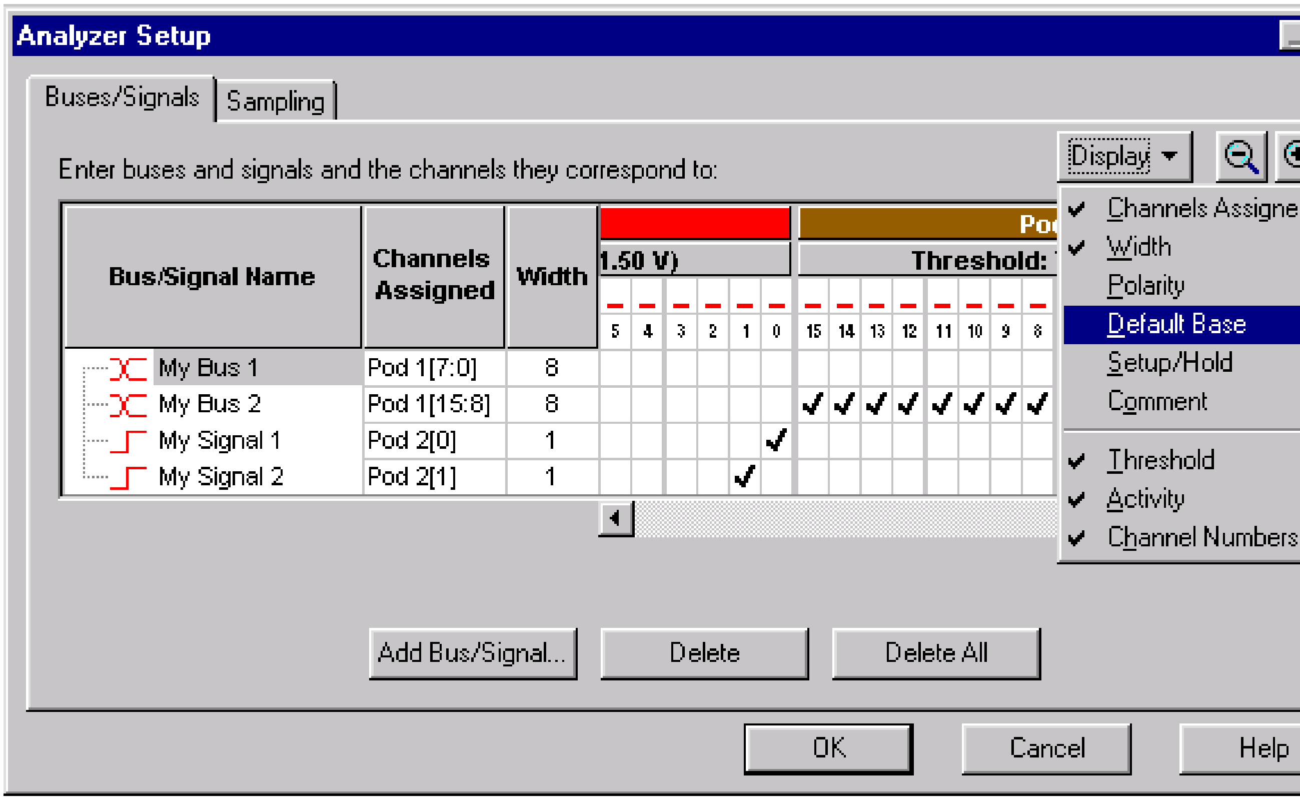


Tip: If clock channels are not connected to clock signals, they can be used as extra data channels. Clock channels are grouped together after the last pod in the channel assignment area.

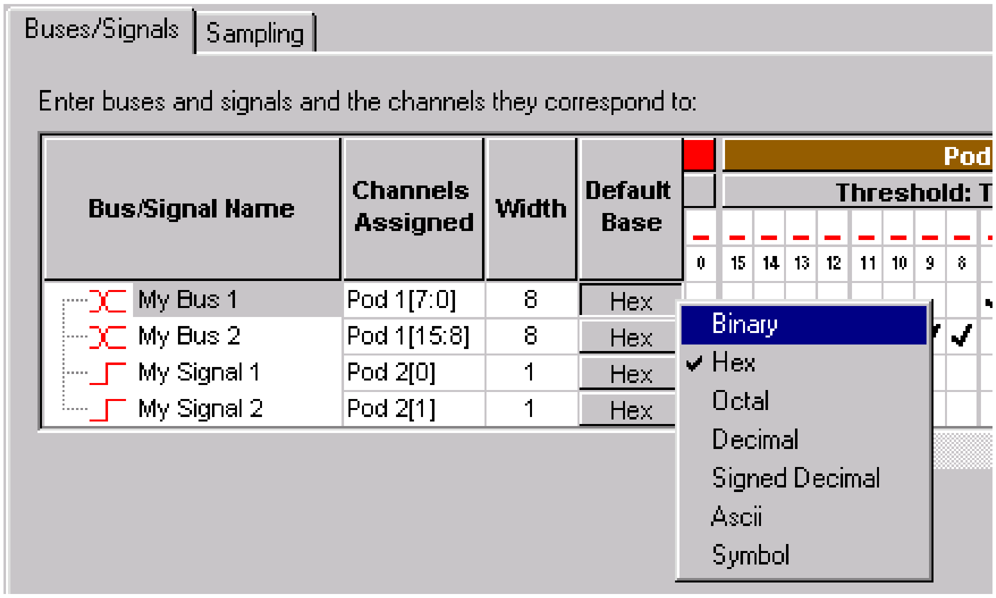
To set numeric base

You can set the default numeric base for a bus when you create the bus. The default base is used to display bus and signal values in the listing and waveform views. Default base only affects new buses and signals; if you change default base for an existing bus or signal you will not see a change unless you add a new copy of the bus or signal to a listing or waveform view.

1. From the menu bar, select Setup>Bus/Signal.
2. In the bus/signal setup dialog, select Display.
3. Select Default Base.



4. To change the default base for a bus or signal, click the default base value.



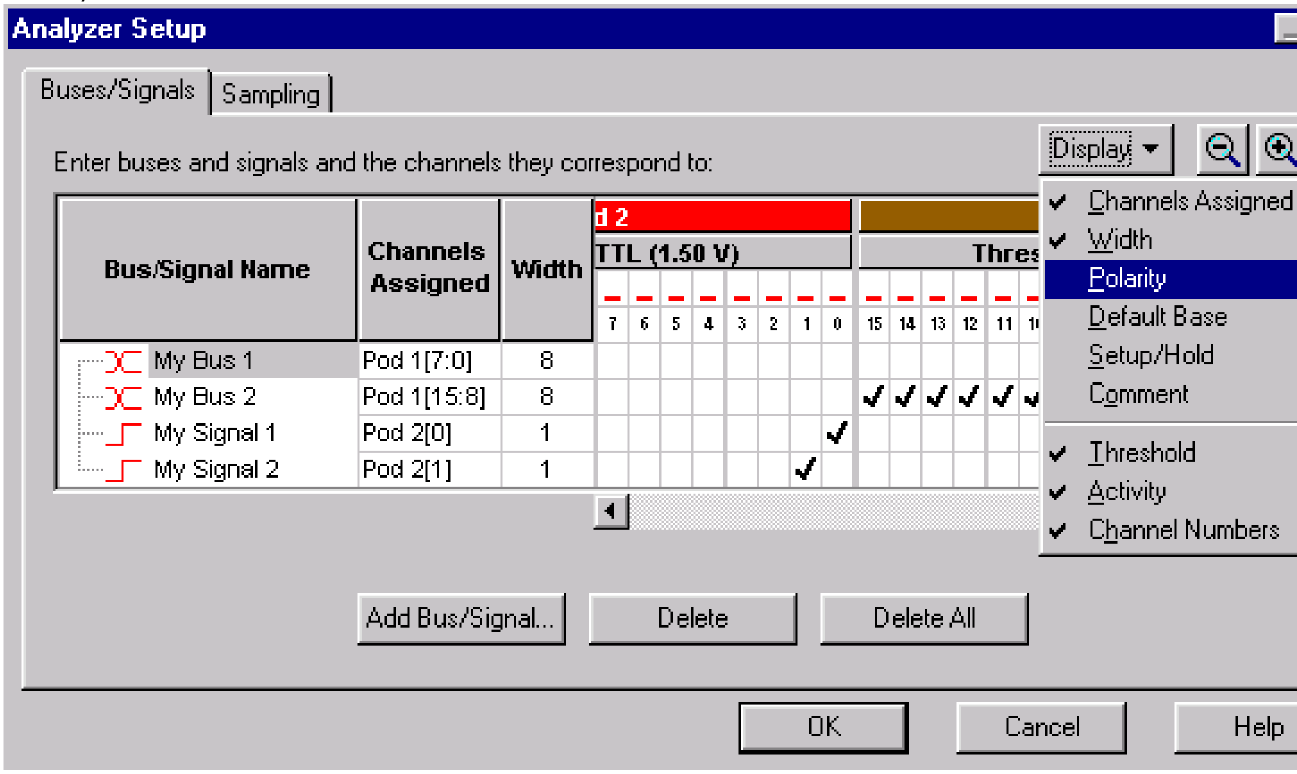
5. Select a new value.
6. Click OK to close the bus/signal dialog.

To set polarity

You can define buses and signals to display with negative or positive polarity. This affects the display of values and waveforms. When a bus or signal is set to negative polarity, an incoming high voltage will be shown with a low waveform and a logical value of 0. The polarity is reflected in all places that use values, such as trigger and symbols.

The default polarity is positive (high = 1).

1. From the menu bar, select Setup>Bus/Signal.
2. In the bus/signal setup dialog, select Display.
3. Select
   Polarity.



4. In the polarity column that appears, toggle between + (positive) and (negative).

To set setup/hold

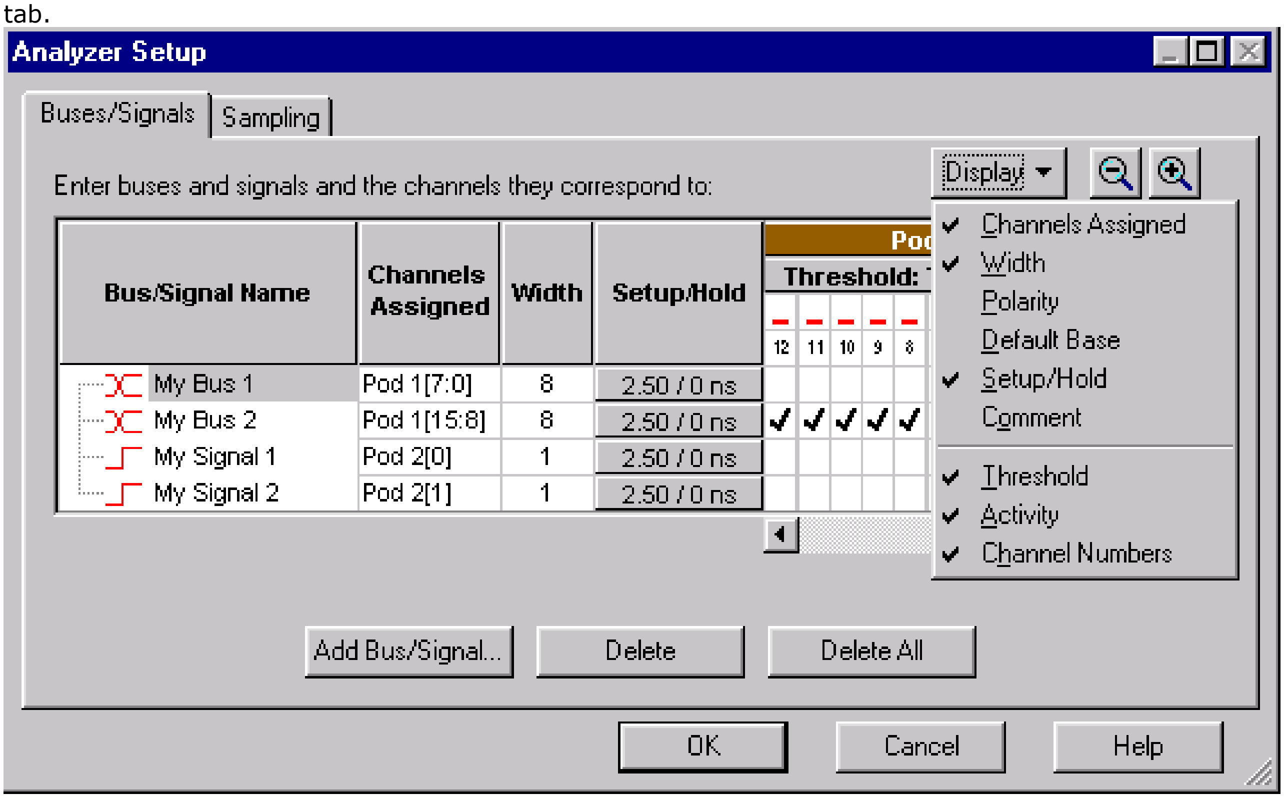
Setup and hold is available in State mode only.

Setup/Hold specifies where the logic analyzer's window should be relative to the clock signal it receives from the target system. The target system needs a window in which the data is valid that is at least as long as the logic analyzer's setup/hold window plus any clock offset. Use the setup/hold setting to move the logic analyzer's setup/hold window relative to the target system's data valid window.

1. Select Setup>Timing/State (Sampling)...
2. Change Acquisition to State - Synchronous Sampling.
3. Set up the clock description.

Note: Single-edge clocks have a smaller setup/hold window than other clocks. Changing to a different setup/hold window changes the setup value. (Hold remains constant.)

4. Select Buses/Signals tab.



5. If the Setup/Hold column is not already displayed, select Display>Setup/Hold.
6. If you are using a slave or demultiplexed clock, change the pod clock type for appropriate buses and signals. You change the pod clock type by selecting Master Clock in the pod column of the assigned channels.
7. Click the bus or signal's Setup/Hold value to adjust it. You can also adjust individual bits, in which case the field shows Individual.

If the data valid window on the target system does not include the time when the clock signal transition, use negative values for the setup or hold. For a single-edge clock, the valid setup/hold window is adjustable from 4.5 ns setup/-2.0 ns hold to -2.0 ns setup/4.5 ns hold in 100 ps increments. For multiple-edge clocks, the valid setup/hold window is adjustable from 5.0 ns setup/-2.0 ns hold to -1.5 ns setup/4.5 ns hold in 100 ps increments.

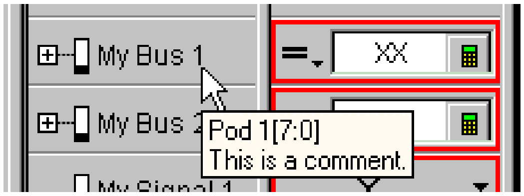
Note: Setup time cannot be changed independently of hold time. The total setup/hold window is a constant, but can be adjusted relative to your data valid window.

See Also

To select the state clock type

To set the state clock qualifier

To add user comments



You can attach comments to buses and signals. The comments show up in the tool tip when you hover the mouse over a bus or signal name in both the waveform and listing windows.

1. From the menu bar, select Setup>Bus/Signal.
2. In the Buses/Signals setup tab that appears, select Display.
3. Select Comment. A new column labelled Comment appears.
4. In the Comment column, type your comment for the bus or signal.
5. Click OK to close the Analyzer Setup dialog.

Note: Comments are intended as a descriptor to embellish a bus/signal name and not as a notepad. Comments can be up to 64 character in length.

To add a folder

The Add Folder... feature adds a windows style folder to the bus/signal list. Use folders to help organize bus and signal names when using many bus/signal names with inverse assemblers.

1. From the menu bar, select **Setup>Bus/Signal**.
2. Right-click on a bus/signal name, then select **Add Folder.**
3. The new folder appears directly below the highlighted name. By default, the new folder has a system generated default name. If desired, rename the new folder in the same way you would a bus/signal name.

See Also

To alias a bus/signal name

To alias a bus/signal name

The Add Alias... feature adds an exact duplicate bus or signal name (same channel, polarity, etc. assignments). Use alias names along with folders to help organize the many bus and signal names with inverse assembly.

1. From the menu bar, select **Setup>Bus/Signal**.
2. Right-click on the desired bus/signal name, then select **Add Alias.**
3. The new alias name appears directly below the highlighted name. The new alias name can be renamed, however, the new name will also be applied to the original name.

See Also

To add a folder

To sort bus/signal names

You can sort bus/signal names and folder names to help organize them.

1. From the menu bar, select **Setup>Bus/Signal...**.
2. Right-click on one of the bus/signal or folder names to be sorted; then, select either**Sort>Ascending** or **Sort>Descending**.

See Also

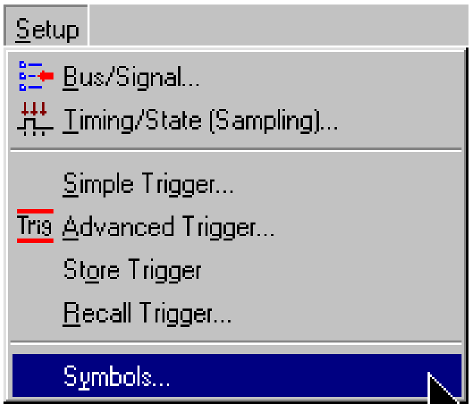
To add a folder

To edit symbols for a bus/signal

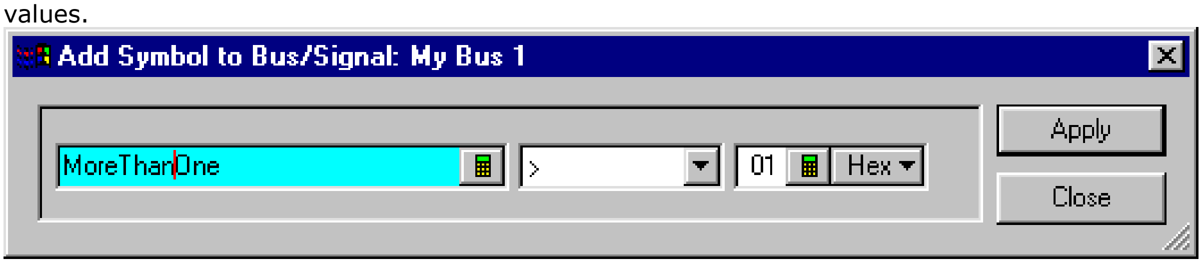
You can edit symbols whenever the Edit symbols dialog is displayed.

To add a symbol

1. Select Setup>Symbols....



2. Select the bus or signal for which the new symbol should be displayed.
   Each symbol is defined for a particular bus/signal.
3. Click Add.
4. Define a value or range of values.



   There are no restrictions on the characters you can use in the name of a symbol. The symbols on a particular bus or signal must have unique names.
5. Click Apply.

To see the symbols in the listing or waveform display, click OK in the Symbols dialog and change the base for the bus/signal to Symbols.

Note: Because XML format logic analyzer configuration files now save and load user-defined symbols, you can also add symbols by (1) using text processing tools to re-format symbol information from software development tools, (2) inserting them into an XML format configuration file, and (3) loading the configuration file into the *Agilent Logic Analyzer* application (see XML Format).

To edit a symbol

1. Select Setup>Symbols....
2. Select the symbol you want to edit.
3. Click Edit.

To delete a symbol

1. Select Setup>Symbols....
2. Select the symbol you want to delete.
3. Click Delete.

To save symbols

Save symbols as part of a configuration file. Symbols are saved in the configuration whether or not you select Setup only in the Save As dialog.

See Also

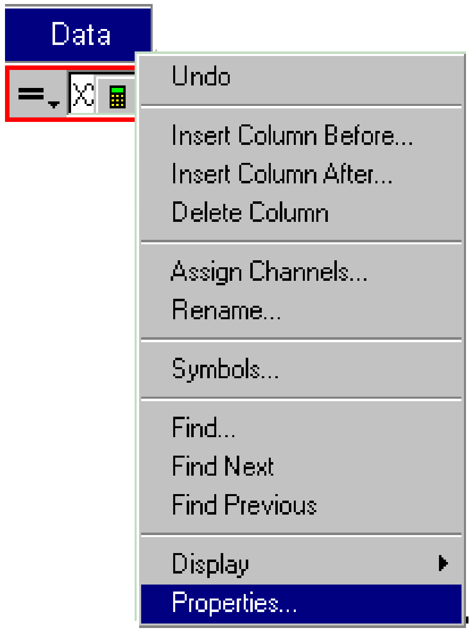
To display symbols

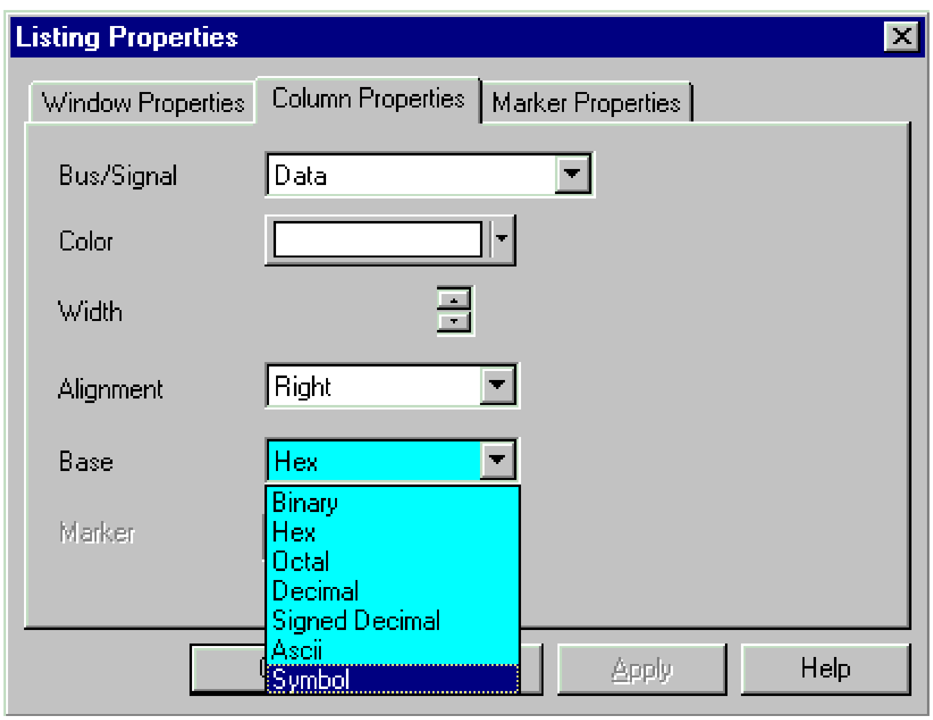
To display symbols

You can display a bus or signal using meaningful names rather than numeric values. Symbols can be displayed in both the Listing and Waveform displays.

To display symbols:

1. Change the numeric base of the bus or signal to Symbols.





2. Define the symbols.

If the symbol is defined as a range, values in the range will be displayed with an offset from the lowest end of the range.

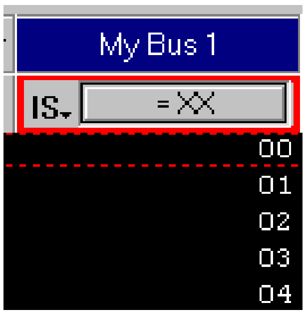
If the definitions of several symbols overlap, the first one listed in the Symbols dialog has precedence over the others.

In the Waveform display, ">>" will be shown when the full name of the symbol will not fit into the space available.

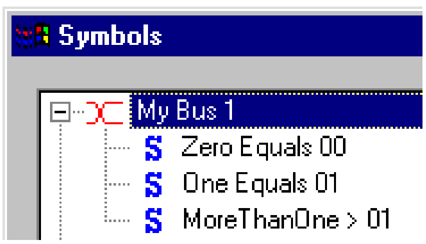
Once you have set up symbols, it's usually a good idea to save the logic analyzer configuration. The symbol definitions will be stored as part of the configuration.

Example

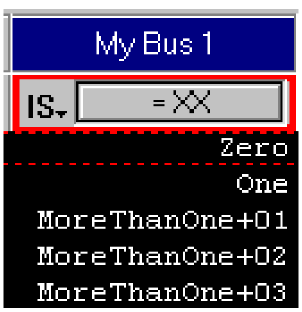
Here is what "My Bus 1" looks like before defining any symbols:



When the symbols have been defined, they are shown in the Symbols dialog:

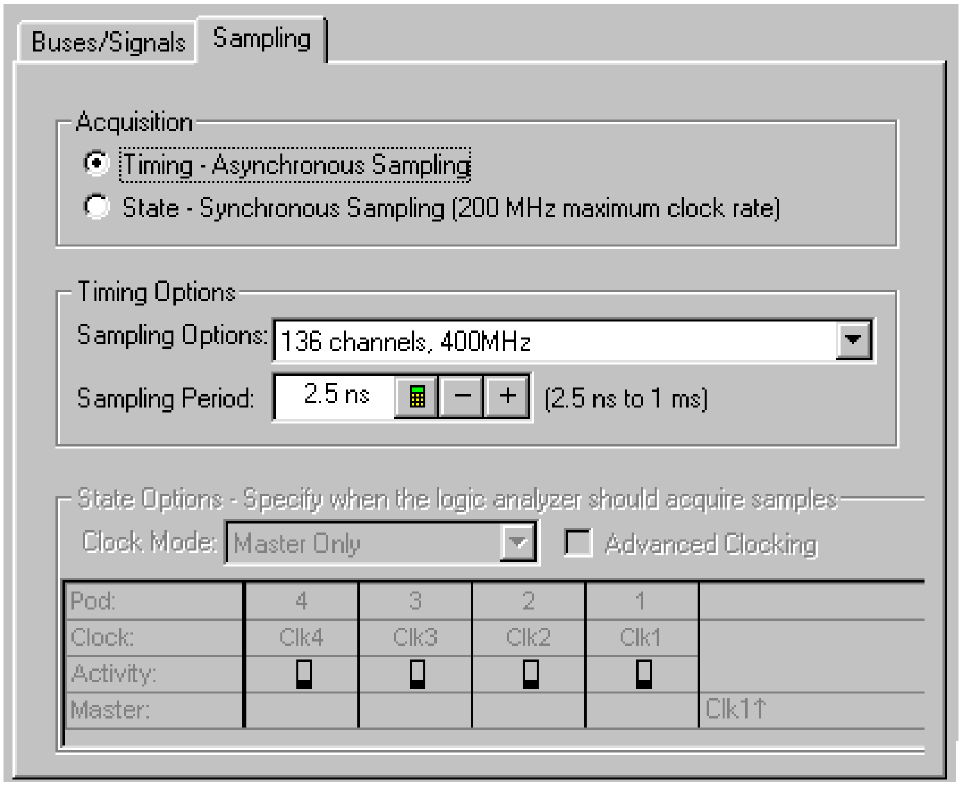


Here is what the bus looks like after the symbols are defined:



Selecting the Sampling Mode

Sampling Setup



The Sampling tab is access through the menu bar's Setup>Timing/State (Sampling). The Sampling setup tab is used to select and configure the acquisition mode.

In the Timing - Asynchronous Sampling mode, you set the sampling period and channel width. You also set the acquisition (memory) depth and the trigger position. Timing mode samples the target system at regular intervals (the sampling period).

In the State - Synchronous Sampling mode, you configure the clocking type and any desired clock qualifier. You also set the acquisition (memory) depth and the trigger position. State mode samples the target system when a signal matching the defined clock occurs.

The following tasks are performed in the Sampling setup tab.

To set the acquisition mode

To set acquisition depth

To set the trigger position

State - Synchronous Sampling mode

To select the state clock type

To set the state clock qualifier

To set up advanced clocking

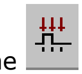
Timing - Asynchronous Sampling mode

To set the sampling period

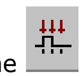
To set the sampling options

To set the acquisition mode

To select the timing analyzer

1. From the menu bar select **Setup>Timing/State (Sampling)**, or click the  icon from the setup toolbar.
2. In the Sampling setup dialog, select the **Timing - Asynchronous Sampling** option.

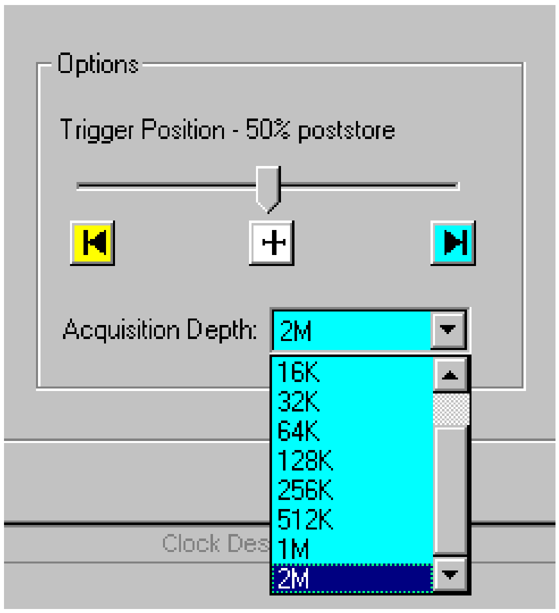
To select the state analyzer

1. From the menu bar select **Setup>Timing/State (Sampling)**, or click the  icon from the setup toolbar.
2. In the Sampling setup dialog, select the **State - Synchronous Sampling** option.

To set acquisition depth

The acquisition depth control allows the user to set the amount of memory that is filled with data on an acquisition. The choices available depend on the maximum memory depth available in the analyzer that is being used.

1. From the menu bar select Setup>Timing/State (Sampling).
2. Set the acquisition mode and any state or timing options. These will affect the available memory choices.
3. In the Options box to the right, set Acquisition Depth.



See Also

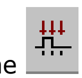
To set the trigger position

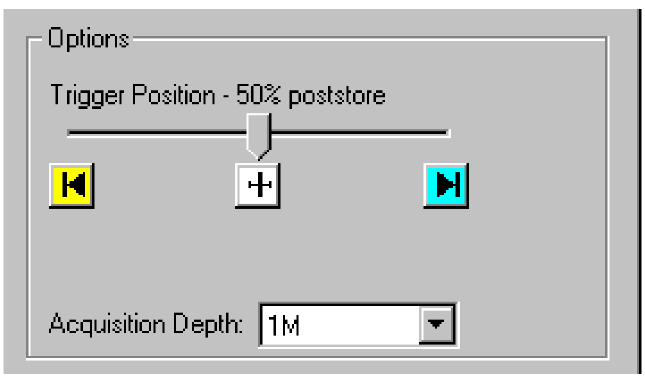
To set the trigger position

The trigger position specifies the amount of trace memory used for samples captured before the trigger. For example when 10% is selected, 90% of trace memory is used for samples captured after the trigger. When 90% is selected, 10 % of trace memory is used for samples captured before the trigger.

In timing mode, the amount of pre-trigger and post-trigger memory will always be what you expect based on your settings, even if your trigger condition occurs before pre-trigger memory is filled. This is happens because in timing mode, the trigger sequencer does not start until pre-trigger memory is full.

However, this is not true for state mode. In state mode, the sequencer starts immediately, in other words, it doesn't wait for prestore memory to be filled. Thus, in state mode, your trigger position may not be where you expect. For example, if you set the trigger position to 50%, but you find the trigger right away, the amount of pre-trigger memory will be less than what you expect.

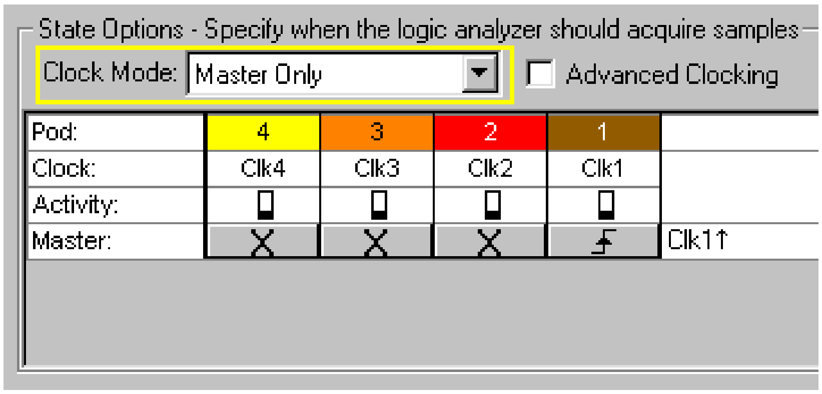
1. From the menu bar select **Setup>Timing/State (Sampling)**, or click the  icon from the setup toolbar.
2. Select the trigger position.



See Also

To set the acquisition depth

To select the state clock type



By default the clock mode is set to Master. If you want to set the clock mode to Master/Slave or Demultiplex, follow the procedure below.

Note: To probe demultiplexed data, use only one pod of a pod pair.

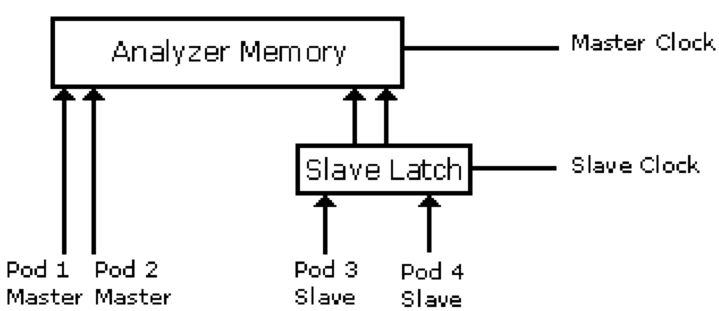
1. From the menu bar, select Setup>State/Timing (Sampling).
2. Select State.
3. In the State Options, change Clock Mode to Master/Slave/Demux.
4. Set up your master and slave clocks.
5. Select the Buses/Signals tab.
6. Select the Clock field under the pod heading.
7. For Demultiplex mode: Set the clock to Demultiplex. The display of the pod and its neighbor changes. For example, if you set Pod 1 to demultiplex, Pod 2 goes away and you see two Pod 1 columns. The first Pod 1 column is labelled Pod 1 (Master Clock) and the second column is Pod 1 (Slave Clock). For Master/Slave mode: Change pod clock fields to Slave as necessary.
8. Assign channels to buses.

Master

In the Master only sampling clock mode, there is one sampling clock signal. When a clock edge occurs, data is captured and saved into one sample of logic analyzer memory. Two additional sampling clock modes let you capture data differently:

Master/Slave

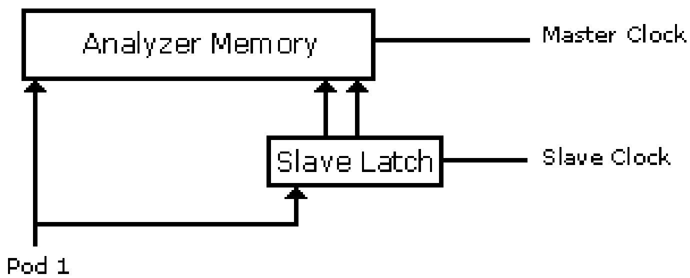
In the Master/Slave mode, you can save data captured on different clock edges into the same sample of logic analyzer memory.



When the slave clock occurs, data captured on the pods that use the slave clock is saved in a slave latch. Then, when the master clock occurs, data captured on the pods that use the master clock, as well as the slave latch data, are saved into logic analyzer memory. If multiple slave clocks occur before the next master clock, only the most recently acquired slave data is saved into logic analyzer memory.

Demultiplex

In the Demultiplex mode, you can demultiplex data being probed by one pod into the logic analyzer memory that is normally used for two pods.   Demultiplex mode uses the master and slave clocks to demultiplex the data.



When the slave clock occurs, data captured on the pod is saved into the slave latch for the other pod in the pod pair. Then, when the master clock occurs, data captured on the pod, as well as the slave latch data, are saved in logic analyzer memory. As with master/slave mode, if multiple slave clocks occur before the next master clock, only the most recently acquired slave data is saved into logic analyzer memory.

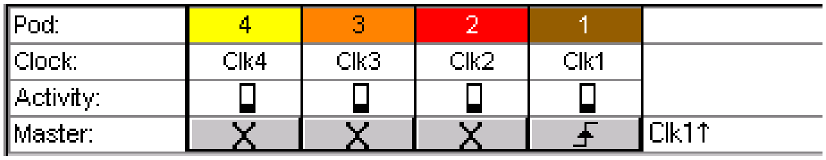
See Also

To set the state clock qualifier

To set up advanced clocking

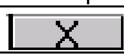
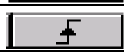
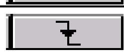
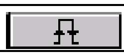
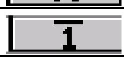
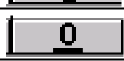
To set the state clock qualifier

The state clock should be set to match the clock signal on your target system. The logic analyzer can handle clock signals comprised of up to four lines. Clocks can be as simple as a single rising edge, or a complicated combination of edges and highs or lows.



In the picture above, the clocks refer to the clock signal lines on pods 1 through 4. Depending on model, your logic analyzer may have more pods. However, only the clock lines on the first 4 pods can be used to generate the state clock signal. Clock lines on extra pods can be used like normal data lines.

1. Attach logic analyzer pods to your target system. Clock signals must be connected to the clock lines on pods 1 through 4.
2. From the menu bar, select Setup>Timing/State (Sampling).
3. Select State - Synchronous Sampling. The State Options area becomes active.
4. Select the state clock type. Most measurements use only the master clock.
5. Set up a clock description to match the clock(s) on your target system.

	Don't care. Clock line not used in this clock.
	Rising edge.
	Falling edge.
	Either edge.
	Qualifier - high.
	Qualifier - low.

A clock description must have at least one edge.

See Also

To select the state clock type

To set setup/hold

To setup advanced clocking

To set up advanced clocking

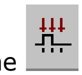
The Advanced Clocking dialog lets you specify more complex clock setups than you can with the normal Master or Slave selections. If you want to use a specific clock channel both as an edge and a qualifier in the same clock description, you need to use advanced clocking.

1. From the menu bar, select Setup>State/Timing (Sampling).
2. Select State mode. The state options become selectable.
3. Next to the clock mode, select Advanced Clocking. The clock controls are replaced by a button.
4. Select Master Clock... or Slave Clock... as appropriate. The Advanced Clocking Setup dialog appears.
5. Choose settings as appropriate. Clock channels can be used both as primary clocking and as clock qualifiers.
6. Click OK to close the dialog. The clock description in the Analyzer Setup window updates.

Note: If you un-check advanced clocking, the clock settings are preserved except that all qualifiers are erased.

To set the sampling period

In timing mode, a logic analyzer takes a sample of the target system's activity once per sample period. You can set this sample period in the Sampling Setup tab. When running in 800 MHz mode, the sample period is not changeable.

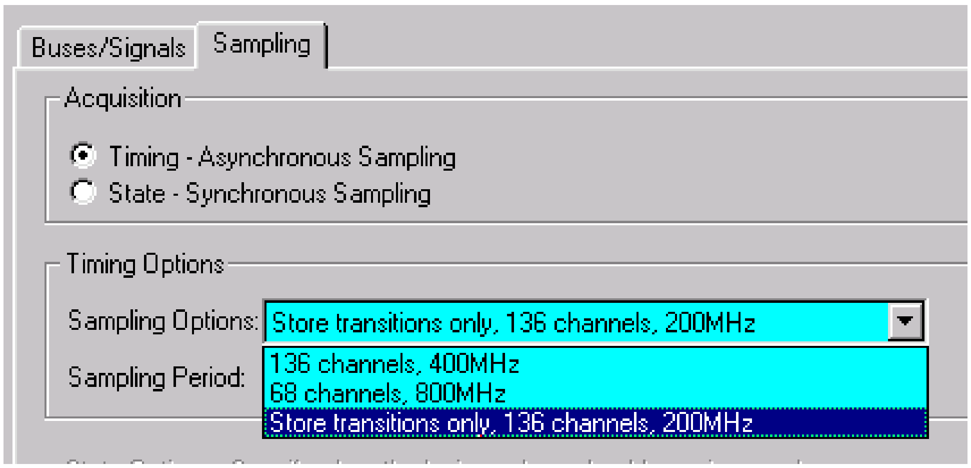
1. From the menu bar select **Setup>Timing/State (Sampling)**, or click the  icon from the setup toolbar.
2. Select Timing - Asynchronous Sampling.
3. In the Timing Mode area of the Sampling setup dialog, increase or decrease the Sample Period.

Note: To capture signal level changes reliably, the sample period should be less than half of the period of the fastest signal you want to measure. Time interval measurements are made by counting the number of samples in the desired waveform area. These measurements are made to a +/- one sample error, so measurement accuracy is improved if the number of samples is maximized.

To set sampling options

In Timing (asynchronous) mode, you can trade off channel width for increased sample time. That is, if you want a faster sampling period, you can choose the option of using only half of the maximum channels available for your analyzer model. The channel count will vary according to the analyzer model you have.

- 1680A/90A - 136 channels
- 1681A/91A - 102 channels
- 1682A/92A - 68 channels
- 1683A/93A - 34 channels



1. From the menu bar, select Setup>State/Timing (Sampling).
2. Select Timing acquisition mode. Timing Options becomes selectable.
3. Select the sampling option you prefer. Your channel count may be different depending on the logic analyzer model.

   Note: Changing the sampling option will affect the sampling period and may affect bus assignments.

136 channels, 400 MHz	Default. All channels are available; the sampling period can be set from 2.5 ns to 1 ms.
68 channels, 800 MHz	Uses one pod from each pod pair. The sampling period is always 1.25 ns.
Store transitions only, 136 channels, 200 MHz	Provides maximum duration of acquisition because data is only stored when a change from the last value is detected. Sampling period ranges from 5.0 ns to 1 ms. See transitional timing.

Note: When you select the timing sampling mode's 400 MHz option (800 MHz option), the trigger marker in captured data may be off by 1 sample (3 samples). This occurs because the logic analyzer hardware uses 2 pipelines (4 pipelines). When triggering on a pattern, the actual sample that causes the trigger may be 1 sample before (within 3 samples before) the trigger marker. When triggering on an edge, the actual sample that causes the trigger may be within () of the trigger marker.

See Also

Transitional timing

Transitional timing

In the Store transitions only mode, the timing analyzer samples data at regular intervals, but only stores data when there is a threshold level transition. Each time a level transition occurs on any of the bits, data on all channels is stored. A time tag is stored with each stored data sample so the measurement can be reconstructed and displayed later.

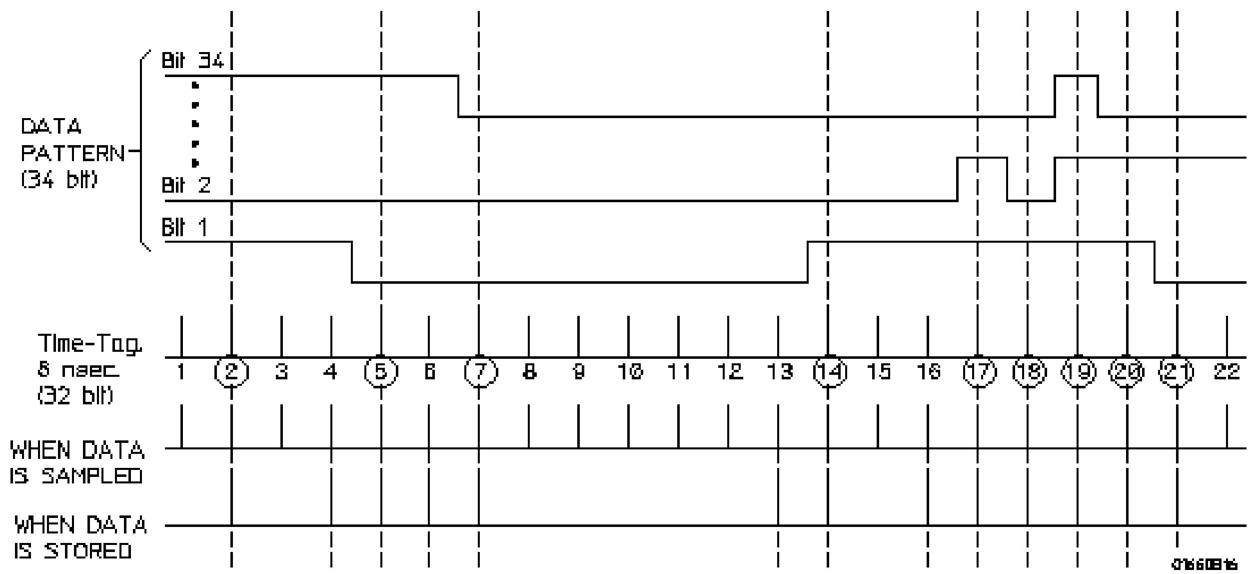
More on storing transitions

Minimum transitions stored

Normally, transitions do not occur at each sample point. This is illustrated below with time-tags 2, 5, 7, and 14. When transitions do occur, two samples are stored for every transition. Therefore, with 2 K samples of memory, 1 K of transitions are stored. You must subtract one, which is necessary for a starting point, for a minimum of 1023 stored transitions.

Maximum transitions stored

If transitions occur at a fast rate, such that there is a transition at each sample point, only one sample is stored for each transition as shown by time tags 17 through 21 below. If this continues for the entire trace, the number of transitions stored is 2 K samples. Again, you must subtract the starting point sample, which then yields a maximum of 2047 stored transitions.



In most cases a transitional timing trace is stored by a mixture of the minimum and maximum cases. Therefore, in this example the actual number of transitions stored will be between 1023 and 2047.

Transitional timing considerations

Data storage

When an edge is detected, two samples are stored across all channels assigned to the timing analyzer. The need of two samples is to avoid loss of data if a second edge were to occur to soon after the first edge for the edge detectors to reset.

Sequence level branching

In transitional timing, only 2 branches are available per sequence level.

Global counters

In transitional timing, only one global counter is available.

Storing Time Tags

Transitional timing requires time tags to recreate the data. Time tags are stored by interleaving them with measurement data in memory.

Increasing Duration of Storage

The analyzer looks for transitions on all bits assigned to a bus. Therefor, to increase usable memory depth and acquisition time, remove bits with transitions on signals like clock or strobe that add little useful information to the measurement when no other signals transition.

Invalid Data

The analyzer only looks for transitions on data lines on buses that are turned on. Data lines on buses that are turned off store data, but only when one of the lines that is turned on transitions. If the data line on a bus is turned on after a run, or the data line is assigned to a new bus, you would see data, but it is unlikely that every transition that occurred

Trigger Position

In transitional timing, no data prestore (samples acquired before trigger) is required. Therefore, much like state mode, the trigger position (start/center/end) will indicate the percentage of memory filled with samples after the trigger. The number of samples acquired/ displayed before trigger will vary between measurements.

Capturing Data from the Device Under Test

A logic analyzer is used to view timing relationships among many signals, or if you need to trigger on patterns of logic highs and lows. A logic analyzer is actually two analyzers in one: a timing analyzer and a state analyzer. A timing analyzer is a logic analyzer that samples based on an internal clock signal. This kind of sampling is also known as asynchronous sampling. A state analyzer is a logic analyzer that samples based on a clock signal (or signals) from the device under test. Typically, the signal used to set up sampling is a state machine or microprocessor clock signal. This type of sampling is also known as synchronous sampling.

How to set up logic analyzer measurements.

- To set the acquisition mode - Before viewing a signal, you must select either a timing analyzer or a state analyzer. A timing analyzer samples based on an internal clock, and a state analyzer samples based on a clock signal from the device under test.
- To set the sampling period - The timing analyzer works by sampling the input waveforms to determine whether they are high or low. The state analyzer has to restrict the sampling of data to times when only the desired data is valid and appears on the signal lines.
- To start/stop measurements - There are two types of run modes; the single run measurement will save captured data to trace memory one time. The run repetitive measurement will save the captured data to trace memory repetitively. Stopping a run determines how much data is captured and placed in trace memory.
- Trigger position - The trigger position specifies the amount of trace memory used for samples captured before the trigger.
- Triggering the logic analyzer at the correct time is important because it allows you to store only the data you want to see and analyze. The more accurate the trigger point, the more precise the captured data set will be surrounding the malfunction in the target system.

  In many measurements the trigger point can be as simple as an occurrence of a data pattern on a bus, or a rising edge on a clock line. In more advanced measurements, the trigger point would occur only after a series of bus patterns, edges, and qualifying time periods.

  Depending on the measurement, the logic analyzer can be triggered in the following ways:

  - Simple Trigger - The Simple Trigger allows you to quickly select a qualifier to compare the incoming bus or signal with.
  - Advanced Trigger - The Advanced Trigger is used to configure triggers that require multiple conditions to be true before a trigger occurs.
  - External Trigger.

See Also

Trigger Functions

Triggering

Triggering the logic analyzer at the correct time is important because it allows you to store only the data you want to see and analyze. The more accurate the trigger point, the more precise the captured data set will be surrounding the malfunction in the target system.

In many measurements the trigger point can be as simple as an occurrence of a data pattern on a bus, or a rising edge on a clock line. In more advanced measurements, the trigger point would occur only after a series of bus patterns, edges, and qualifying time periods.

Depending on the measurement, the logic analyzer can be triggered in the following ways:

Simple trigger

Advanced trigger dialog

External trigger

See Also

Trigger Functions

Set quick trigger with rectangle

Within the waveform and listing displays, you can quickly set up a simple trigger by drawing a rectangle with the mouse. After a simple trigger has been defined, and the analyzer is run, the trigger is stored and can be recalled at any time.

The following guidelines explain how each display interprets the drawn rectangle in regards to the data and how the trigger is set.

General guidelines:

- Any bus/signals with overlapping bits are not included within the trigger specification.

Example: Bus_1 has channels 0 through 7 of pod 1 assigned and Bus_2 has channels 3 through 6 of pod 1 assigned. At this point, you have the same probed signals (channels 3 through 6 of pod 1) assigned in both Bus_1 and Bus_2. Now you draw the rectangle over both bus_1 and bus_2. Since Bus_1 channels 3 through 6 are repeated (overlapped) on Bus_2, they will not be included in the trigger specification.

- Only a single sequence level can be defined by a drawn rectangle.
- As you draw the rectangle, a tooltip is displayed showing the current trigger specification that would be set.

Specific guidelines to the listing display:

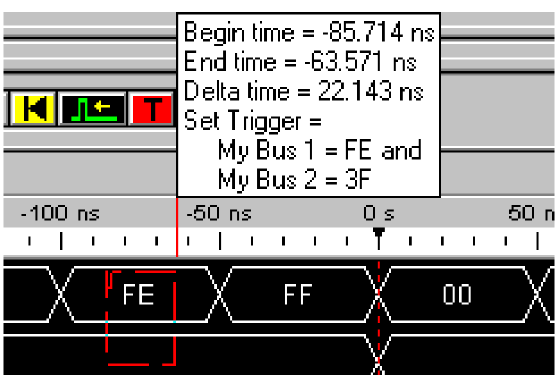
- As you move the mouse left-to-right and top-to-bottom, the signal level or bus value in contact with the top of the rectangle becomes the trigger.

Specific guidelines to the waveform display:

- As you move the mouse left-to-right and top-to-bottom, the signal edge/level or bus value in contact with the left of the rectangle becomes the trigger.
- Only one edge can be set.
- If a bus is expanded into its separate signals, three condition apply: 1. If drawing starts on a bus, none of its expanded signals can be included. 2. If drawing starts on a signal, the bus cannot be included. 3. Edges and levels are mutually exclusive. That is, either one edge can be set, or all levels can be set, but not both at the same time.

To draw the rectangle

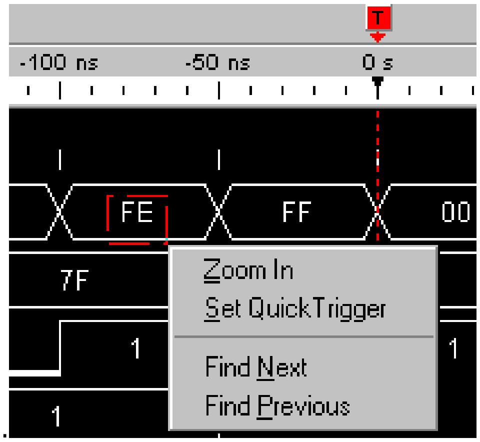
1. Using the mouse, point to the upper-left corner of your desired trigger rectangle.
2. While holding down the mouse button, drag the mouse pointer to the lower-right corner of your desired rectangle, then release the mouse button. As you draw the rectangle, you can monitor the trigger as it is set with the tooltip readout that appears.



Note: In the waveform display, it may be necessary to redraw the rectangle if you do not get your desired trigger points dictated by the left-side line of the rectangle.

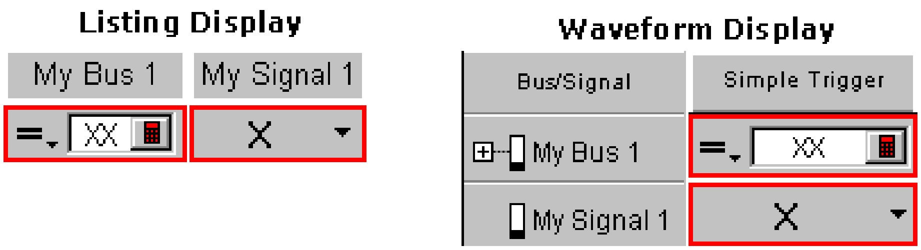
You could also try drawing the rectangle backwards leaving the left-side rectangle line set last.

3. Select Set Trigger.



Specifying Simple Triggers

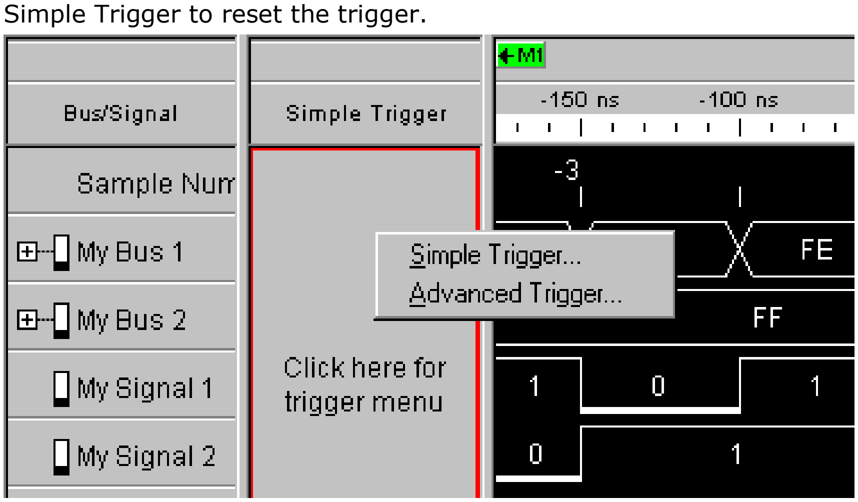
Simple Trigger



The Simple Trigger allows you to quickly create simple triggers like edges and bus patterns within the waveform or listing window. Bus data is compared to a relational operator (example; equal to, not equal to) and a user-defined value. Single signals are compared to options such as a rising or falling edge, or a low or high level. In the state acquisition mode, edges are not available. When a bus pattern and/or signal match the user defined expression, the analyzer triggers.

When both a bus and signal are used in an expression, they are ANDed together with the restriction that only one edge can be set. In cases where multiple edges are set, the last edge set has priority, while all others are changed to don't cares (X).

When a trigger condition requires more than a simple AND/OR expression, such as using multiple sequence steps, or the use of timers and counters, you can choose to use the advance trigger dialog. When the advanced trigger dialog is used, and a trigger expression is configured that surpassed the functional limits of the simple trigger, the simple trigger fields go away. To restore the operation of the simple trigger, you must either change the existing advanced trigger configuration to use bus patterns and edges within the scope of the simple trigger, or, click in the simple trigger field, then choose Simple Trigger to reset the trigger.



Up to the point where the advanced trigger surpasses the functionality limits of the simple trigger, both the simple trigger fields and the advanced trigger dialog will show the current trigger configuration and are available for trigger modification. However, as mentioned above, as soon as changes in the advanced trigger dialog surpass the simple trigger functional limits, the simple trigger fields go away.

In the case where a signal and bus overlap, that is, any signal that is actually part of the group of signals making up the bus, the last change has highest priority. For example, a bus pattern has been set to trigger the analyzer, you then set an overlapping signal (channel 1 of same bus) to trigger on a rising edge. By changing the signal, the bus pattern trigger is now discarded.

To set bus pattern triggers

To set signal trigger options

See Also

Set quick trigger with rectangle

To store a trigger

To recall a trigger

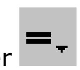
Advanced trigger dialog

Trigger Functions

To set bus pattern triggers

Bus pattern triggers are part of the Simple Trigger and are used to quickly select a qualifier to compare the incoming bus to. Bus data is compared to an operator (example; equal to, not equal to) and a user-defined value. When a bus pattern matches the defined bus pattern trigger, the analyzer triggers. A complete list of operators is listed below.

The Simple Trigger is located in both listing and waveform display windows.

1. In the Simple Trigger field, click on the pattern qualifier  field and set it to the desired operator.
2. Click in the text entry field  and enter the desired data pattern.

Operators

- Equal To
- Not Equal To
- In Range
- Not In Range
- Greater Than
- Greater Than Or Equal To
- Less Than
- Less Than Or Equal To

See Also

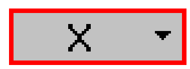
To set signal trigger options

Advanced Trigger Dialog

To set signal trigger options

Signal trigger options are part of the Simple Trigger and are used to quickly select a qualifier to compare the incoming signal to. Signals are compared to options such as a rising edge or falling edge. When the analyzer detects a match with the defined option, the analyzer triggers. A complete list of available options is shown below.

The Simple Trigger is located in both listing and waveform display windows.

1. In the Simple Trigger field, click on the signal option  field and set it to the desired option.

Options

- Rising Edge
- Falling Edge
- Either Edge
- Glitch
- High
- Low
- Don't Care

Note: In State acquisition mode, edge options are not available.

To trigger on a glitch

When Glitch is selected as the signal's trigger option, the analyzer will trigger when a glitch with a minimum width of 1.5 ns appears on the signal line.

Note: Glitches are not drawn on the screen. You need an oscilloscope to further troubleshoot glitches and find out when they occur.

See Also

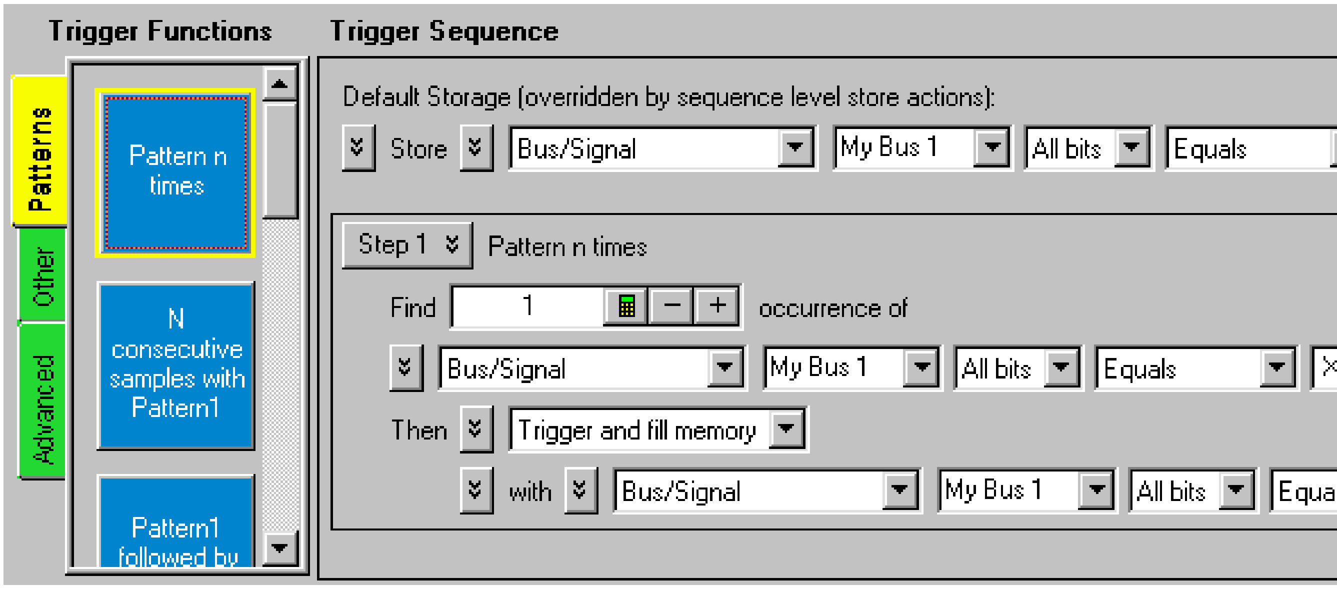
To set bus pattern triggers

Advanced Trigger Dialog

To trigger on one of multiple edges or glitches

Specifying Advanced Triggers

Advanced Trigger Dialog



The Advanced Trigger dialog is used to configure the advanced triggers that require multiple conditions to be true before a trigger occurs. Each acquisition mode (state or timing) has its own set of trigger functions. Trigger functions are represented by blue boxes, and are drag-and-dropped onto the trigger sequence display area in the order in which you want them executed.

The Advanced Trigger dialog is accessed through the menu bar Setup>Advanced Trigger... . The following tasks are performed in the dialog:

To build a trigger sequence

- To set store qualification - (State mode only)
- To insert events and actions
- To negate a function statement
- To modify trigger step display
- ANDing and ORing event statements

To store a trigger

To recall a trigger

See Also

State mode trigger functions

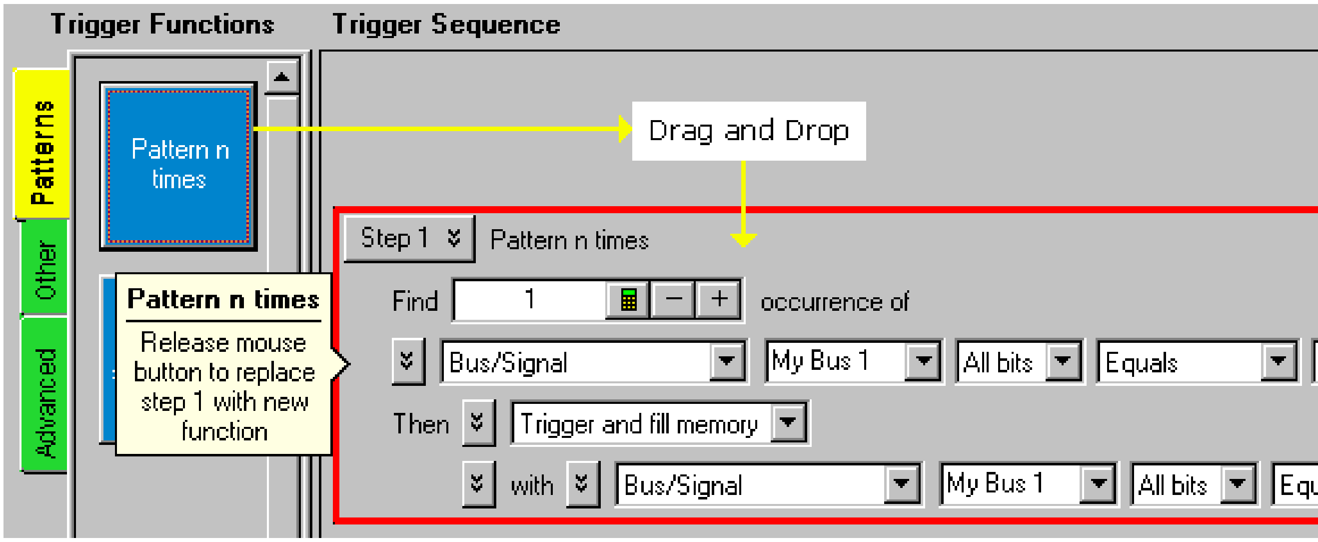
Timing mode trigger functions

Simple Trigger

To build a trigger sequence

The trigger functions included in the advanced trigger dialog give you pre-configured trigger sequences that will work for most measurements. In the case where you have to build a custom trigger sequence for your particular measurements, the following overview shows you how to build a trigger sequence.

1. From the menu bar, click Setup>Advanced Trigger... .
2. From the Advanced Trigger dialog, drag and drop the desired Trigger Function into the Trigger Sequence display area.



Replace - To replace an existing function, drag and drop the new function on top of the old one. A red box around the old function indicates the replace operation.

Add above/below - To add multiple functions, drag and drop the new functions above or below the existing function. When the mouse is positioned above or below an existing function, a red insert bar appears to indicate relative insert location of the new function.

Trigger steps

Each trigger function added to the trigger sequence displays a trigger Step. Trigger steps illustrate in sentence form how the function evaluates data to find the trigger point. The evaluation process continues within the trigger step until either the trigger is found, or an instruction is encountered that sends evaluation to another trigger step. Each trigger function added to the sequence adds another trigger step.

For custom trigger sequences, each trigger step can be modified in the following ways:

To set store qualification - (State mode only)

To insert events and actions

To negate a function statement

To modify trigger step display

See Also

To store a trigger

To recall a trigger

State mode trigger functions

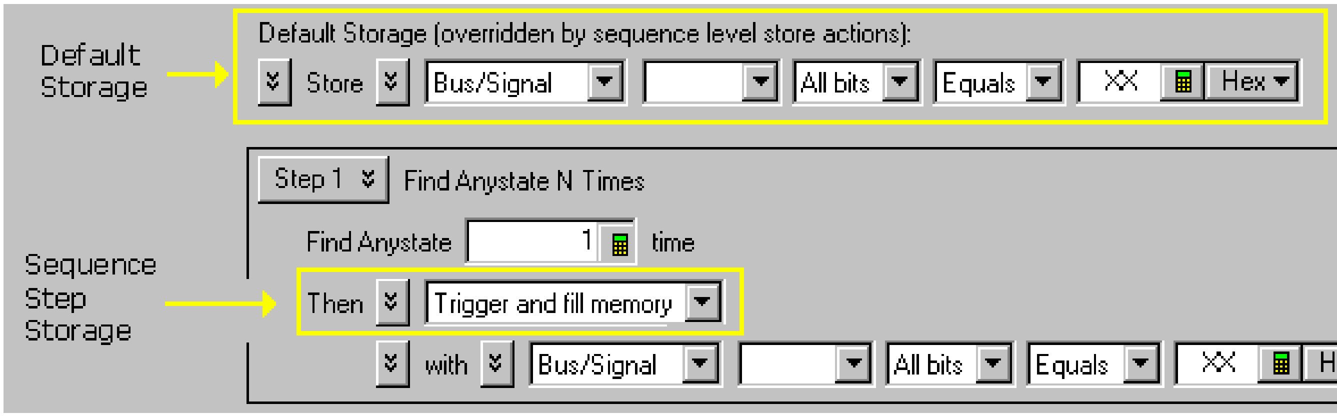
Timing mode trigger functions

To set store qualification

Storage qualification is used to determine if an acquired sample should be stored (that is, placed in memory) or thrown away. This keeps the logic analyzer memory from being filled with samples that are not needed.

Default storage

The simplest method to set up storage qualification is by setting up the Default Storage. This is specified separate from the trigger sequence as shown below.



Default storage means "unless sequence step storage specifies otherwise, this is what should be stored". Sequence step storage always overrides default storage.

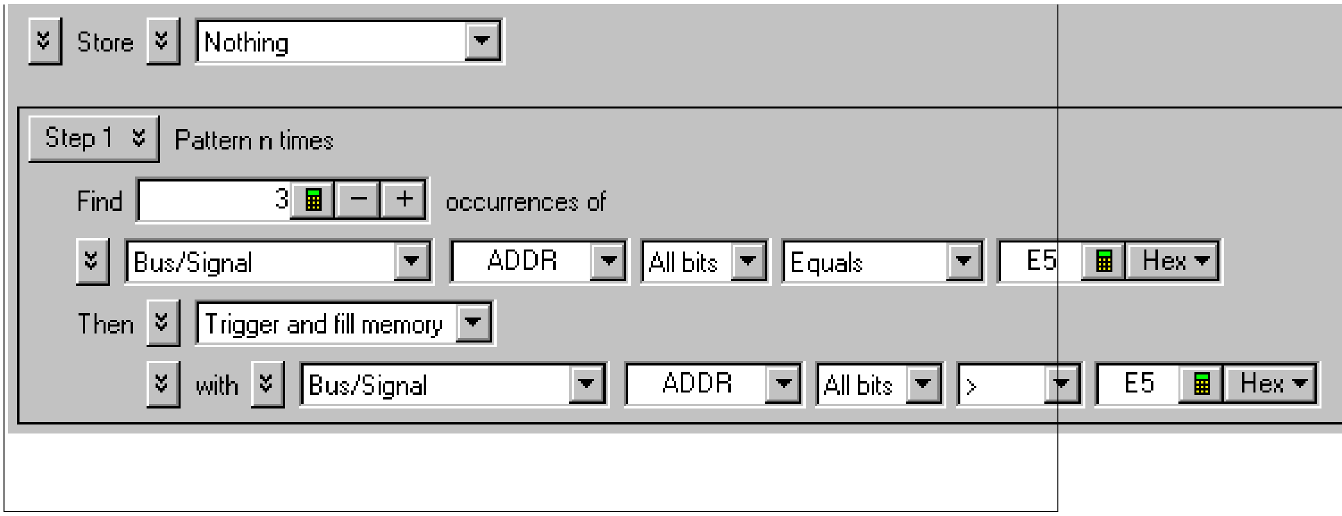
Example: Before the analyzer triggers, you only want to store samples if ADDR is greater than E5. Set the Default Storage as follows:

By default, the default storage is set to store all samples acquired. You can also set the default storage to store nothing, which means that no samples will be stored unless the sequence step storage overrides the default storage.

Sequence step storage

Sequence step storage means that within a particular sequence step only certain samples are stored. This also means that until a "Go To" or "Trigger" action is used to leave the sequence step, the sequence step storage applies. This is useful when you want different sequence step storage qualification for each sequence step.

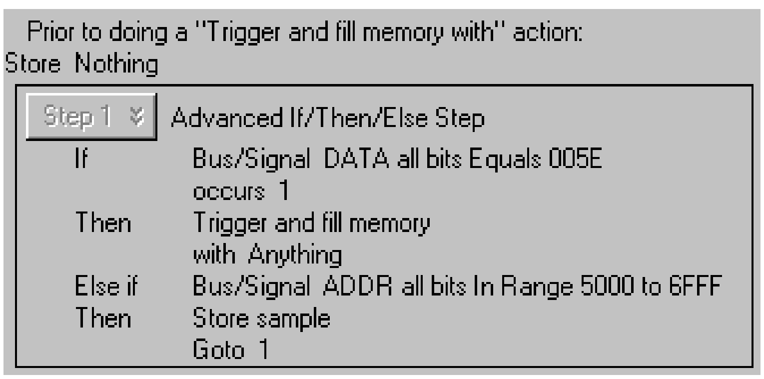
Example: You want default storage to store nothing before trigger, then you want sequence step storage to store samples when ADDR is greater than E5, starting when E5 has occurred for the 3rd time.



Storage interaction

Remember, sequence step storage always overrides default storage, but only for the conditions specifically configured in the sequence step storage. You must be very careful that you account for the interaction between default storage and sequence step storage.

For example, if you want to store only samples with ADDR in the range 5000 to 6FFF while looking for DATA = 005E, the following sequence level could be used in some situations:



Notice the use of the Store sample action. This means "store the most recently acquired sample in memory now". It does not mean, "From now on, start storing". Since the store sample action is never executed unless ADDR is in the range 5000 to 6FFF, this branch essentially means "While in this sequence step, store only samples with ADDR between 5000 and 6FFF".

The above example seems to imply that only samples with ADDR between 5000 and 6FFF will be stored. However, this depends upon how the default storage has been set up. Using the previous example, if the default storage is set to "Store Everything", and a sample is outside of the range 5000 to 6FFF, then the Else If branch is not executed and the Default Storage is applied. In essence, the sequence step has said what to do when a sample has a value in a particular range, but it doesn't say what to do for samples outside the range. Therefore, if you want to specify the sequence step storage unambiguously, use the following:

Prior to doing a "Trigger and fill memory with" action:
Store Nothing

Step 1	Advanced If/Then/Else Step
If	Bus/Signal DATA all bits Equals 005E occurs 1
Then	Trigger and fill memory with Anything
Else if	Bus/Signal ADDR all bits In Range 5000 to 6FFF
Then	Store sample Goto 1
Else if	Bus/Signal ADDR all bits Not In Range 5000 to 6FFF
Then	Don't store sample Goto 1

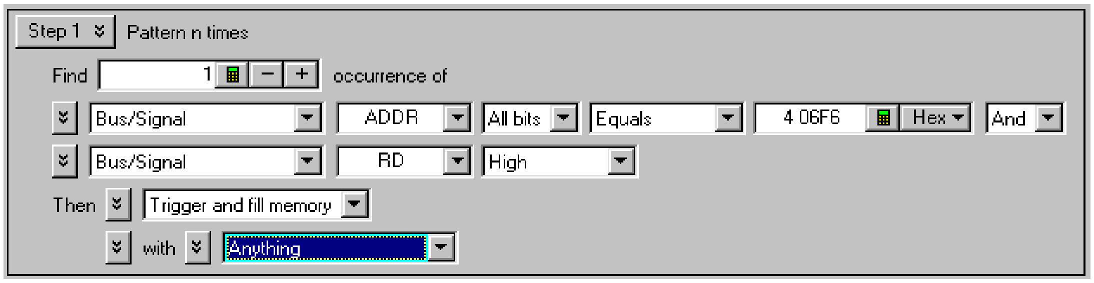
Alternatively, if the default storage is set to "Store Anything", use the following:

Prior to doing a "Trigger and fill memory with" action:
Store Anything

Step 1	Advanced If/Then/Else Step
If	Bus/Signal DATA all bits Equals 005E occurs 1
Then	Trigger and fill memory with Anything
Else if	Bus/Signal ADDR all bits Not In Range 5000 to 6FFF
Then	Don't store sample Goto 1

How to read event and action statements

Your measurement goal is to see what data is stored in memory at the address value 406F6. To do this, you configure the trigger function to look for the pattern 406F6 (hexadecimal) on the address bus, and a rising edge on the RD (memory read) clock line.

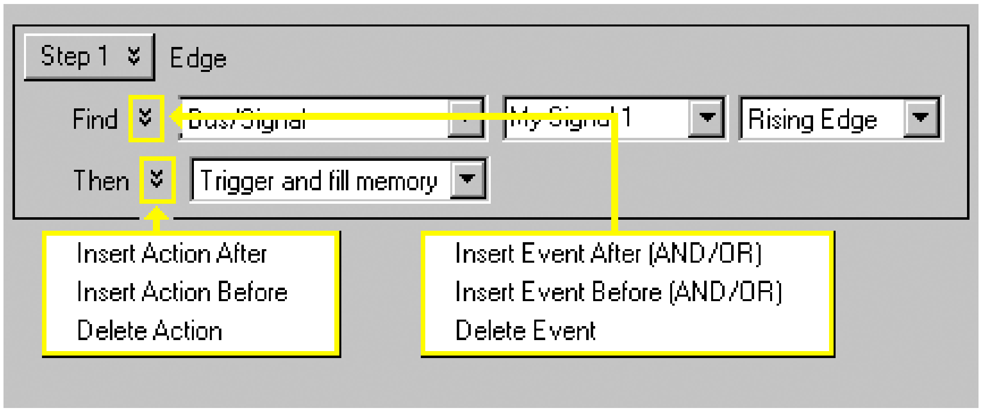


As you configure the trigger function, try to think of it as constructing a sentence that reads left-to-right.

For example:

"Find a Bus named address, and on All bits, a pattern that Equals 406F6 Hex, And a Signal named RD with a Rising Edge. When found, then Trigger and fill memory".

To insert events and actions



Events and Actions are statements, in sentence format, inserted into the trigger step to expand its functionality.

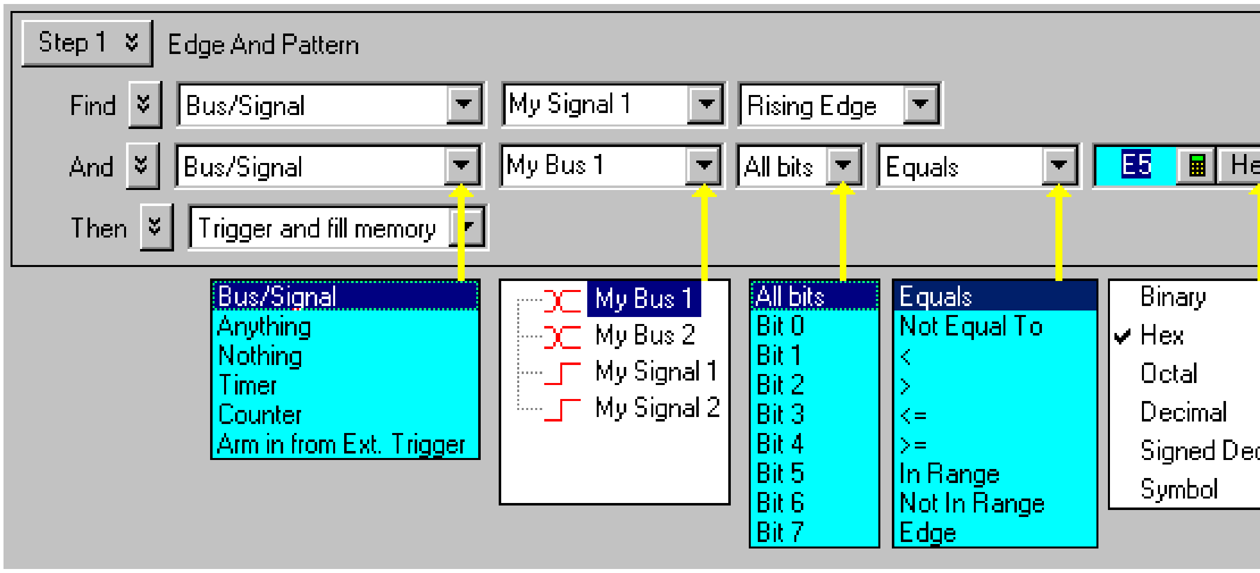
Events are used to qualify what data is being evaluated, and what data or process is being acted upon. When multiple event statements are used, you have the choice to combine them with either the AND or OR logic operators. The following Event options are available:

- Bus/Signal
- Anything
- Nothing
- Timer
- Counter
- Arm in from Ext. trigger

Actions are used to start processes such as timers or counters, or to start filling memory. When multiple action statements are used, they are combined using the AND logic operators. The following Action options are available:

- Timer
- Counter
- Store sample
- Don't store sample

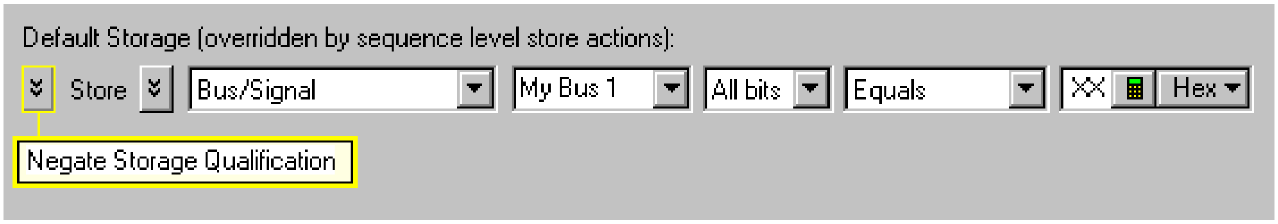
1. Within the trigger step, click the Event and Action buttons, then select insert statements Before and After as desired. If a choice is inappropriate, it is greyed out.
2. Configure the new Action or Event fields that appear. The figure below illustrates the type of selections available. Make choices according to the specific needs of your measurement. See "Example of how to read event and action statements".



See Also

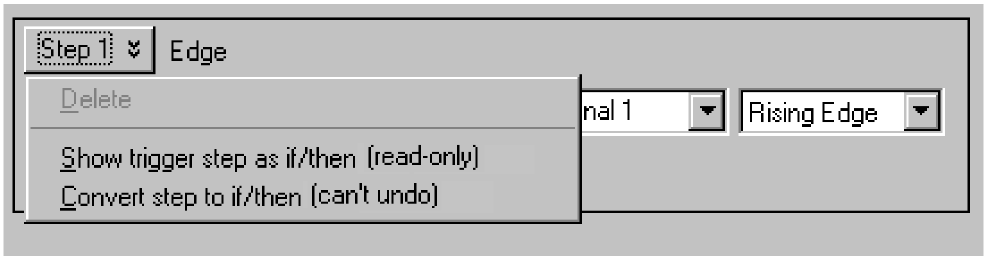
To set store qualification
ANDing and ORing Event statements
To configure a timer
To configure a counter
To store a trigger
To recall a trigger

To negate a function statement



When you negate a statement in a trigger function, you are asking the analyzer to evaluate the statement as an opposite.

To modify trigger step display



All trigger functions by default are displayed in a short form that allows you to modify them when necessary. However, the trigger step can also be displayed in two optional ways that enable you to see them without the graphical buttons and fields, or in an expanded form with graphical buttons and fields.

Note: The Advanced (If/Then, or Branch) trigger functions do not allow alternative display types. By default, they are in the expanded graphical form that cannot be changed. See the If/then (can't undo) choice below.

1. In the Trigger Sequence display area, click the  button in the trigger step.
2. Select the desired display choice.
   - Default - In most cases, the default form of the trigger function has all the necessary elements for a trigger. It appears in a short form which can hide such elements as internal occurrence counter and timer functions.
   - If/then (read-only) - This choice converts the trigger step into read only text. This choice can be switched back to the default form of configurable fields.
   - If/then (can't undo) - This choice converts the trigger step into an Advanced IF/Then form similar to the IF/Then trigger functions under the Advanced tab. This choice cannot be switched back to the default form. However, since this choice is just an expanded form of the default with configurable fields, editing the trigger step in this form is viable.

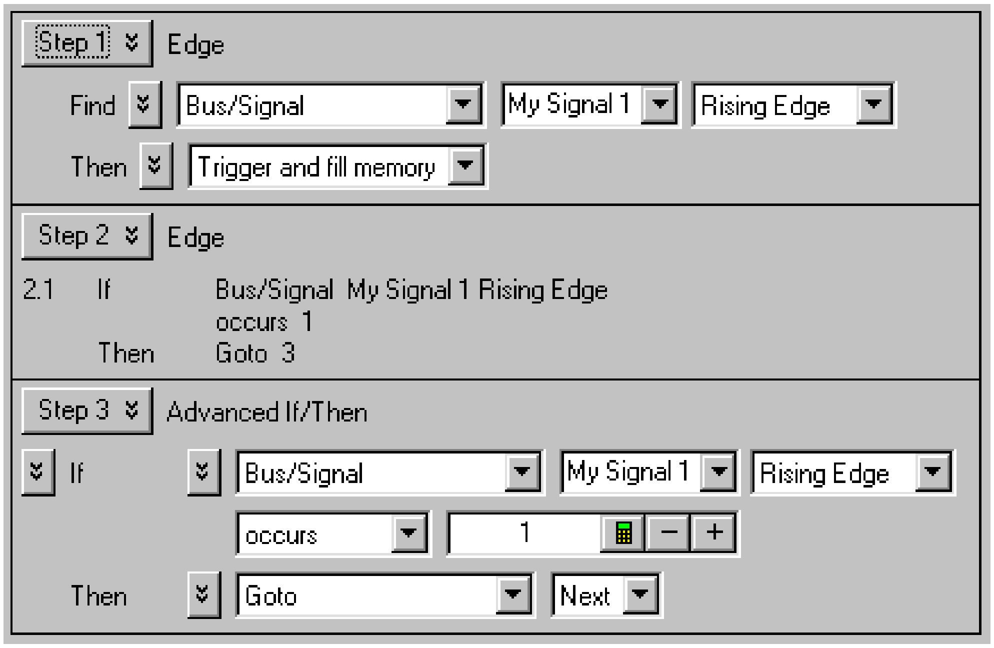
     An advantage of this form is that you have access to elements such as internal occurrence counter and timer functions. The disadvantage is that you cannot convert back to the short form of the trigger function.

The following figure shows the same Edge trigger function in all three forms.

Step 1 = Default

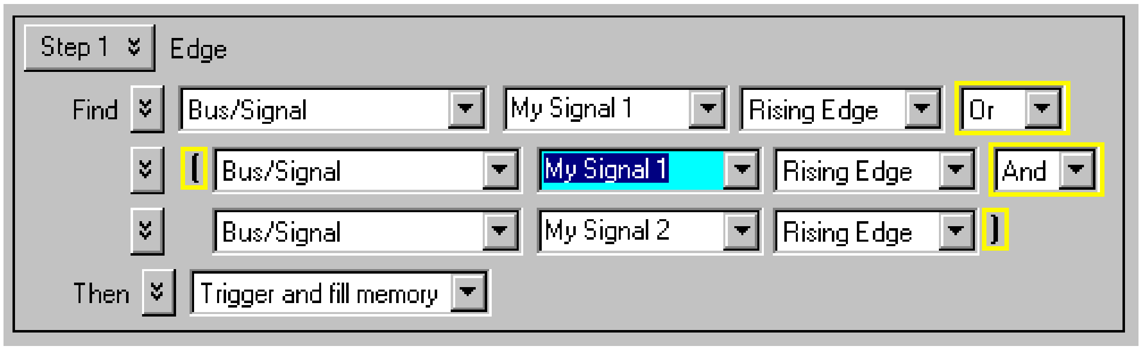
Step 2 = If/then (read only)

Step 3 = If/then (can't undo)


Step 1
Edge
Find
Bus/Signal
My Signal 1
Rising Edge
Then
Trigger and fill memory
Step 2
Edge
2.1 If Bus/Signal My Signal 1 Rising Edge
occurs 1
Then Goto 3
Step 3
Advanced If/Then
If
Bus/Signal
My Signal 1
Rising Edge
occurs
1
Then
Goto
Next

ANDing and ORing Event statements

When you add multiple event statements to form a complex trigger step, the statements are combined using the logical operators AND and OR. Parentheses are also automatically added to help show how the statement is partitioned during evaluation.



See Also

To insert event statements

To configure a counter

Counters are available in both Event and Action statements, and like other events, they evaluate to either true or false. Use counters to create a user-defined delay, or to create a standard against which valid data duration is evaluated. Once configured, a counter persists throughout all the steps of the trigger sequence.

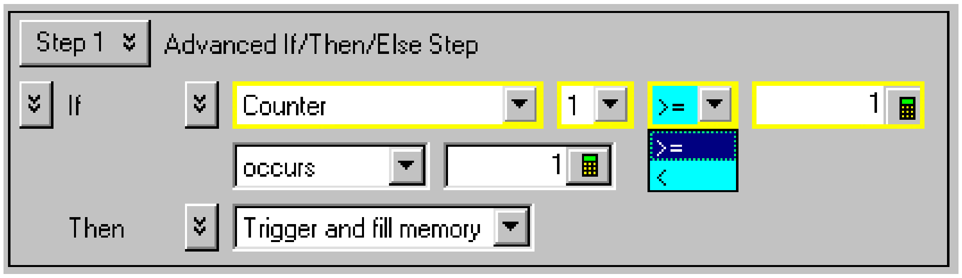
Counter considerations:

- Maximum counters available is 2.
- When using counters in the transitional timing mode, one counter is used internally so only one counter is available in a sequence step.
- Once a counter is configured, you can reuse the counter by selecting its identification number. Each use of the counter must check it for the same value.

Note: The logic analyzer also has occurrence counters, and a reset occurrence counter action. Occurrence counters only exist within steps that contain the "occurs" phrase and are not affected by the other counter actions described on this page.

To insert a counter event

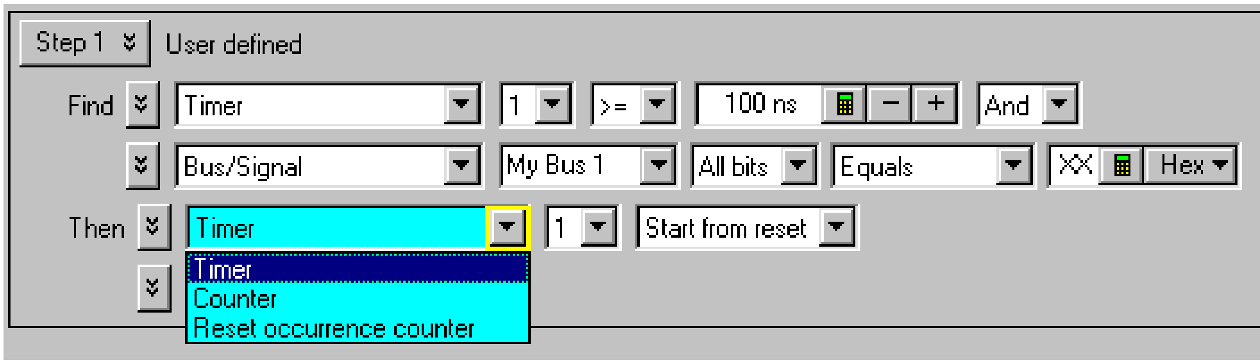
A counter must be started with a counter action before it can be evaluated with a counter event.



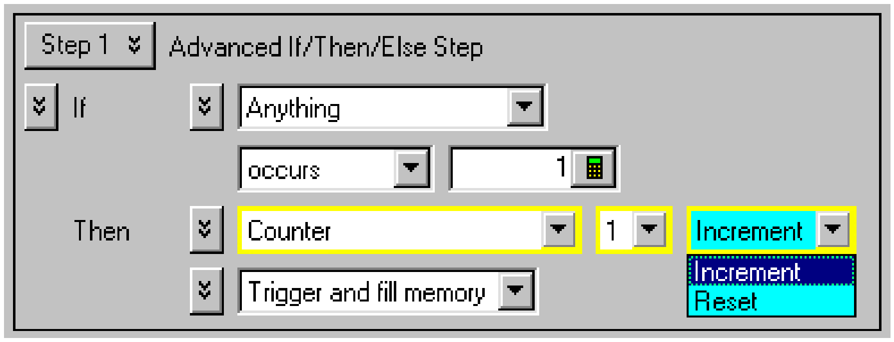
1. From within the trigger Step, select the event type button, and from the selection list that appears, select Counter.
2. Select the counter number button and choose the number of the counter you want to test.
3. Select the operator button and choose either >= or <.
4. Enter the count value.

To insert a counter action

1. From within the trigger Step, select the insert action button, then select Insert Action Before.



2. Select the action type button, and from the selection list that appears, select Counter.



3. Select the counter number button and choose the number of the counter you want to test.
4. Select the operator button and choose either Increment, or Reset.

See Also

To configure a timer

To build a trigger sequence

To modify trigger step display

To negate a function statement

To configure a timer

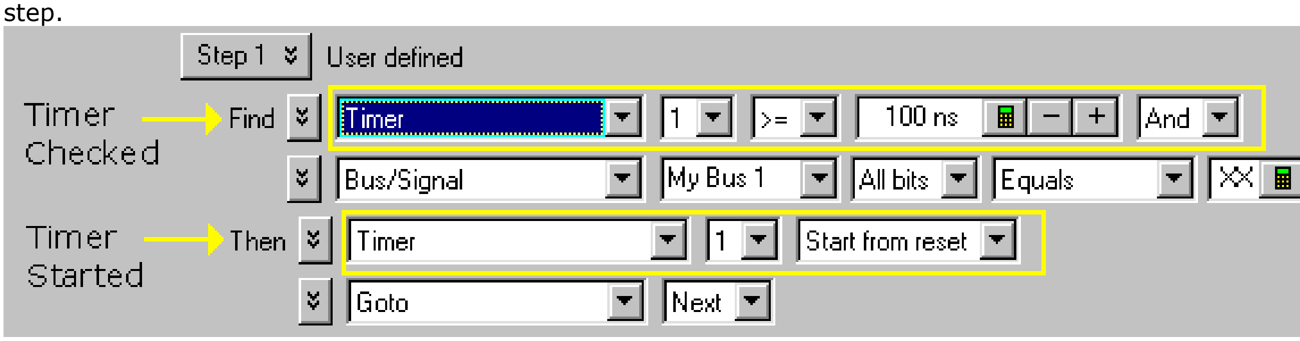
Timers are like stopwatches. Use timers to create either a user-defined delay or a time standard which valid data duration is evaluated against. The timer can Start from reset, Stop and reset, Pause, or Resume.

Timer considerations:

- Depending on the analyzer model and acquisition mode, available timers are as follows:

Model	Timing Acquisition Mode	State Acquisition Mode
1680/90	3 timers available	4 timers available
1681/91	2 timers available	3 timers available
1682/92	1 timer available	2 timers available
1683/93	0 timers available	1 timer available

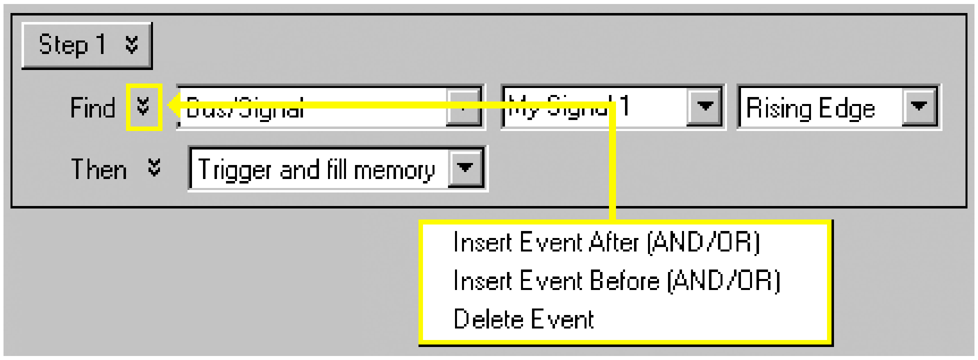
- Timers are checked in event statements, and started in action statements.
- Timers must be started before they can be checked. This is done by either including the timer start action with the timer check event within the same trigger step or starting the timer in a preceding trigger step. The following example shows the timer start action and check event within the same trigger step.



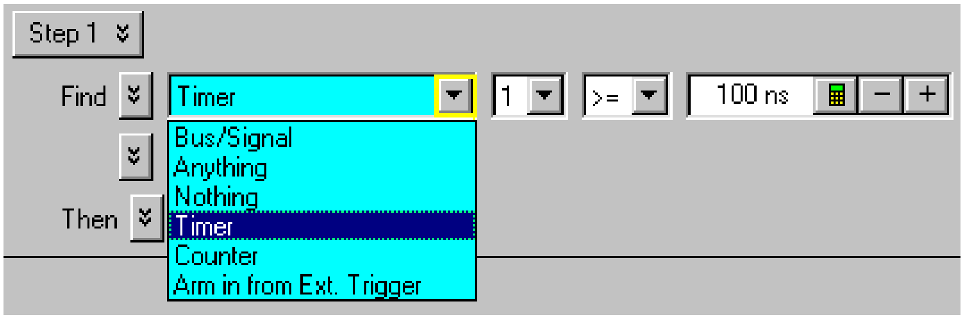
- Once a timer event is configured, you can reuse the timer by selecting its identification number. The same timer must always be checked against the same value. To check for different durations, use different timers.

To insert a timer check event

1. From within the trigger step, select the insert event button, then select Insert Event Before.



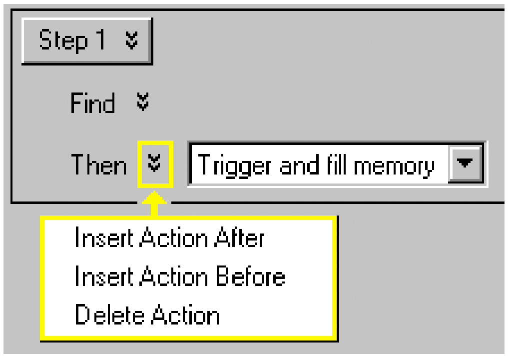
2. Select the event type button, and from the selection list that appears, select Timer.



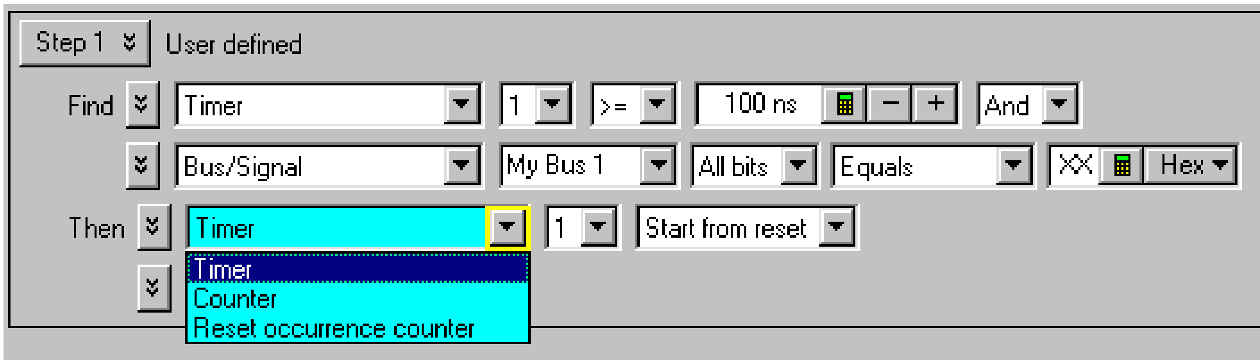
3. Select the timer identification button and choose the number of the timer you want to use.
4. Select the operator button and choose either >= or <.
5. Enter the time value.

To insert a timer start action

1. From within the trigger step, select the insert action button, then select Insert Action Before.



2. Select the action type button, and from the selection list that appears, select Timer.



3. Select the timer number button and choose the number of the timer you want to check.
4. Select the operator button and choose either Start from reset, Stop and reset, Pause, or Resume.

> Note: As stated above in timer considerations, the timer start action can be placed in either the same trigger step as the timer check event, or it can be placed in a preceding trigger step. Checking a timer without starting it will generate an error.

See Also

To configure a counter

To build a trigger sequence

To modify trigger step display

To negate a function statement

External Triggering

There are **Trigger In** and **Trigger Out** BNC connectors located on the logic analyzer (rear panel of 1680-series and front panel of 1690-series). Use them to connect the analyzer to an external instrument and either send or receive a trigger signal.

Trigger signal characteristics

Trigger out signal:

The trigger out signal is designed to drive a 50 Ohm load. It is recommended that for good signal quality, the trigger out signal be terminated in 50 Ohms to ground.

VOH (output high level) = >2.0 V.

VOL (output low level) = <0.5 V.

Pulse width = Approximately 60 ns to 140 ns.

Signal type is set in the System Options dialog. System Options is reached by**Edit>Options**.

Trigger in signal:

TTL, 5.5 V Max.

Edge type is set in the System Options dialog. System Options is reached by **Edit>Options**.

The following tasks show you how to configure the analyzer for external triggering. For an example of a complete analyzer measurement, refer to the measurement examples listed below under See Also.

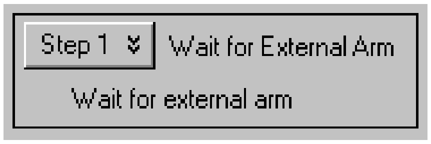
To trigger other instruments - trigger out
To trigger analyzer from another instrument - trigger in

To trigger other instruments - Trigger Out

1. Connect a BNC cable from the **Trigger Out** BNC to the external instrument you want to trigger.
2. Choose **Edit>Options**. In the Options dialog, specify whether the trigger will appear as a rising or falling edge on the**Trigger Out** BNC; then, click OK.
3. Configure the logic analyzer as you would normally for any other measurement. Refer to See Also below.
4. When the analyzer's trigger sequence becomes true and the analyzer triggers, a trigger signal is sent out through the **Trigger Out** BNC to the external instrument.

To trigger analyzer from another instrument - Trigger In

1. Connect a BNC cable from the **Trigger In** BNC to the external instrument that will send the trigger signal.
2. Choose **Edit>Options**. In the Options dialog, specify whether a rising or falling edge on the **Trigger In** BNC will indicate a trigger; then, click OK.
3. Configure the logic analyzer as you would normally for any other measurement. Refer to See Also.
4. From the menu bar select **Setup>Advanced Trigger**.
5. From the Trigger dialog, select the **Other** tab, then select the **Wait for external arm** trigger function.



6. When the logic analyzer receives the external arm signal (trigger signal), it arms and begins to evaluate its trigger sequence. When the trigger sequence becomes true, the analyzer triggers.

See Also

Making a state analyzer measurement

Making a timing analyzer measurement

Wait for external arm (state)

Wait for external arm (timing)

System Options Dialog

To store a trigger

Depending on how you want to recall stored triggers, the logic analyzer enables you to store your trigger specification in two ways.

Most recently used trigger list

Each time you set up a new trigger and run the measurement, the trigger setup is stored automatically in a "Most recently used trigger" list in the analyzer.

Note: The analyzer must be run before the trigger setup is stored. Also, trigger setups are stored as part of the configuration file. If you load a new configuration file, the trigger setups will be overwritten by trigger setups stored with the new file.

To view triggers stored automatically, click Setup>Recall Trigger, then look in the "Most recently used trigger" list. If the list gets to long, you can move them to the "Stored favorite triggers" list. The number of items allowed in the list is set by Edit>Options>Trigger History Depth.

Stored favorite triggers list

When you force a save trigger operation before the measurement is run, the trigger is placed in the "Stored favorite triggers" list.   The number of items allowed in the list is set by Edit>Options>Recent File List Size.

1. From the menu bar, click Setup>Advanced Trigger.
2. From the Advanced Trigger dialog, configure the trigger specification, then click Store.
3. Click Ok.

Optional: You can also store a trigger through the menu bar. Click Setup>Store Trigger.

See Also

To recall a trigger

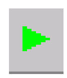
To start/stop measurements

To run the analyzer in single mode - The single run measurement will save captured data to trace memory one time. The amount of data stored during a single run is equal to the amount of trace memory allotted. For example, if trace memory is equal to 2M then the amount of data stored after one run is equal to 2M.

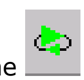
To run the analyzer in repetitive run mode - The run repetitive measurement will save the captured data to trace memory repetitively. The amount of data stored in a repetitive run is the same as a single run. During a repetitive run once the trace memory is full the system clears the trace memory and begins to refill with new data. This cycle will continue until the run is stopped.

To stop the analyzer - When a measurement is stopped the amount of data gathered is equal to the amount of trace memory used up until the stop occurred. For example, if trace memory is equal to 2M and the measurement is stopped exactly half way through the run then the amount of data in trace memory would equal 1M.

To run the analyzer in single mode

1. From the menu bar select **Run/Stop>Run**, or click the  icon from the run/stop toolbar.

To run the analyzer in repetitive run mode

1. From the menu bar select **Run/Stop>Run Repetitive**, or click the  icon from the run/stop toolbar.

To stop the analyzer

1. From the menu bar select **Run/Stop>Stop**, or click the  icon from the run/stop toolbar.

Saving Captured Data (and Logic Analyzer Setups)

You can save logic analyzer setups and captured data to configuration files. Later, the configuration files can be opened to set up the logic analyzer and re-load the data. When saving configuration files, you can choose to save only the logic analyzer setup (that is, without the data).

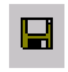
You can also save captured data to comma-separated value (CSV) files. CSV files can be imported into spreadsheet, database, or other data analysis programs.

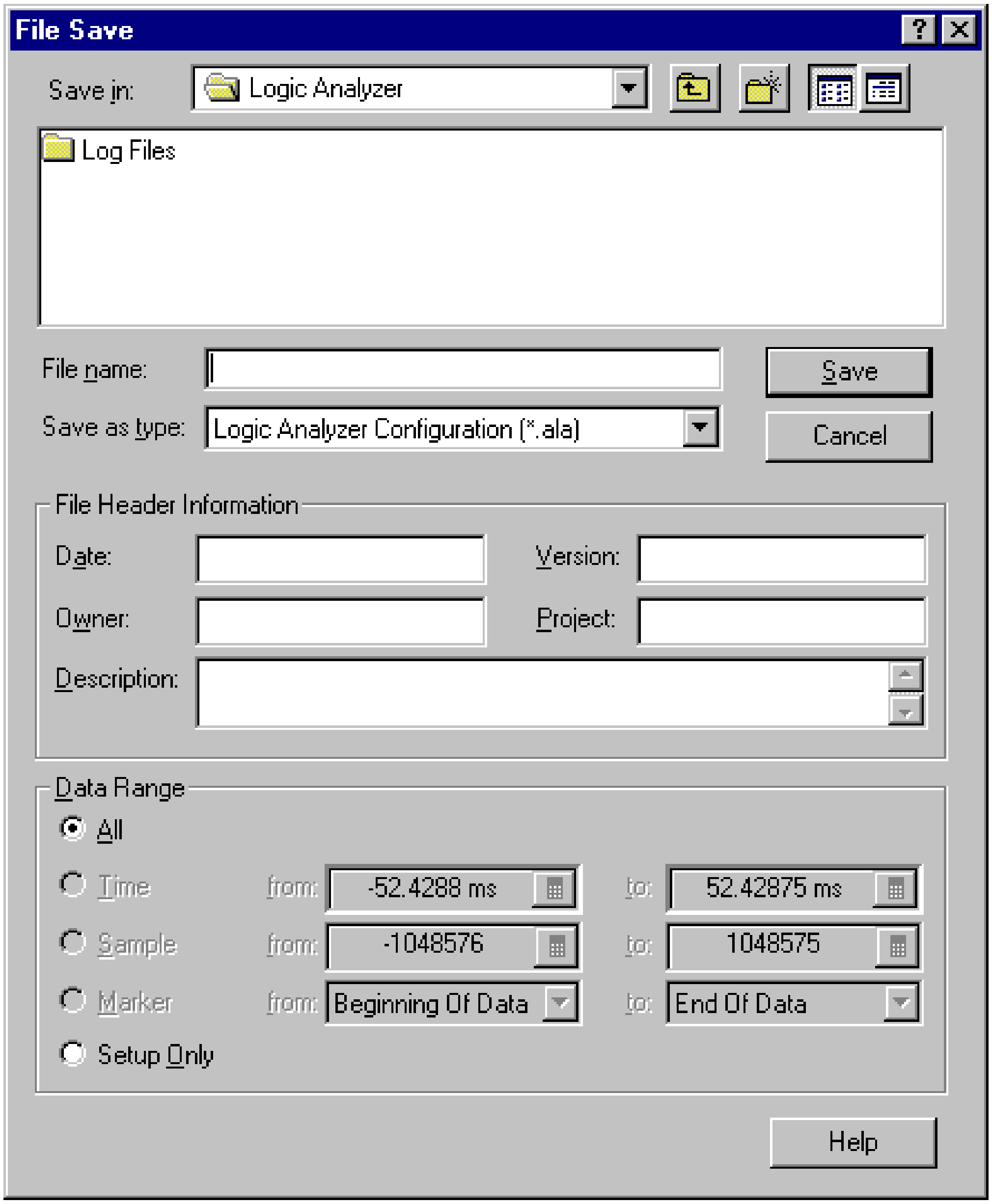
To save a configuration file

To export data to CSV format files

To save a configuration file

The save feature allows you to save a configuration file for later use. The first time a file is saved the logic analyzer configuration file dialog box will appear. The **Save As...** feature allows an existing configuration file to be saved under a different name.

1. From the menu bar, select **File>Save** or select the  icon in the standard toolbar.



2. Enter the file name of the configuration you wish to save.
3. If desired, complete the Date, Version, **Owner**, **Project**, and **Description** fields under the file header information. These fields help identify the configuration file when it is reopened.
4. Select the desired Data Range options. If you are in Offline mode, you will not be allowed to save trace data.

5. Select the **Setup only** checkbox if you wish to save only the instrument settings and not the trace data.

Note: Configuration files that include trace data will be much larger than files that do not contain trace data.

6. Select **Save**.

The **Version** and **Date** fields are automatically generated and cannot be edited. The configuration file was created with the software version shown in the **Version** dialog. The **Date** dialog displays the date in which the configuration file was created.

Note: If you are using the logic analyzer without a keyboard, you can access an on-screen keyboard by selecting **Start>Programs>Accessories>Accessibility>On-Screen Keyboard**.

See Also

Offline Analysis

ALA Format

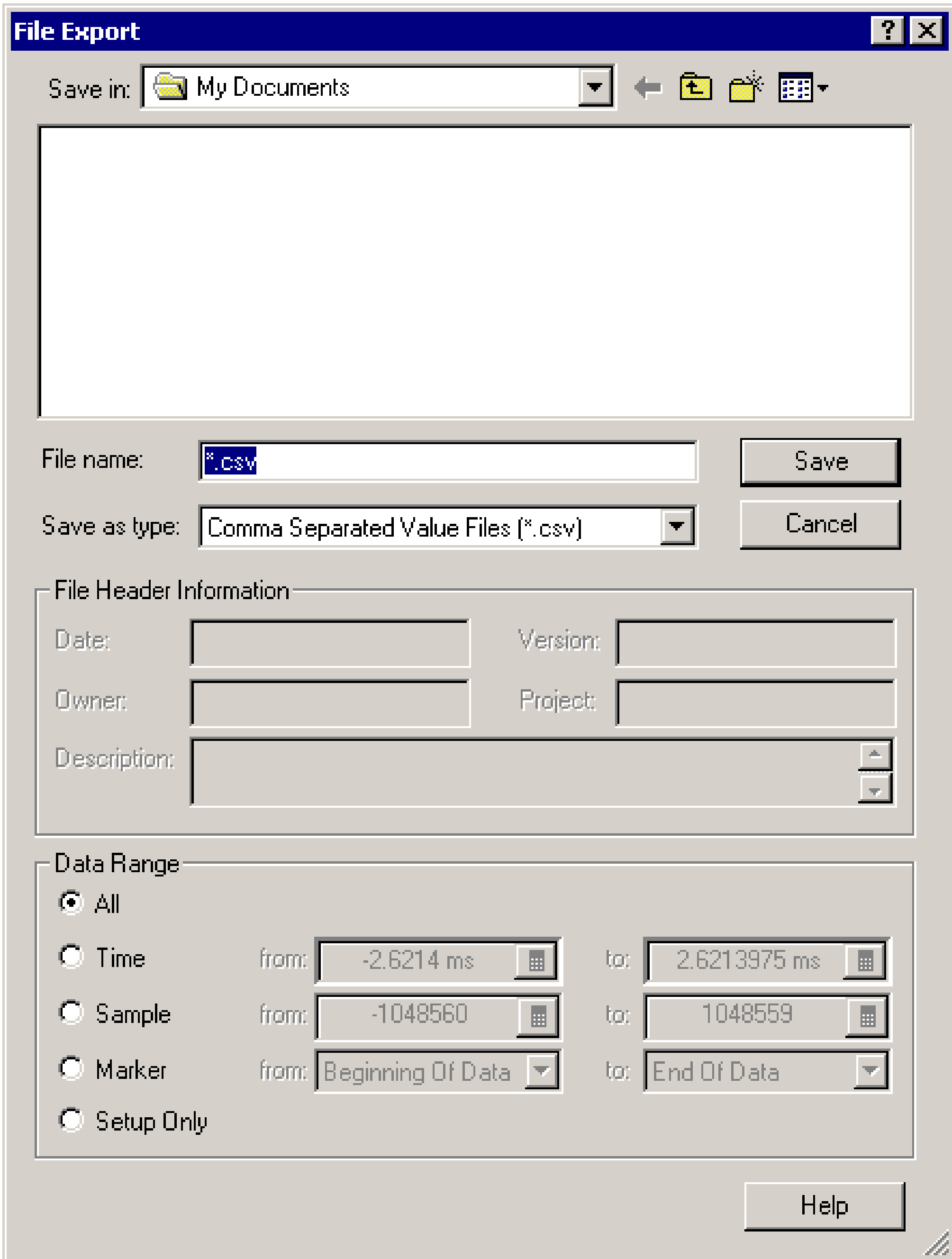
XML Format

To export data to CSV format files

You can export captured data to comma-separated value (CSV) files. CSV files can be imported into spreadsheet, database, or other data analysis programs.

Note: If you are in offline analysis mode, you cannot export data to CSV files.

1. From the menu bar, select **File>Export...**.



2. Enter the CSV file name.
3. Select the desired Data Range options.
4. Select **Save**.

See Also

CSV Format

Analyzing the Captured Data

Offline Analysis (after Loading Captured Data)
Analyzing Waveform Data
Analyzing Listing Data
Comparing Captured Data to Reference Data
Searching the Captured Data
Marking, and Measuring Between, Data Points
Displaying Names (Symbols) for Bus/Signal Values
Printing Captured Data

Loading Saved Data and Setups

You can set up the logic analyzer and load data by opening previously saved configuration files. This lets you return to the stopping point of a previous logic analysis session, load previously saved data for *offline analysis*, or just load saved logic analyzer setups. When opening configuration files that contain data, you can choose to load only the logic analyzer setup (that is, without the data).

You can import 167xx fast binary data for*offline analysis* of data captured on 16700-series logic analyzers.

To open a configuration file

To recall a recently used configuration file

To import 167xx fast binary data

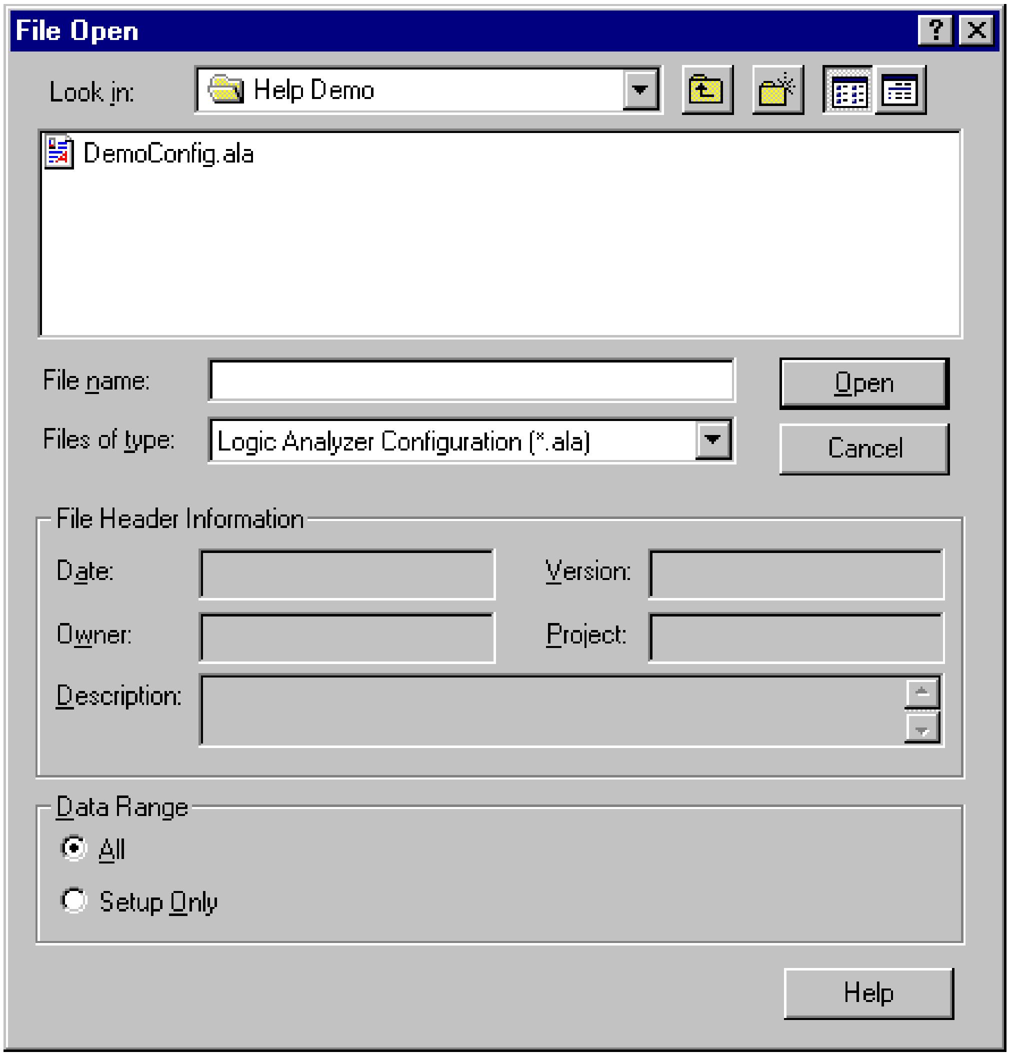
See Also

Offline Analysis

To open a configuration file

You can open configuration files to return to a previous logic analysis session, to load previously saved data for *offline analysis*, or to load saved logic analyzer setups.

1. From the menu bar, select **File>Open...** or select the  icon in the standard toolbar.



2. Select the type of configuration file you wish to open (either *.ala or *.xml).
3. Select the name of the configuration file you wish to open.

   The **Date**, **Version**, **Owner**,**Project**, and **Description** fields are used to identify configuration files. The configuration file was created with the software version shown in the **Version** dialog. The **Date** dialog displays the date the configuration file was created.
4. Select the appropriate Data Range option.

   All will load the logic analyzer setup and data. **Setup Only** will load only the logic analyzer setup.
5. Select **Open**.

Note: If you are using the logic analyzer without a keyboard, you can access an on-screen keyboard by selecting**Start>Programs>Accessories>Accessibility>On-Screen Keyboard**.

See Also

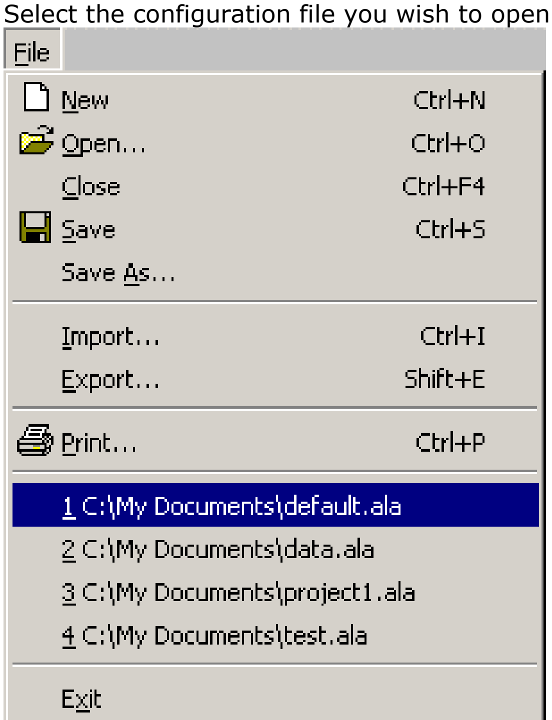
To recall a recently used configuration file

Offline Analysis

ALA Format

To recall a recently used configuration file

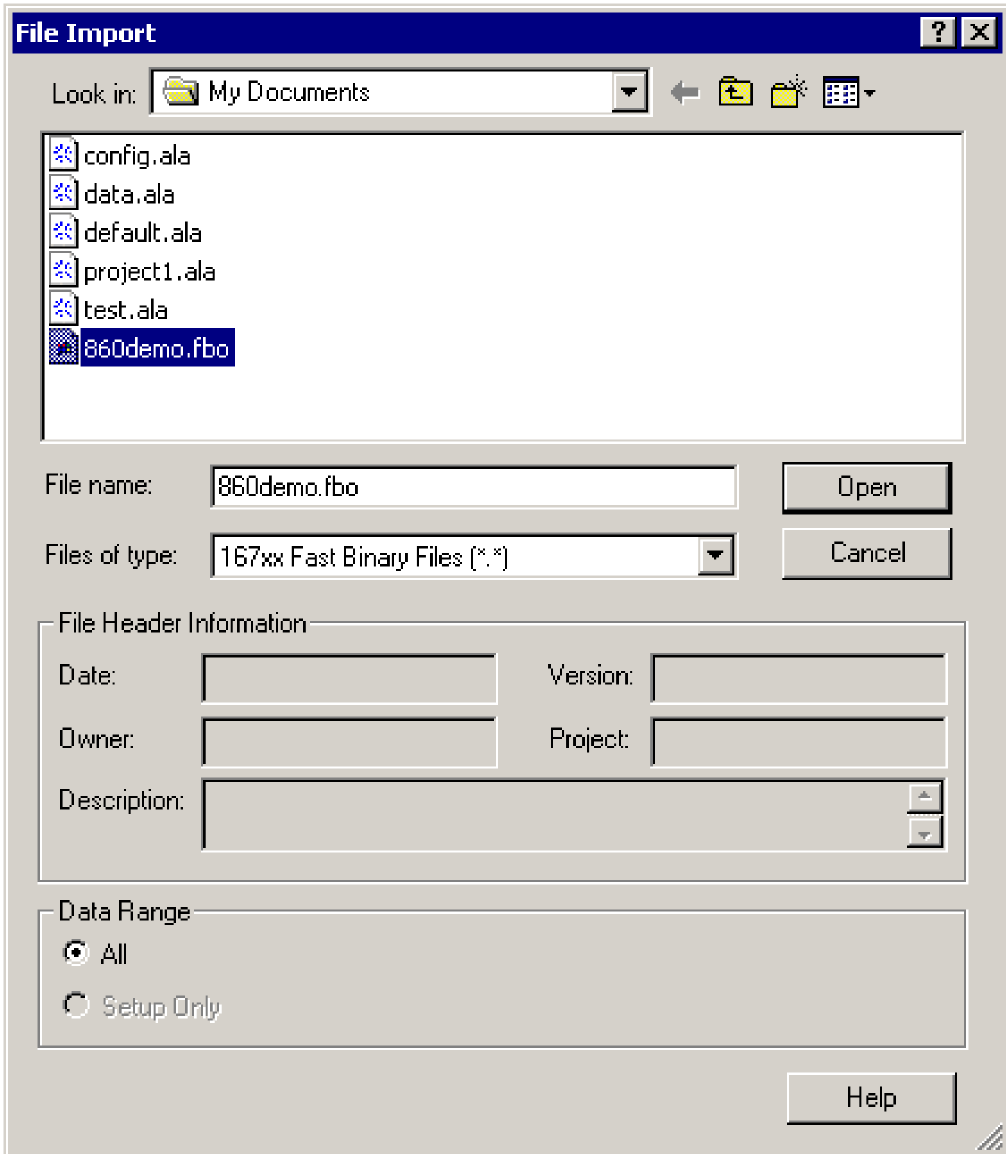
1. From the menu bar, select **File**.
2. Select the configuration file you wish to open from the list provided.



To import 167xx fast binary data

Note: Copy 167xx Fast Binary Files to the local hard disk before importing them. Performance of the *Agilent Logic Analyzer* application is much better when fast binary data files are on the local hard disk than when they are on the network.

1. From the menu bar, select **File>Import...**.



2. Select the name of the 167xx fast binary data file you wish to open.
3. Select **Open**.

Bus/signal names from a 16700-series logic analyzer fast binary data file are organized into a hierarchy of folders based on the module, analyzer, and label names.

See Also

Analyzing 16700-Series Logic Analyzer Data

16700 Pod and Bit Association in Offline Analysis

Fast Binary Data Format
Offline Analysis

Offline Analysis

Offline analysis lets you analyze captured data while the while the logic analyzer's data acquisition hardware is used for making other measurements.

You can use the *Agilent Logic Analyzer* application on stand-alone personal computers or on 1680/90-series logic analyzers to perform offline analysis.

Using shared file systems, offline analysis can be performed from remote locations on the network.

Example: A typical scenario is to capture data in a 16700-series logic analysis system, load the data file into the *Agilent Logic Analyzer* application for offline analysis, and then continue using the 16700-series logic analysis system to look for the next elusive defect or crash. By analyzing the data offline, you can keep your logic analyzer hardware busy making new measurements while you analyze the last one.

Example: Another scenario is to use the *Agilent Logic Analyzer* application to configure and exchange 1680/90-series logic analyzer configuration files containing trigger setups with a team of colleagues located on-site or in remote locations.

General offline analysis considerations

- To analyze data from 16700-series logic analyzers, the data must be saved in the fast binary out (FBO) format. This is done using the File Out Tool (see Analyzing 16700-Series Logic Analyzer Data).
- When analyzing data offline, there is no data data acquisition hardware, so functions such as triggering, hardware assist, and run functions are not available.
- Multiple instances of the *Agilent Logic Analyzer* application can be displayed side-by-side on a 1680/90-series logic analyzer or a personal computer, but their data cannot be time-correlated.
- Offline analysis performance is best with 32K-samples or less on screen.

For more specific information about offline analysis, see:

Analyzing 16700-Series Logic Analyzer Data

Offline Analysis on 1680/90-Series Logic Analyzers

Offline Analysis on Personal Computers

See Also

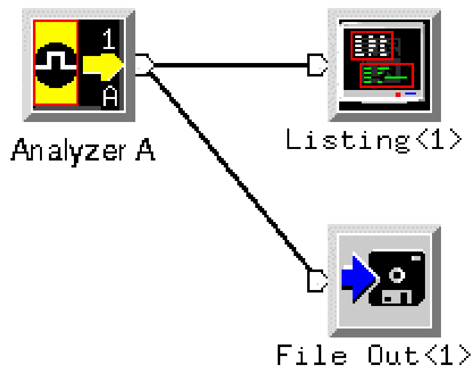
Offline File Formats

Analyzing 16700-Series Logic Analyzer Data

Before you can analyze 16700-series logic analyzer data with the *Agilent Logic Analyzer* application, you must save the measurement data in fast binary out format using the File Out Tool.

The following example shows the general process to use. Refer to the 16700-series logic analysis system online help for any specific information.

1. Configure the 16700-series logic analyzer to capture the desired data.



2. Connect a File Out Tool.

   Data loaded for offline analysis must appear as one data set. Data from a two-machine measurement (as with Pentium 4 processor solution data, for example) must be merged before saving as fast binary output data; in other words, both machines must feed into the same File Out tool.
3. Configure the File Out tool to save the measurement data in fast binary out format.

   For better search performance, limit the size of fast binary data files by using the partial fast binary out option.

   Note: If you want the File Out Tool to save the fast binary out file directly to a shared drive, be sure to configure allLAN connections to enable file sharing.
4. Run the 16700-series logic analyzer to capture the data.
5. Copy the fast binary data file to the local hard disk of the personal computer or 1680/90-series logic analyzer on which the *Agilent Logic Analyzer* application runs.

   Performance of the *Agilent Logic Analyzer* application is much better when fast binary data files are on the local hard disk than when they are on the network.
6. In the *Agilent Logic Analyzer* application, use the **File>Import...** command to import from the fast binary data file.

More 16700 considerations

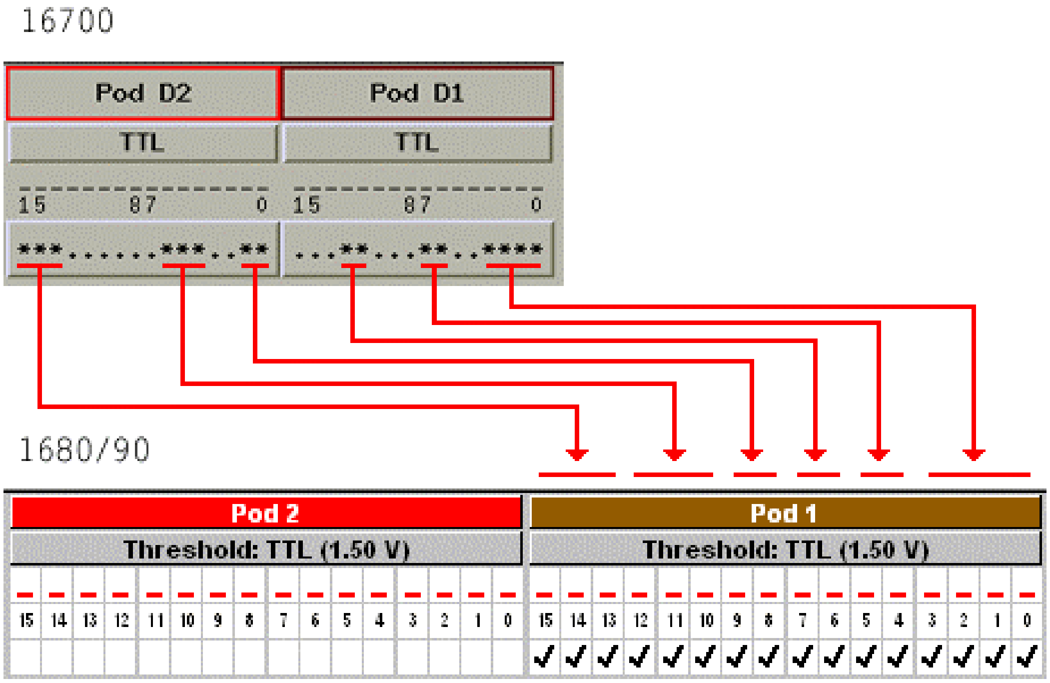
- Only labels and measurement data are saved in the fast binary out format.
- The 16700 pod and bit association iscollapsed when viewed in the *Agilent Logic Analyzer* application.
- After importing fast binary data files, you cannot use the Bus/Signal Setup dialog to add new bus/signal names or change the channel assignments of the imported bus/signal names. You can, however, change bus/signal polarity, rename or delete buses/signals, and add comments.
- Also, because 16700-series logic analyzer sampling mode options cannot be used to set up a 1680/90-series logic analyzer, the Timing/State Sampling Mode Setup dialog is not available.

See Also

16700 Pod/Bit Association in Offline Analysis
Fast Binary Data Format

16700 Pod and Bit Association in Offline Analysis

The offline analysis application will display the 16700 pod and bit association differently. For any given label, all assigned bits across all pods in the 16700 interface is converted to sequentially ordered bits under sequentially ordered pods, starting with pod 1.



Offline Analysis on 1680/90-Series Logic Analyzers

You can perform offline analysis with 1680/90-series logic analyzers, in general, by running a second instance of the *Agilent Logic Analyzer* application and loading previously saved data into that instance. With two instances of the application running, one in *Online* mode and one in *Offline* mode, you can continue making measurements in one instance while you perform offline analysis in the other.

You can perform offline analysis on fast binary data files saved from 16700-series logic analyzers as well as configuration files (.ala format) from any 1680/90-series logic analyzer.

Keep these things in mind when performing offline analysis with a 1680/90-series logic analyzer:

- You can start multiple instances of the*Agilent Logic Analyzer* application.

  If logic analyzer acquisition hardware is present (this should always be the case with 1680-series logic analyzers), the first instance opens in*Online* mode. If acquisition hardware is not present (which may be the case when a 1690-series logic analyzer is powered OFF or is disconnected from the personal computer), the first instance opens in*Offline* mode.

  All instances after the first open in *Offline* mode.
- Logic analyzer run functions do not work in *Offline* mode.
- In the *Offline* mode, you can continue to create triggers and save them in configuration files that can be opened by other instances of the *Agilent Logic Analyzer* application.
- If you import 167xx fast binary data into an *Online* instance of the *Agilent Logic Analyzer* application, the application automatically switches to *Offline* mode, and all hardware functions are automatically turned off.
- If you open a 1680/90-series logic analyzer configuration (.ala) file for offline analysis in an *Online* instance of the *Agilent Logic Analyzer* application, run functions will overwrite the data that has been loaded.

Note: To open, copy, or save files directly from shared disk drives, make sure to configure all LAN connections to enable file sharing.

See Also

To open a configuration file

To import 167xx fast binary data

To save a configuration file

Offline Analysis on Personal Computers

A personal computer (PC) with the *Agilent Logic Analyzer* application installed can perform offline analysis on fast binary data files saved from a 16700-series logic analyzer as well as configuration files (.ala format) from any 1680/90-series logic analyzer.

When using the *Agilent Logic Analyzer* application by itself on a PC:

- The logic analyzer run functions are not available (because there is no acquisition hardware).
- You can save logic analyzer setups (including trigger sequences) to .ala format configuration files, and you can pass these files between personal computers (running the *Agilent Logic Analyzer* application) and other 1680/90-series logic analyzers.
- You can have more than one instance of the*Agilent Logic Analyzer* application running. Without logic analyzer acquisition hardware, all instances will open in *Offline* mode.

Note: To open, copy, or save files directly from shared disk drives, make sure to configure all LAN connections to enable file sharing.

See Also

To open a configuration file

To import 167xx fast binary data

To save a configuration file

Minimum PC Requirements

Minimum PC Requirements

Minimum PC requirements: 500 MHz, 128 M RAM
Operating System: NT or Win 2000

Offline File Formats

.ala and .xml Formats

You can open logic analyzer configuration files (.ala extension) or generic configuration files (.xml extension) for offline analysis. These files are saved by the *Agilent Logic Analyzer* application on 1680/90-series logic analyzers.

The logic analyzer configuration (.ala) file format is an internal format used by the *Agilent Logic Analyzer* application for saving and re-opening setups and data. These files contain hardware setup (including channels assigned to labels), labels, data, and user-defined symbols.

The generic configuration (.xml) file format is an eXtensible Markup Language format that can be edited (using an ASCII text editor) and post-processed by scripts (or other tools) and re-opened by the*Agilent Logic Analyzer* application. These files contain labels, channels assigned to labels, data, and user-defined symbols.

Fast Binary Data Format

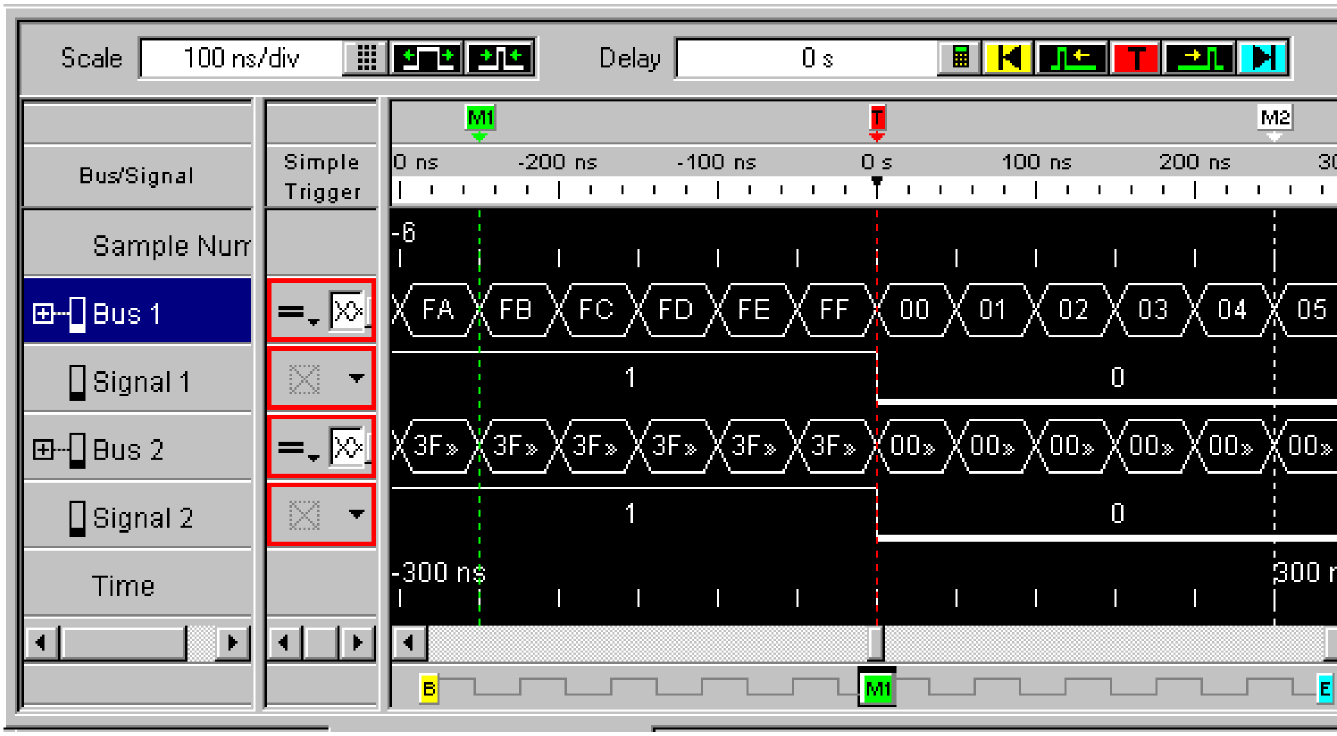
You can import Fast Binary Data Format files (no common file extension) for offline analysis. These files are created on a 16700-series logic analysis system using the *File Out* tool. These files contain labels and data from the 16700-series logic analyzer.

CSV Format

You can export captured data to CSV (Comma-Separated Values) Format files (.csv extension) for offline analysis in other applications like Excel. These files contain labels and data.

Analyzing Waveform Data

Waveform Display Window



The Waveform window is accessed through the menu bar's Window>Waveform. If you have Tabbed Windows turned on, you can also select a tab at the bottom of the window.

The Waveform window displays captured data as a digital waveform. You can configure the window to display selected buses and signals with time or pattern markers in the data. You can also set up bus pattern triggers and signal trigger options.

The Waveform window consists of the following four areas:

- Bus/Signal configuration
- Simple Trigger
- Marker display bar
- Waveform display area

Bus/Signal configuration

To access the following Bus/Signal configuration options, right-click on any bus or signal name in the Bus/Signal column.

Undo - Undo the last action performed.

Insert Row Before - Inserts a predefined bus/signal before the highlighted row.

Insert Row After - Inserts a predefined bus/signal after the highlighted row.

Delete Row - Deletes the bus/signal in the highlighted row.

Expand - Expands the highlighted bus into separate displayed channels.

Collapse - Collapses displayed channels to a single displayed bus.

Assign Channels - Access to the Buses/Signals tab of the Analyzer Setup dialog for mapping (assigning) the highlighted bus/signal to the desired pod and channel connection of the probes.

Rename - Access a keypad to rename the highlighted bus/signal.

Group into Bus - Groups highlighted signals into a bus.

Overlay - Overlays the highlighted bus or signal with another selected bus or signal.

Overlay Remove - Separates overlaid bus/signals.

Symbols - Access to edit symbols dialog for selected bus/signal.

Properties - Access to properties dialog for waveform window, waveform data, and markers.

To reposition bus/signal names

To reposition bus/signal names in the display window, click and hold the mouse cursor over the name to move, then drag and drop the name to the new position. The name is placed above the red position indicator that appears.

Simple Trigger

To set bus pattern triggers

To set signal trigger options

Marker display bar

To access the Marker bar options, right-click anywhere in the marker display bar.

New - To create new markers.

Place - To place markers in data.

Go To - Go To a predefined marker.

Center About - To center the display about a marker.

Delete - To delete a marker.

Rename - To rename a marker.

Send to back - To send a marker to the back.

Properties - To set marker properties.

Drag and drop markers - To move markers in the marker display bar.

Waveform display area

To access waveform display options, right-click anywhere in the display area.

Undo - Undo the last action performed.

Zoom Out

Zoom In

Go To

Place Markers

Center About Markers

Find

  Find Next

  Find Previous

Properties

Drawing Rectangle in data

  Zoom In

Set Quick Trigger - Alternative way to set a Simple Trigger.

Find Next - Data value on left edge of rectangle becomes Find search criteria and next occurrence of that data value is placed at center screen.

Find Previous - Data value on left edge of rectangle becomes Find search criteria and previous occurrence of that data value is placed at center screen.

Change Delay

Change Scale

See Also

Working with Markers

Advanced trigger dialog

To edit symbols

To add or delete display windows

Turning window tabs on/off

Tooltips

To set waveform window properties

Use the Waveform Properties dialog to modify the attributes of the waveform window or the data that appears in each row of the display. To access the Waveform Properties dialog click Edit>Window Properties in the menu bar.

Tip: You can also access the Properties dialog by a right-click in the display area, then select Properties.

To set window properties

The following properties effect the entire waveform window, including displayed data, bus/signal text, and simple trigger text.

To set background color

To set font size

To set an overlay color

To show activity indicators

To show tool tip - values

To show tool tip - transition width

To set markers - snap to edge

To set markers - move edge on screen

To set markers - place on edge

To set row properties

The following properties effect only the row data of the selected bus/signal unless <all> is selected.

To select the Bus/Signal

To set data color

To set row height

To set numeric base

To show data values

To show soft glitch

To set bus options

To set background color

The background color property sets the background color of the data display area of the waveform window.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click the background color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set font size

The font size property adjusts the size of font used in the data display, bus/signal text, and simple trigger text. Fonts can range from size 6 through 72. As the font size is changed, the row height is automatically adjusted to a minimum height to fit the new font size.

1. From the Waveform Properties dialog, click the Windows Properties tab.

2. Click the font size up/down arrows to adjust the font to the desired size.
3. Click Ok.

To set an overlay color

Multiple bus/signals can be inserted (Overlay) within a single row. All overlaid bus/signals are displayed in a different color than the primary bus or signal.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click the overlay color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To show activity indicators

Activity indicators are displayed to the left of the bus or signal name. The Show Activity Indicators property controls whether the indicators are shown or hidden. The indicators show either a low bar (low level), high bar (high level), or a transition arrow (transitioning signal).

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Show Activity Indicators box to display the indicators.
3. Click Ok.

To show tool tip - values

When the mouse is moved within the display area, and is positioned directly over a waveform, a tool tip (text readout) appears that displays the data value of that waveform at its current position. The Show Values property controls whether the tool tip is shown or hidden.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Show Values box under Tool Tips.
3. Click Ok.

To show tool tip - transition width

When the mouse is positioned directly over a waveform, a tool tip (text readout) appears that displays both the data value and the transition width of that waveform at its current position. The Transition Width property controls whether the tool tip includes the transition width. Transition width is displayed in parenthesis.

Note: The Transition Width property is only available when Show Values is on.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Transition Width box under Tool Tips.
3. Click Ok.

To set markers - snap to edge

The Markers - Snap to Edge property enables easy placement of markers on waveform edges by using graphical guidance. When a marker is grabbed and dragged in the data display area, the cursor changes to a green "direction arrow" indicating the direction of the next valid edge. A red "valid edge" bar is placed on the next edge that the marker will be placed on if you decide to release the mouse button. In other words, the green arrow points to the red bar where the marker will be placed.

If the mouse cursor is not positioned over any waveforms when you begin the drag and drop operation, the marker can be dropped anywhere between edges with no snap-to-edge action.

1. From the Waveform Properties dialog, click the Windows Properties tab.

2. Check the Snap to Edge box under Markers.
3. Click Ok.

To set markers - move edge on screen

When a marker is snapped to an edge that is off-screen, the data display is shifted so the marker and the new edge it is snapped to, appear at center screen. The waveform delay is adjusted to the time of the new edge.

Note: The Move Edge on Screen property is only available when Snap to Edge is on.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Move Edge on Screen box under Markers.
3. Click Ok.

To set markers - place on edge

With the Place on Edge property set, when a marker is placed using the right-click>Place Marker shortcut, the marker will be placed on the closest edge from where the mouse right-click occurred. When the Place on Edge property is disabled, the marker is set where the mouse right-click occurred.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Place on Edge box under Markers.
3. Click Ok.

To select the Bus/Signal

The Bus/Signal field lists all signals that have been inserted into the display in the Buses/Signals tab. Use the Bus/Signal property to select the bus or signal that you want the property changes applied to. In addition to the individual signals listed, you also have the selections of "all" and "selected". The "all" selection will apply property changes to all bus/signals. The "selected" option is only available when multiple bus/signals are selected (highlighted), and will apply property changes only to the bus/signals highlighted, and only to the common properties to each of the selected bus/signals. If a property change is not valid to all selected bus/signals, the fields will be greyed out.

Property changes to a bus effects all signals within the bus. For example, if the color of a bus is changed, then if you were to expand the bus, all individual signals would have their color changed.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click in the Bus/Signal name field, then select the desired bus/signal name.
3. Click Ok.

To set data color

The waveform Color property sets the color of the waveform in the data display.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click the color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set row height

The Height property controls the height of the row measured in pixels. The minimum row height is set by the font size. The maximum height is 1000 pixels.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click the Height box, then enter the desired height value.

3. Click Ok.

Tip: You can also set row height by hovering mouse over separator line, when cursor changes to a resizing cursor, click and drag row border to new width.

To set numeric base

The Base property sets the numeric base used to display the data. If you change to a base that creates a longer number format (example; Hex to Binary), all fields are automatically adjusted. Also, if you have data values displayed within the waveform, and you switch from Hex to Binary or Symbols, ">>" may appear to indicate that more text will be displayed if you expand the scale.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Click the Base field, then select the desired numeric base.
3. Click Ok.

To show data values

The Show Value property enables an integrated display of both the waveform and the data value of the waveform at that point. You also can set a color for the waveform data value text.

Note: If the waveform time scale is small, ">>" may appear in the data value to indicate that more text will be displayed if you expand the scale.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Show Value box, then click in the color box and select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To show soft glitch

The Show Soft Glitch property enables you to be visually cued when multiple transitions are occurring at the same point in time. It does this by displaying screen pixels that are common to multiple transitions being drawn on screen. These common pixels are drawn in a different color to create an eye pattern to compare against an expected visual pattern. Soft glitches are seen when you zoom out the time scale.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Show Soft Glitch box, then click in the color box and select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set bus options

The Bus Options button is only available when a bus is selected for property changes. The options listed below are applied only to the selected bus.

1. From the Waveform Properties dialog, click the Windows Properties tab.
2. Check the Show Value box, then click in the color box and select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set waveform style - Bus

The Bus property sets the waveform appearance to a conventional bus shape. When Bus is selected, and MSB ordering is checked, the ordering of the signals in the bus are changed from least significant bit first to most significant bit first.

To set waveform style - Chart

The Chart property sets the waveform appearance to a samples versus time (chart mode) display. When Chart is selected, and Show Axis is checked, a small axis is drawn in the center of the waveform which represents the center of the range of values being displayed. When Show Axis is checked, you can also set an axis color.

When the Chart property is selected, you can also set the following Waveform Limits:

- Max/Min - sets the range limits of the displayed axis.
- Lock to Setup - sets the range limits based on the width of the bus. For example, an 8-bit bus is set to a range of 0-255.
- Show Clipped - enables out-of-range data values to be displayed in a user defined color.

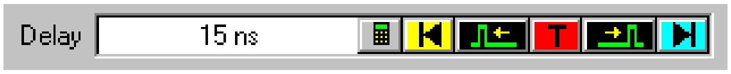
To expand into signals

The Expand into Signals property expands the bus into individual signals. This function is the same as if you selected the Expand (+) field to the left of the bus name.

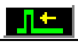
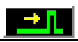
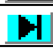
See Also

To set marker properties

Change delay



The delay adjusts the display window relative to the waveform data. The display window's relative position in time is dependant on the trigger point, and the beginning and end of data. Use the following delay controls to position the display window over the desired data.

	Use the keypad to enter a numeric value. If the value you enter is greater than or less than the time of the data range, the window will be moved to the beginning or end limit.
	Moves the window over the beginning of data.
	Scrolls the window towards the beginning of data.
	Moves the window over the trigger point.
	Scrolls the window towards the end of data.
	Moves the window over the end of data.

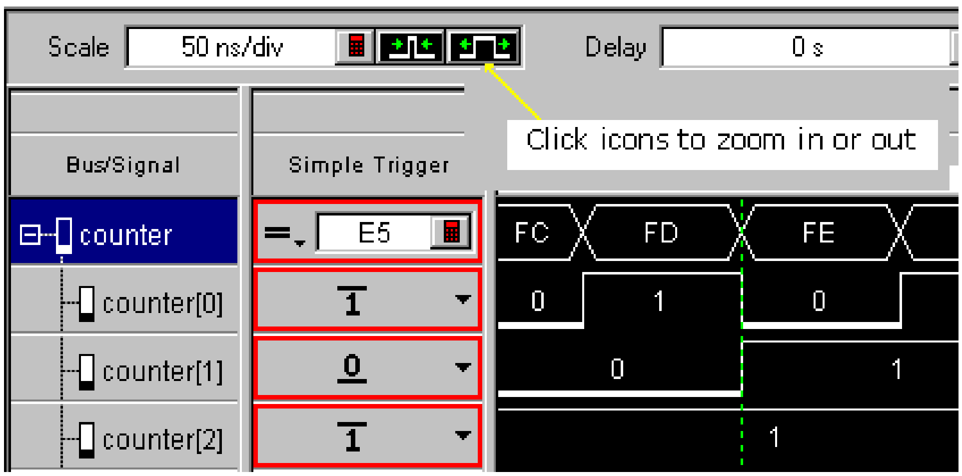
Change scale (time/division)

Data from a timing analyzer is displayed similar to oscilloscope data. That is, they both display waveforms on a horizontal time axis. Therefor, to zoom in or out on a waveform, you simply change the Scale (time/division) of the time axis that the waveform is viewed with.

- Change the scale
- Draw a rectangle in the data

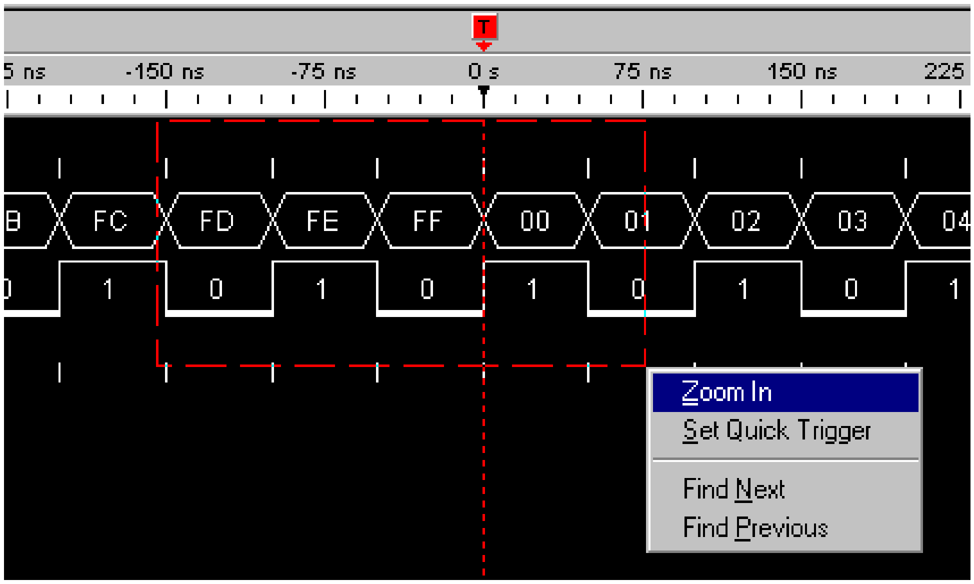
Change the scale

1. Click the zoom out icon to expand the signals to where you want them. The scale ranges from 1 ps/div to 1ks/div. You can also change the scale by clicking the keypad icon and entering a numeric value.



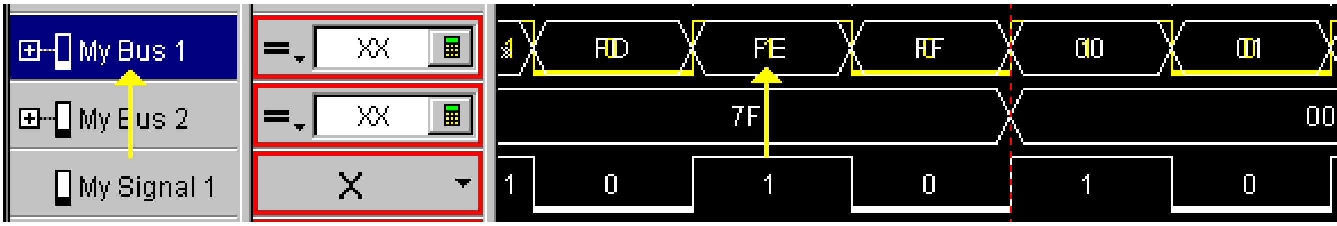
Draw a rectangle in the data

1. Point the mouse to the upper-left corner of the desired view area, then click and hold while moving the mouse to the lower-right corner, then release the mouse button.
2. Select Zoom In. The new display scale is adjusted to the width of the box drawn.


T
5 ns
-150 ns
-75 ns
0 s
75 ns
150 ns
225
B
FC
FD
FE
FF
00
01
02
03
04
0
1
0
1
0
1
0
1
0
1
Zoom In
Set Quick Trigger
Find Next
Find Previous

To overlay bus/signals

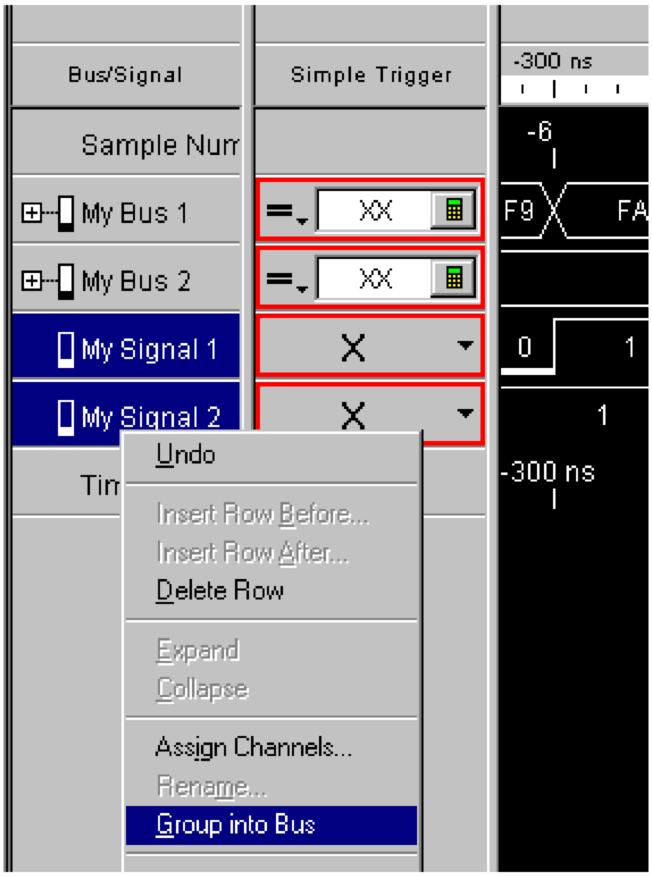
Use the Overlay feature to place multiple bus or signals into one row of displayed data. When multiple signals are overlaid, you can see the relationships visually between all signals. The overlaid bus or signal is drawn first, then the main bus/signal is drawn last as to overwrite the overlaid bus/signals for clarity.



1. Right-click on the bus or signal you want to overlay another bus or signal onto, then select Overlay... .
2. From the Overlay selection dialog that appears, select the bus or signal you want to overlay onto the highlighted bus or signal.
3. If you want to change the color of the overlaid bus or signal, go to Window Properties and change Overlay color.

To group signals into a bus

1. While holding the shift key down, click on all desired signals.
2. With the mouse pointer over any one of the highlighted signals, right-click and select Group into Bus.



To add or delete display windows

You can add new listing and waveform display windows to the interface. As new windows are added, they appear in the list under Window in the menu bar. The active window will have a check mark. All available windows can be accessed either through the menu bar or through the use of tabs.

To add a new display windows

1. In the menu bar, click Window>New Listing or New Waveform. If the windows are tabbed, you can also right-click on the tab, then select New Listing or New Waveform.

To delete display windows

1. From the menu bar, click Window>Close. If windows are tabbed, you can also right-click on the tap, then select Close.

See Also

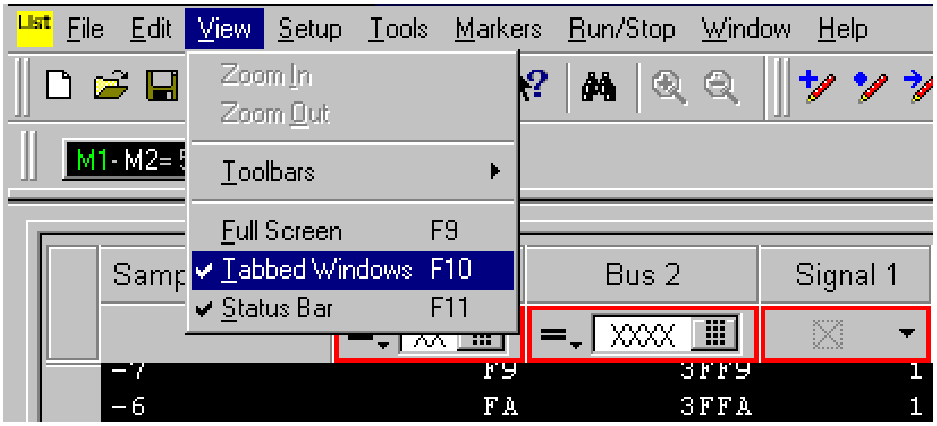
Turning window tabs on/off

Viewer toolbar

Turning window tabs on/off

By default, the Listing and Waveform display windows are tabbed for ease of switching between displays.

To turn on or off window tabs, select View>Tabbed Windows.



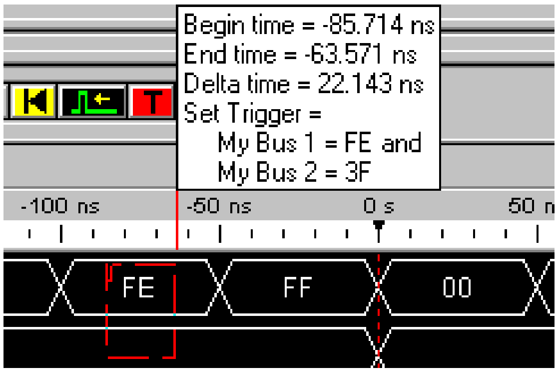
To switch display windows when tabs are turned off, you must select Window>"*display window name*".

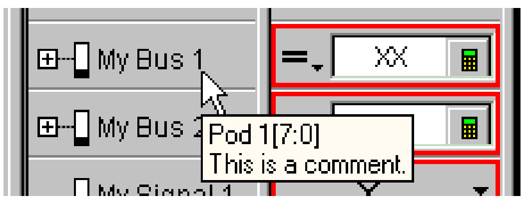
See Also

To add or delete display windows

Tool Tips

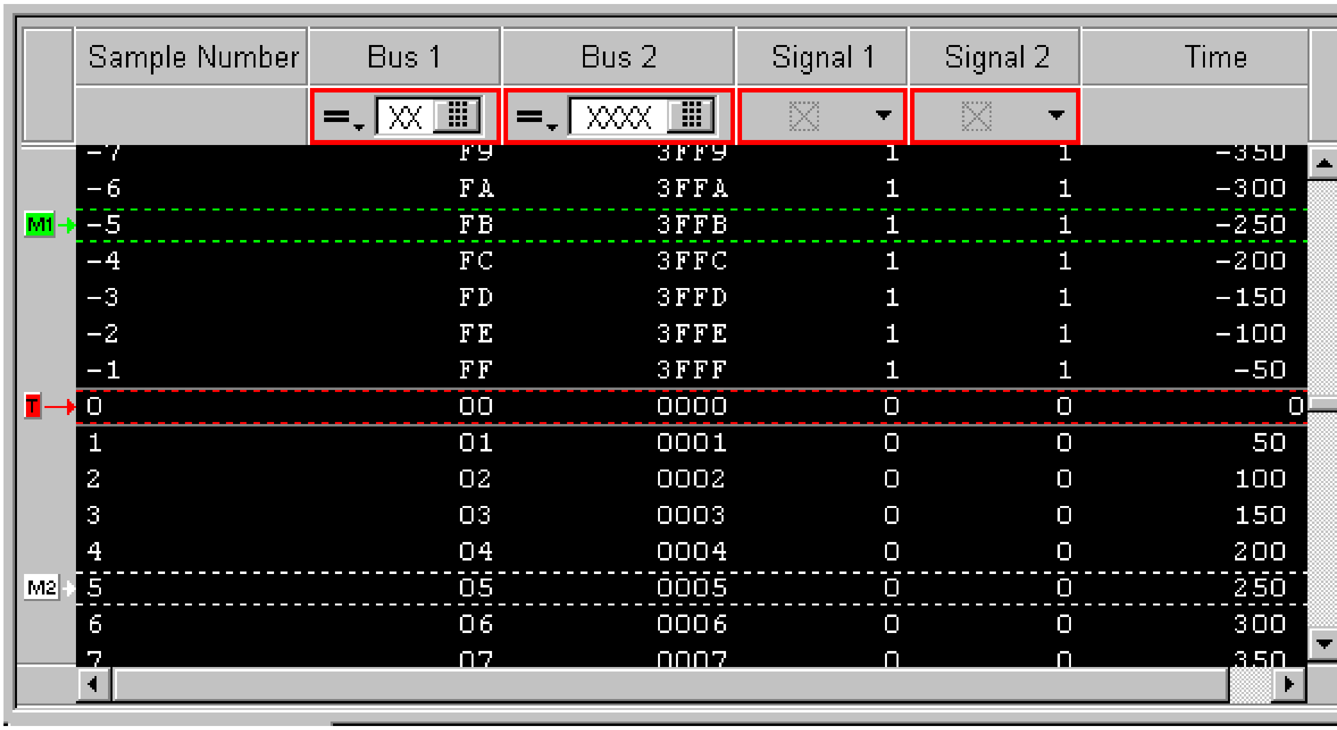
Tool tips are small information displays (text readout) that appear during mouse operations such as hovering over a waveform or bus/signal name, moving markers, or drawing a rectangle in data. Use them as comments or to read current positions, waveform transition widths, or trigger specifications (when setting trigger with mouse).





Analyzing Listing Data

Listing Display Window



The Listing window is accessed through the menu bar's Window>Listing. If you have Tabbed Windows turned on, you can also select a tab at the bottom of the window.

The Listing window displays your captured data as a state listing. You configure the window to display selected buses and signals in columns. Within the listed data, you can insert time or pattern markers. You can also configure the bus pattern triggers and signal trigger options.

The Listing window consists of the following four areas:

- Column configuration
- Simple Trigger
- Marker display bar
- Listing display area

Column configuration

To access the following column configuration options, right-click on any bus or signal name in the column head.

Insert Column Before

Insert Column After

Delete Column

Assign Channels

Rename

Symbols

Display

Properties

To reposition bus/signal names

To reposition bus/signal names in the display window, click and hold the mouse cursor over the name to move, then drag and drop the name to the new position. The name is placed to the left of the red position indicator that appears.

Simple Trigger

To set bus pattern triggers

To set signal trigger options

Marker display bar

To access the marker display options, right-click anywhere in the marker display bar.

New - To create new markers.

Place - To place markers in data.

Go To - Go To a marker.

Center About - To center the display about a marker.

Delete - To delete a marker.

Rename - To rename a marker.

Send to back - To send a marker to the back.

Properties - To set marker properties.

Drag and drop markers - To move markers in the marker display bar.

Listing display area

To access the Listing display options, right-click anywhere in the display area.

Undo - Same as Edit>Undo.

Go To

Place Marker

Find

Find Next

Find Previous

Properties

Draw Rectangle in data

Set Quick Trigger - Alternative way to set a simple trigger.

Find Next - Data value on top edge of rectangle becomes Find search criteria and next occurrence of that data value is placed at center screen.

Find Previous - Data value on top edge of rectangle becomes Find search criteria and previous occurrence of that data value is placed at center screen.

Copy Text - Copies data as text into the system clipboard.

See Also

To display symbols

Working with Markers

To add or delete display windows
- Turning window tabs on/off
- Tool Tips

Column configuration

To access the tasks in this area, right-click on any bus or signal name in the column head of the listing window.

Insert Column Before

Insert Column After

Delete Column

Assign Channels

Rename

Symbols

Display

Properties

Insert Column Before

Accesses a list of predefined bus and signal names. The selected name is inserted to the left of the highlighted column. To create new bus and signal names go to To add a new bus or signal.

Insert Column After

Accesses a list of predefined bus and signal names. The selected name is inserted to the right of the highlighted column. To create new bus and signal names go to To add a new bus or signal.

Delete Column

Deletes the highlighted column.

Assign Channels

To assign channels to pods go to Buses/Signal Setup.

Rename

Accesses a keypad to rename the highlighted bus or signal name.

Symbols

For information on editing symbols, go to Symbols.

Display

Turns on or off the activity indicators, default base, and the simple trigger. When turned on, they are visible in the column head of the listing window.

Properties

For information on setting listing window properties, refer to To set listing window properties.

To set listing window properties

Use the Listing Properties dialog to modify the attributes of the listing window or the data that appears in each column of the display. To access the Listing Properties dialog click Edit>Window Properties from the menu bar.

Tip: You can also access the Properties dialog by a right-click in the display area, then select Properties.

To set window properties

The following properties effect the window as well as all displayed row data.

To set background color

To set font size

To show column display options - activity indicators

To show column display options - column base

To show column display options - simple trigger

To show center rectangle

To set column properties

The following properties effect the displayed column data.

To select the Bus/Signal name

To set data color

To set column width

To set column alignment

To set numeric base

To set marker relative

To set background color

The Background Color property sets the background color of the data display area of the Listing window.

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Click the background color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set font size

The font size property adjusts the size of font used in the data display, bus/signal text, and simple trigger text. Fonts can range from size 6 through 72. As the font size is changed, the column width is automatically adjusted to a minimum width to fit the new font size.

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Click the font size up/down arrows to adjust the font to the desired size.
3. Click Ok.

Column display options

The Column Display Options properties enable viewing of the activity indicators, the column base, and the simple trigger configuration.

Display activity indicators

Activity indicators are displayed to the left of the bus or signal name. The indicators show signal activity as either a low bar (low level), high bar (high level), or a transition arrow (transitioning signal).

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Check the Display Activity Indicators box to display the indicators.
3. Click Ok.

Display column base

The base field displays the numeric base for which the data is displayed in. The numeric base is set in the Buses/Signals tab of the Analyzer.

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Check the Display Column Base box to display the indicators.
3. Click Ok.

Display simple trigger

The Display Simple Trigger property enables configuration and viewing of the simple trigger. For more information on configuring the simple trigger refer to Simple Trigger.

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Check the Display Simple Trigger box to display the indicators.
3. Click Ok.

To show center rectangle

The center rectangle is the box that is drawn around the one state displayed at center screen. The Display Rectangle property enables the rectangle to be displayed or hidden.

1. From the Listing Properties dialog, click the Windows Properties tab.
2. Under Center Rectangle, check the Display Rectangle box to display the center rectangle.
3. Click the rectangle color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
4. Click Ok.

To select the Bus/Signal name

The Bus/Signal field lists all signals inserted into the display. The names are listed in the same order as they appear in the display. Use the Bus/Signal property to select the bus or signal that you want the property changes applied to. In addition to the individual signals listed, you also have the selections of "all" and "selected". The "all" selection will apply property changes to all bus/signals. The "selected" option is only available when multiple bus/signals are selected (highlighted in display), and will apply property changes only to the bus/signals highlighted, and only to the common properties to each of the selected bus/signals. If a property change is not valid to all selected bus/signals, the fields will be greyed out.

Property changes to a bus effects all signals within the bus. For example, if the color of a bus is changed, then if you were to expand the bus, all individual signals would have their color changed.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click in the Bus/Signal name field, then select the desired bus/signal name.
3. Click Ok.

To set data color

The Color property sets the color of the data in the listing display.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click the color box, then select the desired color from the palette. If you want to create a custom color, click Other... to access the custom color dialog.
3. Click Ok.

To set column width

The Width property controls the width of the column (specified in pixels). The minimum column width is 1 pixel, while the maximum width is 1000 pixels.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click the width box, then type in the desired pixel width value.
3. Click Ok.

Tip: You can also set column widths by placing the mouse pointer over the right border of the column header box; when the pointer icon changes to a resizing pointer, click and drag the column border to the new width.

Tip: You can autosize columns by placing the mouse pointer over the right border of the column header box; then, when the pointer icon changes to a resizing pointer, double-click.

Tip: If your keyboard has a numeric keypad, you can also autosize columns by selecting the column header box (to highlight it) and by pressing **Ctrl** and "+" on the numeric keypad.

To set column alignment

The Alignment property sets the display of data to be left-justified, right-justified, or centered within the column.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click the Alignment box, then select the desired alignment choice.
3. Click Ok.

To set numeric base

The Base property sets the numeric base used to display the data. If you change to a base that creates a longer number format (example; Hex to Binary), all fields are automatically adjusted.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click the Base box, then select the desired numeric base choice.

   Note: If the Time column has been selected instead of a data column, your choices change from a numeric format to Absolute, Relative Previous, or Relative Marker.

3. Click Ok.

To set marker relative

The Marker property only applies when a Time column is selected and the base of the Time column is set to Relative Marker. This field allows the selection of a marker to act as the location where time should be relative from.

1. From the Listing Properties dialog, click the Column Properties tab.
2. Click the Marker box, then select the desired marker.
3. Click Ok.

See Also

To set marker properties

Comparing Captured Data to Reference Data

By comparing data from different acquisitions, you can look for differences between a known-good device under test and a device under test with a problem or one that is operating under different conditions.

To compare captured to reference data:

1. Capture (or load) the data you want to use as the reference data.
2. Select **Window>New Compare...** to open a new Compare display window.
3. In the Compare display window, click the **Copy...** button to select the current data that should be copied to the reference buffer.
4. Capture (or load) the data that you want to compare to the reference.

   Differences are highlighted in the Compare window.

For more information on comparing captured data to reference data, see:

- To copy data to the reference buffer
- To find differences in the compared data
- To compare only a range of samples
- To offset the reference data
- To run until a number of compare differences
- To set the Compare display options

See Also

Compare Display Window

Capturing Data from the Device Under Test

Loading Saved Data and Setups

To copy data to the reference buffer

1. In the Compare display window, click the **Copy...** button.
2. In the Select Buses/Signals dialog:
   1. From the available buses and signals, select the ones to be copied to the reference buffer and click **Add>>**.

      To remove buses and signals from the selected list, select them and click **<<Remove**.
   2. Select either *All* data or a range of data using markers.
3. **Note:** Copying generated bus/signal columns, such as those created by an inverse assembler or an analysis tool, takes longer because of the extra processing to re-create the data.

   Note: If your logic analyzer has deep memory, it takes a while to copy data to the reference buffer.

   Note: The copying process *cannot* be stopped once it is started.

   When you are ready to begin the copy, click **OK**.

To find differences in the compared data

In the Compare display window:

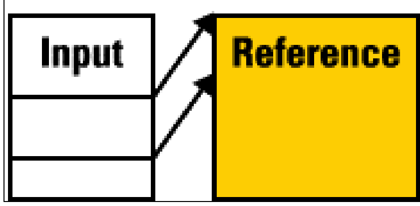
- Click the **>>** button to find the next difference (below the center reference).
- Click the **<<** button to find the previous difference (above the center reference).
- Click a blue tick mark in the **Compare Overview** bar (between the vertical scroll bar and the Marker Overview bar on the right side of the window) to go to that difference.

Note: When a difference occurs on a subrow (for example, when the data is inverse-assembled or decoded by an analysis tool), the next and previous buttons go to the sample row instead of the subrow.

To compare only a range of samples

1. In the Compare display window, click the**Range & Offset...** button.
2. In the Range & Reference dialog, select either *All* data or a range of data using markers.
3. Click **OK**.

Note: When you specify a range to compare, the range is compared to the*top* of the reference buffer (unless thereference data has been offset by a number of samples) and not the same range in the reference buffer as you might expect. This behavior allows multiple ranges in the input data to be compared with the reference data.



To offset the reference data

When there are differences in the number of samples captured before the trigger, or when you are comparing a range of samples, you can offset the reference data so that the samples being compared are properly aligned.

1. In the Compare display window, click the**Range & Offset...** button.
2. In the Range & Reference Offset dialog, enter the number of samples to offset the reference by.
3. Click **OK**.

See Also

To compare only a range of samples

To run until a number of compare differences

The Compare display lets you stop a repetitive run or send e-mail after a run has more than a specified number of differences when compared to the reference data.

1. In the Compare display window, click the**Run Until...** button.
2. In the Run Until dialog, select the **Run Until** check box and enter the number of differences that will stop a repetitive run or send an e-mail event.
3. To stop a repetitive run after the number of differences have been found, select the **Stop repetitive run** check box.
4. To send e-mail after the number of differences have been found, select the **Send e-mail** check box; then, click the **E-mail...** button and enter the address to which e-mail will be sent.

   You can specify multiple recipients by separating each e-mail address with a semicolon (;).

   Note: E-mail must be set up on the computer running the *Ailent Logic Analyzer* application before this feature will work. Refer to the operating system online help or your mail application's online help for information on setting up e-mail. Some mail applications block e-mail that is sent by running programs, so test this feature with your mail application before you use it.
5. Click **OK** in the Run Until dialog.
6. Start the repetitive run measurement.

See Also

To start/stop measurements

To set the Compare display options

1. In the Compare display window, right-click on the bus/signal column name; then, select **Properties...** from the popup menu.
2. In the Compare Properties dialog:
   - The **Window Properties** tab lets you select the reference data background color, the background color that indicates no reference data, and the difference foreground and background colors.
   - The **Column Properties** tab's Display field lets you display **All** of the reference data, just the reference data where a difference was found (**Difference Pair**), or only the highlighted differences in the data being compared (**Input Only**).

   All other Compare property options are the same as in the Listing window.

See Also

To set Listing window properties

Searching the Captured Data

Searching

The logic analyzer allows you to search for data patterns, time values, or sample numbers.

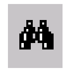
To search for a value - This search option locates a specified data pattern or value. You can qualify your search by specific bits, data patterns, equality, and range operators. The search result is placed at the center of the display.

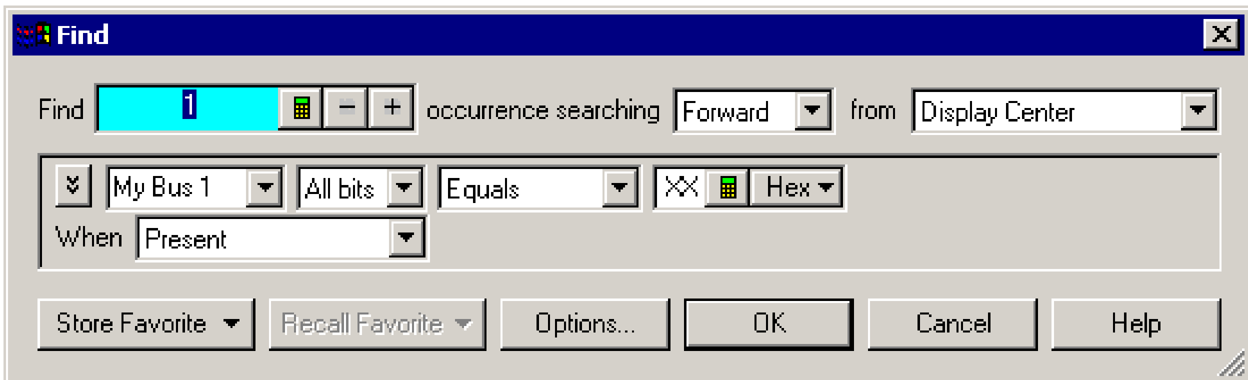
To perform complex pattern searches - This search option locates specified data patterns or values by expanding the search criteria to include more than one event. You can qualify your search by specific bits, data patterns, equality, and range operators. The search result is placed at the center of the display.

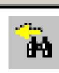
To Go To a specific position in the acquisition - This search option locates an individual time value, sample number, or a marker. The search result is placed at the center of the display.

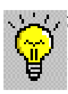
To search for a value

This search option locates a specified data pattern. You can qualify your search by specific bits, data patterns, equality, and range operators. The search result is placed at the center of the display.

1. From the menu bar select, **Edit>Find**or click the  icon in the standard toolbar.
2. In the **Find** dialog, select the number of the occurrences you wish to find, select whether you want to search forward or backward from the start location, then select the start location.
3. Configure the event you wish to locate.
4. Click **OK**.



5. Click the Previous  or Next  icons to see more occurrences.



Tip: As you configure the find function, try to think of it as constructing a sentence that reads left-to-right. For example:

"Find **1** occurrence **Forward** from the**Display Center** of a bus named **My Bus 1**, and on **All bits** a pattern that **Equals XX Hex**, display the event When all criteria is **Present**."

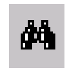
Note: The find qualifiers with an operator (Present>, Present>=, Present<=, Present>) and the "Present for Range" and "Not Present for Range" allow you to specify a time duration. This means that the find event specified in the expression area will be found based upon the given time and operator. The other qualifiers (Present, Not Present, and Entering) do not allow a time duration.

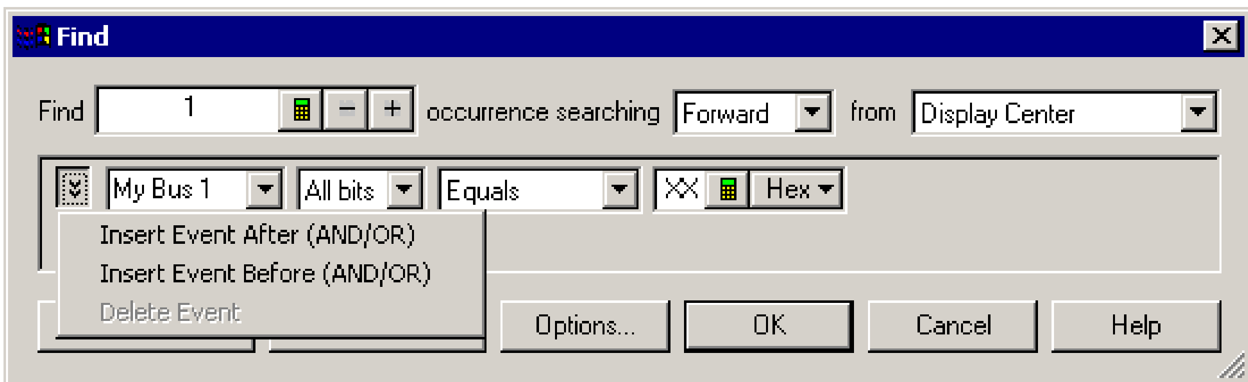
See Also

To perform complex pattern searches
Store Favorite
Recall Favorite
Options

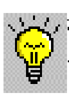
To perform complex pattern searches

This search option locates specified data patterns by expanding the search criteria to include more than one event. You can qualify your search by specific bits, data patterns, equality, and range operators. The search result is placed at the center of the display.

1. From the menu bar select, **Edit>Find** or click the  icon in the standard toolbar.
2. In the **Find** dialog, select the number of the occurrences you wish to find, select whether you want to search forward or backward from the start location, then select the start location.
3. Configure the event you wish to locate.
4. Select the drop down menu to choose the type of event you wish to add. Select **Insert Event After** to insert a new find event after the current event, or select **Insert Event Before** to insert a new find event before the current event. The **Delete Event** option will delete the current event only if there is more than one event present.



5. Select either **AND** or **OR** from the box that appears after you add a search option. AND'ed searches find occurrences of both events, while OR'ed searches find occurrences of either event.
6. Select the bus or signal name and enter the value you want to locate.
7. Click **OK**.
8. Click the Previous  or Next  icons to see more occurrences.



Tip: As you configure the find function, try to think of it as constructing a sentence that reads left-to-right. For example:

"Find **1** occurrence **Forward** from the **Display Center** of a bus named **My Bus 1**, and on **All bits** a pattern that **Equals XX Hex**, display the event When all criteria is **Present**."

Note: The find qualifiers with an operator (Present>, Present>=, Present<=, Present>) and the "Present for Range" and "Not Present for Range" allow you to specify a time duration. This means that the find event specified in the expression area will be found based upon the given time and operator. The other qualifiers (Present, Not Present, and Entering) do not allow a time duration.

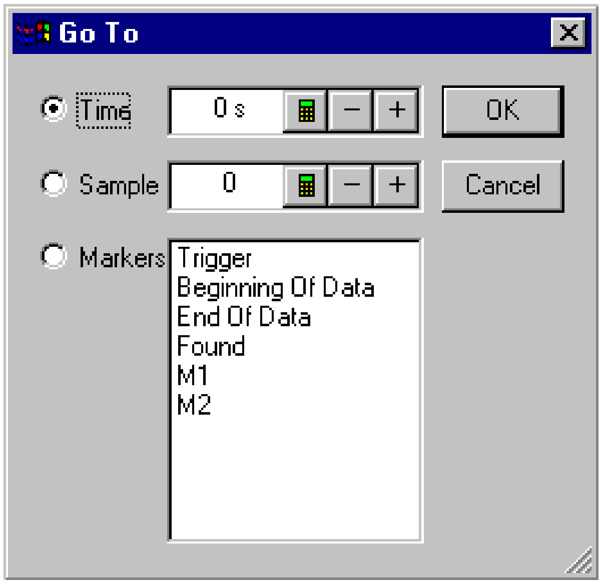
See Also

To search for a value
Store Favorite
Recall Favorite
Options

To Go To a specific position in the acquisition

This search option locates a specified time value, sample number, or a marker. The search result is placed at the center of the display.

1. From the menu bar, select **Edit>Go To**.
2. From the Go To dialog, choose either **Time**, **Sample**, or **Markers**.



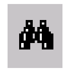
3. Enter the value you want to locate or select the marker you want to locate.
4. Click **OK**.

Tip: You can also access the Go To dialog by right-clicking in the display area.

Favorite Find Pattern

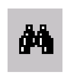
To store a favorite a pattern
To recall a favorite pattern
To delete a favorite pattern

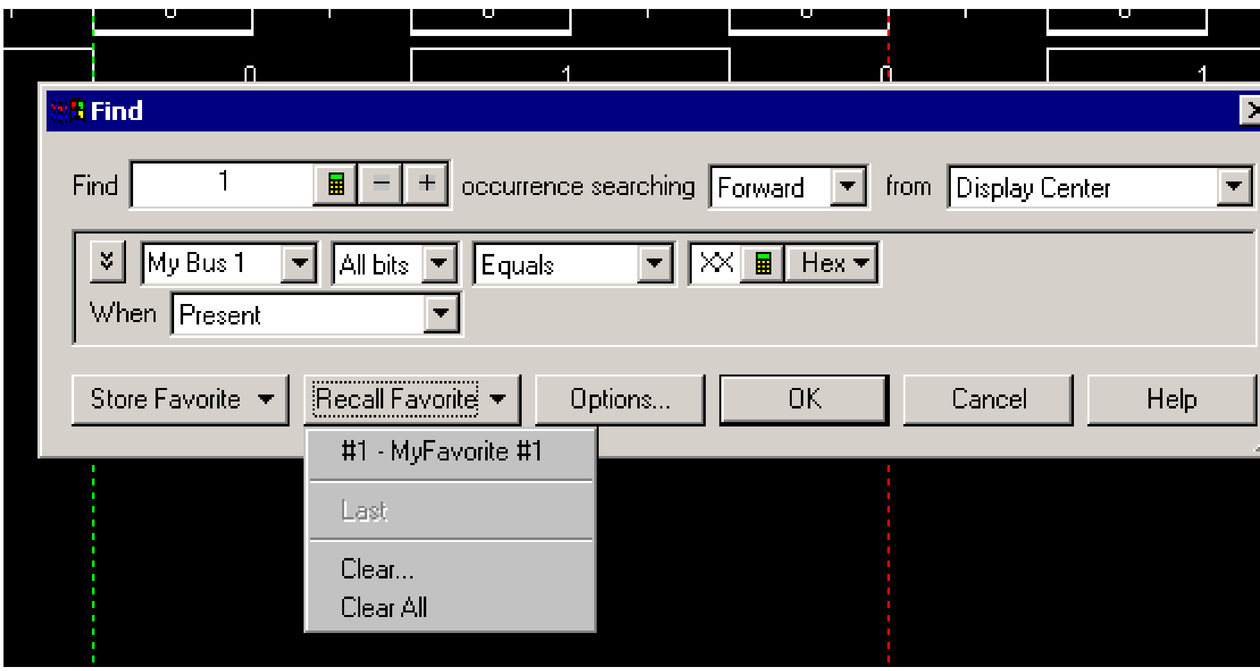
To store a favorite pattern

1. From the menu bar select, **Edit>Find**, or click the  icon.
2. Set up the pattern you want to find.
3. Click **Save Favorite**.

4. Enter the name of the pattern.
5. Click **OK** to save the pattern.

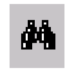
To recall a favorite pattern

1. From the menu bar select, **Edit>Find**, or click the  icon.
2. Select **Recall Favorite**, then select the pattern you want to use from the drop down menu.



3. Click **OK**.

To delete a favorite pattern

1. From the menu bar select, **Edit>Find**, or click the  icon.
2. Select **Recall Favorite**, then select **Clear**.

Note: To clear all of the patterns, select **Clear All**.

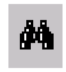
3. Select the pattern you wish to delete from the list provided.
4. Select **Clear**.

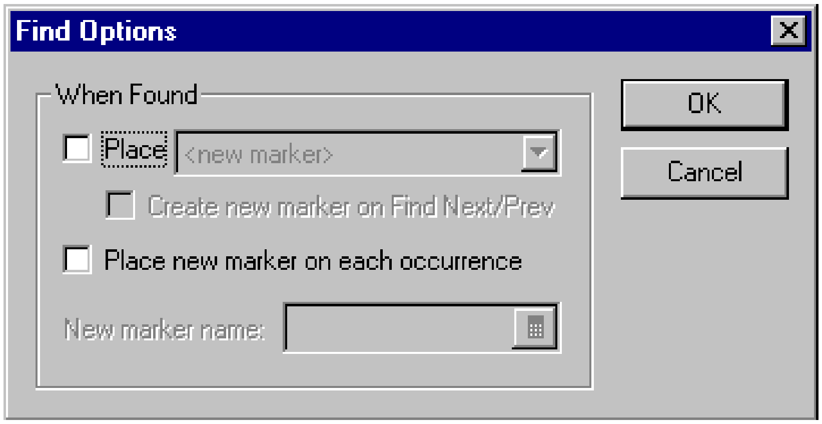
See Also

Searching
Find Options

Find Options

When searching for a pattern you can place a marker at the occurrence.

1. From the menu bar select, **Edit>Find**, or click the  icon in the standard toolbar.
2. Setup the pattern you want to find.
3. Click **Options**.



To use the Place function

1. Select **Place**, then choose the marker you want to place from the drop down menu.
    a. For **<new marker>**, enter the name of the new marker.
    b. Select **Create new marker...** if you want a new marker at each occurrence of the pattern.
2. Select **OK**.

To use the Place new marker on each occurrence function

1. Select **Place new marker on each occurrence**.
2. Enter the name of the new marker.
3. Select **OK**.

See Also

Searching
Favorite Find Pattern

Marking, and Measuring Between, Data Points

Working with Markers

Once a marker is created, you can use it as a reference point in the data when measuring intervals or viewing the data value at the marker.

To create new markers

To place markers in data

Go To marker

To center the display about a marker

To delete a marker

To rename a marker

To send a marker to the back

To set marker properties

To drag and drop markers in data

Marker snap to edge

To create a new interval measurement

To create a new value at measurement

See Also

Markers display bar

Reading off-screen markers

Marker measurement display bar

Marker menu

Marker toolbar

Making marker measurements

Once a marker is created, you can use it as a reference point in the data when measuring intervals or viewing the data value at the marker.

To create a new interval measurement

To create a new value at measurement

See Also

Working with markers

To create new markers

When you create a new marker, you have the option of customizing it with a user-defined name and color. You can also position it precisely in data by either a time value or sample number, or by a data value. Up to 1024 markers can be created.

1. From the menu bar, select Markers>New.
2. From the New Marker dialog, either accept the new default marker name and color, or assign a custom name and color.
3. Configure the position of the new marker in the data by one of the following methods:
   - **Time** - Position the marker by a time value from a reference point. Reference points are the Trigger, Beginning of Data, End of Data, or another Marker.
   - **Sample** - Position the marker by a number of samples from a reference point. Reference points are the Trigger, Beginning of Data, End of Data, or another Marker.
   - **Value** - Position the marker at a user defined value. Click the **Occurs** field and configure the find data options.

Using abbreviated names

Marker names can be very descriptive. The problem with descriptive marker names is that they are long and take up room on the display. To keep a descriptive marker name and also minimize the amount of display real estate used, an abbreviated name can be defined. The abbreviated name is defined by including "[]" within the descriptive name.

Example:

The name for the trigger marker is defined as: Trigger[T]. The word "Trigger" is the descriptive name that appears within the marker tool tip. The "[T]" is the abbreviated name that appears within the marker display bar. If an abbreviated name is not defined within a descriptive name, the descriptive name will be displayed within the marker display bar.

See Also

To place markers in data

Go To markers

Reading off-screen markers

To center the display about a marker

To delete a marker

To rename a marker

To sent a marker to the back

To set marker properties

To drag and drop markers in data

Marker snap to edge

To place markers in data

Use Place Markers to quickly position a pre-defined marker in the data. Depending on how you access the Place Markers feature, the marker is placed in the data a little differently.

You can also move markers by dragging with the mouse or using the front-panel knobs.

Place marker at center screen

1. From the menu bar click Markers>Place On Screen.
2. Select the desired marker, then select OK. The marker will be placed at mid-screen.

Place marker at mouse cursor

1. Point the mouse to the desired data point in the display.
2. Right-click the mouse, and select Place Markers.
3. Select the desired marker, then select OK. The marker is placed in the data at the point where the mouse cursor was pointing.

See Also

To create new markers

Go To markers

Reading off-screen markers

To center the display about a marker

To delete a marker

To rename a marker

To send a marker to the back

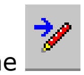
To set marker properties

To drag and drop markers in data

Marker snap to edge

Go To Markers

Use the Go To markers feature to quickly find a previously set marker in the data, or the beginning of data, end of data, or the trigger point. The selected marker appears at the center of the display.

1. From the menu bar, select **Markers>Go To...** or select the  icon in the markers toolbar.
2. Select the marker you wish to find from the list provided.
3. Click **OK**.

See Also

To create new markers

To place markers in data

To center the display about a marker

To delete a marker

To rename a marker

To sent a marker to the back

To set marker properties

To drag and drop markers in data

Marker snap to edge

To center the display about a marker

Use the center about feature to center the display around a selected marker pair. If the marker pair is separated by a large time or sample amount, the scale of the display is automatically changed so both markers appear on screen.

Since the center about feature centers the display around a pair (two) markers, if you have three or more markers defined, you will have available choices for all possible combinations of two.

1. From the menu bar, select Markers>Center About.
2. Select the desired marker combination, then select OK. The center of data between the selected markers is set at mid-screen.

See Also

To create new markers

To place markers in data

Go To markers

To delete a marker

To rename a marker

To sent a marker to the back

To set marker properties

To drag and drop markers in data

Marker snap to edge

To delete a marker

1. In the Marker Display Bar, point the mouse cursor over the marker you want to delete.
2. Right-click the mouse, then select Delete. The marker under the mouse cursor is deleted. To delete all markers, select Delete All.

See Also

To create new markers

To place markers in data

Go To markers

To center the display about a marker

To rename a marker

To sent a marker to the back

To set marker properties

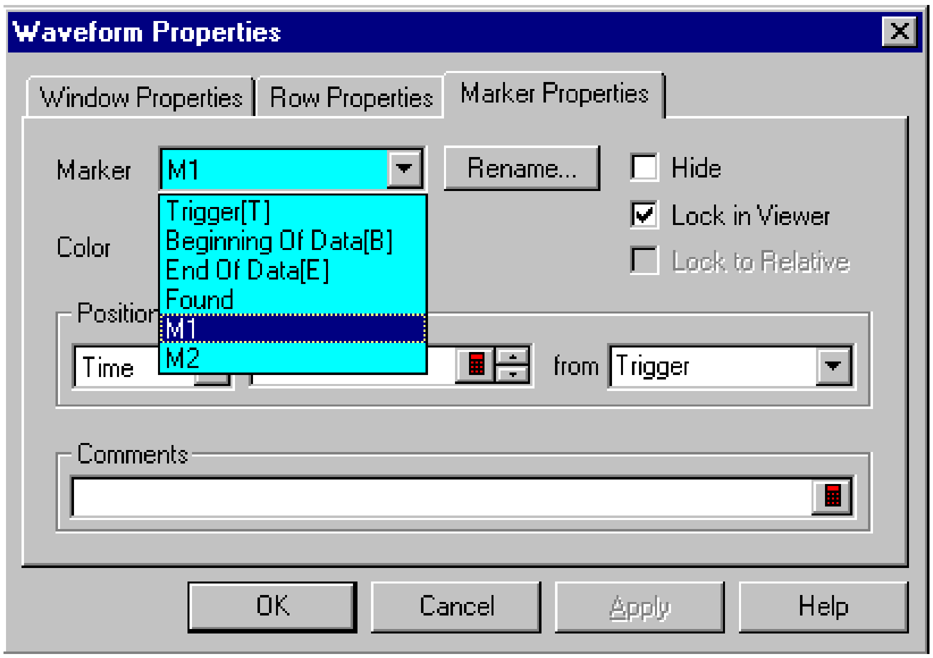
To drag and drop markers in data

Marker snap to edge

To rename a marker

The rename marker feature allows you to change the marker name.

1. Click in the Marker box, then select the marker you wish to rename.
2. Select **Rename...**, then from the pop-up keypad that appears, type in the new name.
3. Select Ok.



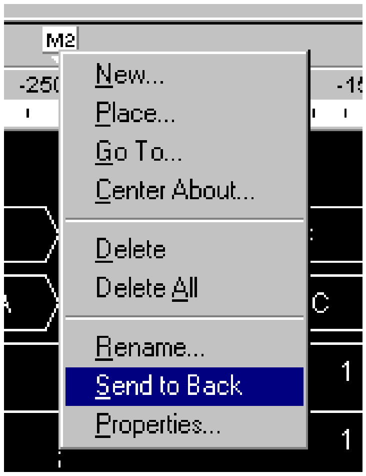
See Also

To create new markers
To place markers in data
Go To markers
To center the display about a marker
To delete a marker
To send a marker to the back
To set marker properties
To drag and drop markers in data
Marker snap to edge

To send a marker to the back

This feature is used to toggle between markers that are located in the same position. This feature can be accessed by right clicking on the marker you want move.

1. Right click on the marker you wish to send to the back.
2. Select **Send to Back** from the pop-up menu.



See Also

To create new markers
To place markers in data
Go To markers
To center the display about a marker
To delete a marker
To rename a marker
To set marker properties
To drag and drop markers in data
Marker snap to edge

To set marker properties

Once a marker is created, you can modify any of its attributes from the Marker Properties tab. To access marker properties, click Markers>Properties from the menu bar, then click the Markers Properties tab.

Tip: You can also access the Properties dialog by a right-click in the display area, then select Properties.

To rename the marker

To change the marker color

To lock marker in viewer

To change the marker type

Marker snap to edge

To rename the marker

1. From the Marker properties tab, select the Marker to rename.
2. Select the Rename button, and from the Rename keypad that appears, type the new name, then select Ok.

Using abbreviated names

Marker names can be very descriptive. The problem with descriptive marker names is that they are long and take up room on the display. To keep a descriptive marker name and also minimize the amount of display real estate used, an abbreviated name can be defined. The abbreviated name is defined by including "[]" within the descriptive name.

Example:

The name for the trigger marker is defined as: Trigger[T]. The word "Trigger" is the descriptive name that appears within the marker tool tip. The "[T]" is the abbreviated name that appears within the marker display bar. If an abbreviated name is not defined within a descriptive name, the descriptive name will be displayed within the marker display bar.

To change the marker color

1. From the Marker properties tab, select the Marker you want to change.
2. Select the marker Color button, and from the color palette that appears, select the new color. For custom colors, select the Other... button and create the desired color from the color dialog that appears.

To hide the marker

1. From the Marker properties tab, select the Marker you want to hide.
2. Check the Hide box. When a marker is hidden, all assigned marker properties are retained, it is just hidden from view in the display.

To lock marker in viewer

When viewing data in multiple display viewers, you can make a marker in one display move and remain viewable in another display. In other words, when a marker is locked in the viewer, it will follow the movement of the same maker as it is moved in another viewer.

This applies only when a marker is dragged within the immediate data viewing area. If a marker is moved by defining a new location using the markers properties dialog, it is not guaranteed to stay in view within any of the viewers.

1. From the Marker properties tab, select the Marker you want to lock in the display.
2. Check the Lock in Viewer box, then click Ok.

To change the marker type

1. From the Marker properties tab, select the Marker you want to change.
2. In the "Position" area of the properties dialog, select the marker type name (Example; Time) and from the selection list that appears, select the new type. When the marker type is changed between Time and Sample, the marker position in the data does not change, just the unit of measure.

See Also

To create new markers

To place markers in data

Go To markers

To center the display about a marker

To delete a marker

To rename a marker

To sent a marker to the back

To drag and drop markers in data

Reading off-screen markers

To drag and drop markers in data

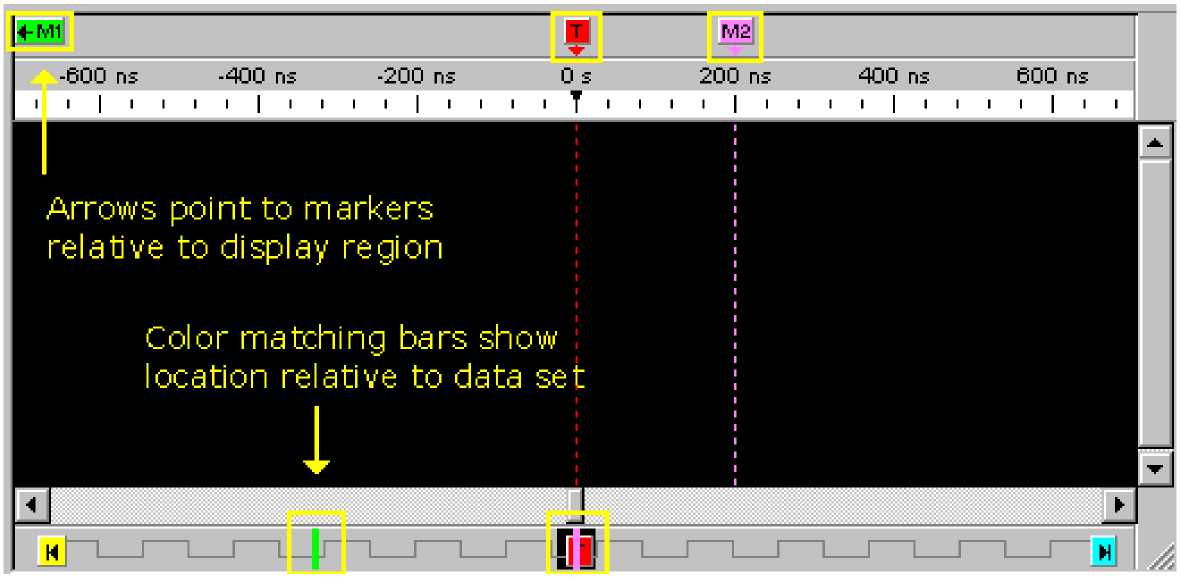
Using the drag and drop feature you can move markers to new positions in the data.

1. Click and hold down the mouse button on the marker you wish to move.
2. Move the mouse cursor to the new position.
3. Release the mouse button to reposition the marker.

See Also

To create new markers
To place markers in data
Go To markers
To center the display about a marker
To delete a marker
To rename a marker
To sent a marker to the back
To set marker properties
Marker snap to edge

Reading off-screen markers



In the upper marker display bar, markers are color coded and displayed with arrows that point to the marker's relative location to the locally (current window) displayed data. In the lower marker display bar, markers are displayed as color coded bars showing relative location within the data record (global).

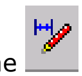
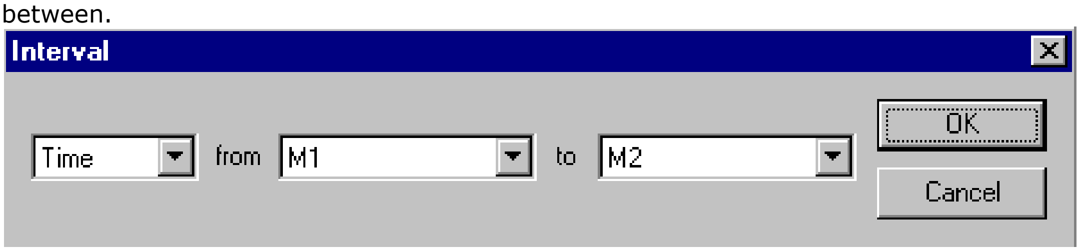
In the waveform display as shown above, the marker display bars appear on the top/bottom of the display window. In the listing display, the marker display bars appear on the left/right sides of the display window in a similar way.

Tip: You can quickly display a different region of data from the data set by double-clicking on the global marker display bar at the bottom (waveform) or right side (listing).

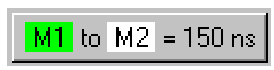
To create a new interval measurement

Use the new interval measurement feature to measure a time interval, or the number of samples between two specified points in data. Measurement results are displayed in the marker measurement display bar.

Note: Marker measurements use either the default M1 and M2 markers, or any other user-defined markers.

1. From the menu bar select **Markers>New Interval Measurement,** or click the  icon in the markers toolbar.
2. Select either time or sample, then select the markers you want to measure between.


3. Click **OK**.

The result of the interval measurement 

is displayed in the marker measurements display bar.

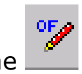
See Also

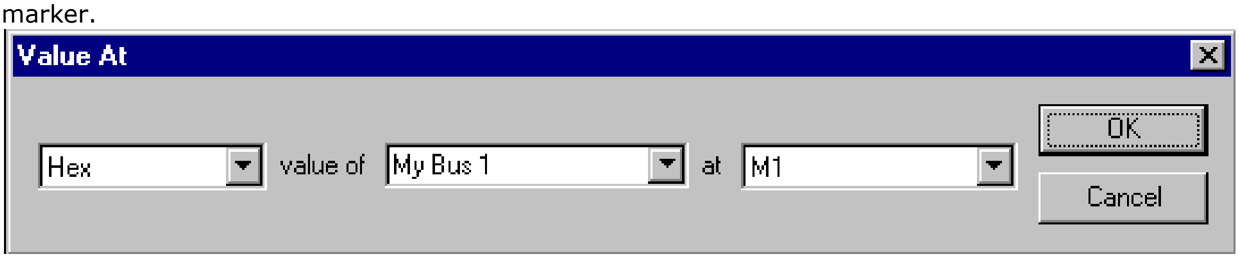
To create a new value at measurement

To create a new value at measurement

Use the new value at measurement feature to measure the value of a bus or a single signal at a specified marker location in data. Measurement results are displayed in the marker measurement display bar.

Note: Marker measurements use either the default M1 and M2 markers, or any other user-defined markers.

1. From the menu bar select **Markers>New Value At Measurement**, or click the  icon in the markers toolbar.
2. Select the numeric base of the data, the bus or signal source, then the marker.



3. Click **OK**.

The result of the value at measurement 

is displayed in the marker measurement display bar.

See Also

To create a new interval measurement

Displaying Names (Symbols) for Bus/Signal Values

Printing Captured Data

To print data and screens

There are three ways to create printed documentation of your measurement.

- Print data

  From the logic analyzer's File>Print dialog, print the current measurement data in memory from either the listing or waveform display viewer windows. You can print All data, or print just a defined range of Time (waveform) or Samples (listing). Data is printed from smallest time/sample to largest.

- Copy text to clip board

  When in the listing display window, you can draw a rectangle in the listing display area, then copy the region. Text is copied to the system clip board. To print the text, you must paste the text into a word processor or excel spreadsheet and print from that application.

- Copy screen to clip board

  Use the Edit>Copy Screen function to copy a bitmap of the active screen into the system clip board. To print the screen, you must paste the bitmap into a graphics editing program and print from that application.

Note: The first time you access the print dialog, you are asked to install a printer. Follow the directions in the printer install dialogs that appear.

To print data

1. From the menu bar, select File>Print... .
2. In the "Print What" section, select the desired display viewer window.
3. Click Options..., then select the desired bus/signals to print. To select a bus or signal, click on the name in the left pane, then click the right-arrow button to move the selection to the right pane. Click Ok.
4. In the "Print range" section, set the desired range using the "from" and "to" values.
5. Click **OK** to print the specified data.

To copy and print text

1. From the listing display area, position the mouse cursor over the upper-left corner of the desired display region.
2. Click and hold the left mouse button, then drag the mouse cursor to the lower-right corner. Release the mouse button. A rectangle is drawn around the defined region (snaps to state lines and bus/signal columns).
3. From the shortcut list that appears, click Copy Text.
4. Open a word processor or excel spreadsheet program, paste the text into the program, then print the screen.

To copy and print a screen

1. Click Edit>Copy Screen. The currently displayed window is copied into the windows

clip board buffer.

2. Paste the contents of the clip board buffer into a graphics editing program of your choice.
3. Print the screen from the graphics program.

See Also

To Install a printer

To connect a LAN

To install a printer

Local and network printers are installed outside of the logic analyzer environment using the Windows printer install wizard.

1. Click Start>Settings>Printers.
2. Click on an existing printer, or click Add Printer.
3. Follow the Windows printer install wizard instructions.

To connect a LAN

Local area networks (LAN) are install outside of the logic analyzer environment using the Windows network configuration wizard.

1. Click Start>Settings>Network and Dial-up Connections.
2. Click on an existing connection, or click Make New Connection.
3. Follow the Windows network install wizard instructions.

Solving Problems

Interpreting Error Messages
Translating 167xG-Series Logic Analyzer Configurations
Running the Logic Analyzer Self Tests

Error Messages

To locate the error you received, use ctrl+F and search for a key word.

Error messages
Warning messages
Informational messages

Error messages

Acquisition errors

An acquisition error has occurred due to state clock edges occurring too close together. This could be the result of: - Poor state clock quality (signal integrity). - Inadequate probe grounding (try multiple grounds around clock signals). - State clock edges spaced closer than specifications allow. - Multiple clocks selected and spaced closer than specifications allow.

When in the state acquisition mode, the logic analyzer requires a clear clock signal no faster than 200 MHz (edges at least 2.5 ns apart). Poor state clock quality may be caused by loading on the target system. It may also be caused by a clock setup in the Sampling Menu that is a combination of several signals which combined together violate the clock specification. When your clock setup uses multiple edges, the analyzer cannot sample faster than 3.0 ns. When you are using a clock speed near the specification, grounding every second or third probe connection is recommended.

Bus/Signal errors

Maximum of 128 channels per Bus.

The logic analyzer cannot handle buses that contain more than 128 channels (signals). If you require wider buses, try breaking the bus into two or more buses, for example Data_HI and Data_LO.

Cannot group into Bus. Maximum of 128 channels per Bus.

The logic analyzer cannot handle buses that contain more than 128 channels (signals). If you require wider buses, try breaking the bus into two or more buses, for example Data_HI and Data_LO.

The following Bus/Signals are required to have a specific number of assigned channels because they are locked. Please correct the following Bus/Signals: name (has num1 channels, requires num2 channels)

Some tools may "lock" buses and signals that are necessary to produce their output. The locked buses and signals may have their specific channel assignments changed, but the total number of channels on each bus or signal must stay the same. Please change the channel assignment for each indicated bus or signal so that the width is num2. This message sometimes appears in combination with the next one. In these cases, you may have changed a configuration to use half the pods for sampling. Check the sampling tab.

Every Bus/Signal requires at least one assigned channel. Please assign channels to the following Bus/Signals:

Every bus or signal requires at least one channel. If you do not see the Bus/Signal named in the error dialog, try scrolling the Bus/Signal listing. Certain tools may also have created buses or signals within folders. If you are trying to avoid showing extra information on the viewer, delete the row or column the bus or signal is in. This removes the information from the viewer without losing the bus/signal setup information. This message sometimes appears in combination with the previous one. In these cases, you may have changed a configuration to use half the pods for sampling. Check the sampling tab.

Minimum of one Bus/Signal with assigned channels required. Please add a Bus/Signal.

The bus/signal setup cannot be closed because you have deleted all buses and signals. Folders only contain buses and signals, but do not represent data mappings of themselves. In order to close the dialog, select Add Bus/Signal. Assign at least one channel to the new bus or signal. Alternatively, you can select Cancel and revert to the previous bus and signal assignments.

name is locked and cannot be deleted because it is required by another tool in the application. In order to unlock it, the following tools must be deleted: tool

Some tools may "lock" buses and signals that are necessary to produce their output. Until the tool is deleted via Tools>Overview, you cannot delete or rename the bus or signal.

name is locked and cannot be renamed because it is required by another tool in the application. In order to unlock it, the following tools must be deleted: tool

Some tools may "lock" buses and signals that are necessary to produce their output. Until the tool is deleted via Tools>Overview, you cannot delete or rename the bus or signal.

Cannot change pod selection while there are locked Bus/Signals.

Some tools may "lock" buses and signals that are necessary to produce their output. Until the tool is deleted via Tools>Overview, you cannot modify the bus or signal by changing the pod in use.

Cannot delete folder because it contains one or more locked children.

Some tools may "lock" buses and signals that are necessary to produce their output. The folder you have tried to delete contains a unique copy of at least one locked bus or signal. You can move the locked buses or signals outside the folder, and then delete the folder. Alternatively, you can delete the tool locking the buses or signals via Tools>Overview, and then delete the folder.

Unable to set setup and hold times for this Bus/Signal since no channels have been assigned. Please assign channels before using setup and hold.

The possible valid range of setup and hold values depends on the clock setup used by the pods that the channels are attached to. Without knowing which pods' channels are part of the bus or signal, it is impossible for the logic analyzer to set appropriate ranges. Please assign channels to the bus or signal, and then set setup and hold.

Please enter a user threshold value.

All pods will be set to the same threshold value. If you select OK without setting a value, the current threshold values (at least one of which is different from the rest) are retained. Please check the dialog and be sure all fields are filled in.

Please enter a threshold value.

All pods will be set to the same threshold value. If you select OK without setting a value, the current threshold values (at least one of which is different from the rest) are retained. Please check the dialog and be sure all fields are filled in.

File errors

Error trying to remove file: "directory/hardware_log.txt".

When the logic analyzer is started up, it replaces the old hardware_log.txt file. For some reason, this time the old hardware log was not able to be deleted. This could indicate a problem with the disk that the file is stored on.

File "filename" could not be opened.

When the logic analyzer is started up, it creates a new hardware_log.txt file. For some reason, this time the log was not able to be opened after creation. This could indicate a file system or disk problem.

Hardware errors

Analyzer Calibration Failed [time] - Instrument may need service

The logic analyzer's pre-measurement calibration failed. Any data collected after receiving this error message is possibly incorrect. If the failure is transient, cycling power may fix the problem. If the failure is persistent, run Help>Self-test>Analyzer self-test or call your Agilent Sales Office to arrange for service.

High speed system clock failure - Instrument may need service. Sleep Duration and Count: duration, counted

The internal 100 MHz clock did not pass initialization tests. Any measurements are likely to be faulty. Please contact Agilent Technologies sales or support at http://www.agilent.com/find/contactus for information on getting the instrument repaired.

Contact with the analyzer hardware has been lost. This application will be terminated. You will have an opportunity to save your configuration.

[1690A-series analyzers only] Something has interrupted the IEEE-1394 connection between the computer and the logic analyzer. Save your current work in a configuration (*.ala) file, then check the power to the logic analyzer hardware and the connections. A lost connection cannot be resumed; you will need to re-start the logic analyzer application.

I/O Channel Error: Invalid Request Argument.

[1690A-series analyzers only] There is a problem with the data being sent via the IEEE-1394 connection.

I/O Channel Error: Offset Out of Bounds.

[1690A-series analyzers only] There is a problem with the data being sent via the IEEE-1394 connection.

I/O Channel Error: Timeout.

[1690A-series analyzers only] There is a problem with the data being sent via the IEEE-1394 connection.

I/O Channel Error: System I/O Error.

[1690A-series analyzers only] There is a problem with the data being sent via the IEEE-1394 connection.

I/O Channel Error.

[1690A-series analyzers only] There is an unspecified problem with the data being sent via the IEEE-1394 connection.

Help file errors

Help file information was not found in the registry. You may need to reinstall the tool.

The logic analyzer could not find a registry entry for the help file associated with the tool. If you have done a custom installation of the tool, you must also install the help file to access help. If the problem persists after re-installing, please contact Agilent Technologies sales or support at http://www.agilent.com/find/contactus for assistance.

Help file information not found in registry. Cannot display help.

The logic analyzer could not find a registry entry for the help file associated with the tool. If you have done a custom installation of the tool, you must also install the help file to access help. If the problem persists after re-installing, please contact Agilent Technologies sales or support at http://www.agilent.com/find/contactus for assistance.

Help file not found. Cannot display help.

The help file was not found where specified by the registry. It may have been deleted or moved. You can search the drive where the logic analyzer software is installed for .chm files, or re-install the tool.

The HTML Help file "filename" was not found. You may need to re-install the product.

The help file was not found where specified by the registry. It may have been deleted or moved. You can search the drive where the logic analyzer software is installed for the file, or re-install. To re-install, close the logic analyzer application and run the setup program on the logic analyzer CD.

The HTML Help file "filename" was not found. You may need to re-install the tool.

The help file for the tool was not found where specified by the registry. It may have been deleted or moved. You can search the drive where the logic analyzer software is installed for .chm files, or re-install the tool.

Import/Export and Translator errors

Refer to the import files: file1 and file2 for more details.

The configuration translator could not complete the translation. An explanation will be listed in file1 or file2 between "<!--" and ">" delimiters.

The specified file is NOT a 167xG Analyzer configuration file: filename

The configuration translator was not able to translate the specified file because it was not in an understood format. The configuration translator only translates configuration files generated by 1670G, 1671G, 1672G, and 1673G logic analyzers.

Cannot read configuration file filename

The configuration translator was unable to read the configuration file indicated due to an internal error in the configuration file.

Cannot import an empty file. Import terminated.

The file you tried to import has no content.

Invalid non-ascii character read. Import terminated.

The logic analyzer only imports ASCII files. The file you tried to import contains a non-ascii character. You can edit the file in any text editing program, such as Notepad, to remove the character. Be careful to not change header data or the number of samples.

There are one or more locked bus/signals required by tool(s) currently loaded in the application. In order to unlock any of these bus/signals, you must unload every tool listed for that bus/signal. ... Import terminated.

Some tools may "lock" buses and signals that are necessary to produce their output. Until the tool is deleted via Tools>Overview, you cannot delete, rename, or modify the bus or signal. A side effect of this is that you cannot import a file that uses different buses and signals, or analyzer channel count.

There were fewer time data samples than indicated by the NumberOfSamples attribute of Module tag. Import terminated.

When you import a saved data file, the logic analyzer verifies that the data is consistent. To fix this error, you can edit the file in any text editing program, such as Notepad.

Syntax error in import file: error. Import terminated.

The error description indicates the syntax problem. Most often, it results from mismatched tags. Check the import file for any accidental deletions.

No bus/signals were valid for importing. Import terminated.

The import file was created on a logic analyzer model with more pod pairs than this one, and all the buses and signals were defined on pod pairs this model does not possess. You can attempt to modify the file by changing the assigned channels for the buses and signals. You can use any text editing program (such as Notepad) to edit import files. For more information on the format of import files, see XML Format .

Could not create file for export

The logic analyzer was unable to create the file. Possible reasons include not enough disk space or insufficient permissions to create the file where indicated.

Could not find required tag1 section contained within tag2 section

The import file is required to have a section with the heading tag1 completely contained within the section delimited by <tag2> and </tag2>. You can repair the import file by adding <tag1></tag1> at the beginning of the tag2 section. You can use any text editing program (such as Notepad) to edit import files. For more information on the format of import files, see XML Format .

Invalid or missing attribute attribute in tag XML tag

The import file requires the XML tag tag to include the keyword attribute and a value. For example, if attribute is Acquisition and tag is Sampling, the file has an XML tag of the form <Sampling></Sampling> but requires <Sampling Acquisition="State"></Sampling>. You can repair the import file by adding attribute to the specified tag and giving it a value. (Try exporting a similar configuration to see standard values.) You can use any text editing program (such as Notepad) to edit import files. For more information on the format of import files, see XML Format .

Invalid attribute value for attribute attribute in tag XML tag

The import file requires the XML tag tag to include the keyword attribute and a value. For example, if attribute is Acquisition and tag is Sampling, the file might have an XML tag of the form <Sampling Acquisition="Time"></Sampling> but requires <Sampling Acquisition="Timing"></Sampling>. You can repair the import file by editing the value. (Try exporting a similar configuration to see standard values.) You can use any text editing program (such as Notepad) to edit import files. For more information on the format of import files, see XML Format .

Naming errors

To edit channel assignment, click on the pod cells to the right.

This error occurs when you select the Channels Assigned column in Bus/Signal setup. This column is for reference only. To assign channels to a bus or signal, use the grid to the right. For more detail on channel assignment, see To assign channels . To hide the Channels Assigned column, select Display> Channels Assigned in the Buses/Signals setup tab.

The object name cannot be an empty string.

When you rename a viewer, tool, or bus, you must give it a name at least one character in length. The blank or empty name you tried was not accepted by the logic analyzer.

Object name must be unique.

You have tried to rename a viewer, tool, or bus, but the name you entered is already being used and so was not accepted by the logic analyzer. You may appear to have identical names on some buses or signals, but these either are truncated or refer to the same bus (aliases).

The name "toolname" is already in use. Please choose a different name.

You have tried to rename a tool, but the name you selected is already in use. If you do not rename the tool, it will revert back to its previous name when you close the Tool Overview dialog.

Tool errors

For errors generated by specific inverse assemblers or bus analysis tools, go to the appropriate tool help.

Could not load the component - you may need to reinstall

This error occurs when there is a problem with the tool file. Possible reasons are the tool file was renamed, or permissions changed so that the logic analyzer cannot open it. To reinstall the tool file, close the logic analyzer and run the setup program on the logic analyzer CD.

Could not get license information for component

The logic analyzer attempted to determine if the tool was licensed or freely available, but could not find the information. Try re-installing the tool.

Could not obtain license information for component

This error means the logic analyzer is missing some information it needs in order to check the license. Licenses are created by the lmtools.exe program. You can run this to see what information is missing, and to check licenses.

Could not obtain a license for component

This error occurs when the license is not in the expected directory. When the tool is installed, the license is written into a predefined directory. Moving or deleting the file prevents you from using the component. If you do not believe the license was deleted, check your hard drive for *.lic files.

License for component is invalid

This error means that a license file exists, but that the information in it does not match. Licenses are specific to equipment; you cannot transfer a license for a tool or a logic analyzer between tools or logic analyzers.

Could not create licensed component

There was a valid license for the tool earlier in the install process, but something has gone wrong. Start the tool installation process over again. If this error persists, please contact Agilent Technologies sales or support at http://www.agilent.com/find/contactus for assistance.

Could not create component - unknown error

A license for the tool exists in the proper directory, but the internal information is inconsistent. Licenses are specific to equipment; you cannot transfer a license between tools or between logic analyzers. If this is a corrupt license, try reinstalling the tool again.

The stored tool could not be restored from file. The tool may have been uninstalled.

The configuration file you are trying to load includes a licensed tool. The logic analyzer was not able to find this tool, so some information will be missing. All buses and signals that were based on physical data will be loaded; buses and signals created by the tool will not.

The selected tool: Name could not be loaded.

The configuration file you are trying to load includes a licensed tool. The logic analyzer was not able to find this tool, so some information will be missing. All buses and signals that were based on physical data will be loaded; buses and signals created by the tool will not.

The selected tool could not be loaded.

The configuration file you are trying to load includes a licensed tool. The logic analyzer was not able to find this tool, so some information will be missing. All buses and signals that were based on physical data will be loaded; buses and signals created by the tool will not.

The tool could not be created - you may need to reinstall the tool.

The configuration file you are trying to load includes a licensed tool. The logic analyzer was not able to find this tool, so some information will be missing. All buses and signals that were based on physical data will be loaded; buses and signals created by the tool will not.

The analysis tool (toolname) could not be restored from the configuration file.

The configuration file you are trying to load includes a licensed tool. The logic analyzer was not able to find this tool, so some information will be missing. All buses and signals that were based on physical data will be loaded; buses and signals created by the tool will not.

Trigger Errors

Only one action per timer per branch is allowed.

A branch is the collection of actions after a "Then" in an Advanced Trigger step. Some steps, such as Advanced 2-Way Branch, may have multiple branches. Within a branch, only one of Start from reset, Stop and reset, Pause, or Resume is allowed per timer. For more on timers, see To configure a timer.

Only one action per counter per branch is allowed.

A branch is the collection of actions after a "Then" in an Advanced Trigger step. Some steps, such as Advanced 2-Way Branch, may have multiple branches. Within a branch, you can not both Increment and Reset the same counter. You can increment one and reset the other. For more on counters, see To configure a counter.

Only one store action per branch is allowed.

A branch is the collection of actions after a "Then" in an Advanced Trigger step. Some steps, such as Advanced 2-Way Branch, may have multiple branches. Within a branch, you can set Store sample or Don't store sample but not both in the same branch. If you do not specify any store actions, default storage is used.

Only one reset occurrence counter action per branch is allowed.

A branch is the collection of actions after a "Then" in an Advanced Trigger step. Some steps, such as Advanced 2-Way Branch, may have multiple branches. Within a branch, you can only specify Reset occurrence counter once.

No more edge resources available for this pod pair

The logic analyzer hardware can only handle two edge statements per pod pair in 400 MHz or 800 MHz timing mode, or one edge statement per pod pair in transitions only timing mode. The pod pairs are pod 1 & pod 2, pod 3 & pod 4, pod 5 & pod 6, and pod 7 & pod 8 (some models have fewer pods). If the edges are on different signals, try probing one of the signals with another channel on another pod pair. If all the edges are being used on the same signal, replace the "either edge" terms with "rising edge OR falling edge". See To insert events for how to replace "either edge".

No more pattern resources available for this pod pair

The logic analyzer hardware has a limited number of pattern (bus value) variables per pod pair. The pod pairs are pod 1 & pod 2, pod 3 & pod 4, pod 5 & pod 6, and pod 7 & pod 8 (some models have fewer pods). If the values you are checking for are on different buses, try probing one of the buses with another pod pair.

Branch expression is too complex

The expression in one of the branches of the trigger specification is too complicated for the logic analyzer. The logic analyzer first combines all AND terms and then ORs the expressions together. AND terms that have more than 4 events use twice the resources. Try rewriting the branch expression to use more OR terms, or delete some events.

Trigger Specification is too complex

Although no single branch expression is too complex, the total number of ANDs and ORs has exceeded the logic analyzer's resources. Try simplifying some expressions in some steps, or removing steps altogether.

Replacement failed. Maximum number of sequence levels exceeded.

The logic analyzer translates the trigger you specified into internal sequence levels. Different trigger functions use different numbers of internal sequence levels. Also, the "trigger and fill memory" action requires an additional internal sequence level each time it is used in state acquisition mode. One possible way to simplify the trigger specification is to replace all other "trigger and fill memory" actions with a "goto N" action that points to a "Find anything then trigger and fill memory" step.

Unable to insert level. The maximum number of sequence levels are already allocated.

The logic analyzer translates the trigger you specified into internal sequence levels. Different trigger functions use different numbers of internal sequence levels. Also, the "trigger and fill memory" action requires an additional internal sequence level each time it is used in state acquisition mode. One possible way to simplify the trigger specification is to replace all other "trigger and fill memory" actions with a "goto N" action that points to a "Find anything then trigger and fill memory" step.

Too many sequence levels.

The logic analyzer translates the trigger you specified into internal sequence levels. Different trigger functions use different numbers of internal sequence levels. Also, the "trigger and fill memory" action requires an additional internal sequence level each time it is used in state acquisition mode. One possible way to simplify the trigger specification is to replace all other "trigger and fill memory" actions with a "goto N" action that points to a "Find anything then trigger and fill memory" step.

Goto action specifies an undefined level.

The last step in the trigger sequence includes the action "Goto next". Because there is no next level, the logic analyzer cannot run and look for a trigger. Select Setup>Advanced Trigger, and change the action for the last trigger step.

Counter event specified both true and false in the same product term

In the trigger specification, at least one branch ANDs together "bus equals X" and "bus not equal X". Because this condition can never be true, the logic analyzer will not trigger and does not start the acquisition. If you intend to have it run until you press stop, use the trigger function Run Until User Stop, found under the "Other" tab in advanced trigger.

Cannot use <,<=,>,>= for a bus with clock bits that spans pod pairs

You have defined a bus that both spans pod pairs, and includes a clock bit. The pod pairs are pod 1 & pod 2, pod 3 & pod 4, pod 5 & pod 6, and pod 7 & pod 8. (Some models have fewer pod pairs.) The clock bits are numbered the same as the pod they are located on, and it is possible for them to be the channel that is not on the same pod pair as the others. Check the channel assignment in the bus/signal setup window. The logic analyzer will not run until this problem is corrected.

Cannot specify a range on a bus with clocks bits that spans pod pairs

You have defined a bus that both spans pod pairs, and includes a clock bit. The pod pairs are pod 1 & pod 2, pod 3 & pod 4, pod 5 & pod 6, and pod 7 & pod 8. (Some models have fewer pod pairs.) The clock bits are numbered the same as the pod they are located on, and it is possible for them to be the channel that is not on the same pod pair as the others. Check the channel assignment in the bus/signal setup window. The logic analyzer will not run until this problem is corrected.

Warning Messages

You are currently running "Offline," so running the analyzer is not possible. If you wish to create "fake" data while offline, go to "Edit -> Options" and select "Create Data When Offline". Note: This setting is persistent from session to session.

The logic analyzer is running in offline mode. Offline mode means that the logic analyzer software does not have access to the 1680 or 1690-series hardware. All logic analyzer software loaded onto PCs without a 1690-series logic analyzer is in offline mode. If you have a logic analyzer attached, please check the connection. For more on running with fake data, see System Options Dialog. Fake data is useful when learning how to use the logic analyzer software.

This module is already being used by another instance of the application. You are now working Offline.

The logic analyzer hardware is attached to an open instance of the logic analyzer software. If you need to acquire data, locate that instance from the Windows taskbar. The online/offline indicator is at the bottom of the application window. In offline mode, the software can still work with saved data.

Event specified both true and false in the same product term

In the Advanced Trigger dialog, one of the branches for one of the steps checks that an event is both true and not true. An event may be a bus or signal equal to a value, a timer expiring, or a count exceeding some value. Because of the AND combination, the branch cannot be true. You may want to modify the trigger to use either an OR combination of the events, or separate them into different branches or steps. For more on constructing complex triggers, see To build a trigger sequence. For more on how to interpret the trigger sequence, see How to read event and action statements.

Timer n value checked as an event, but no start action specified.

In the Advanced Trigger dialog, the trigger sequence checks the value of a timer that was never started. Timers need to be explicitly started in a previous trigger step. See To configure a timer for more information.

Counter n value checked as an event, but no increment action specified.

In the Advanced Trigger dialog, the trigger sequence checks the value of a counter that is never incremented. Counters need to be incremented in the action statements of a trigger step. See To configure a counter for more information.

The Bus/Signals listed below could not be loaded from the configuration file. Please recheck your Trigger since it may have changed.

The configuration file you just loaded was created on a logic analyzer with more pods than this model. Because of this, some buses and signals which rely on the additional pods could not be loaded. If these buses or signals were used in the trigger sequence, the trigger sequence will have changed. You may be able to work around this by assigning different channels to the affected buses and signals, and re-creating the trigger sequence.

Slow or missing clock in Trigger Step n...

The logic analyzer is not able to detect the state clock, and is therefore unable to take samples and evaluate the trigger sequence. If your target system's clock is bursty, this may be expected behavior. If it is not, please check all probing connections. To verify the clock signal is being received, you can assign the clock channels to a bus in timing acquisition mode and acquire data.

Informational messages

Filling memory after trigger...

The logic analyzer has triggered and is filling memory. Due to either a slow clock or storage qualification in state acquisition mode, or infrequent transitions in transitions-only timing acquisition mode, the logic analyzer is taking enough time to fill memory that this message is showing.

Trigger inhibited during prestore...

The logic analyzer is in timing acquisition mode. In timing acquisition mode, the logic analyzer fills the designated amount of memory before searching for the trigger. If this message is showing, the logic analyzer is filling memory and has not yet begun to compare data to the trigger sequence. To capture triggers that happen during the beginning of a target system's boot sequence, be sure to set the trigger position in the Sampling tab to 100% poststore.

Waiting in Trigger Step n

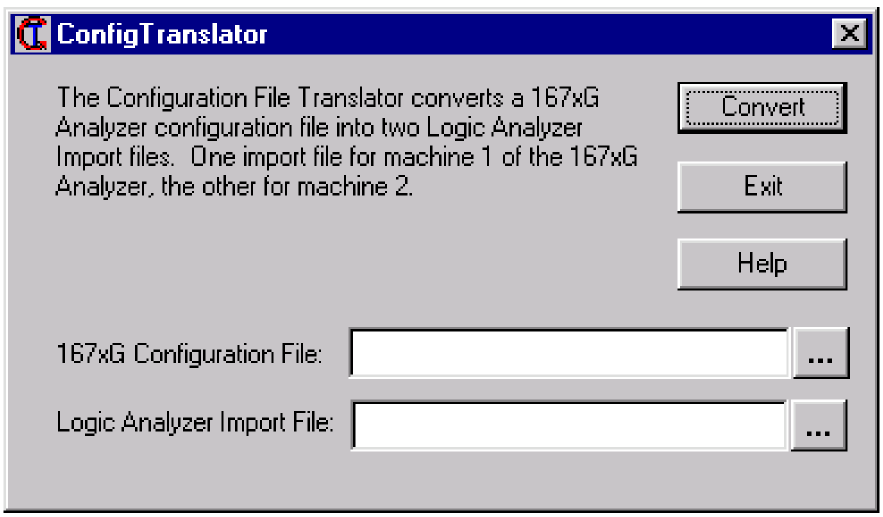
The logic analyzer is waiting for a sample that matches the events defined in step n of the trigger sequence. Sometimes the event is rare, causing long waits. If you feel that the logic analyzer should have triggered already, check the trigger sequence in Advanced Trigger. For more on triggering, see Advanced Trigger Dialog.

Config Translator Application

Config Translator is a helper application for the Agilent Logic Analyzer. In the first release, shipped with your logic analyzer, it only translates 1670G-series Logic Analyzer configuration files. You can download updates along with updated logic analyzer software at our website, http://software.cos.agilent.com/1680_1690/.

To run Config Translator, select the Agilent Logic Analyzer menu from the Windows2000 Start bar. Config Translator can be run without any logic analyzer hardware.

To convert a configuration file



1. Open Config Translator by selecting Start>Programs>Agilent Logic Analyzer>Agilent 167xG Config Translator.
2. Type in the name of the configuration file you want to convert. Use the ... button to browse if you are unsure of the name.

> Note: 167xG configuration files end in ._A, but this suffix is also used by other models. If possible, confirm that it has a file type of 167xdn_config when viewed on a 1670-series logic analyzer.
>
> Only setup information is translated, not saved data.

3. Type in the name you want to save the new configuration files under.
    1. Because the 1670-series logic analyzers split resources between two measurement engines by default, two output files are created. The default filename is the same as the 167xG configuration file, but has 1.txt or 2.txt appended. Both files are created even if the configuration only used one of the measurement engines.
    2. Files are saved in the same directory as the input file unless otherwise specified.
    3. The output files are in XML format, and can be opened like other XML configuration files.

4. Select Convert.

Information not converted from file

- In timing acquisition mode, the sampling period and sampling option.
- In state acquisition mode, the clock mode and clock description.
- All data.
- Any tool information.
- Interface layout.
- Marker information.

See Also

XML Format

To open a configuration file

Self Tests

The Self Test menu checks the major hardware functions of the analyzer to verify that the analyzer is working correctly. A description of each test is found below.

If after completing the self tests you have failures, or you have questions about the performance of the logic analyzer, contact Agilent Technologies sales or support at http://www.agilent.com/find/contactus.

For more contact information, refer to Product overview.

To access self test menu

1. From the menu bar select Help>Self Test>Analyzer Self Test.
2. The first time the self test menu is accessed, a warning message comes up, "Running self tests will invalidate acquired data." Select OK to bring up the menu.
3. Select either a single test or to run all tests. As the test is running, test progress is reported in the lower art of the dialog.

Self test descriptions

Register Test

Memory Test

Comparator Test

Trigger Bus Test

Trigger Arm Test

Clock Paths Test

Memory Modes Test

Calibration Test

Register Test

The Register Test verifies that the registers of each acquisition IC are operating properly. Test patterns are written to each register on each acquisition IC, read, and compared with known values. The registers are reset, and verified that each register has been initialized. Test patterns are then written to ensure the chip address lines are not shorted or opened. Finally test data is written to registers of individual acquisition ICs to ensure each acquisition IC can be selected independently.

Passing the Register Test implies that the acquisition IC registers can store acquisition control data to properly manage the operating of each IC.

Memory Test

The Memory Test verifies that each bit in the acquisition memory IC can be written with a logic “0” and logic “1” through the Serial Access Memory port. Test data is generated using a shifting test register in the acquisition ICs. The serialized test patterns are then sent to the memory port of each acquisition memory IC and stored. The data in the acquisition memory ICs are then downloaded and compared with known values.

Passing the Memory Test implies the acquisition memory can store data written through the memory port. This test along with the Memory Modes Test provides complete testing of the memory ICs.

Comparator Test

The Comparator Test ensures the data signal comparators in the module front end can be set to their maximum and minimum thresholds and that they recognize activity at the signal inputs. A clock signal is routed to a test port on each comparator. The threshold is then set to the minimum value. The comparator output is then read and compared with a known value. The threshold is then set to a maximum value. The comparator output is again read and compared with a known value.

Passing the Comparators Test implies that the front-end comparators are operating properly, can recognize both a logic “0” and logic “1”, and can properly send the acquisition data downstream to the acquisition ICs.

Trigger Bus Test

The Trigger Bus Test verifies the trigger resource lines that run between each acquisition IC. The test ensures that the trigger resource lines can be both driven as outputs and read as inputs. The resource registers are written with test patterns, read back, then compared with known values. The resource registers are then written with test patterns, read back from a different acquisition IC, then compared with known values.

Trigger Arm Test

The Trigger Arm Test verifies that the local arm signal can be received by the master acquisition IC on the acquisition board. The test also verifies the global arm signal can be driven by each acquisition IC on a master board and received by all acquisition ICs on the card. The arm lines are asserted and read at the acquisition ICs to ensure each acquisition IC recognizes the signal.

Passing the Trigger Arm Test implies any acquisition IC can arm the card and that all acquisition ICs can recognize the arm signal.

Clock Paths Test

The Clock Paths Test verifies that the system Master, Slave, and Psync clocks are functional between the acquisition ICs. The module is configured to take a simple measurement. Test data is then created at the comparators and an acquisition taken. The resulting data is then downloaded and compared with known values.

Passing the Clock Paths Test implies that all acquisition IC clock lines can be driven by each acquisition IC and can be received by each acquisition IC in the module. Consequently each acquisition IC can reliably acquire data in response to the acquisition clock signal

Memory Modes Test

The Memory Modes Test verifies the CPU interface can properly manage the acquisition memory unload in full-channel, half- channel, count only, and interleaved modes. Test data is written to acquisition memory. Different unload modes are selected, then the data is read and compared with known values.

Passing the Memory Modes Test implies that the data can be reliably read from acquisition memory in full-channel, half-channel, count only, and interleaved mode. This test along with the Memory Test provides complete testing of acquisition memory downloading through the 1394 interface.

Calibration Test

The Calibration Test ensures that each acquisition IC in the module can perform an operational accuracy self-calibration. Various self-calibration routines are initiated. The results of each self-calibration routine are then checked to see if the self-calibration was successful or not.

Passing the Calibration Test implies that the module can reliably perform an operational accuracy self-calibration. Consequently the incoming data path is optimized to reduce channel-to-channel skew so the acquisition ICs can reliably capture the incoming data.

Concepts

Logic analysis basics

- When should I use an oscilloscope
- When should I use a logic analyzer
- What is a logic analyzer
    - Timing analyzer
        - Clocking
        - Sampling
        - Triggering
    - State analyzer
        - Clocking
        - Sampling
        - Triggering
- Probing options

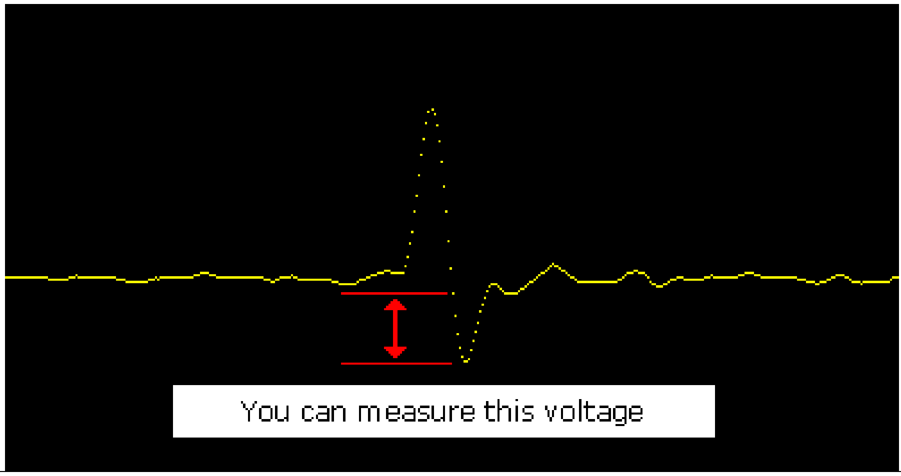
When should I use an oscilloscope

Tutorial Home | Next Topic | Previous Topic

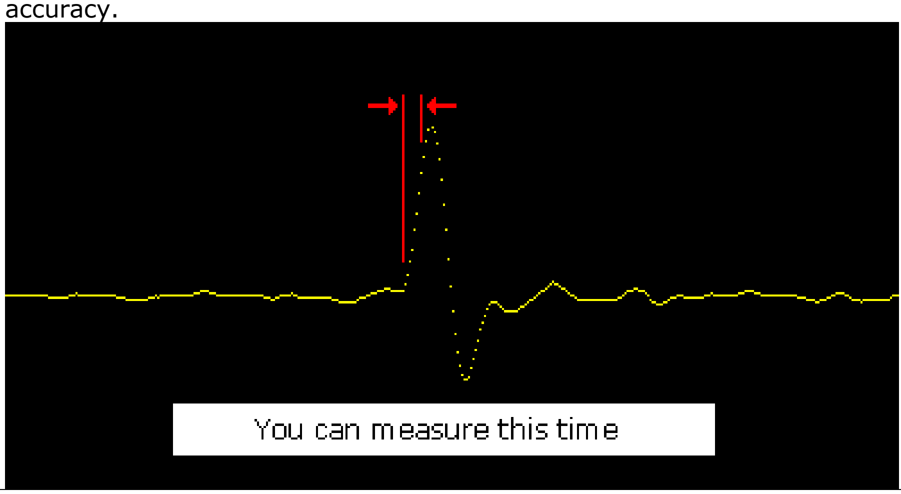
Generally, an oscilloscope is used when you need precise parametric information such as time intervals and voltage readings.

More specifically:

- When you need to measure small voltage excursions on your signals such as undershoot or overshoot.



- When you need high time-interval accuracy. Oscilloscopes can capture precise parametric information such as the time between two points on a rising edge of a pulse with very high accuracy.



When should I use a logic analyzer

Tutorial Home   Next Topic   Previous Topic

Generally, a logic analyzer is used to view timing relationships among many signals, or if you need to trigger on patterns of logic highs and lows. A logic analyzer reacts the same way as the logic circuits do when a voltage threshold is crossed by a signal in the device under test. It will recognize the signal to be either low or high.

More specifically:

- When you need to see many signals at once. Logic analyzers are very good at organizing and displaying multiple signals. A common task is to group multiple signals into a bus and assign a custom name. Good examples are address, data, and control buses.
- When you need to look at signals in your system the same way your hardware does. Signals are displayed on a time axis so you can see when transitions occur relative to other bus signals or clock signals.
- When you need to trigger on a unique bus pattern or signal edge. Logic analyzers can be configured to store data when the high or low values of a group (bus) of signals match a predefined pattern. Logic analyzers can be configured to store data when a specific edge or level is detected on a single signal.

What is a Logic Analyzer

Tutorial Home | Next Topic | Previous Topic

Now that we've talked a little about when to use a logic analyzer, let's look in more detail at what a logic analyzer is. Up to now, we've used the term "logic analyzer" rather loosely. In fact, most logic analyzers are really two analyzers in one.

What is a timing analyzer

A timing analyzer is the part of a logic analyzer that is analogous to an oscilloscope. As a matter of fact, they can be thought of as close cousins.

The timing analyzer displays information in the same general form as a scope, with the horizontal axis representing time and the vertical axis as voltage amplitude. Because the waveforms on both instruments are time-dependent, the displays are said to be in the "time domain".

The basic areas of functionality in a timing analyzer are as follows:

Clocking data in the timing analyzer

Sampling in the timing analyzer

Triggering the timing analyzer

What is a state analyzer

A state analyzer is very good at tracking down bugs in software or defective components in hardware. It can help eliminate the question whether a problem is in the software code or some hardware device.

Most often, state analyzers are used to find out what logic levels are present on a bus when a particular clock signal occurs. In other words, you want to know what "state of activity" is present when the clock occurs and data is suppose to be valid. Data captured in memory is displayed in a listing format with a time tag attached to every state.

The basic areas of functionality in a state analyzer are as follows:

Clocking data in the state analyzer

Sampling in the state analyzer

Triggering the state analyzer

Clocking data in the timing analyzer

Tutorial Home | Next Topic | Previous Topic

The timing analyzer uses its own internal clock to control the sampling of data. This type of clocking makes the sampling of data in the logic analyzer asynchronous to the clocking in the device under test.

More specifically:

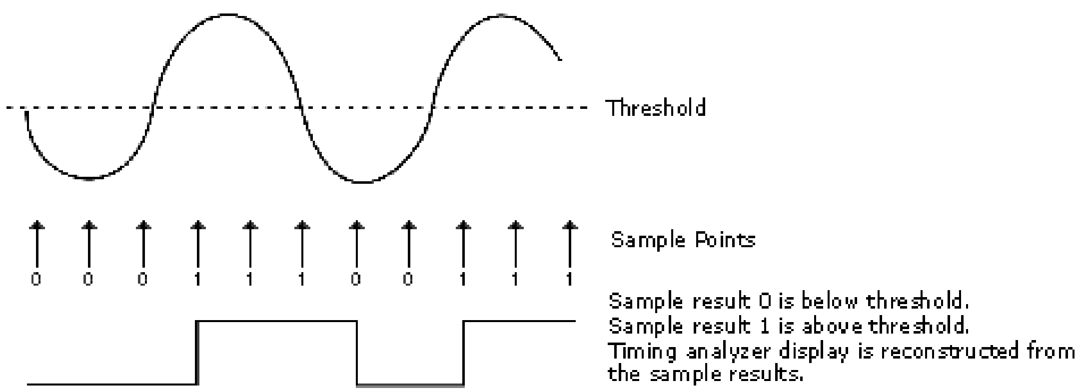
- A timing analyzer is good at showing you "When" signal activity occurs "Relative to other signals".
- A timing analyzer is more interested in viewing the timing relationships between individual signals, than the timing relationships to the signals that are controlling execution in the device under test.
- This is why a timing analyzer can sample data "out of sync", or asynchronous to the target system clock signals.

Sampling in the timing analyzer

Tutorial Home | Next Topic | Previous Topic

The timing analyzer works by sampling the input waveforms to determine whether they are high or low. It determines a high or low by comparing the voltage level of the incoming signal to a user-defined voltage threshold. If the signal is above that threshold when it samples, it will be displayed as a 1 or high by the analyzer. By the same criterion, any signal sampled that is below threshold is displayed as a 0 or low.

The figure below illustrates how a logic analyzer samples a sine wave as it crosses the threshold level.

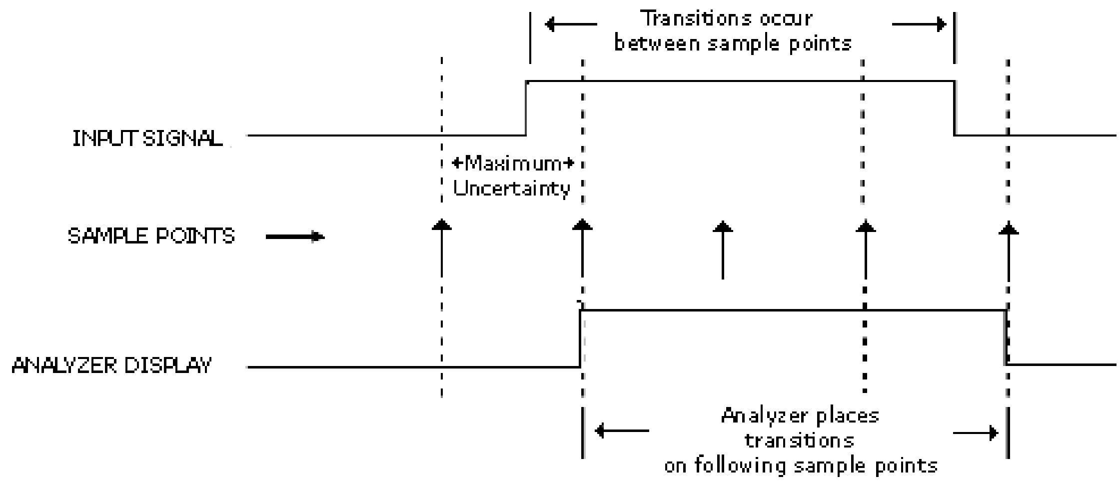


The sample points are then stored in memory and used to reconstruct a more squared-off digital waveform.

This tendency to square everything up would seem to limit the usefulness of a timing analyzer. However, a timing analyzer is not intended as a parametric instrument. If you want to check rise time of a signal with an analyzer, you should use a scope. But if you need to verify timing relationships among several or hundreds of signals by seeing them all together, a timing analyzer is the right choice.

Sampling accuracy

When the timing analyzer samples an input channel, it is either high or low. If the channel is at one state (high or low) on one sample, and the opposite state on the next sample, the analyzer "knows" that the input signal has transitioned sometime between the two samples. It doesn't know when, so it places the transition point at the next sample, as shown in the figure below.



This presents some ambiguity as to when the transition actually occurred and when it is displayed by the analyzer.

Worst case for this ambiguity is one sample period, assuming that the transition occurred immediately after the previous sample point.

With this technique however, there is a trade-off between resolution and total acquisition time. Remember that every sampling point uses one memory location. Thus, the higher the resolution (faster sampling rate), the shorter the acquisition window.

Triggering the timing analyzer

Tutorial Home | Next Topic | Previous Topic

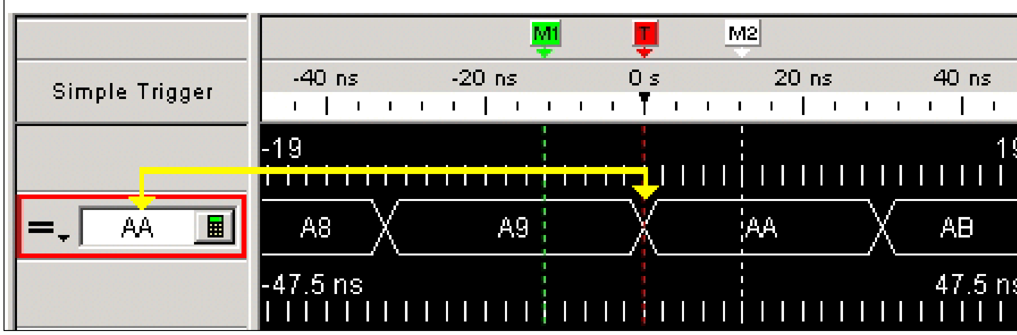
At some point in a measurement, the logic analyzer has to know when to capture (store) the data that is flowing through its memory. This is know as the trigger point.

One way to get the analyzer to trigger is to configure the analyzer to look for either a pattern of highs and lows from a group of signals (bus), or a rising or falling edge from a single signal. When the analyzer sees the specified patterns or edges in data, it triggers.

Pattern Trigger

Pattern triggers are used to find specific patterns of highs and lows across a bus. You can specify different kinds of criteria such as equal, not equal, in or out of a range, or greater than/less than.

Example: You have a bus containing 8 signal lines. You configure the Simple Trigger to specify that the analyzer triggers when the incoming data is equal to a pattern of "AA".



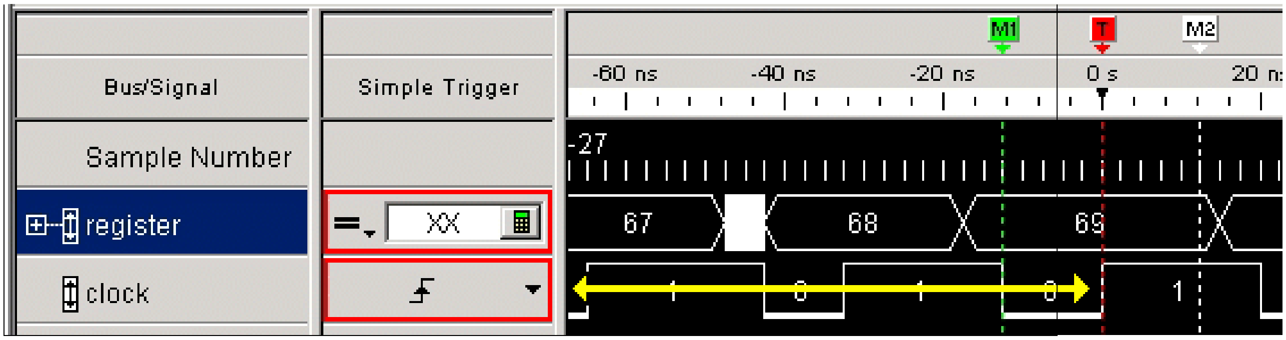
To make things easier for some users, the trigger point on most analyzers can be set not only in Hex, but in binary (1's and 0's), octal, ASCII, or decimal. For instance, the Hex trigger value of AA could also be set to an equivalent binary trigger value of 0101 0101. However, using hex for the trigger point is particularly helpful when looking at buses that are 16, 24, 32, or 64 bits wide.

Edge Trigger

Edge triggering is a familiar concept to those accustomed to using an oscilloscope. When adjusting the "trigger level" knob on a scope, you could think of it as setting the level of a voltage comparator that tells the scope to trigger when the input voltage crosses that level. A timing analyzer works essentially the same on edge triggering except that the trigger level is preset to a logic threshold.

While many logic devices are level dependent, clock and control signals of these devices are often edge-sensitive. Edge triggering allows you to start capturing data as the device is clocked.

Example: Take the case of an edge-triggered shift register that is not shifting data correctly. Is the problem with the data or the clock edge? In order to check the device, we need to verify the data when it is clocked – on the clock edge. The analyzer can be told to capture data when the clock edge occurs (rising or falling) and catch all of the outputs of the shift register.


Bus/Signal
Simple Trigger
Sample Number
register
clock
XX
M1
T
M2
-60 ns
-40 ns
-20 ns
0 s
-27
67
68
69
1
0
1
0
1

Clocking data in the state analyzer

Tutorial Home | Next Topic | Previous Topic

The state analyzer requires a clock signal from the target system. This type of clocking makes the sampling of data in the logic analyzer synchronous to the clocked events on the device under test.

More specifically:

- A state analyzer is good at showing you "What" the signal activity is during a "Valid clock or control signal".
- A state analyzer is more interested in viewing signal activity during specified times of target system execution, than signal activity unrelated to the target system timing.
- This is why a state analyzer wants to sample data that is "synchronized" or synchronous to the target system clock signals.

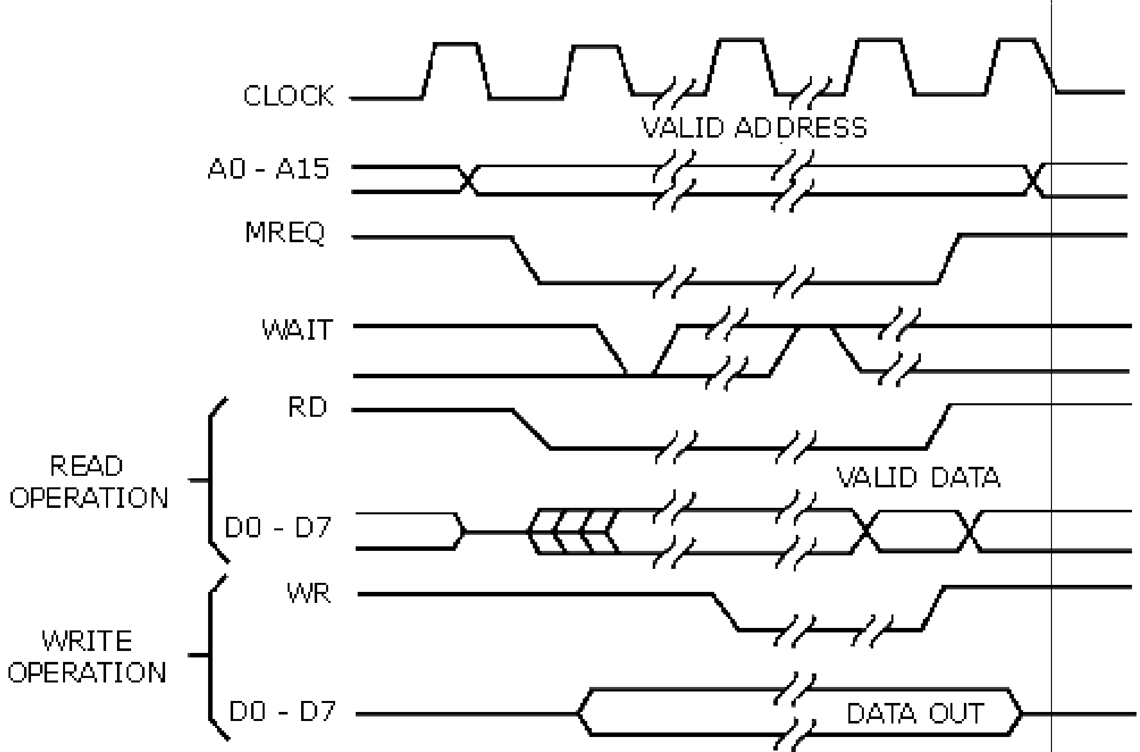
Sampling in the state analyzer

Tutorial Home | Next Topic | Previous Topic

In the world of microprocessors, you can have both data and address appearing on the same signal lines. To capture the correct data, the state analyzer has to restrict the sampling of data to times when only the desired data is valid and appears on the signal lines. It does this by sampling data from the same signal lines but with different clocks.

Example: The following timing diagram shows that to capture addresses, we want the analyzer to sample when MREQ line goes low.

To capture data, we want the analyzer to sample when the WR line goes low (write cycle) or when RD goes low (read cycle).

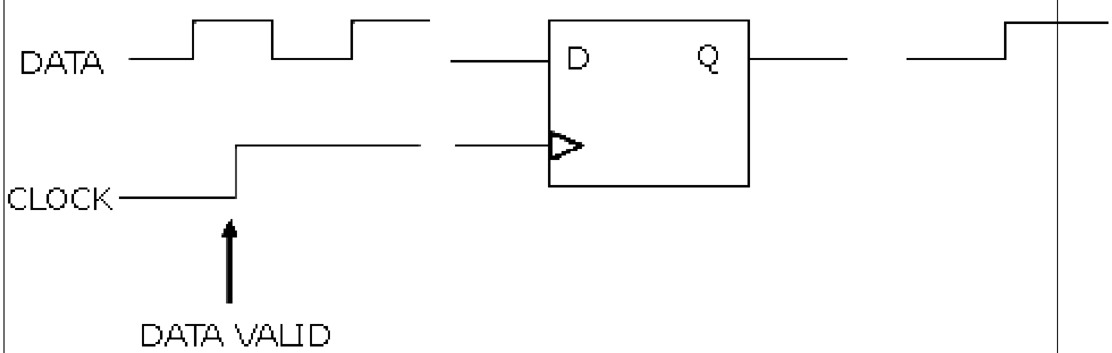


Triggering the state analyzer

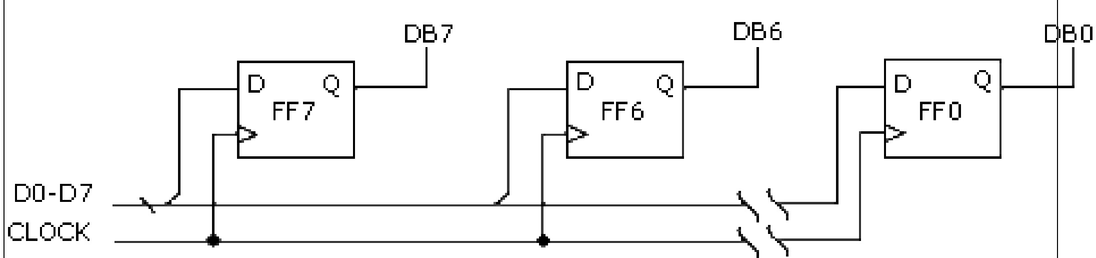
Tutorial Home | Next Topic | Previous Topic

Similar to a timing analyzer, a state analyzer has the capability to qualify the data we want to store. If we are looking for a specific pattern of highs and lows on the address bus, we can tell the analyzer to start storing when it finds that pattern and to continue storing until the analyzer's memory is full.

Simple Trigger Example: Looking at the "D" flip-flop shown below, data on the "D" input is not valid until after a positive-going clock edge occurs. Thus, a valid state for the flip-flop is when the clock input is high.

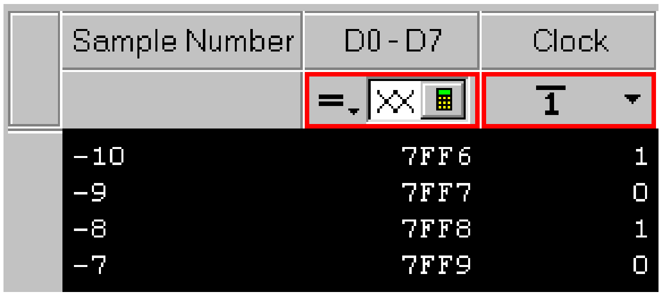


Now imagine that we have eight of these flip-flops in parallel. All eight are connected to the same clock signal as shown below.

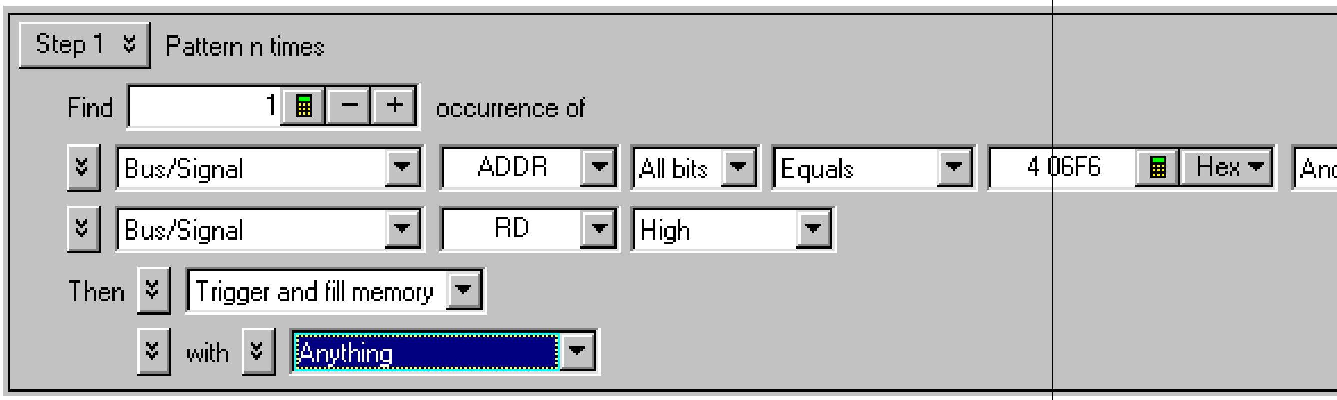


When a high level occurs on the clock line, all eight capture data at their "D" inputs. Again, a valid state occurs each time there is a positive level on the clock line.

The following simple trigger tells the analyzer to collect data on lines D0 - D7 when a high level is on the clock line.



Advanced Trigger Example: You want to see what data is stored in memory at the address value 406F6. You configure the advanced trigger to look for the pattern 406F6 (hexadecimal) on the address bus and a high level on the RD (memory read) clock line.



As you configure the Edge And Pattern trigger dialog, try to think of it as constructing a sentence that reads left-to-right.

"Find the first occurrence of a Bus named ADDR, and on All bits a pattern that Equals 406F6 Hex, And a Signal named RD with a High level. Then Trigger and fill memory with Anything.

Probing options

Tutorial Home Next Topic Previous Topic

- General Purpose Probing (Standard)
- Adapter to board connectors (Optional)
- Analysis probes (Optional)

So far we've talked about some of the differences between scopes, timing and state analyzers. Before we're ready to apply these new tools, we should talk about one more subject – the probing system.

A scope probe is designed to gain easy access to the target system while minimizing the signal distortion. Since we want to look at parametric information like voltage levels and rise times, it is important that the probe doesn't load the circuit under test significantly. A typical scope probe has 1 M ohm impedance shunted by 10 pF, depending on the bandwidth required.

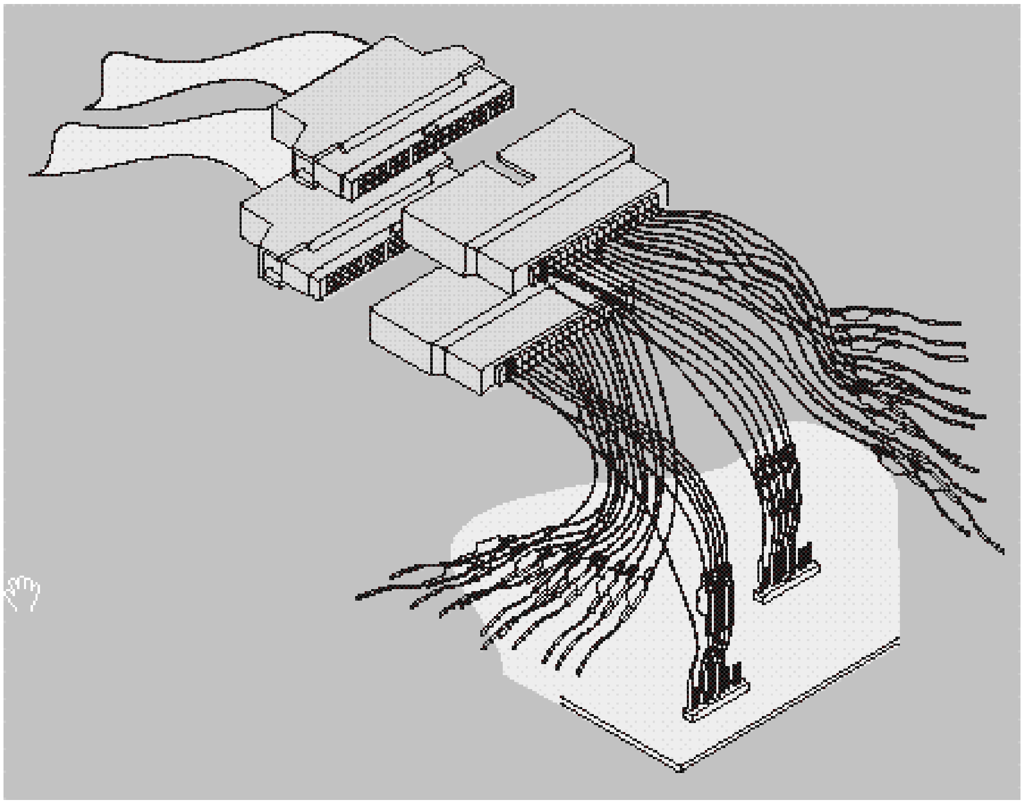
Logic analyzer probes are designed to allow connection of a high number of channels to the target system easily by trading off amplitude accuracy of the signal under test. Remember that a logic analyzer only distinguishes between two voltage levels!

Traditionally, logic analyzers used active probe pods, which had an integrated signal detection circuitry for eight channels integrated. From these pods, we could connect with leads to the circuit under test.

The typical impedance of a logic analyzer probe is in the area of 100 k ohm shunted by 8 pF at the input of the active pod. The connecting wires, however, add another 8 pF stray capacitance, giving a total of 16 pF per channel.

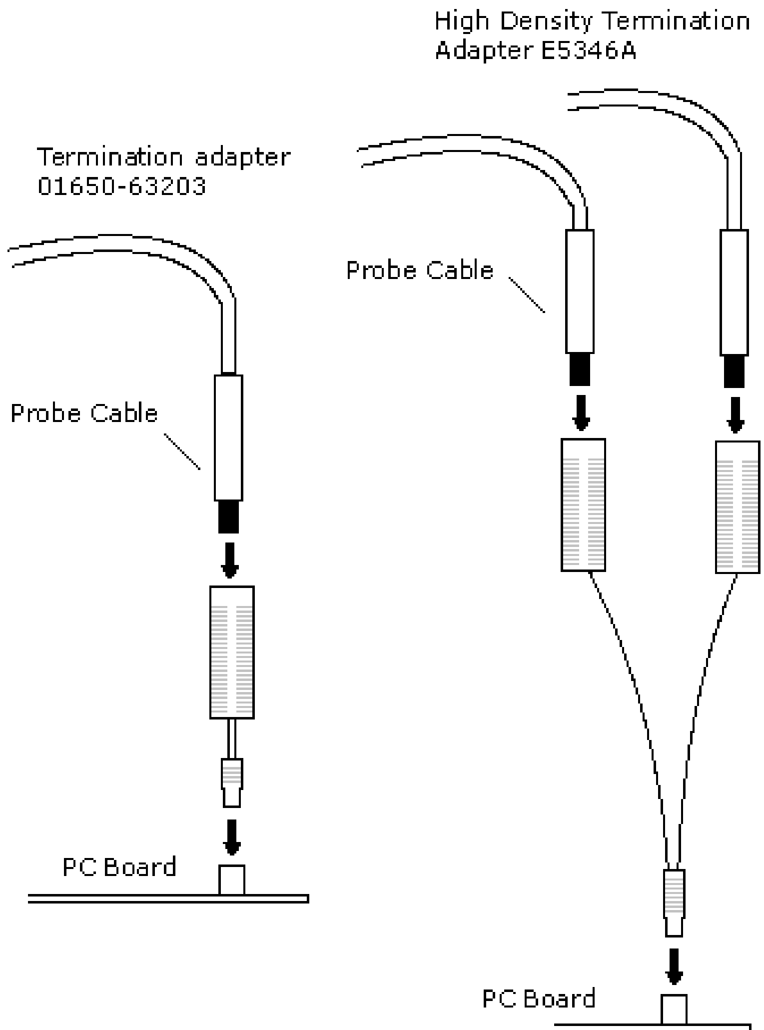
General Purpose Probing

Physical connections to digital systems must be reliable and convenient to deliver accurate data to the logic analyzer with minimum intrusion to the target system. The standard general purpose probing solution shown below is shipped with the logic analyzer. Each channel is terminated at both ends with 100k ohm and 8 pF.



The standard set plugs directly into any .1-inch grid with 0.026 to 0.033-inch diameter round pins or 0.025-inch square pins. All probe tips work with the Agilent Technologies 5059-4356 surface mount grabbers and the Agilent Technologies 5959-0288 through-hole grabbers.

Adaptor to board connectors



Both the 01650-63203 and the E5346A adapters include termination for the logic analyzer. The 01650-63203 termination adapter plugs into a 2 x 10 pin header with 0.1 inch spacing. The E5346A high-density adapter connects to an AMP "Mictor 38" connector.

Analysis Probes

Connecting a state analyzer to a microprocessor system requires some effort in terms of mechanical connection and clock selection. Remember, we have to clock the state analyzer whenever data or addresses on the bus are valid. With some microprocessors it might be necessary to use external circuitry to decode several signals to derive the clock for the state analyzer.

Analysis probes (formerly called preprocessors) are microprocessor-specific interfaces that make it easier to probe buses. Generally, analysis probes consist of a circuit board that attaches to the microprocessor (possibly through an adapter) and a configuration file. The configuration file sets up the logic analyzer's clocks, buses, and signals correctly, and may include an inverse assembler. The circuit board provides access to logical groups of pins through headers designed to connect directly to the logic analyzer.

Reference

Product Overview
Menus
Toolbars
Marker Display Bar
Marker Measurement Display Bar
Windows
Dialogs
Trigger Functions
Specifications and Characteristics

Product Overview

The Agilent Technologies 1680/90-series logic analyzers provide a variety of channel widths, memory depths, and state and timing acquisition speeds (see tables below).

The 1680A/AD-series comes with a large integrated 12.1-inch color flat panel display which can show up to 22 waveforms on screen simultaneously.

The 1690A/AD-series is a PC-hosted model which allows you to carry out your measurement and debug work in your PC environment.

Both model series have the familiar Windows-based user interface which takes the complexity out of making logic analyzer measurements. You can perform all operations directly from one window. See "Intrinsic Support" below.

Agilent's Simple, Quick and Advanced Trigger functions take the complexity out of triggering. Use Simple Trigger's pull down menus to define events in terms of edges and patterns. With Quick Trigger you can see if a suspect event ever reoccurs by just drawing a box around the event in the display. Quick trigger will do the rest! Use Advanced Trigger's drag and drop graphical icons with sentence-like structures to customize complex trigger scenarios.

Table of 1680A/AD-series channel, memory, and speed

Table of 1690A/AD-series channel, memory, and speed

Front Panel Operation

Keyboard Commands

Specifications

Characteristics

Supplied Accessories

1680A/AD-series	**1690A/AD-series**
PS2 mouse	IEEE 1394 PCI card and cable
Mini keyboard	Laptop IEEE 1394 cable
Front panel cover	Accessory pouch
Accessory pouch	

Optional Accessories

- Rack Mount Kit - Option 1CM
- Additional IEEE 1394 PCI card and cable

Documentation

Quick Start/Installation Guide

The Quick Start/Installation Guide gives you information on how to connect system peripherals and probing. Also included is an overview of the interface and information on installing software upgrades. Use this guide to quickly get familiar with the analyzer and also as a future reference for keeping your analyzer up-to-date and running properly.

Online Help System

The Online Help gives you product reference and feature information. Also included is a tutorial showing you how to make a basic measurement and links to time-saving features and concepts.

Agilent Technologies Web Sites

Corporation/Contact

Corporation - http://www.agilent.com

Contact Us - http://www.agilent.com/find/contactus/

Notify Me - http://www.agilent.com/find/notifyme

Product Information

Logic Analysis - http://www.agilent.com/find/logicanalyzer/

1680/90 Analyzer - http://www.agilent.com/find/digitaldesign/

Software Updates - http://www.software.cos.agilent.com/1680-1690

Documentation

Web-based documentation - http://www.cos.agilent.com/manuals/

Intrinsic Support

Because the Agilent 1680A/AD-series products operate in a Microsoft Windows 2000 Professional environment, intrinsic support shall only cover the Agilent Logic Analyzer application. Intrinsic support shall also cover any Windows 2000 Professional operating system services utilized by the Agilent Logic Analyzer application:

- Print from the Agilent Logic Analyzer application.
- Networking.
- File management from the Agilent Logic Analyzer application.

Because the Agilent 1690A/AD-series products operate on a hosted desktop PC, support shall only cover the Agilent Logic Analyzer application and the IEEE 1394 interface to the host PC. Intrinsic support shall not cover any other Windows 2000 Professional operating system issues other than those listed above. Other Windows 2000 Professional issues shall be considered Microsoft issues.

> Note: Any customer-installed applications on an Agilent 1680A,AD-series product shall not be supported by Agilent. Customers must contact the software vendor for support.

See Also

Tutorial - Getting to know your logic analyzer

Table of 1680-series channel, memory, and speed

Model	Channel Count and Memory Depth
1680A	136-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1680A/D	136-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1681A	102-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1681A/D	102-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1682A	68-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1682A/D	68-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1683A	34-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1683A/D	34-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep

Table of 1690-series channel, memory, and speed

Model	Channel Width and Memory Depth
1690A	136-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1690A/D	136-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1691A	102-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1691A/D	102-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1692A	68-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1692A/D	68-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep
1693A	34-channel State: 200 MHz, 256 K deep Timing: 400 MHz/800 MHz (full/half channel), 512 K/1 M deep (full/half channel) Transitional timing: 200 MHz, 256K deep
1693A/D	34-channel State: 200 MHz, 1 M deep Timing: 400 MHz/800 MHz (full/half channel), 2 M/4 M deep (full/half channel) Transitional timing: 200 MHz, 1 M deep

Front Panel Operation

The front panel interface consist of knobs and buttons that you use to setup and run measurements. There are also shortcut buttons that quickly access commonly used dialogs in the interface.

All functions available with the front panel knobs and buttons can also be performed in the graphical user interface (GUI). Unlike in the GUI, where you can block access by disabling menus and toolbars, the front panel buttons/knobs cannot be blocked. However, when a front panel action is not valid, an audible "beep" will sound.

Run/Stop

Item	Description
Run Single	Runs a single acquisition. The Run Single button turns green indicating when a Run action is valid. While the analyzer is running, the light goes out.
Run Rep. (Repetitive)	Runs a repetitive acquisition. The Run Repetitive button turns green indicating when a Run Repetitive action is valid. While the analyzer is running, the light goes out.
Stop	Stops the current acquisition. The Stop button turns red during a Run cycle indicating when the Stop action is valid.

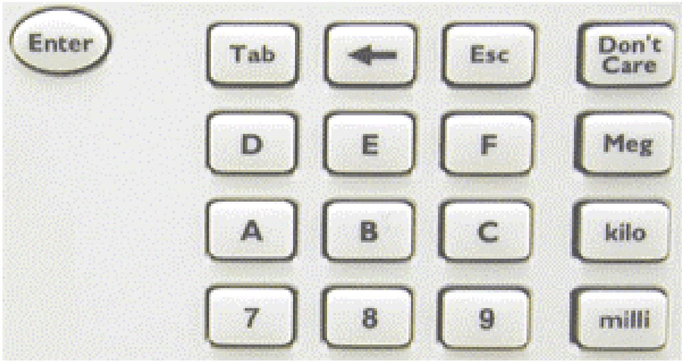
Save/Open Setup

Item	Description
Open Setup	Accesses the Open Configuration dialog.
Save Setup	Accesses the Save Configuration dialog.
Default Setup	Resets setup to the default power up configuration.

General purpose knob

The general purpose knob acts on the field that has the current focus. Fields that have the current focus have the blue background. The general purpose knob is typically used to increase/decrease numeric values such as waveform scale and delay.

Alphanumeric Keypad



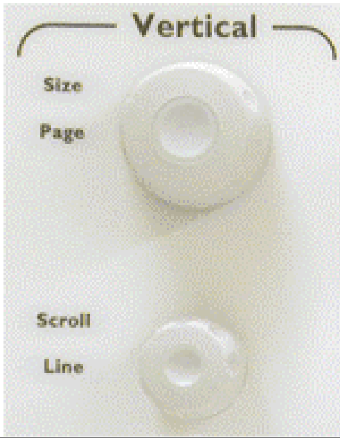
Item	Description
Enter	Accepts value or configuration change and exits dialog.
Tab	Scrolls configuration fields left-to-right and top-to-bottom.
Backspace (left arrow)	Backspaces cursor in an alphanumeric assignment field.
Esc	In Ok/Cancel dialogs, the escape key acts as a cancel operation and exits the dialog.
Keypad	Used for alphanumeric entry.
Units	Sets unit of measure.

Shortcuts

Item	Description
Setup	Accesses Buses/Signals tab in the Analyzer Setup dialog.
Waveform	Accesses Waveform display window. When the Waveform display is active, the button turns green.
Listing	Accesses Listing display window. When the Listing display is active, the button turns green.
Find	Accesses advanced search dialog.
Trigger	Accesses advanced trigger dialog.
File Mgr	Accesses the Explore file manager dialog.
Print	Accesses the Print dialog.

Help	Accesses the online help system's main window. Same as F1 key.

Vertical



Item	Description
Size	Adjusts height of all waveform rows. The selection lights up green indicating it's part of the waveform group, and that the waveform group is currently active.
Page	Scrolls a page at a time of Listing data. The selection lights up green indicating it's part of the listing group, and that the listing group is currently active.
Scroll	Scrolls row at a time of Waveform data. The selection lights up green indicating it's part of the waveform group, and that the waveform group is currently active.
Line	Scrolls a line at a time of Listing data. The selection lights up green indicating it's part of the listing group, and that the listing group is currently active.

Horizontal

Item	Description
Time/Div	Changes time/division scale of Waveform display. The selection lights up green indicating it's part of the waveform group, and that the waveform group is currently active.

Bus/Sig	Scrolls first column to last column in Listing Display. The selection lights up green indicating it's part of the listing group, and that the listing group is currently active.
Delay	Changes delay of Waveform display. The selection lights up green indicating it's part of the waveform group, and that the waveform group is currently active.
Column	Scrolls a column at a time of Listing data. The selection lights up green indicating it's part of the listing group, and that the listing group is currently active.

Marker

Item	Description
Move marker knob	Moves selected marker in the display.
Choose marker button	Selects marker for "Move" operation. Press button to scroll through available markers. If no markers are defined, operating the knob an Choose button will create an M1 marker for you.

For more marker information, refer to "Working with markers".

Keyboard Commands

Access Menus
File Operations
Edit Operations
Search Operations
View operations
Run/Stop Operations
Window Operations
Help Operations
Miscellaneous

Access Menus

Alt+F	Access to File menu
Alt+E	Access to Edit menu
Alt+V	Access to View menu
Alt+S	Access to Setup menu
Alt+T	Access to Tools menu
Alt+M	Access to Markers menu
Alt+R	Access to Run/Stop menu
Alt+W	Access to Window menu
Alt+H	Access to Help menu

File Operations

The following operations are located under File in the menu bar.

Ctrl+N	File - New
Ctrl+O	File - Open
Ctrl+F4	File - Close
Ctrl+S	File - Save
Ctrl+P	File - Print

Edit Operations

The following operations are located under Edit in the menu bar.

Ctrl+Z	Edit - Undo
Ctrl+X	Edit - Cut
Ctrl+C	Edit - Copy
Ctrl+V	Edit - Paste
Alt+I	Edit - Insert Bus/Signal into Window
Alt+P	Edit - Current Window Properties

Search Operations

The following operations are located under Edit in the menu bar.

Ctrl+F	Edit - Find
Shift+F3	Edit - Find Previous
F3	Edit - Find Next
Ctrl+B	Edit - Go To Beginning
Ctrl+T	Edit - Go To Trigger
Ctrl+E	Edit - Go To End
Ctrl+G	Edit - Go To

View Operations

The following operations are located under View in the menu bar.

Shift+O	View - Zoom Out
Shift+I	View - Zoom In
F9	View - Full Screen
F11	View - Toggle Tabbed Windows
F12	View - Toggle Status Bar

Run/Stop Operations

The following operations are located under Run/Stop in the menu bar.

F5	Run/Stop - Run
Ctrl+F5	Run/Stop - Run Repetitive
F8	Run/Stop - Stop
Shift+F8	Run/Stop - Cancel
Shift+Ctrl+F8	Run/Stop - Resume

Window Operations

The following operations are located under Window in the menu bar.

Alt+P	Window - Properties
F6	Window - Toggle to Next
Shift+F6	Window - Toggle to Previous

Help Operations

The following operations are located under Help in the menu bar.

F1	Help - Help Topics

Miscellaneous

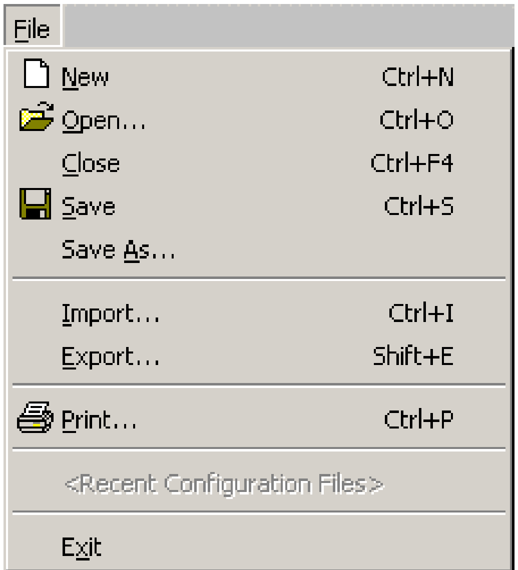
The following operations are located throughout the interface.

Ctrl+esc	Shows Windows Start bar

Menus

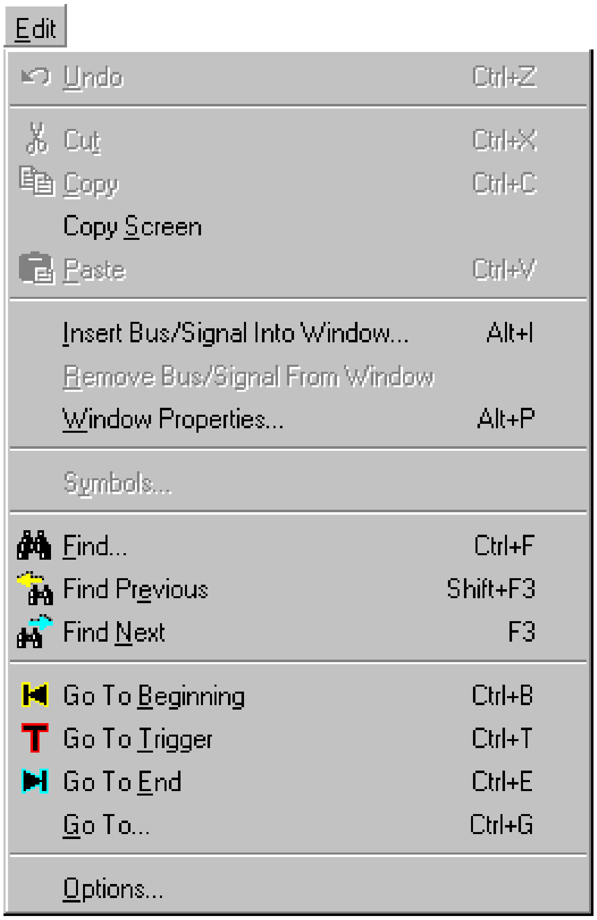
File Menu
Edit Menu
View Menu
Setup Menu
Tools Menu
Markers Menu
Run/Stop Menu
Window Menu
Help Menu

File Menu



New	Creates a new logic analyzer configuration file.
Open	Opens a previously saved logic analyzer configuration file. Secondary display windows are not restored.
Close	Closes the active window after asking whether to save its data.
Save	Saves changes to the currently open configuration file.
Save As	Saves the currently open configuration file to a new name.
Import	For offline analysis, imports fast binary output data from 16700-series logic analyzers.
Export	Saves captured data to comma-separated value (CSV) files. CSV files can be imported into spreadsheet, database, or other data analysis programs.
Print	Prints displayed data within a defined range.
Recent Configuration Files	Lists recently opened files for quick reference or access.
Exit	Closes the logic analyzer user interface window.

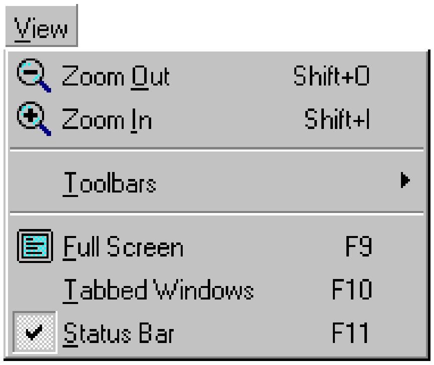
Edit Menu



Undo	Undo the last user action. This includes any properties that have changed such as column color, column width, column move, column insert, column delete, etc. Items that cannot be undone include scrolling, acquisition runs and simple trigger modifications.
Cut	Cuts the selection from alphanumeric fields in listing and waveform windows. Alphanumeric fields in lower level modal dialogs are cut using keyboard commands (accelerator keys). Cut selections are pasted to the clipboard.
Copy	Copies the selection from alphanumeric fields in listing and waveform windows. Alphanumer fields in lower level modal dialogs are copied using keyboard commands (accelerator keys). Copied selections are pasted to the clipboard.
Copy Screen	Copies the current screen to a bitmap and places it on the system clip board.
Paste	Pastes the cut or copied data that is stored in the clipboard into the alphanumeric field. Alphanumeric data is pasted into fields in lower level modal dialogs using keyboard commands (accelerator keys).
Insert Bus/Signal Into Window	Inserts a predefined bus or signal into the display.
Remove Bus/Signal	Deletes the highlighted bus or signal from the display window.

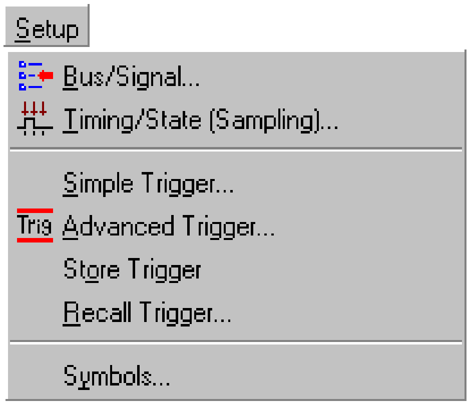
From Window	
Window Properties	Accesses the window properties dialog.
Symbols	Accesses the Symbols dialog.
Find	Locates specific data in the acquisition.
Find Previous	Locates the previous occurrence of the specified data.
Find Next	Locates the next occurrence of the specified data.
Go To beginning	Places the beginning of the captured data trace at center screen.
Go To Trigger	Places the trigger point at center screen.
Go To End	Places the end of the captured data trace at center screen.
Go To	Accesses the Go To selection list.
Options	Accesses the System Options dialog.

View Menu



Zoom Out	Zooms out on an active window.
Zoom In	Zooms in on an active window.
Toolbars	Access the Toolbar dialog window.
Full Screen	Enables or disables full screen display.
Tabbed Windows	Enables or disables Listing and Waveform tabs.
Status Bar	Enables or disables the status bar.

Setup Menu



Bus/Signal	Accesses the Bus/Signal setup dialog.
Timing/State (Sampling)	Accesses the Sampling setup dialog.
Simple Trigger	See "Simple Trigger".
Advanced Trigger	Accesses the Advanced Trigger dialog.
Store Trigger	Stores current trigger.
Recall Trigger	Accesses a list of most recently used triggers.
Symbols	Accesses the Symbols dialog.

Tools Menu

All add-in tools are grouped under the tools menu. Your Agilent Logic Analyzer comes with a filter/colorize tool built in. If you are using inverse assemblers, bus analysis tools, or other third-party tools, the tools will show up in the Tools menu under New.

As tools are created, they are added to the top of the Tools menu. The menu above shows one active tool.

1 *tool name*	Edit an existing tool.
New	Creates a new analysis tool.
Overview	Lets you manage the active tools.
Find	Locates specific data in the acquisition.

See Also

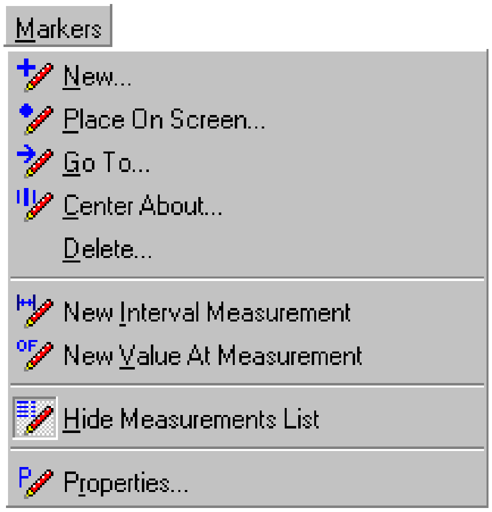
To add a new tool

To change a tool

To delete a tool

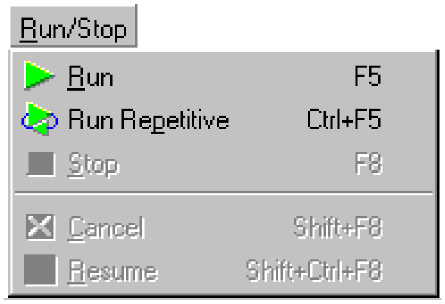
To filter or colorize data

Markers Menu



New	Creates a new marker.
Place on Screen	Places a selected marker on screen over the trigger point.
Go To	Goes to a selected marker.
Center About	Centers the display around a selected marker.
Delete	Deletes selected markers.
New Interval Measurement	Creates a new interval measurement.
New Value At Measurement	Creates a new value at measurement.
Hide Measurements List	Hides the marker measurement display bar.
Properties	Accesses the markers properties dialog

Run/Stop Menu

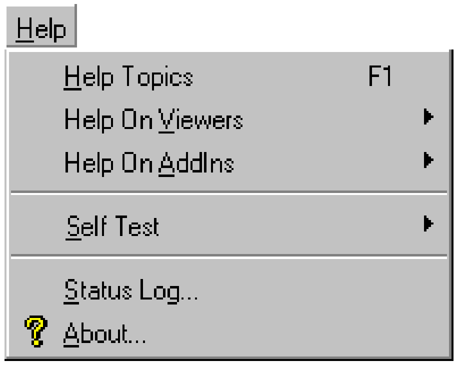


Run	Starts sampling, fills logic analyzer memory with samples around the trigger, and stops.
Run Repetitive	Starts sampling, fills logic analyzer memory with samples around the trigger, and repeats.
Stop	Stops a logic analyzer measurement that is in progress.
Cancel	Cancels the current Run.
Resume	Resumes the current Run after a Cancel.

Window Menu

New Listing	Creates an additional listing window.
New Waveform	Creates an additional waveform window.
Rename	Allows the user to change the active window name.
Properties	Accesses the display properties dialog.
Close	Closes the active window.
Next	Displays the next window.
Previous	Displays the previous window.
Cascade	Displays all opened windows in an overlaid and offset format.
Tile Horizontally	Displays all opened windows so the horizontal display space is equally divided.
Tile Vertically	Displays all opened windows so the vertical display space is equally divided.
Arrange Icons	All minimized listing and waveform windows are arranged at the bottom of the analyzer window.
1. 2.	Lists all open display windows for reference or access.

Help Menu



Help Topics	Accesses the online help.
Help On Viewers	Opens online help for the Waveform, Listing, or Compare windows.
Help On AddIns	Opens online help for AddIns.
Self Test	Accesses the Logic Analyzer Self-Tests dialog.
Status Log	Accesses the Info Manager dialog.
About	Displays product version and copyright information.

Toolbars

Toolbars are located under the menu bar, and are used to quickly access a function or perform a task. By default, not all toolbars, or individual tools within a given toolbar are displayed. For a complete list of all available toolbars, click View>Toolbars>. For a complete list of all tools within a given toolbar, click View>Toolbars>Customize>Commands.

Standard

Setup

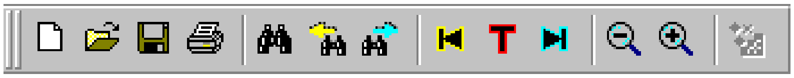
Markers

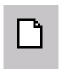
Run/Stop

Viewers

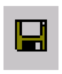
Customize

Standard Toolbar

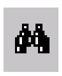


 New - Creates a new logic analyzer configuration file.

 Open - Opens a previously saved logic analyzer configuration file. Secondary display windows are not restored.

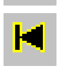
 Save - Saves changes to the currently open configuration file.

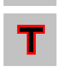
 Print - Prints displayed data within a defined range.

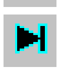
 Find - Locates specific data in the acquisition.

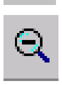
 Find Previous - Locates the previous occurrence of the specified data.

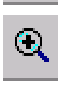
 Find Next - Locates the next occurrence of the specified data.

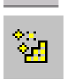
 Go to Beginning - Centers the beginning of the acquisition data.

 Go to Trigger - Centers the trigger point of the acquisition.

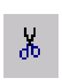
 Go to End - Centers the end of the acquisition data.

 Zoom Out - Zooms in on an active window.

 Zoom In - Zooms out on an active window.

 Tools Overview - Shows an overview of the current Tools.

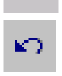
Note: The following are optional standard toolbar icons.

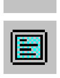
 Cuts the selection and places it on the clipboard.

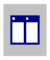
 Copies the selection and places it on the clipboard.

 Pastes the data that is stored on the clipboard.

 Provides information about the Agilent 16xx Product.

 Undo last user action.

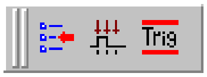
 Enables or disables full screen display.

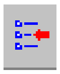
	Activate the next window.
	Activates the previous window.
	Arranges windows as overlapping tiles.
	Arranges windows as non-overlapping horizontal tiles.
	Arranges windows as non-overlapping vertical tiles.

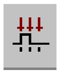
See Also

To create a custom toolbar

Setup Toolbar



 Bus/Signal - Accesses the Bus/Signal setup dialog.

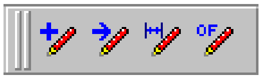
 Timing/State (Sampling) - Accesses the Sampling setup dialog.

Advanced Trigger - Accesses the Advanced Trigger dialog.

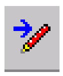
See Also

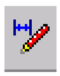
To create a custom toolbar

Markers Toolbar



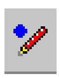
 New - Creates a new marker.

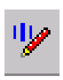
 Go To - Centers the display around the selected marker.

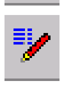
 Creates a new interval measurement.

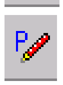
 Creates a new value at a measurement.

Note: The following are optional markers toolbar icons.

 Place Maker - Places the selected marker on the screen.

 Center About - Centers the display around two selected markers.

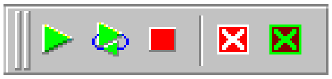
 Turns the marker measurement list on and off.

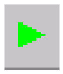
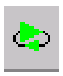
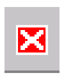
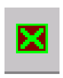
 Accesses the markers properties dialog.

See Also

To create a custom toolbar

Run/Stop Toolbar

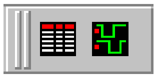


	Starts sampling, fills logic analyzer memory with samples.
	Starts Sampling, fills logic analyzer memory with samples around the trigger, and repeats.
	Stops a logic analyzer measurement in progress, for example, when the trigger condition is not found.
	Stops the current action.
	Resumes the cancelled action.

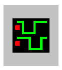
See Also

To create a custom toolbar

Viewers Toolbar



 Creates a new listing viewer.

 Creates a new waveform viewer.

See Also

To create a custom toolbar

To create a custom toolbar

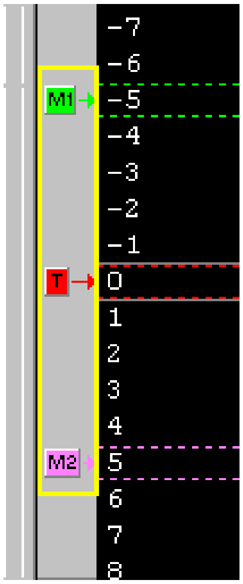
To add tool icons

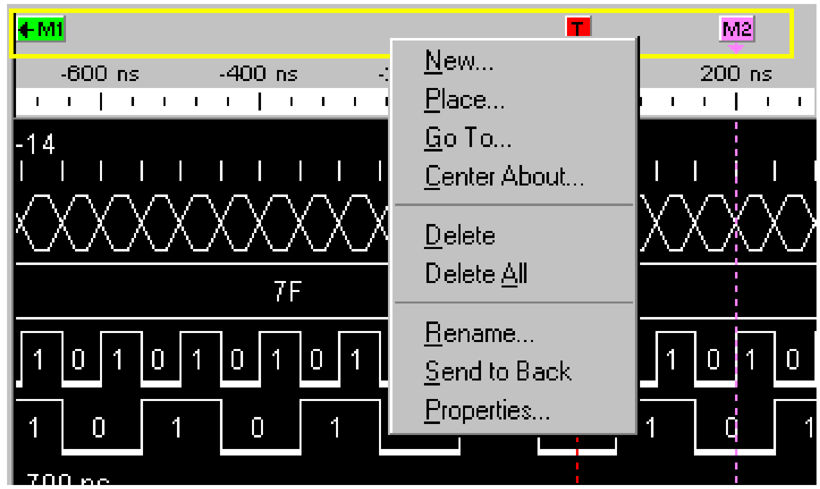
1. From the menu bar, select **View>Toolbars**.
2. Click **Customize**, then select the **Commands** tab.
3. Select the toolbar you wish to customize.
4. Drag the desired icon to the correct position on the selected toolbar. Release the mouse button to insert the tool icon.
5. Repeat for any other icons you wish to add.
6. Select **OK**.

To remove tool icons

1. From the menu bar, select **View>Toolbars**.
2. Click **Customize**, then select the **Commands** tab.
3. Select the toolbar you wish to customize.
4. Drag the icon from the toolbar and drop it onto the **Customize** dialog box.
5. Repeat for any other icons you wish to remove.
6. Select **OK**.

Markers Display Bar





To access these tasks, right-click anywhere in the marker display bar.

New - To create new markers.

Place - To place markers in data.

Go To - Go To a marker.

Center About - To center the display about a marker.

Delete - To delete a marker.

Delete All - To delete all makers.

Rename - To rename a marker

Send to back - To send a marker to the back

Properties - To set marker properties.

See Also

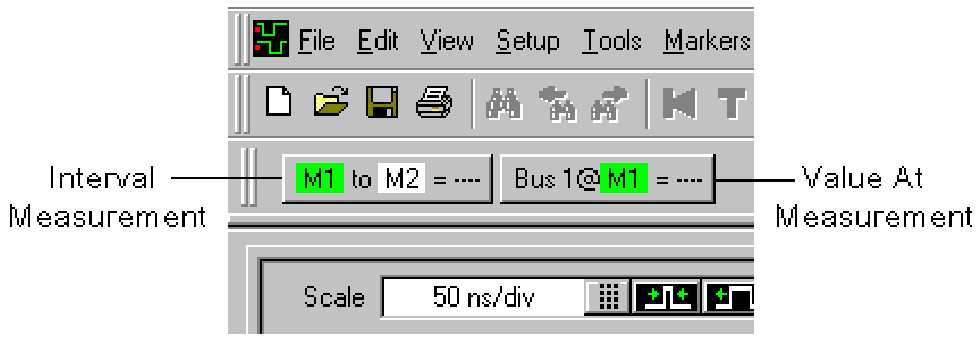
Reading off-screen markers

Markers menu

Markers toolbar

Marker Measurement Display Bar

Marker "interval" and "value at" measurements are displayed below the menu bar with the other toolbars.



To create a new interval measurement

To create a new value at measurement

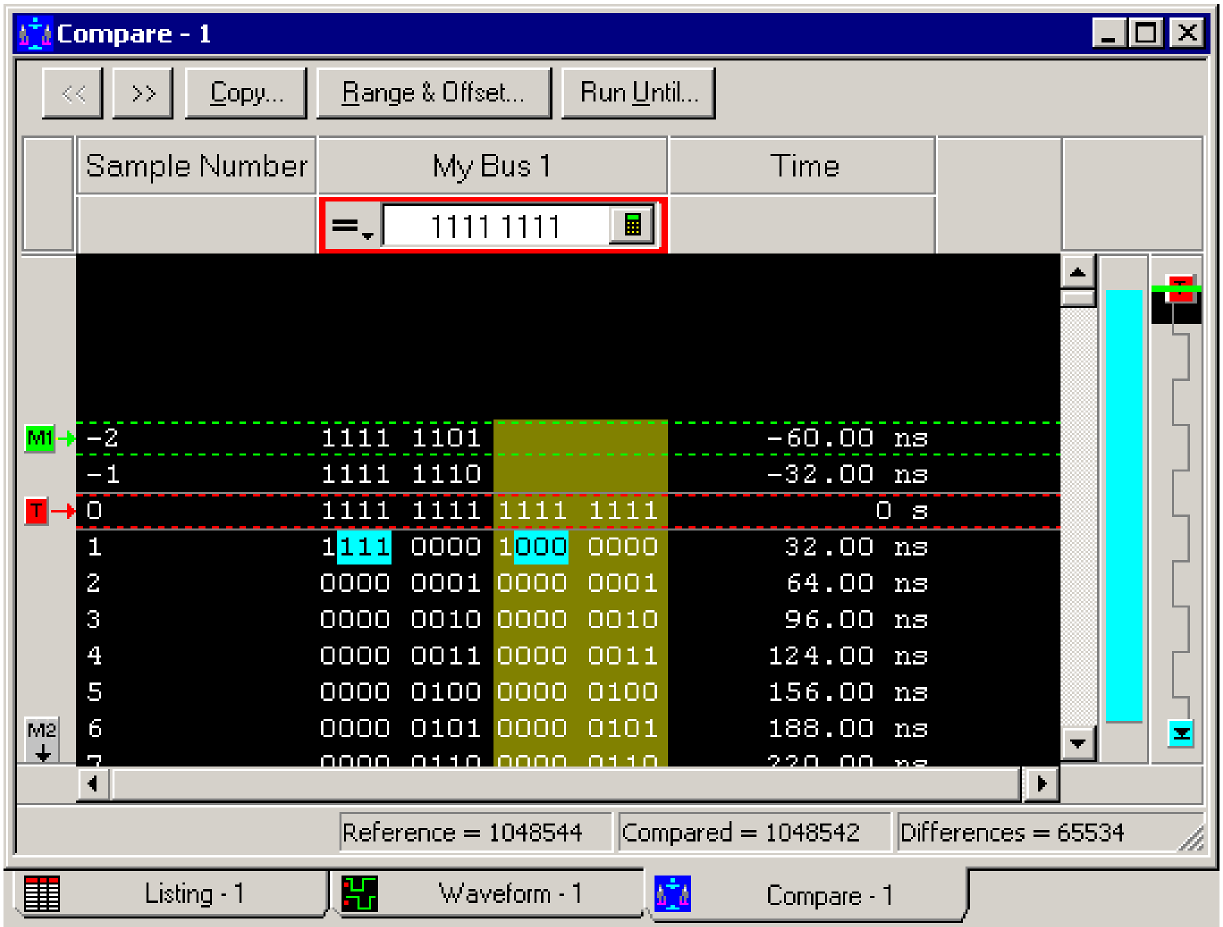
To hide/show measurement display bar

Windows

Waveform Display Window
Listing Display Window
Compare Display Window

See Also
Viewers Toolbar

Compare Display Window



The Compare window lets you compare acquired (input) data to data that has been saved in a reference buffer. The reference data has a colored background, and differences between between the input data and the reference data are highlighted.

The Compare window is accessed through the menu bar's**Window>Compare**. If you haveTabbed Windows turned on, you can also select a tab at the bottom of the window.

Except for the Compare window's ability to display the differences between captured data and reference data, and its inability display colorized data (from the Filter/Colorize tool), the Compare window is just like the Listing window.

See Also

Comparing Captured Data to Reference Data

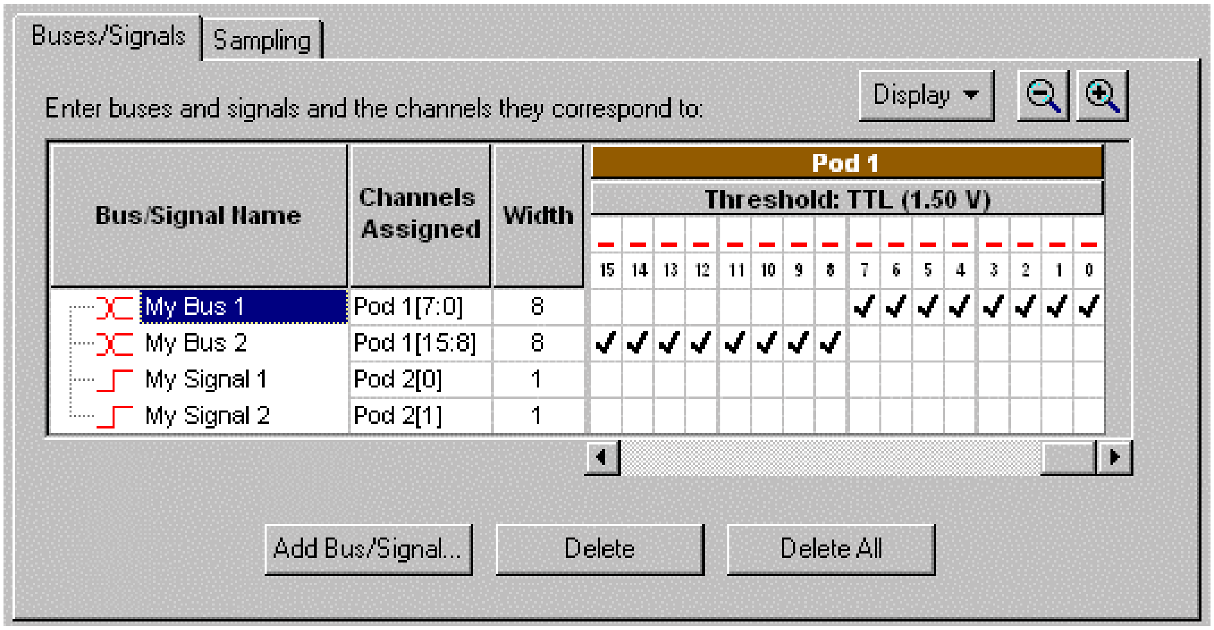
- To copy data to the reference buffer
- To find differences in the compared data
- To compare only a range of samples
- To offset the reference data
- To run until a number of compare differences
- To set the Compare display options

Listing Display Window

Dialogs

Buses/Signals Setup
Sampling Setup
Printing Data
Advanced Trigger
System Options
Recall Trigger
Waveform Window Properties
Listing Window Properties
Markers Properties
Symbol Selection
Symbols
Filter/Colorize
Tool Overview

Analyzer Setup Dialog



The Analyzer Setup dialog is accessed through the menu bar's Setup>Bus/Signal.

The dialog consists of the following two tabs.

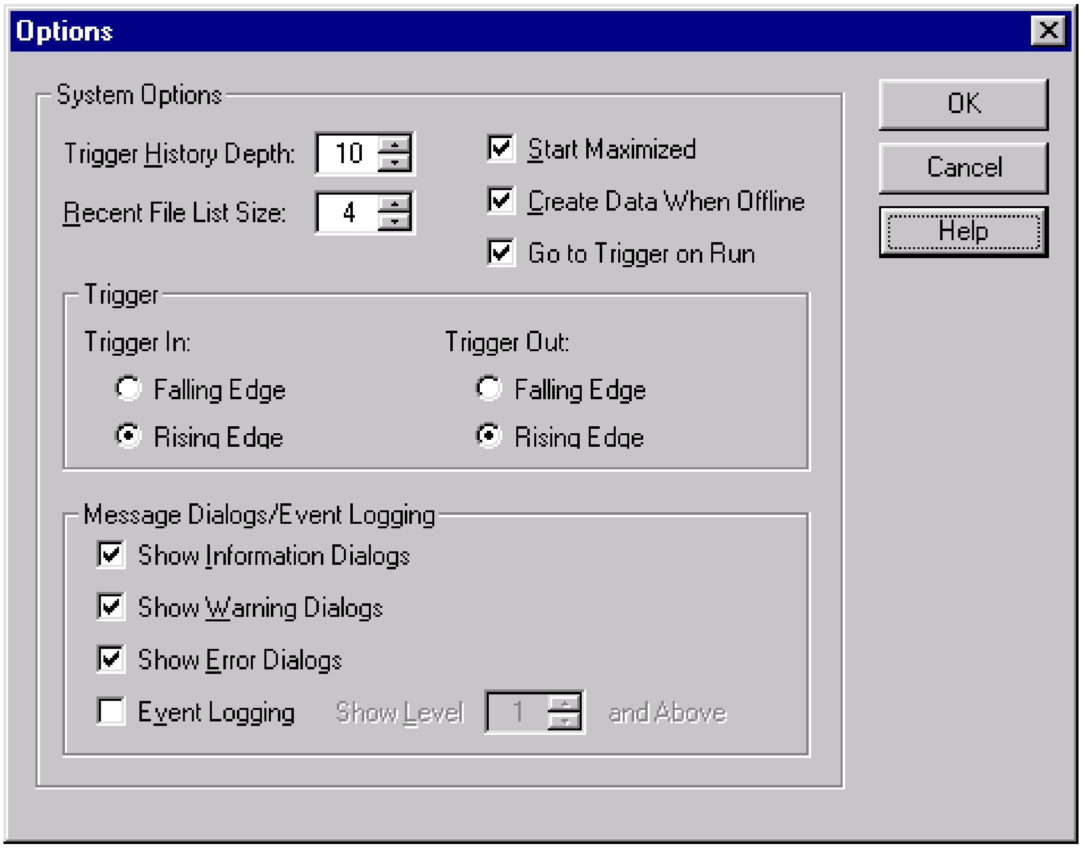
Buses/Signal

The Buses/Signals tab is used to map bus and signal names in the interface to the pod and channel connections of the probes. Also, you can set a pod threshold, and assign a numeric base and polarity to the bus or signal.

Sampling

The Sampling tab is used to name the analyzer, and select and configure the acquisition mode. In the Timing Acquisition Mode, you set the channel width and sampling rate. In the State Acquisition Mode you configure the state clocks and qualifiers.

System Options Dialog



To change your system options, select Edit>Options from the menu bar. System options are written in the Windows registry file and persist across sessions.

To change how many triggers are saved

Set the **Trigger History Depth**. You can keep as many as 50 of the most recently used triggers. See To store a trigger for more information on re-using triggers. Trigger history is saved in the configuration files. The default is 10.

To change the length of the file history

Set the **Recent File List Size**. This sets how many recently-loaded configuration files are shown in the File menu. The default is 4.

To create data when offline

Check the Create Data When Offline box. When you run the analyzer, fake data will be created. This mode is useful when learning how to use the logic analyzer software.

Note: The logic analyzer does not trigger with fake data. You can set the triggers, but will not get the same results you would with a real acquisition.

To set Trigger In and Trigger Out

Trigger In allows you to trigger the logic analyzer from another source. It expects to see a TTL signal, not to exceed 5.5 volts, with a minimum signal amplitude of 500 mV.

Trigger Out sends a signal to another device when the logic analyzer triggers. Low is 0 volts, high is 3.3 volts.

To set message level

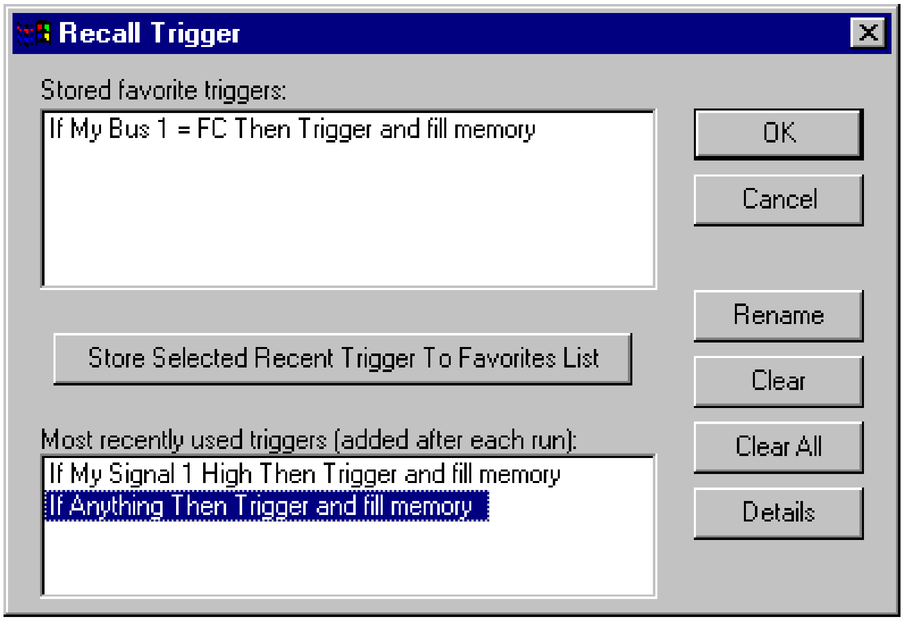
As with any other program, the Agilent Logic Analyzer generates messages about events. You can choose which messages are displayed. Check the appropriate box to indicate you wish the dialogs displayed.

Show Information Dialogs	Information dialogs offer tips such as the location of Simple Trigger, and do not indicate a failure.
Show Warning Dialogs	Warning dialogs occur when some setting may affect your data, such as being offline.
Show Error Dialogs	Error dialogs occur when an operation cannot be completed as specified.
Event Logging	You can choose to have all events recorded in a log file. Event logging will slow down your logic analyzer. Event logs can be viewed through Help>Status Log. Set the event logging level according to the directions of your Agilent Technologies support person. Be sure to turn off event logging when resuming normal use.

Recall Trigger Dialog

After a trigger is stored, it is placed in a recall list for use at a later time. Use the Recall Trigger dialog to recall a previously used trigger, save a trigger in a favorites list, view trigger details, or rename the trigger to a custom name.

Note: Saved triggers are part of a the current configuration. If you load a new configuration file, saved triggers in the current recall list are lost.



Rename	Allow you to edit the name of the highlighted trigger.
Clear	Clears the highlighted trigger from the list.
Clear All	Clears all triggers from recall lists.
Details	Shows complete definition of the highlighted trigger.

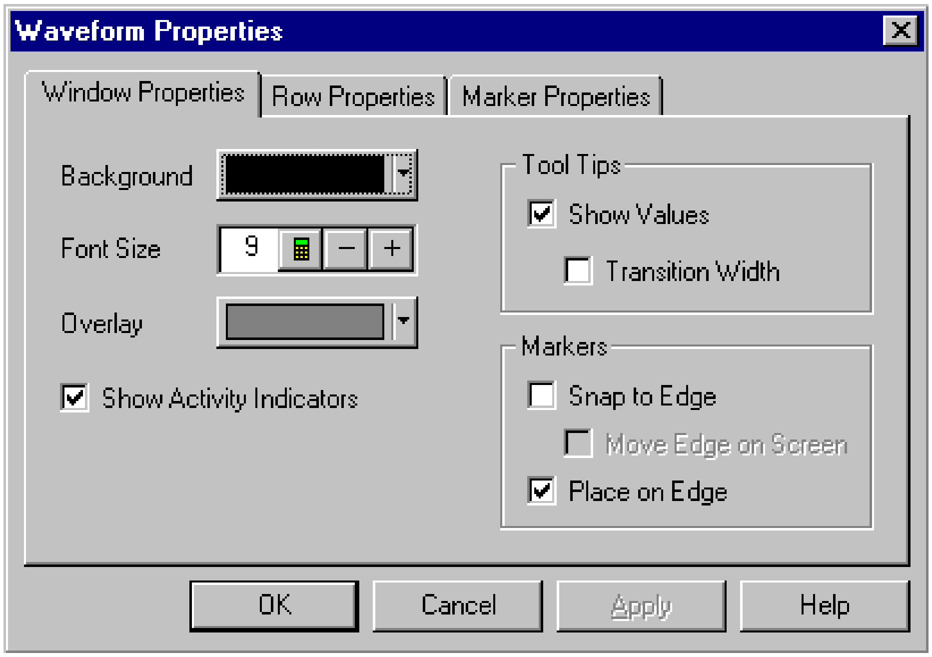
To recall a trigger

1. From the menu bar click Setup>Recall Trigger...
2. Select the desired trigger, then click Ok.

See Also

To store a trigger

Properties Dialog



The Properties dialog is accessed through the menu bar's Window>Properties. Use it to set up how the window and the displayed data appear.

To set waveform window properties

To set waveform display properties

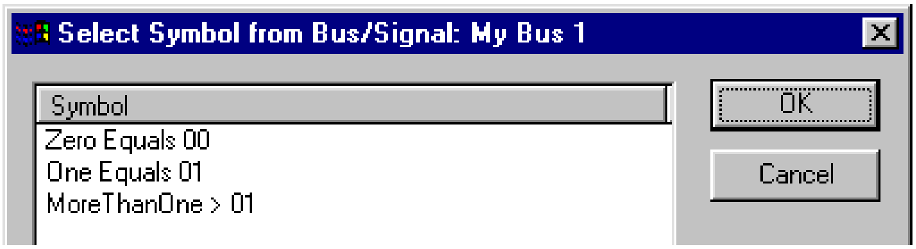
To set listing window properties

To set listing column properties

To set marker properties

Symbol Selection

Use the Select Symbol dialog to choose a symbol to use when the numeric base is set to Symbols.



To choose a symbol:

1. Look for the bus for which the symbol is defined.
2. Select the symbol.

If the symbol you want is not listed, you can:

- Check that you are looking at the right bus, or
- Load a configuration file in which the symbol is defined. or
- Add the symbol.

The Select Symbol dialog is used whenever you need to choose a symbol. For example, the Select Symbol dialog will be displayed when you use the Symbols numeric base in the following dialogs:

- Find
- Filter/Colorize
- Simple Trigger
- Advanced Trigger Dialog

Trigger Functions

Timing mode trigger functions
State mode trigger functions

Timing mode trigger functions

The following trigger setup examples are available as Trigger Functions in the Advanced Trigger dialog when in the timing acquisition mode. To see these trigger setups in the context of an example measurement refer to "Making a timing analyzer measurement".

Edge

- Edge
- "N" number of edges
- Edge and Pattern
- Edge followed by edge
- Edges too far apart
- Edge followed by pattern
- Pattern too late after edge

Bus Pattern

- Pattern
- Edge And Pattern
- Pattern present for > "T" time
- Pattern present for < "T" time
- Pattern absent for > "T" time
- Pattern absent for < "T" time
- Edge followed by pattern
- Pattern too late after edge

Other

- Find anything "N" times
- Reset and start timer
- Width violation on pattern or pulse
- Wait "T" seconds
- Run until user stop
- Wait for external arm

Advanced

- Advanced If/Then
- Advanced 2-Way Branch
- Advanced 3-Way Branch
- Advanced 4-Way Branch
- Pattern "AND" Pattern
- Pattern "OR" Pattern

State mode trigger functions

The following trigger setup examples are available as Trigger Functions in the Advanced Trigger dialog when in the state acquisition mode. To see these trigger setups in the context of an example measurement refer to "Making a state analyzer measurement".

Patterns

- Pattern "N" times
- "N" consecutive samples with Pattern1
- Pattern1 followed by Pattern2
- Pattern1 immediately followed by Pattern2
- Pattern1 followed by Pattern2 before Pattern3
- Too few states between Pattern1 and Pattern2
- Too many states between Pattern1 and Pattern2
- Pattern2 occurring too soon after Pattern1
- Pattern2 occurring too late after Pattern1

Other

- Reset and start timer
- Find anything "N" times
- Run until user stop
- Wait for external arm
- Wait "N" external clock states

Advanced

- Advanced If/Then
- Advanced 2-Way Branch
- Advanced 3-Way Branch
- Advanced 4-Way Branch
- Pattern AND Pattern
- Pattern OR Pattern

See Also

External Triggering

Timing Mode

Timing mode trigger functions

Edge

- Edge
- "N" number of edges
- Edge and Pattern
- Edge followed by edge
- Edges too far apart
- Edge followed by pattern
- Pattern too late after edge

Bus Pattern

- Pattern
- Edge and Pattern
- Pattern present for > "T" time
- Pattern present for < "T" time
- Pattern absent for > "T" time
- Pattern absent for < "T" time
- Edge followed by pattern
- Pattern too late after edge

Other

- Find anything "N" times
- Reset and start timer
- Width violation on pattern or pulse
- Wait "T" seconds
- Run until user stop
- Wait for external arm

Advanced

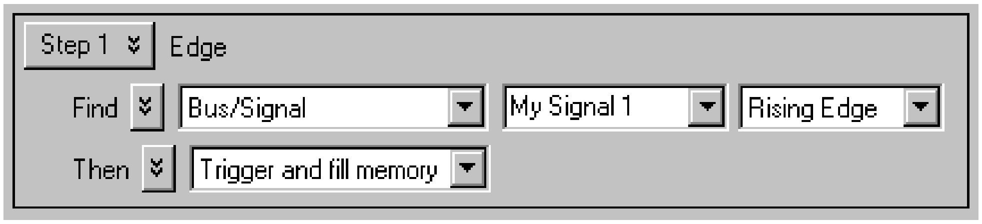
- Advanced If/Then
- Advanced 2-Way Branch
- Advanced 3-Way Branch
- Advanced 4-Way Branch
- Pattern "AND" Pattern
- Pattern "OR" Pattern

See Also

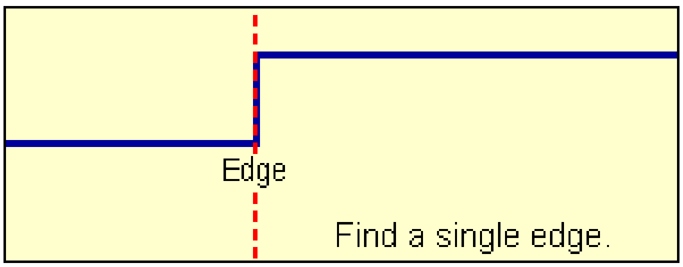
State mode trigger functions
To build a trigger sequence
To store a trigger

To recall a trigger
Simple Trigger

Edge



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a user-defined edge occurs.



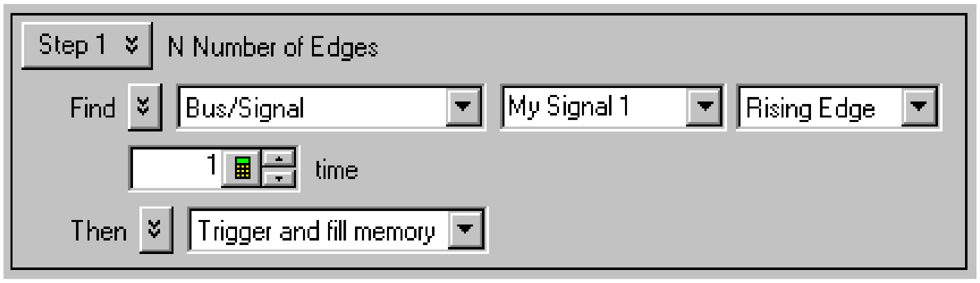
To edit this function

To insert events and actions

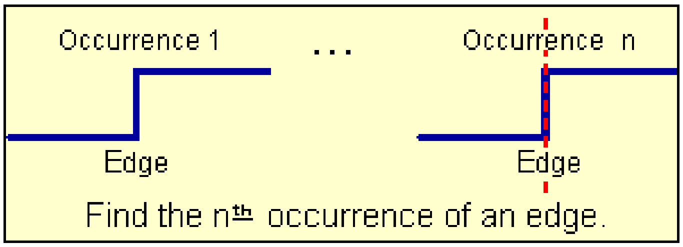
To modify trigger step display

To negate a function statement

"N" number of edges



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when the "Nth" occurrence of a user-defined edge occurs.



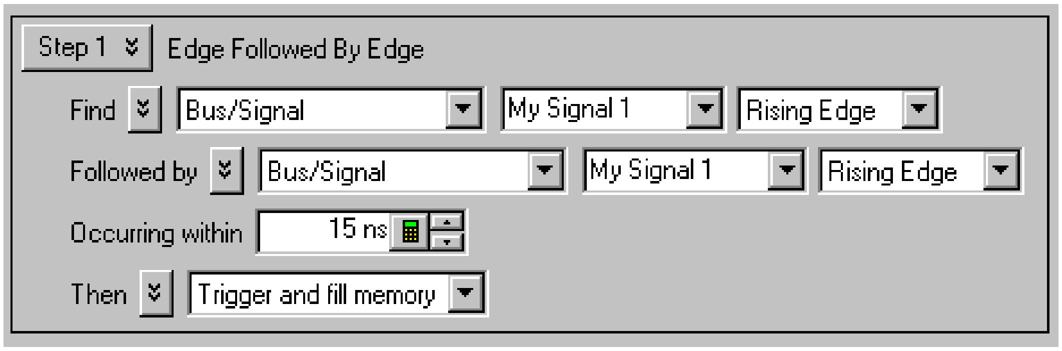
To edit this function

To insert events and actions

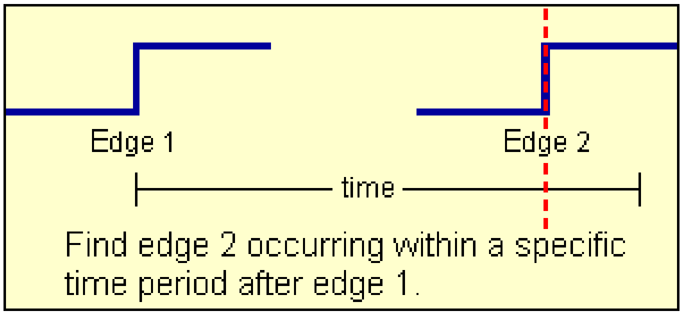
To modify trigger step display

To negate a function statement

Edge followed by edge



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when edge 2 occurs within a specified time period after edge 1.



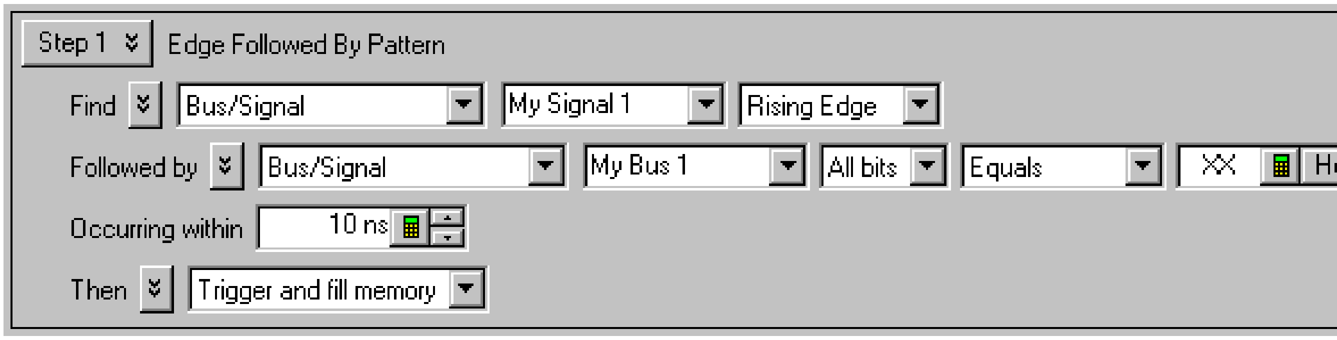
To edit this function

To insert events and actions

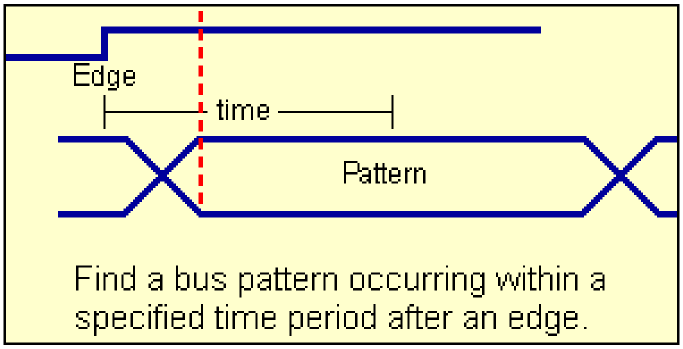
To modify trigger step display

To negate a function statement

Edge followed by pattern



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a bus pattern occurs within a specified time period after an edge.



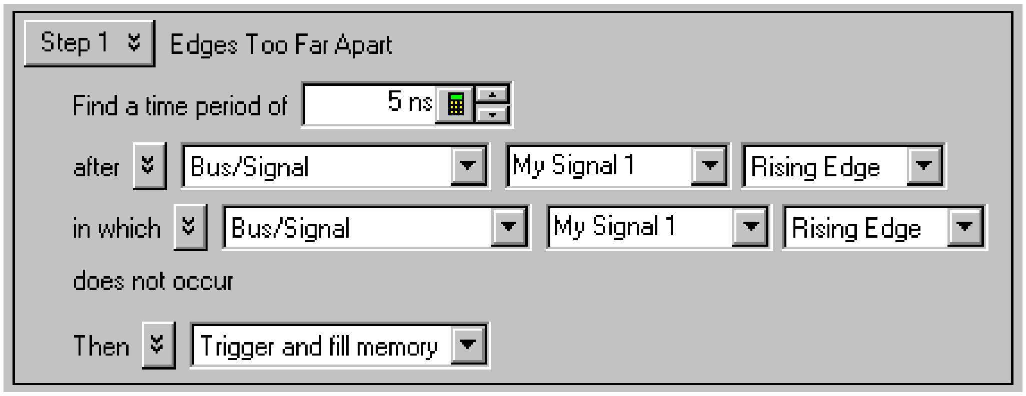
To edit this function

To insert events and actions

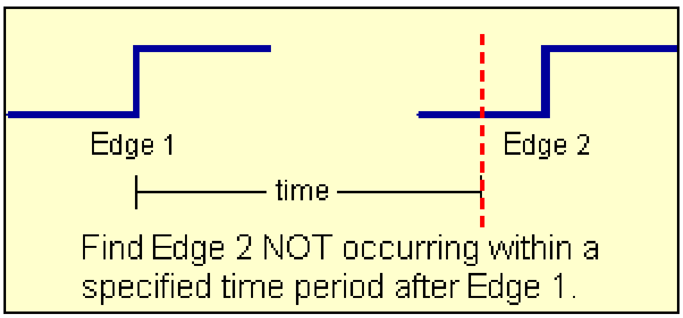
To modify trigger step display

To negate a function statement

Edges too far apart



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when edge 2 does not occur within a specified time period after edge 1.



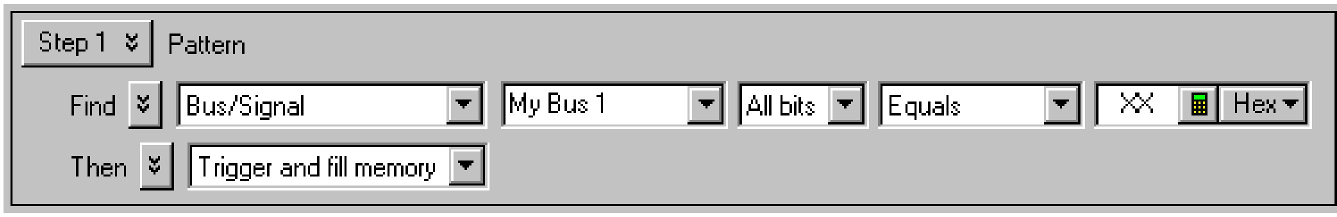
To edit this function

To insert events and actions

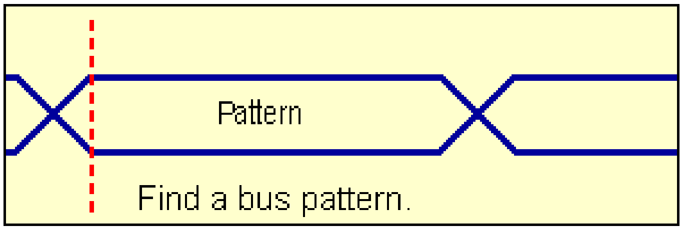
To modify trigger step display

To negate a function statement

Pattern



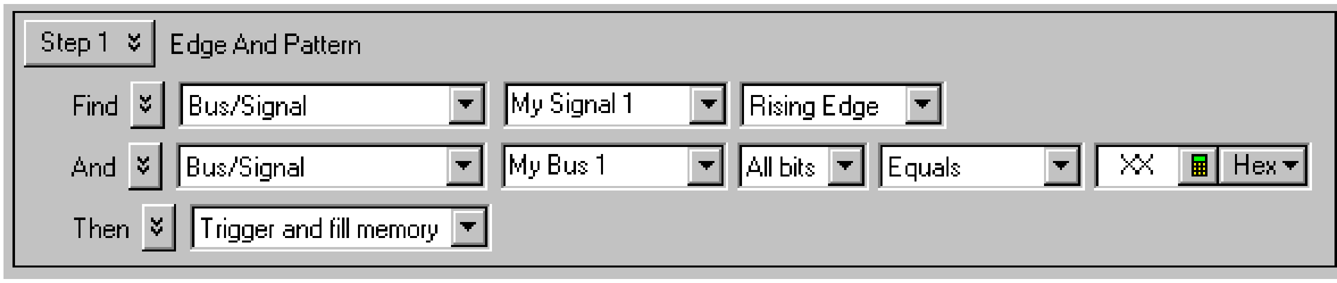
This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a designated bus pattern occurs.



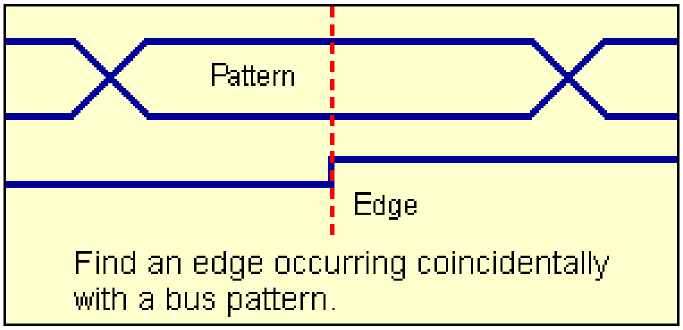
To edit this function

To insert events and actions
To modify trigger step display
To negate a function statement

Edge and Pattern



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when both a user-defined edge and bus pattern occur at the same time.



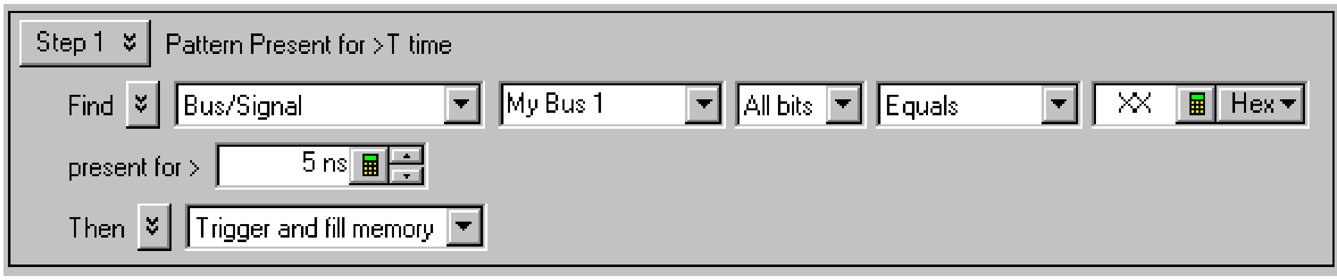
To edit this function

To insert events and actions

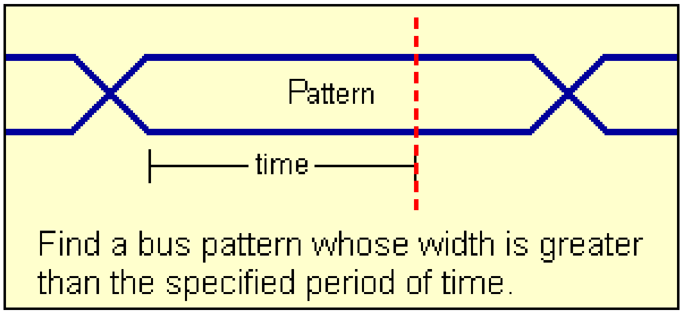
To modify trigger step display

To negate a function statement

Pattern present for > "T" time



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a user-defined bus pattern is present greater than a specified time period.



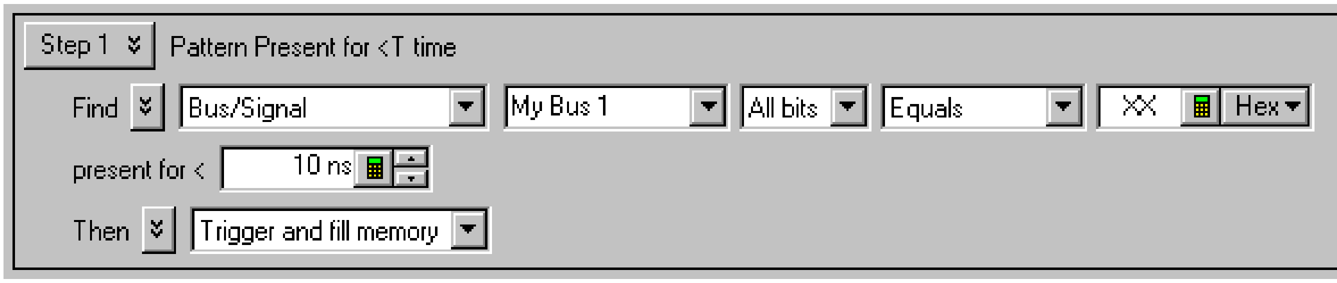
To edit this function

To insert events and actions

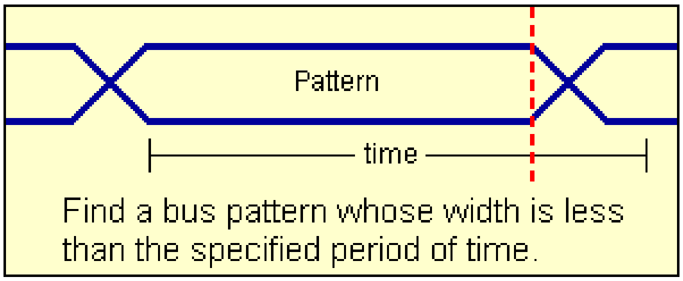
To modify trigger step display

To negate a function statement

Pattern present for < "T" time



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a user-defined bus pattern is present less than a specified time period.



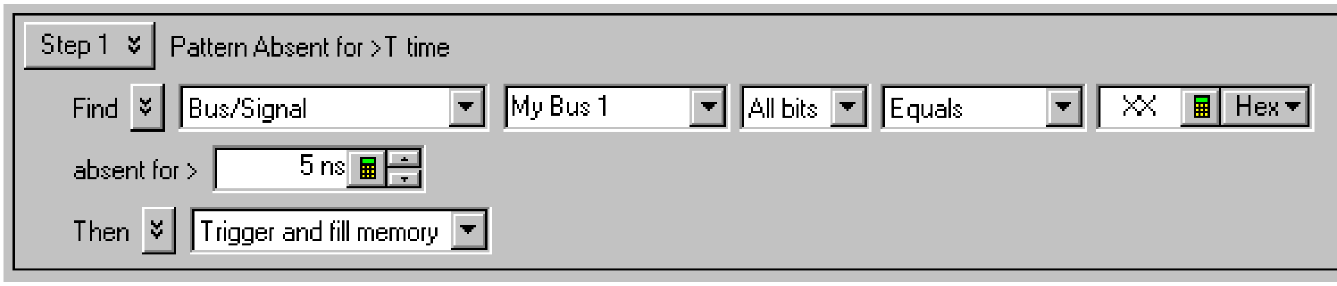
To edit this function

To insert events and actions

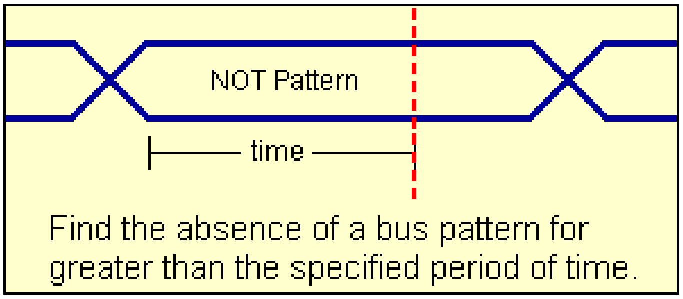
To modify trigger step display

To negate a function statement

Pattern absent for > "T" time



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a user-defined bus pattern is absent greater than a specified time period.



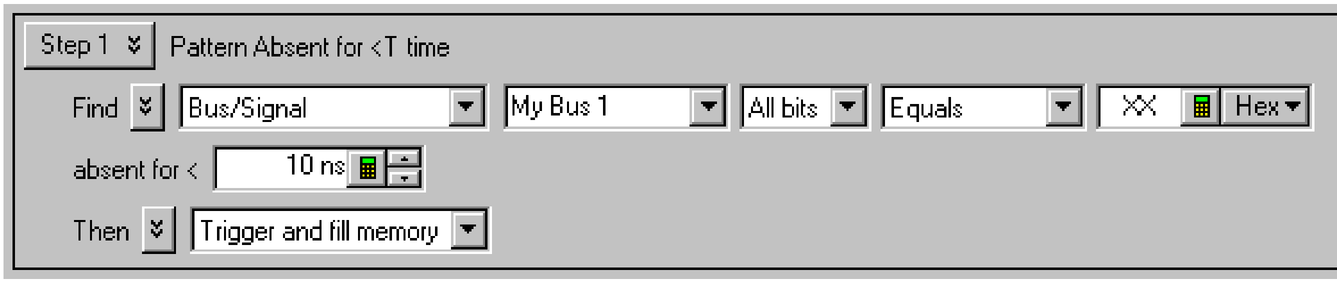
To edit this function

To insert events and actions

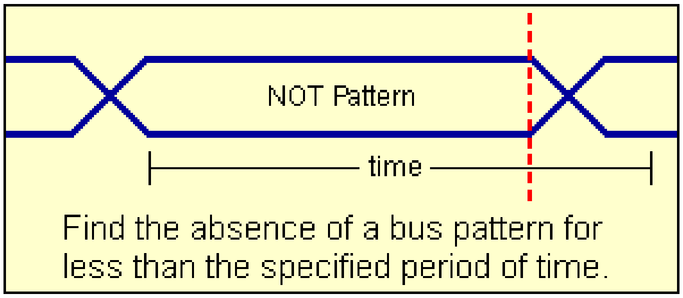
To modify trigger step display

To negate a function statement

Pattern absent for < "T" time



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a user-defined bus pattern is absent less than a specified time period.



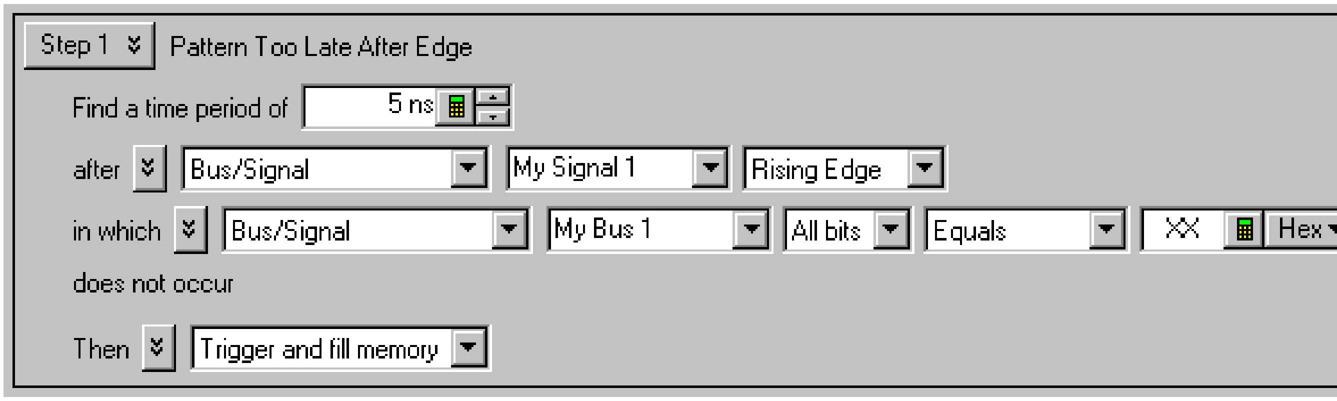
To edit this function

To insert events and actions

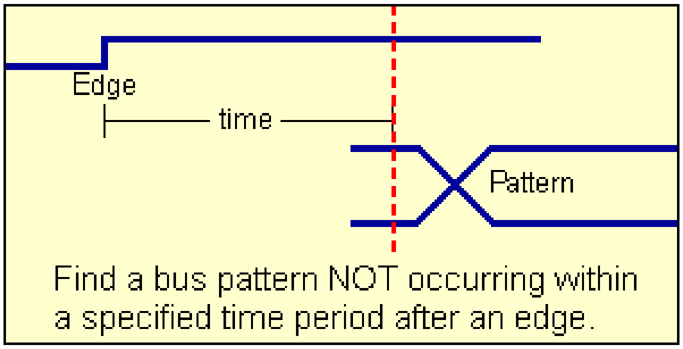
To modify trigger step display

To negate a function statement

Pattern too late after edge



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a specified bus pattern does not occur within a specified time period after an edge.



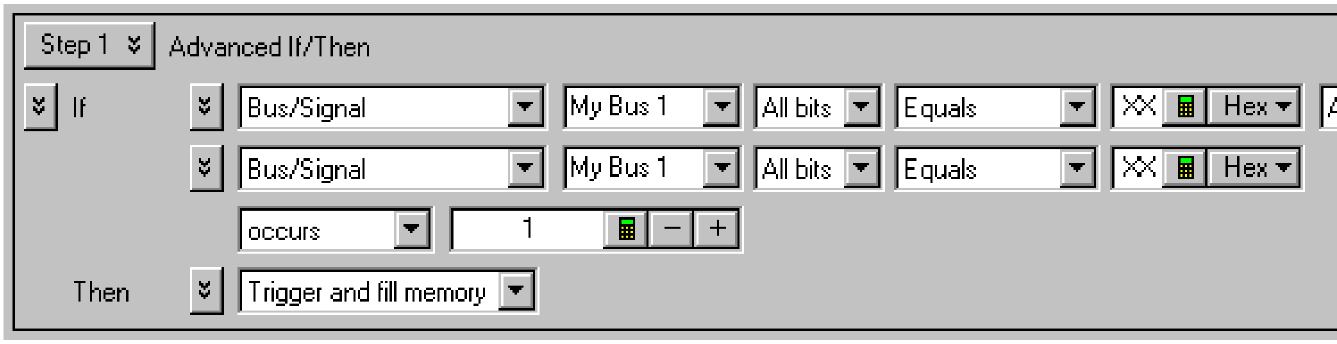
To edit this function

To insert events and actions

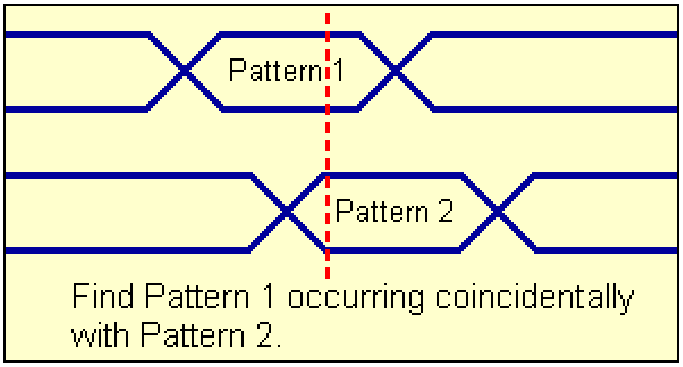
To modify trigger step display

To negate a function statement

Pattern "AND" Pattern (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when both pattern1 "AND" pattern2 occur at the same time, and for the specified numbers of samples (occurs).



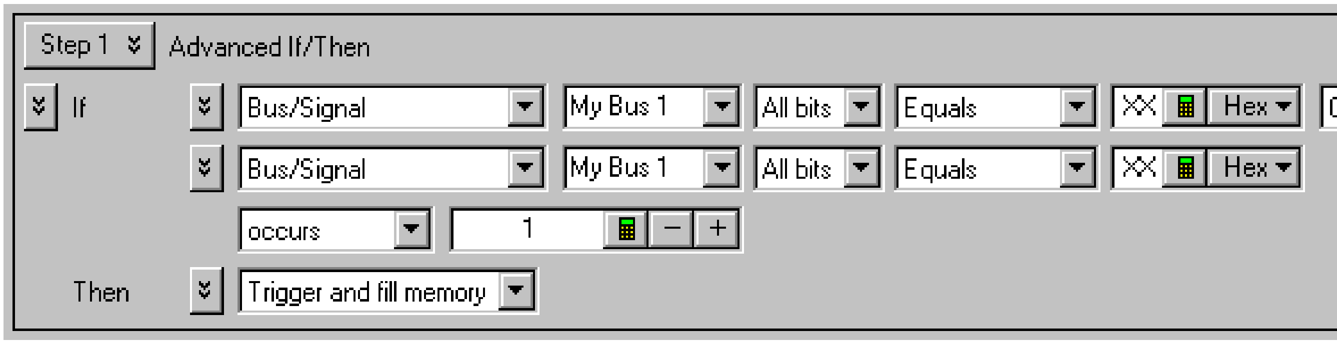
To edit this function

To insert events and actions

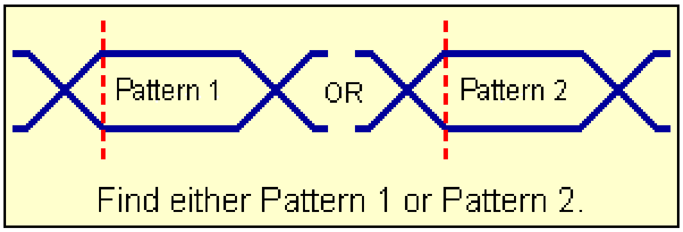
To modify trigger step display

To negate a function statement

Pattern "OR" Pattern (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when either pattern1 "OR" pattern2 occurs for the specified numbers of samples (occurs).



To edit this function

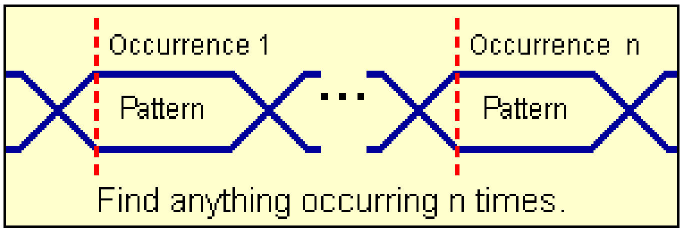
To insert events and actions

To modify trigger step display

To negate a function statement

Find anything "N" times (timing)

This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when it sees any data (Anything) for the Nth time.



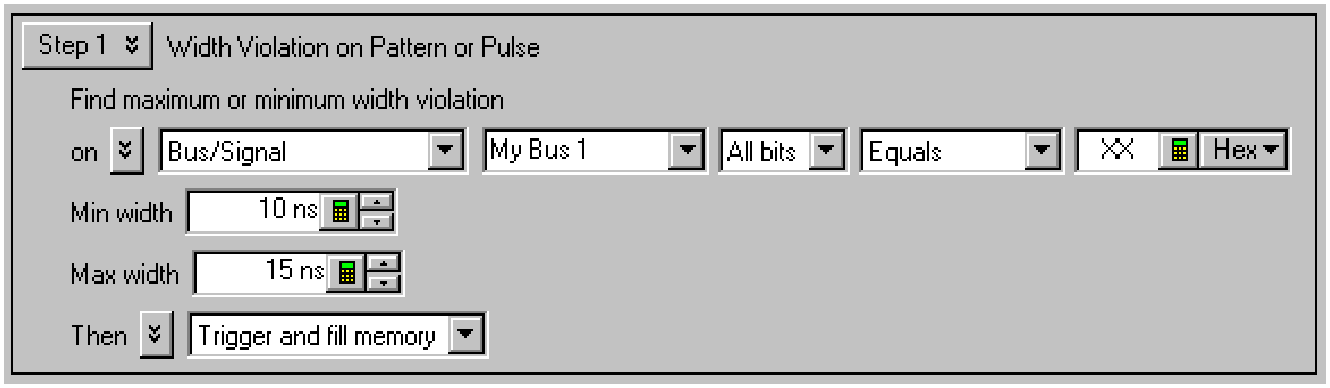
To edit this function

To insert events and actions

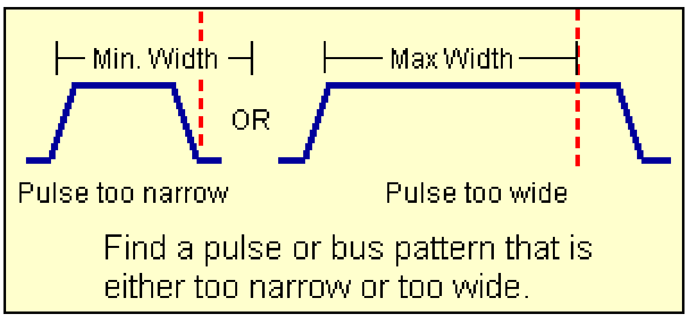
To modify trigger setup display

To negate a function statement

Width violation on pattern or pulse



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when a pulse or bus pattern is found that is either too narrow or too wide.



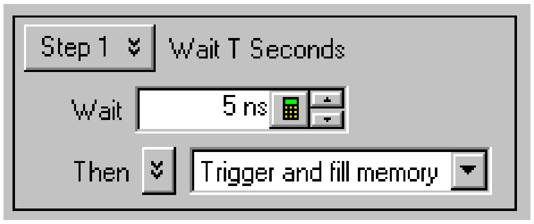
To edit this function

To insert events and actions

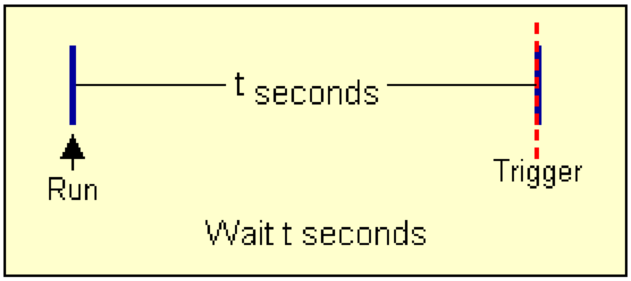
To modify trigger step display

To negate a function statement

Wait "T" seconds



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers after the specified time period expires.



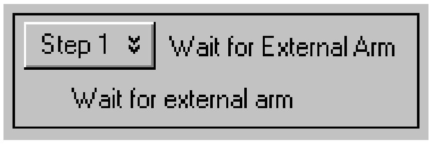
To edit this function

To insert events and actions

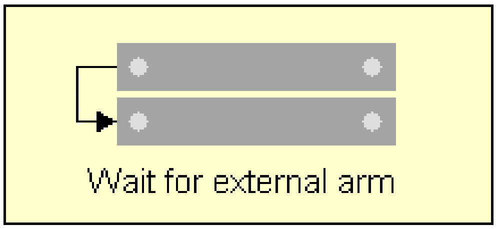
To modify trigger step display

To negate a function statement

Wait for external arm (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when an external arming signal appears through the external trigger in port. The external trigger port is located on the rear panel of the 168X models and the front panel of the 169X models.



To edit this function

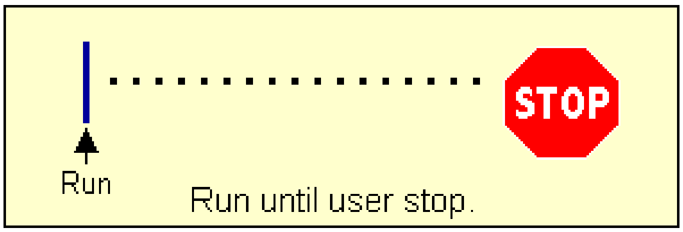
To insert events and actions

To modify trigger step display

To negate a function statement

Run until user stop (timing)

This trigger function is available when the acquisition mode is set to Timing - Asynchronous. This trigger function sets up to never trigger. You must select the stop button to view the captured data.



To edit this function

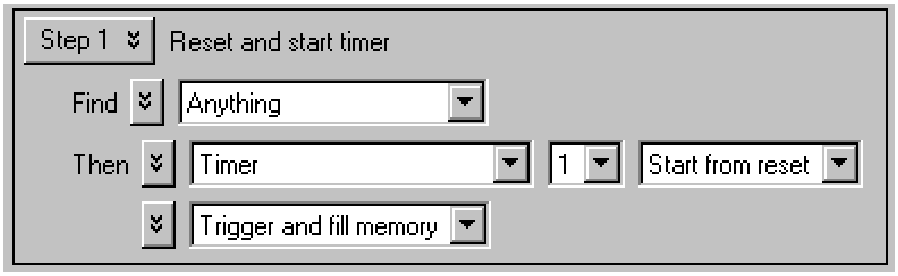
To set store qualification

To insert events and actions

To modify trigger step display

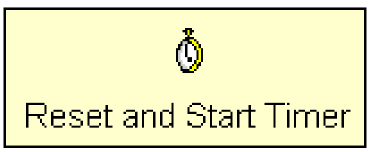
To negate a function statement

Reset and start timer (timing)



Note: This trigger function is not available in the 1683A/93A models because they do not have timers available.

This trigger function is available when the acquisition mode is set to Timing - Asynchronous. This trigger function resets a timer, then starts the timer for a specified period of time. This trigger function requires that the timer value be set in either the same trigger step, or another trigger step that follows. When the timer stops, the analyzer triggers. For more information refer to "To configure a timer".



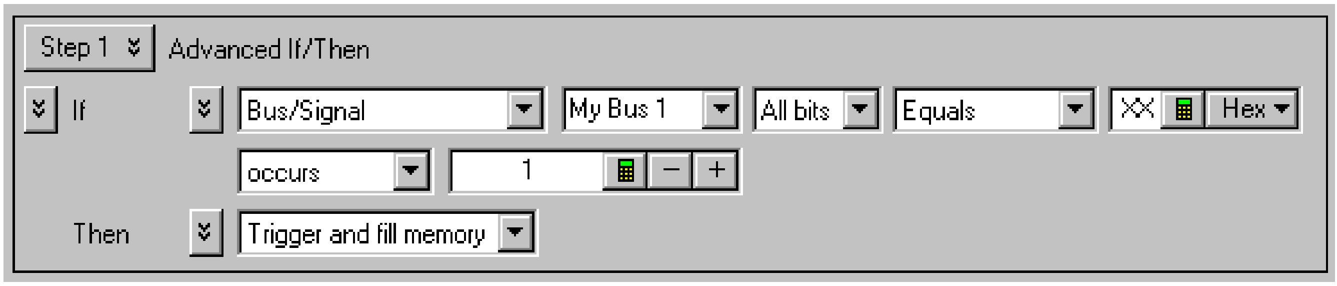
To edit this function

To insert events and actions

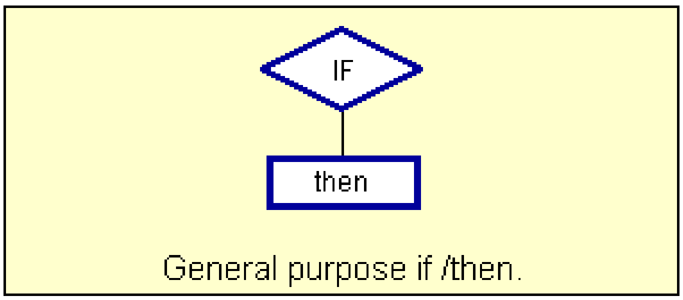
To modify trigger step display

To negate a function statement

Advanced If/Then (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The analyzer triggers when the "If" clause becomes true.



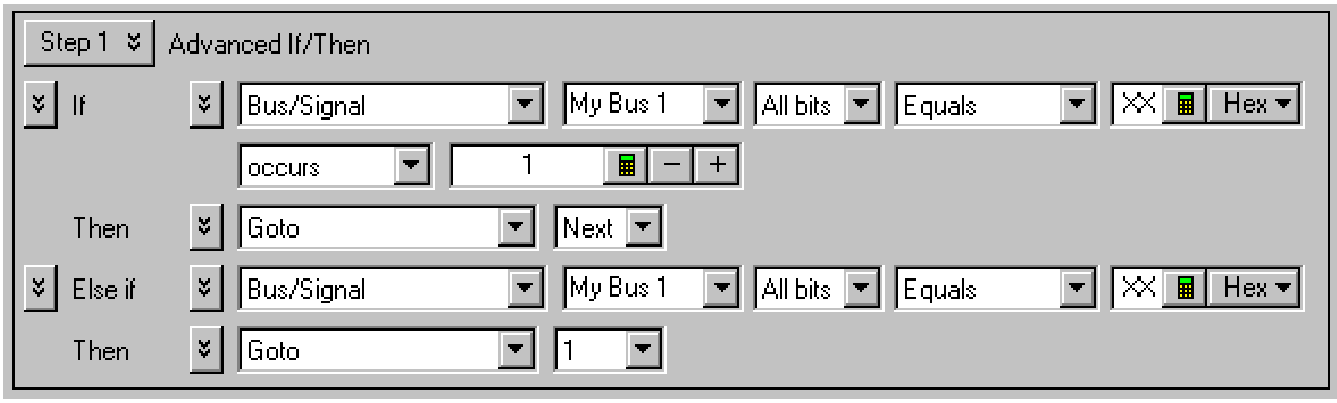
To edit this function

To insert events and actions

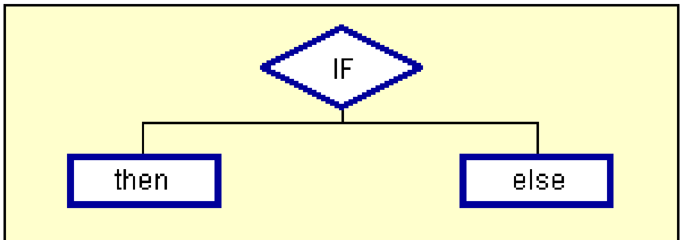
To modify trigger step display

To negate a function statement

Advanced 2-Way Branch (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The two-way branch is evaluated true when either of two patterns (if or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



General purpose if /then /else.

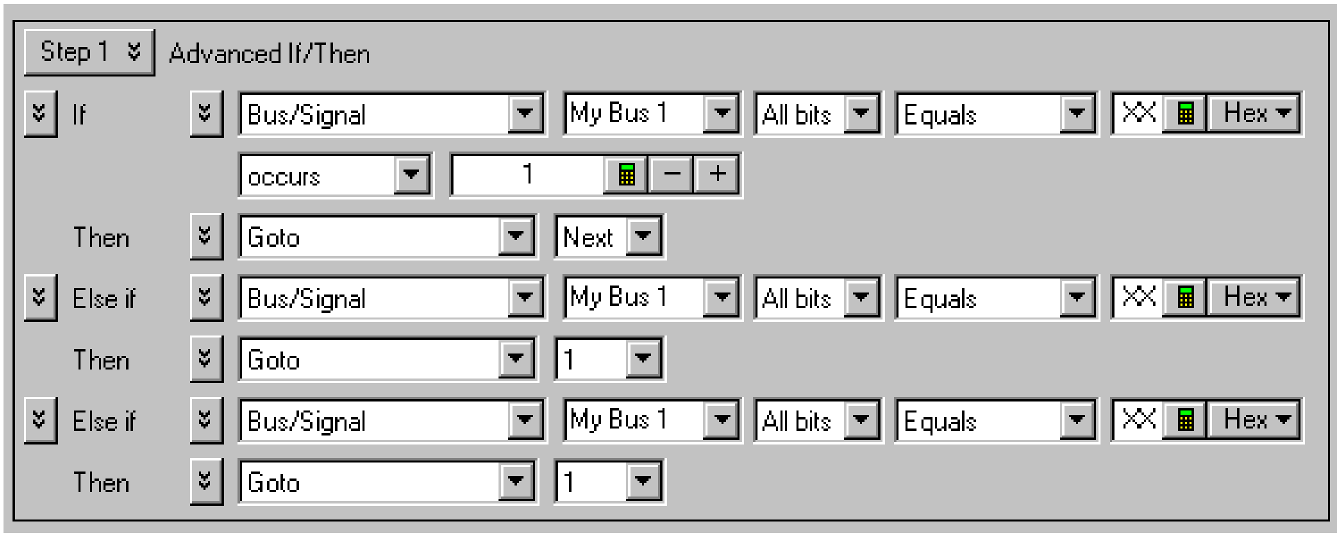
To edit this function

To insert events and actions

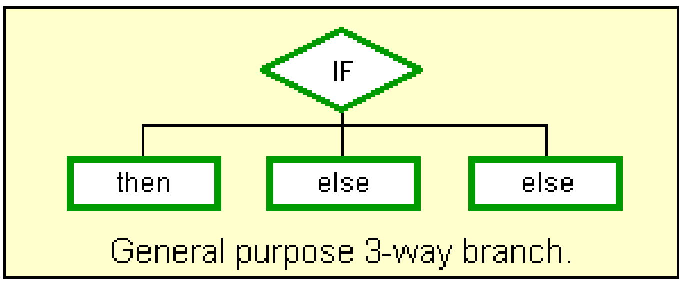
To modify trigger step display

To negate a function statement

Advanced 3-Way Branch (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The three-way branch is evaluated true when either of three patterns (If or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



General purpose 3-way branch.

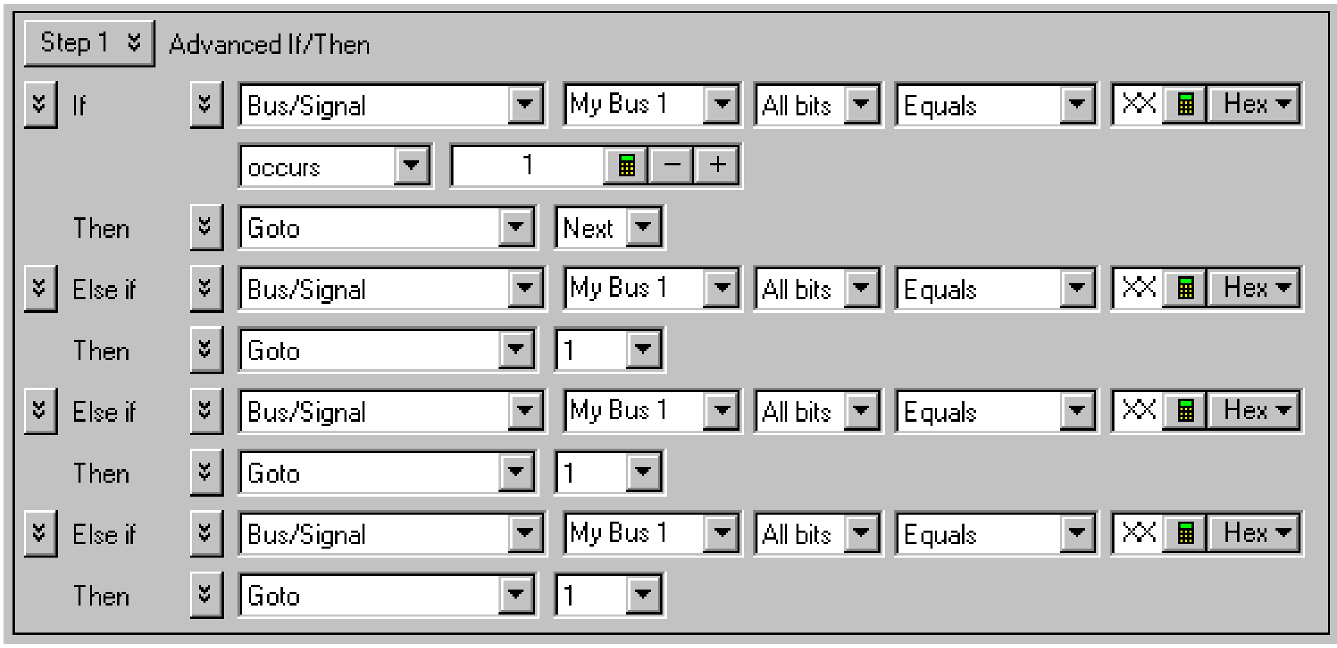
To edit this function

To insert events and actions

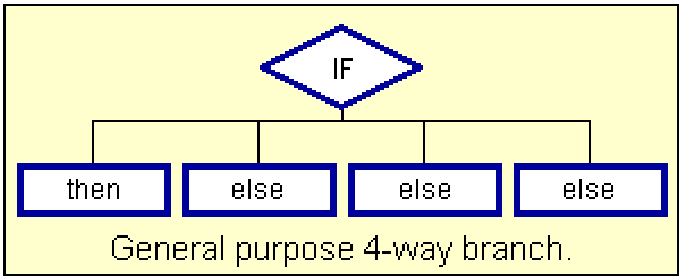
To modify trigger step display

To negate a function statement

Advanced 4-Way Branch (timing)



This trigger function is available when the acquisition mode is set to Timing - Asynchronous. The four-way branch is evaluated true when either of four patterns (If or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



To edit this function

To insert events and actions

To modify trigger step display

To negate a function statement

State Mode

State mode trigger functions

Patterns

- Pattern "N" times
- "N" consecutive samples with Pattern1
- Pattern1 followed by Pattern2
- Pattern1 immediately followed by Pattern2
- Pattern1 followed by Pattern2 before Pattern3
- Too few states between Pattern1 and Pattern2
- Too many states between Pattern1 and Pattern2
- Pattern2 occurring too soon after Pattern1
- Pattern2 occurring too late after Pattern1

Other

- Reset and start timer
- Find anything "N" times
- Run until user stop
- Wait for external arm
- Wait "N" external clock states

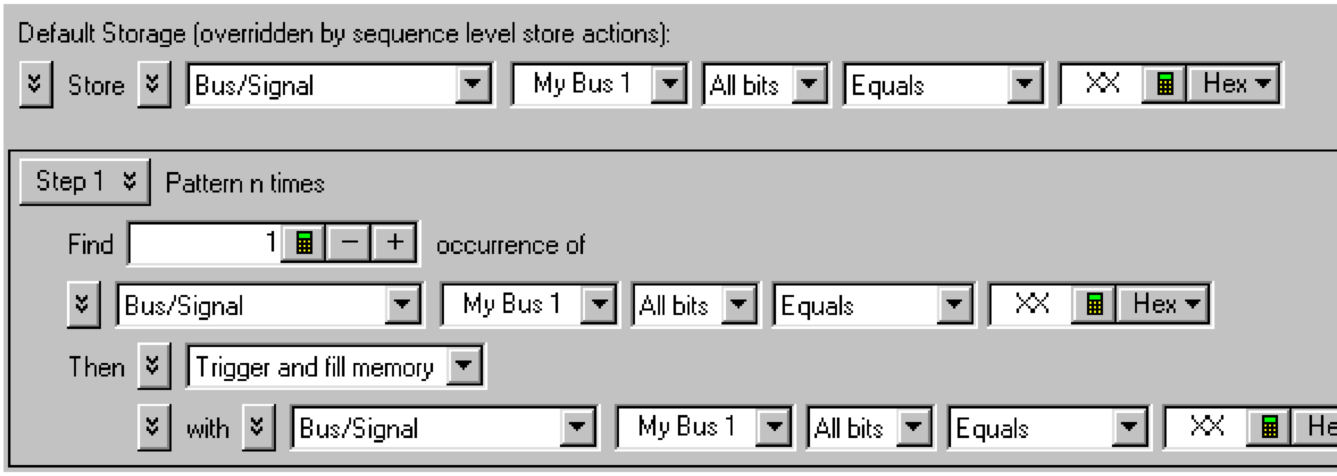
Advanced

- Advanced If/Then
- Advanced 2-Way Branch
- Advanced 3-Way Branch
- Advanced 4-Way Branch
- Pattern "AND" Pattern
- Pattern "OR" Pattern

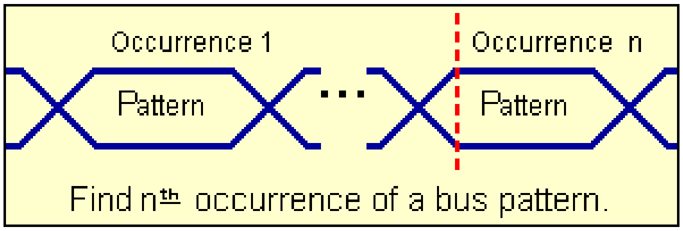
See Also

Timing mode trigger functions
To build a trigger sequence
To set store qualification
To store a trigger
To recall a trigger
Simple Trigger

Pattern "N" times



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when it finds the nth occurrence of a bus pattern as shown below.



To edit this function

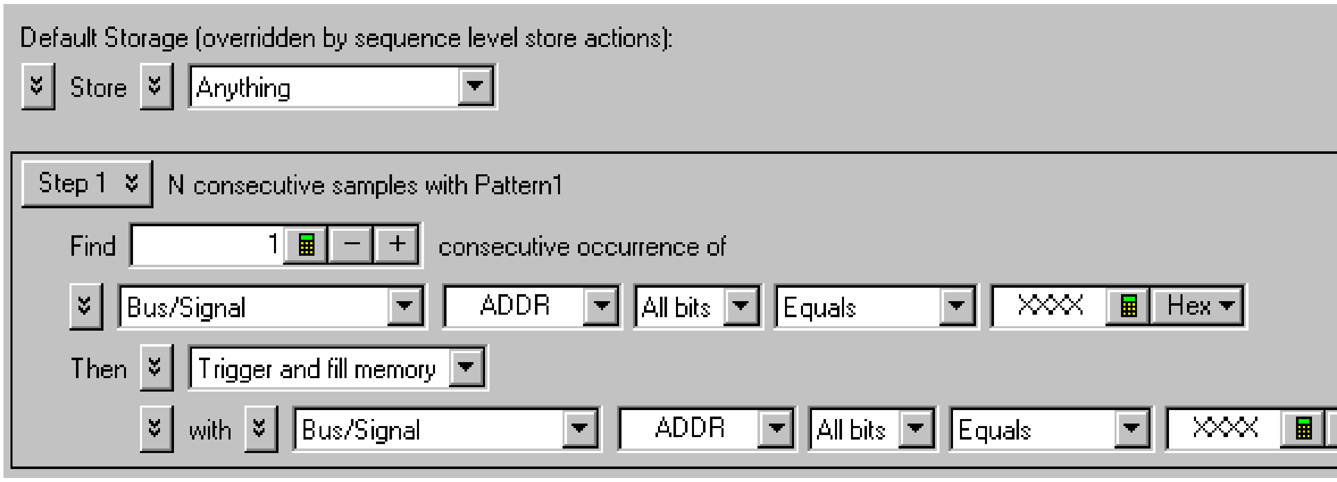
To set store qualification

To insert events and actions

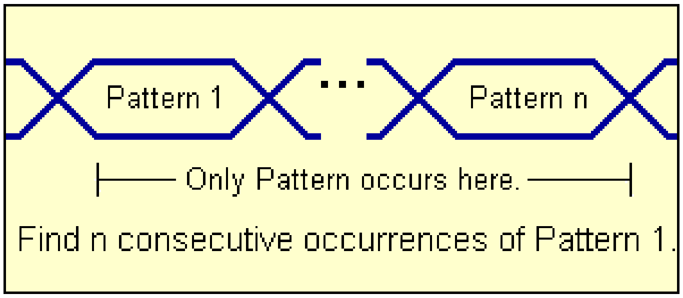
To modify trigger step display

To negate a function statement

"N" consecutive samples with Pattern1



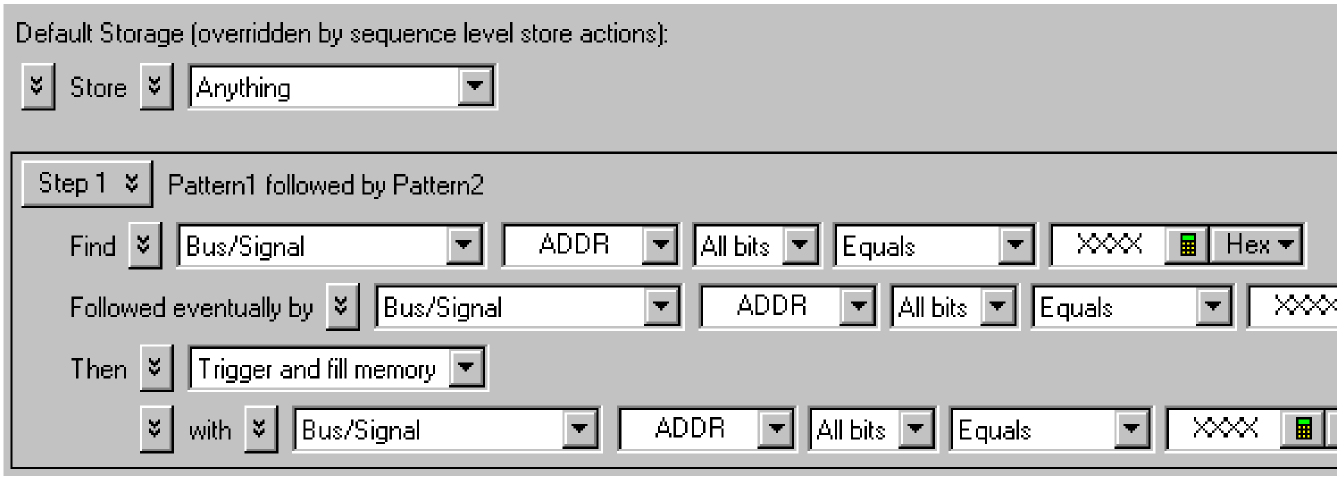
This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when a bus pattern occurs a specified number times.



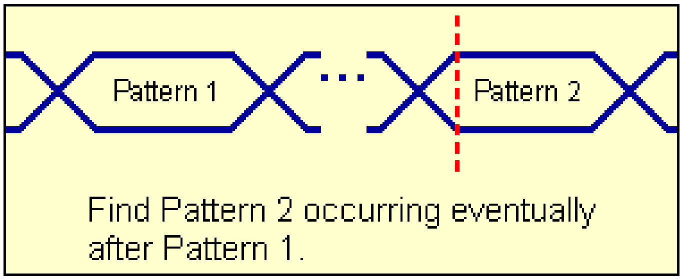
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Pattern1 followed by Pattern2



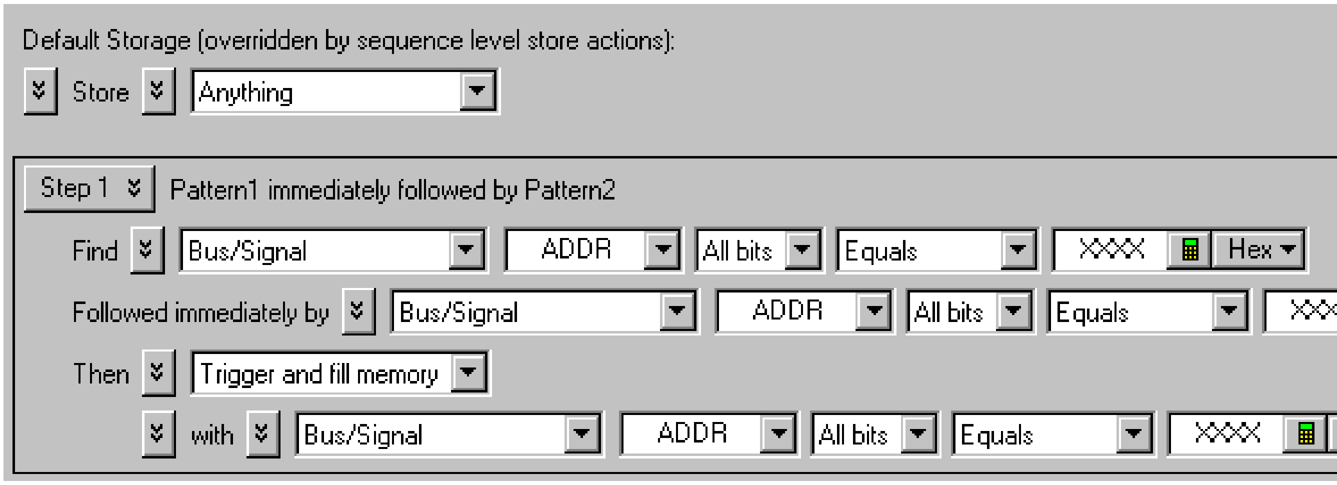
This trigger function is available when the acquisition mode is set to State - Synchronous.
It will trigger the logic analyzer when pattern2 occurs eventually after pattern 1.



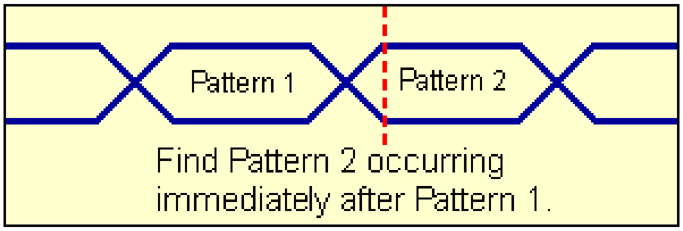
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Pattern1 immediately followed by Pattern2



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern 2 is found immediately after exiting pattern 1.



To edit this function

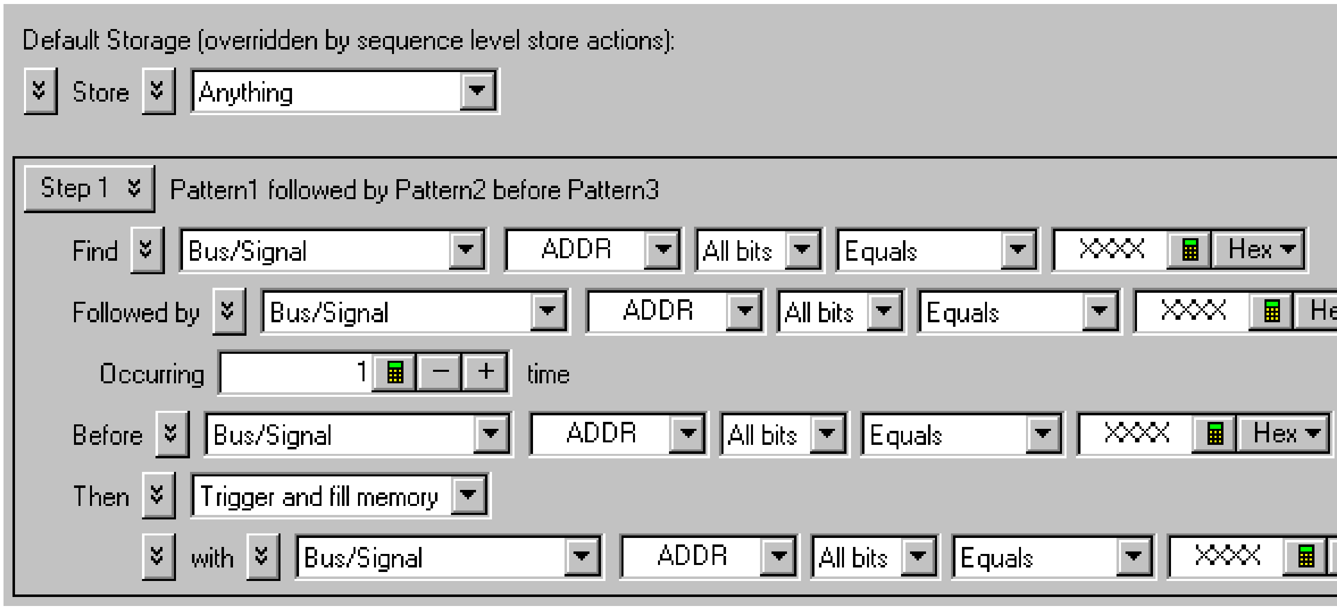
To set store qualification

To insert events and actions

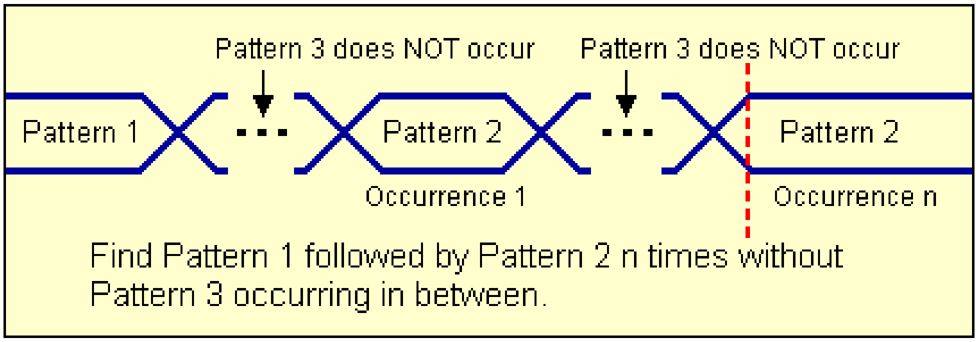
To modify trigger step display

To negate a function statement

Pattern1 followed by Pattern2 before Pattern3



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern2 occurs eventually after pattern1, for a specified number of times, without pattern3 occurring in between.



To edit this function

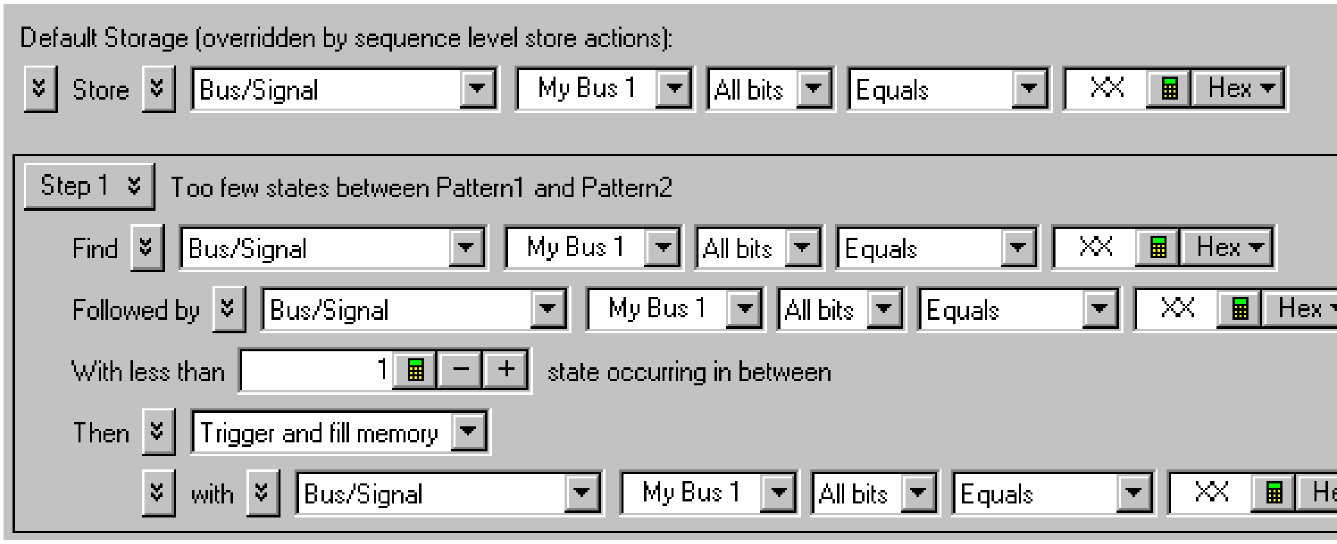
To set store qualification

To insert events and actions

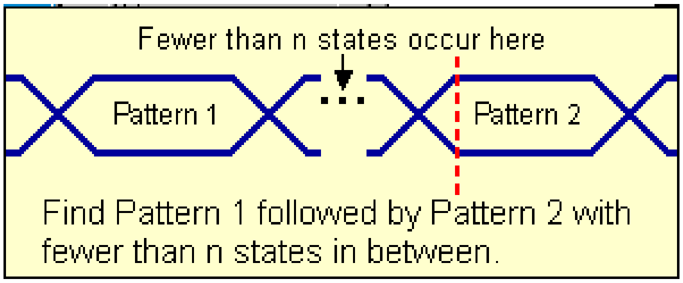
To modify trigger step display

To negate a function statement

Too few states between Pattern1 and Pattern2



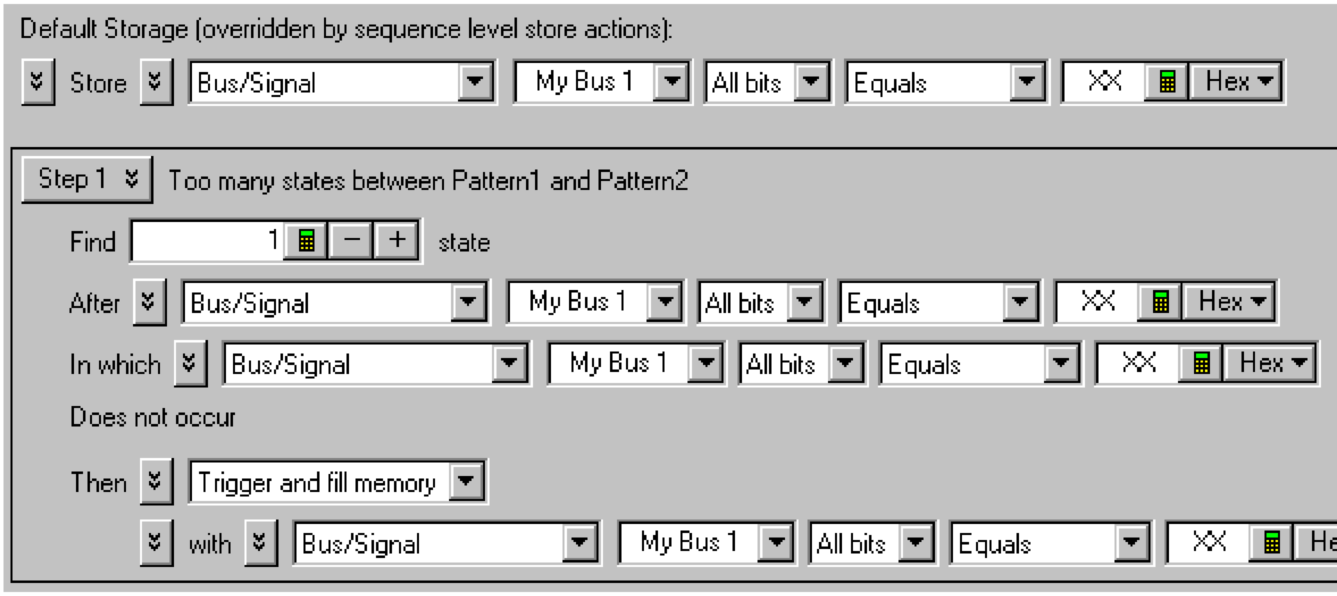
This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern1 is followed by pattern2 with fewer than "N" specified states in between.



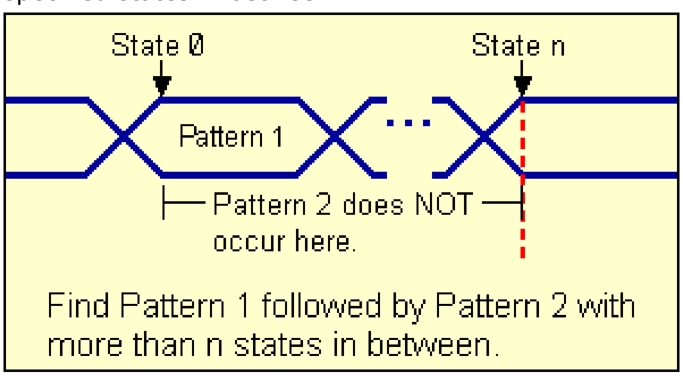
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Too many states between Pattern1 and Pattern2



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern1 is followed by pattern2 with more than "N" specified states in between.



To edit this function

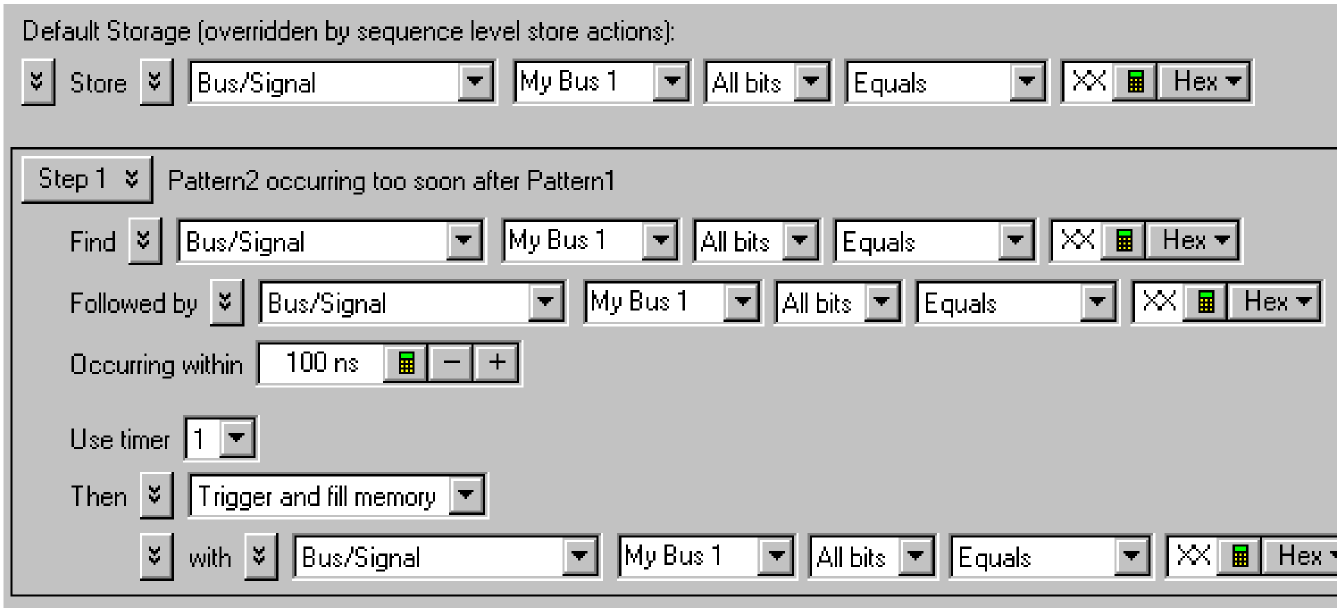
To set store qualification

To insert events and actions

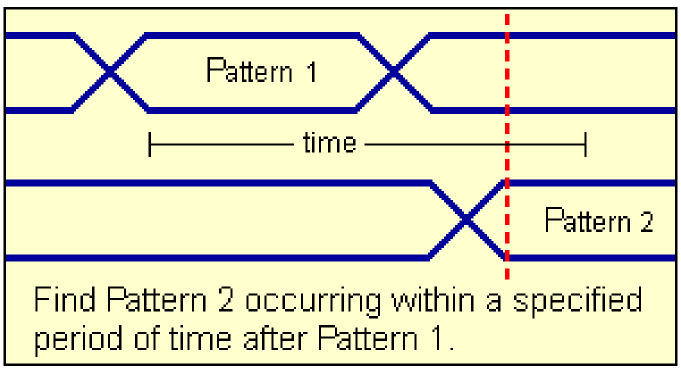
To modify trigger step display

To negate a function statement

Pattern2 occurring too soon after Pattern1



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern2 occurs within a specified time period after pattern1.



To edit this function

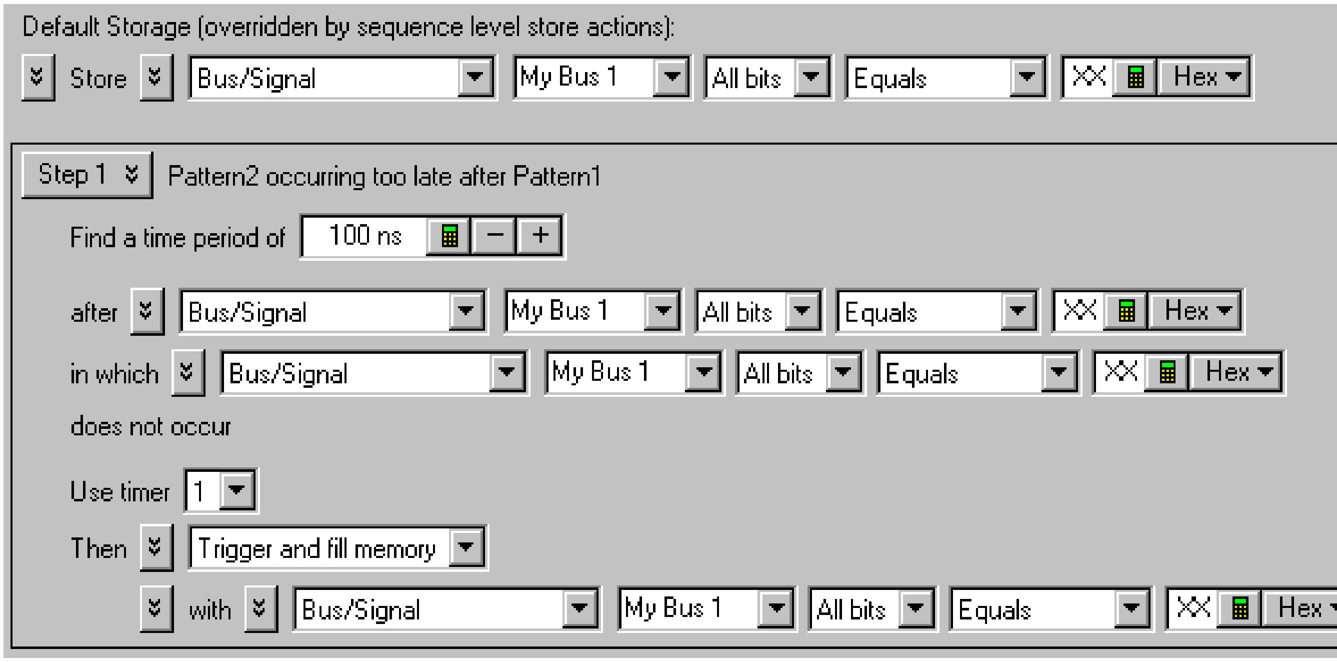
To set store qualification

To insert events and actions

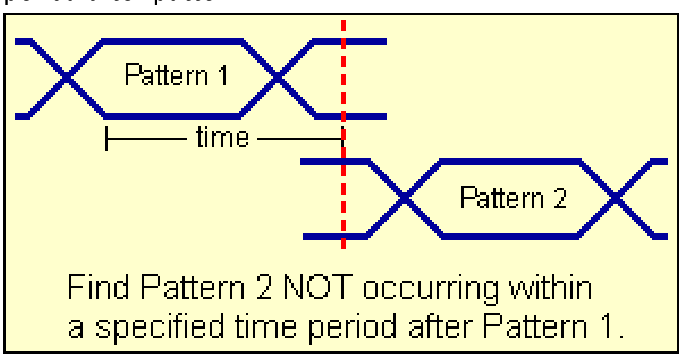
To modify trigger step display

To negate a function statement

Pattern2 occurring too late after Pattern1



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when pattern2 does not occur within a specified time period after pattern1.



To edit this function

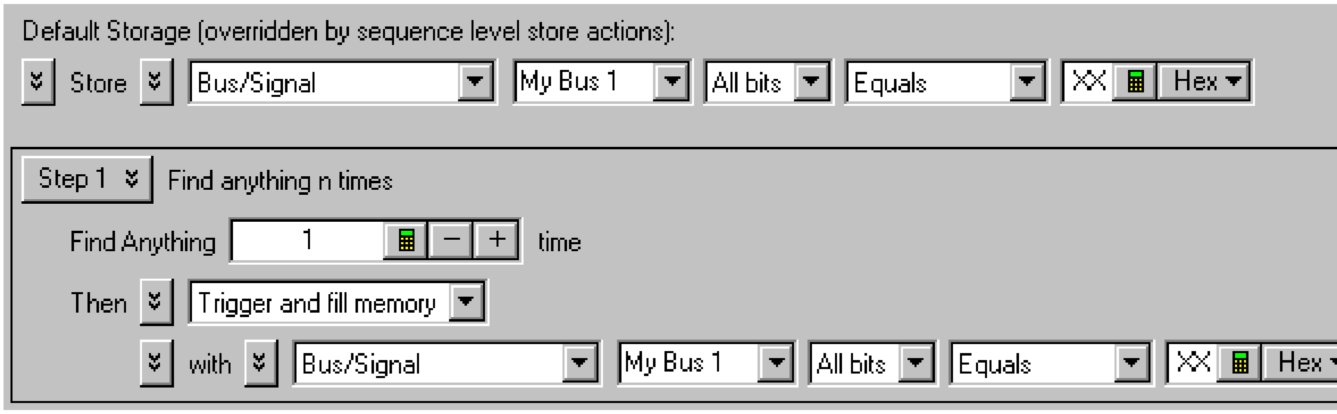
To set store qualification

To insert events and actions

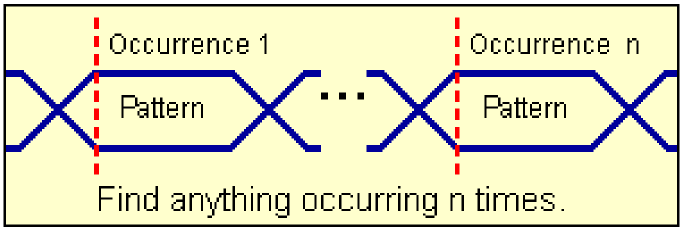
To modify trigger step display

To negate a function statement

Find anything "N" times (state)



This trigger function is available when the acquisition mode is set to State - Synchronous. It will trigger the logic analyzer when any data (Anything) is seen for the Nth time. It is commonly used to create an immediate trigger, or a trigger after a user-defined delay.



To edit this function

To set store qualification

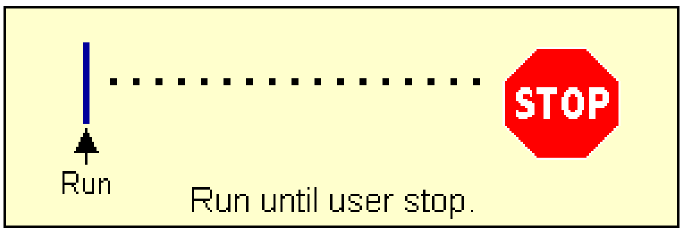
To insert events and actions

To modify trigger step display

To negate a function statement

Run until user stop (state)

This trigger function is available when the acquisition mode is set to State - Synchronous. This trigger function sets up to never trigger. You must select the stop button to view the captured data.



To edit this function

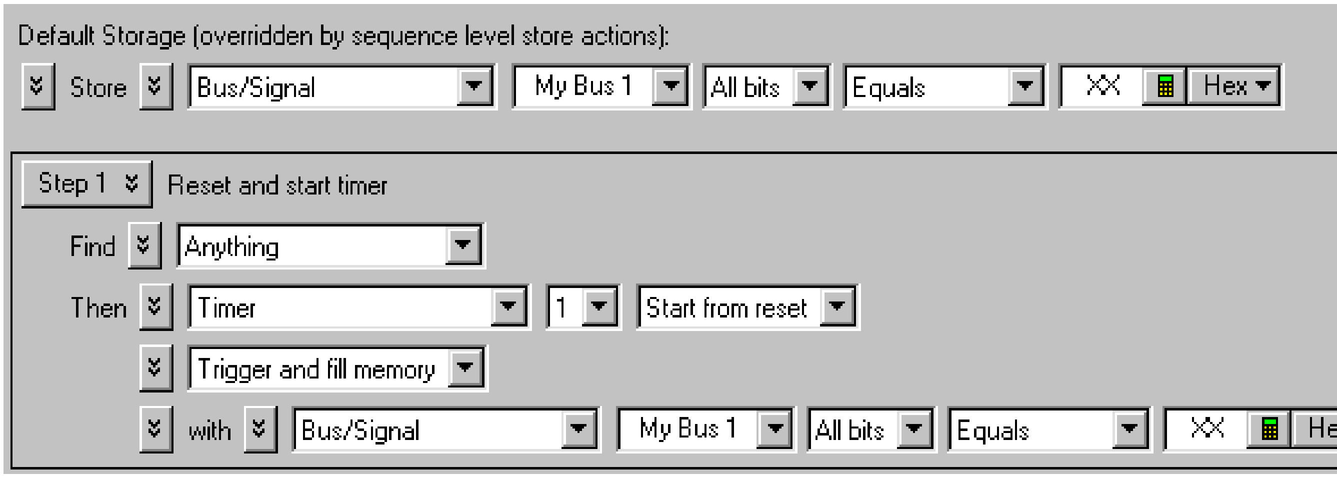
To set store qualification

To insert events and actions

To modify trigger step display

To negate a function statement

Reset and start timer (state)



Note: This trigger function is not available in the 1683A/93A models because they do not have timers available.

This trigger function is available when the acquisition mode is set to State - Synchronous. This trigger function resets a timer, then starts the timer for a specified period of time. This trigger function requires that the timer value be set in either the same trigger step, or another trigger step that follows. When the timer stops, the analyzer triggers. For more information refer to "To configure a timer".

To edit this function

To set store qualification

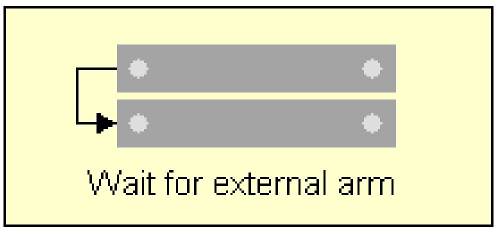
To insert events and actions

To modify trigger step display

To negate a function statement

Wait for external arm (state)

This trigger function is available when the acquisition mode is set to State - Synchronous. The analyzer triggers when an external arming signal appears through the external trigger in port. The external trigger port is located on the rear panel of the 168X models and the front panel of the 169X models.



To edit this function

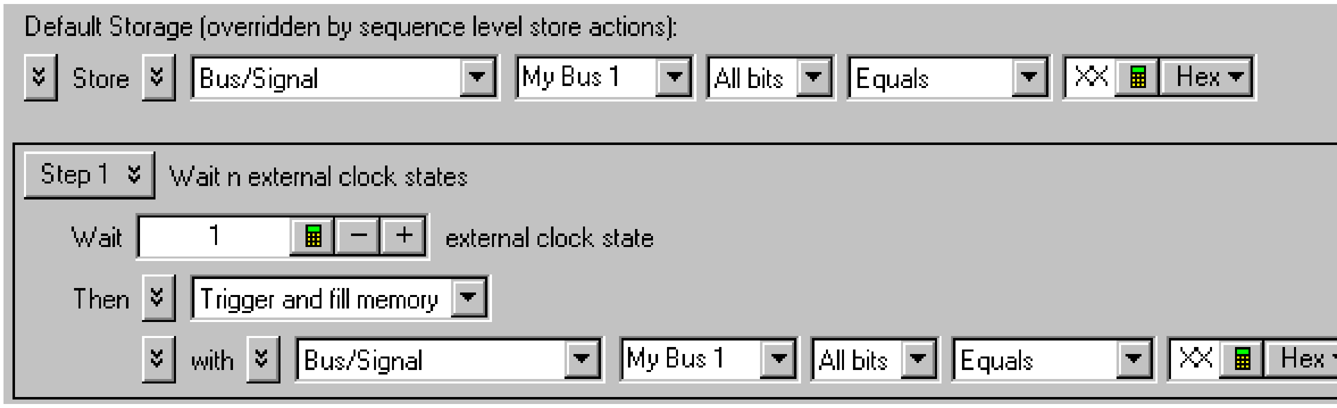
To set store qualification

To insert events and actions

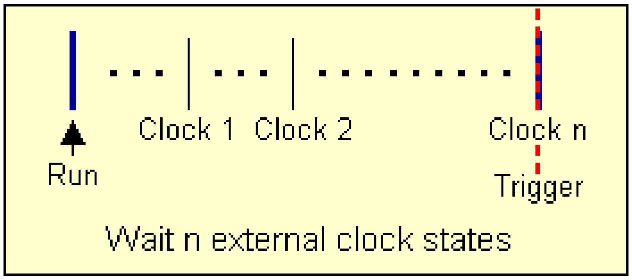
To modify trigger step display

To negate a function statement

Wait "N" external clock states



This trigger function is available when the acquisition mode is set to State - Synchronous. The analyzer triggers on the "Nth" occurrence of the external clock signal (plus any user-defined clock qualification) from the device under test.



To edit this function

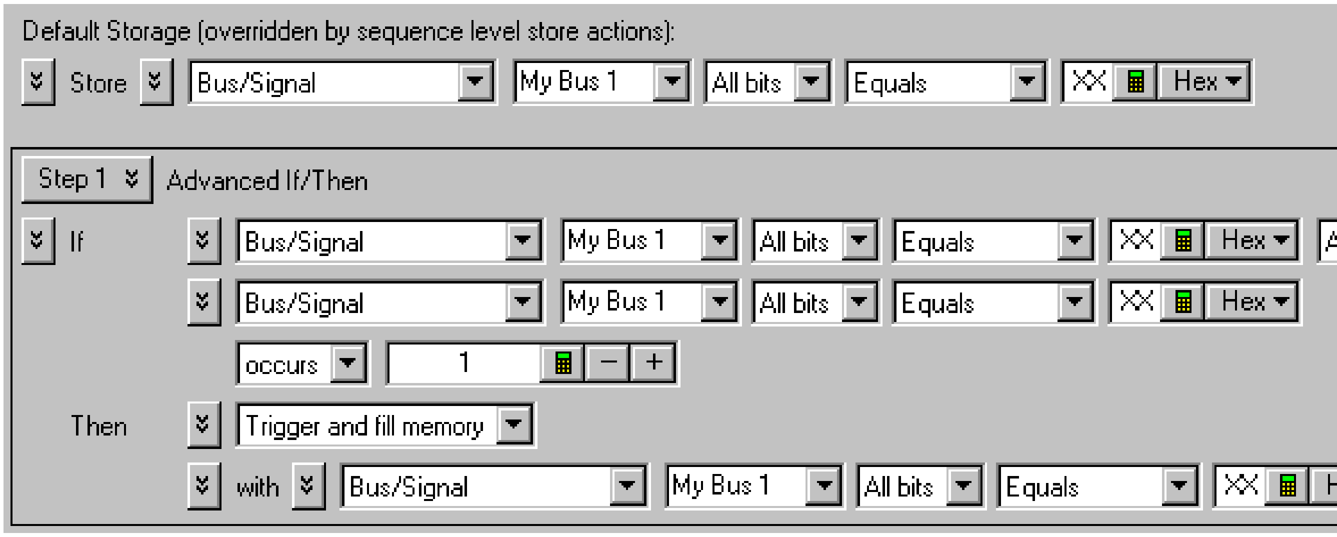
To set store qualification

To insert events and actions

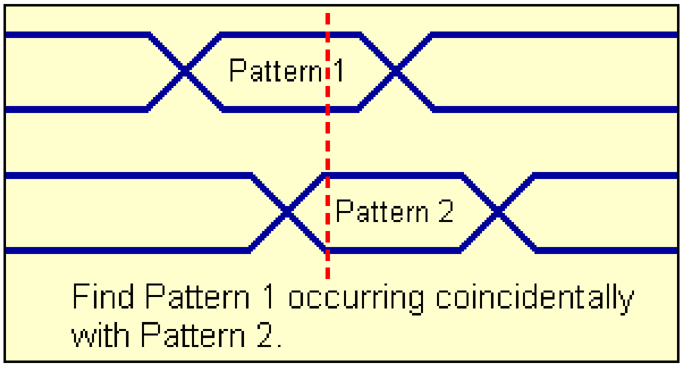
To modify trigger step display

To negate a function statement

Pattern "AND" Pattern (state)



This trigger function is available when the acquisition mode is set to State - Synchronous. The analyzer triggers when both pattern1 "AND" pattern2 occur at the same time, and for the specified numbers of samples (occurs).



To edit this function

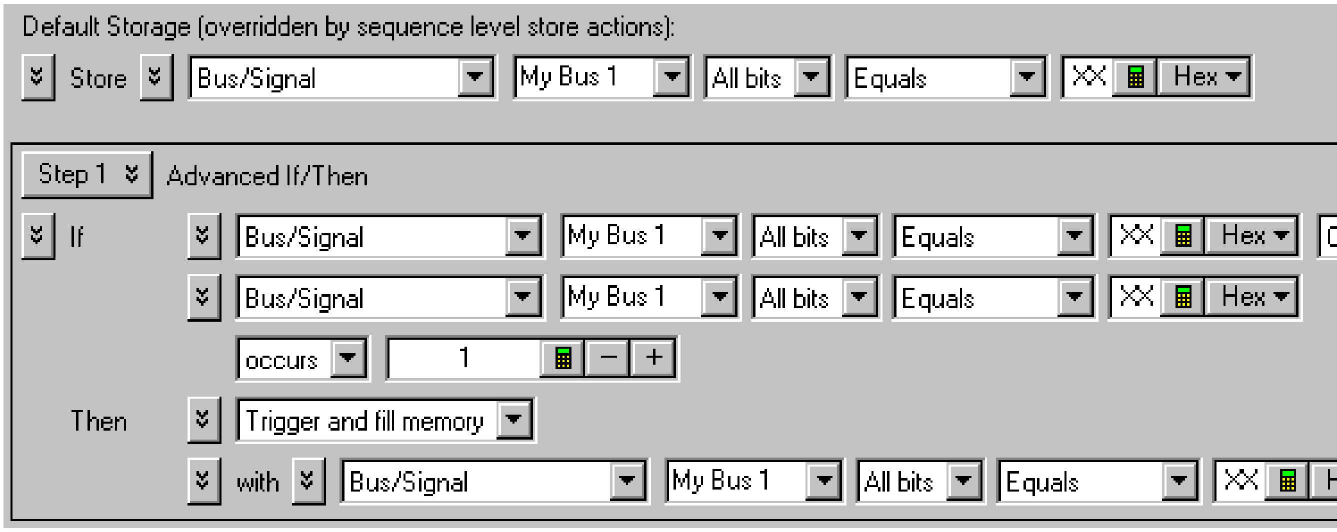
To set store qualification

To insert events and actions

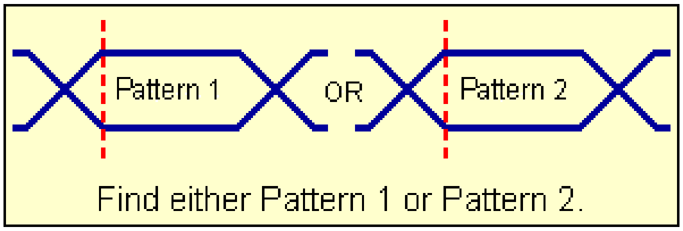
To modify trigger step display

To negate a function statement

Pattern "OR" Pattern (state)



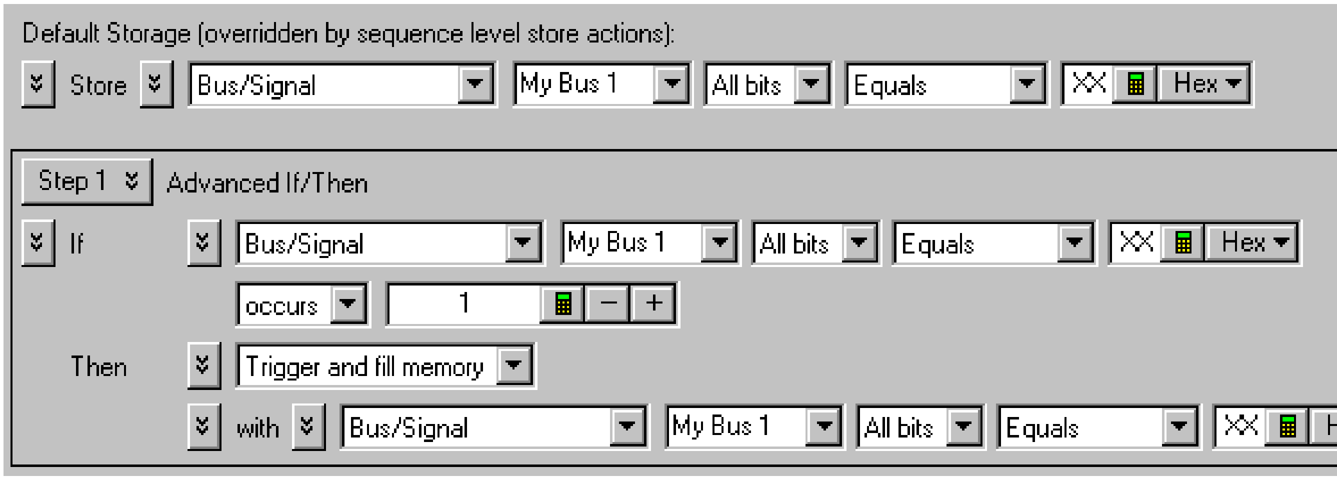
This trigger function is available when the acquisition mode is set to State - Synchronous. The analyzer triggers when either pattern1 "OR" pattern2 occurs for the specified numbers of samples (occurs).



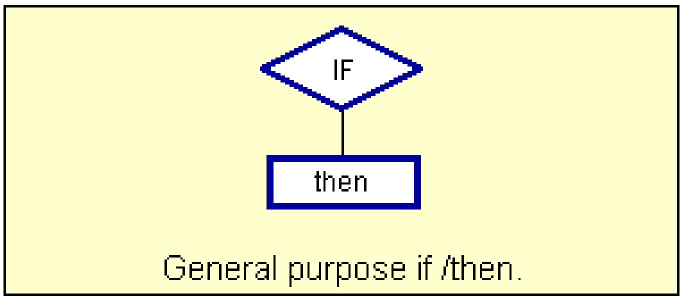
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Advanced If/Then (state)



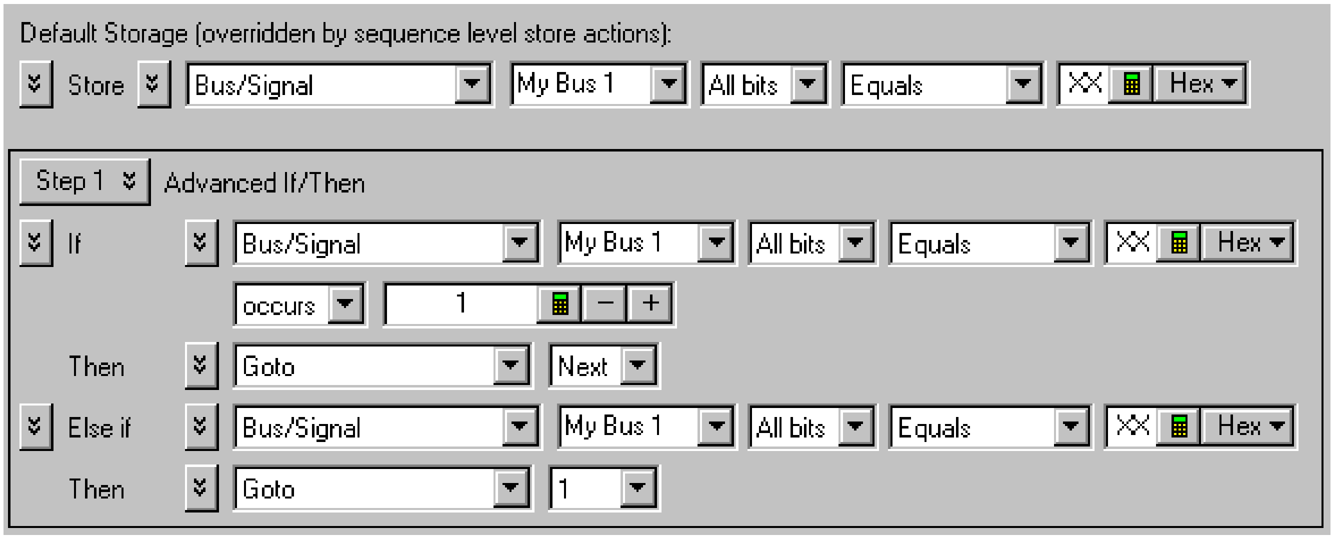
This trigger function is available when the acquisition mode is set to State - Synchronous.
The analyzer triggers when the "If" clause becomes true.



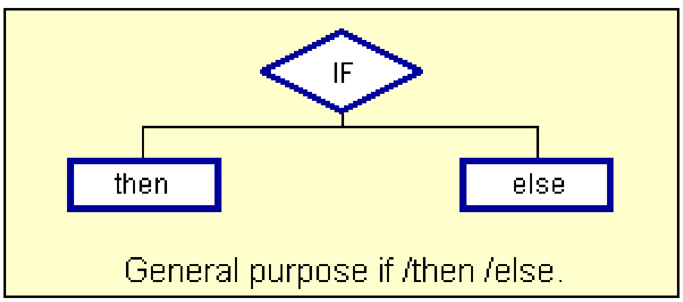
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Advanced 2-Way Branch (state)



This trigger function is available when the acquisition mode is set to State - Synchronous. The two-way branch is evaluated true when either of two patterns (if or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



To edit this function

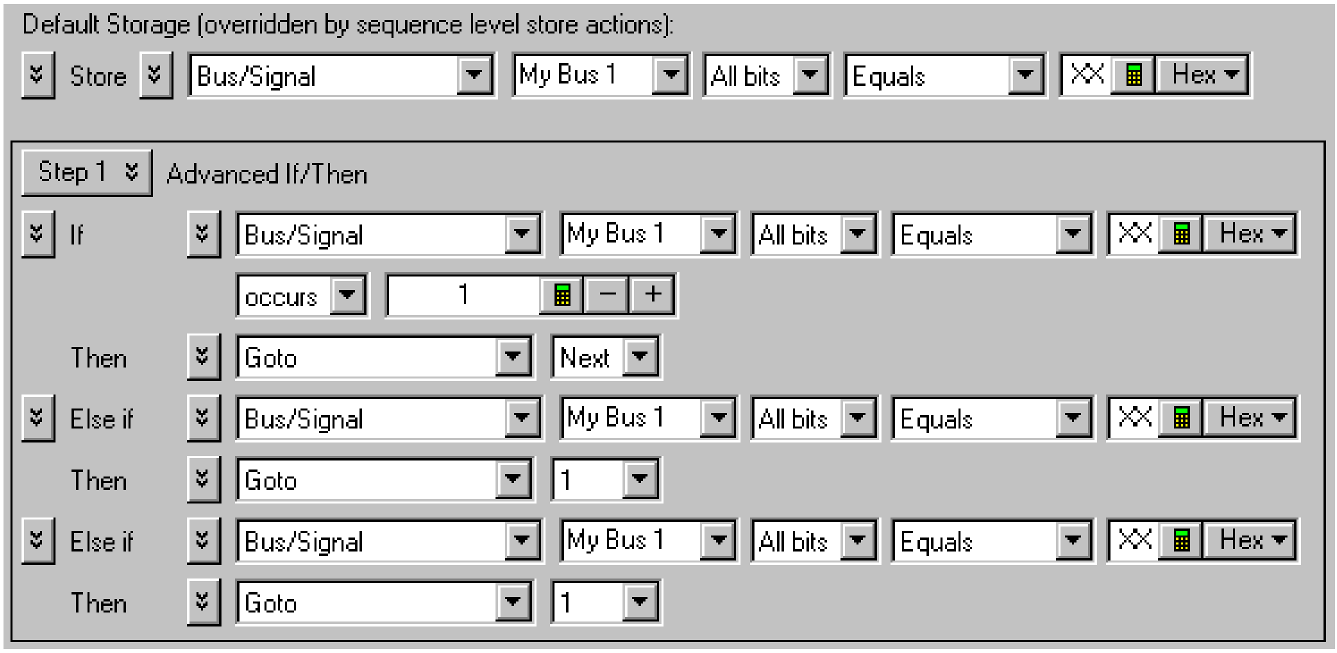
To set store qualification

To insert events and actions

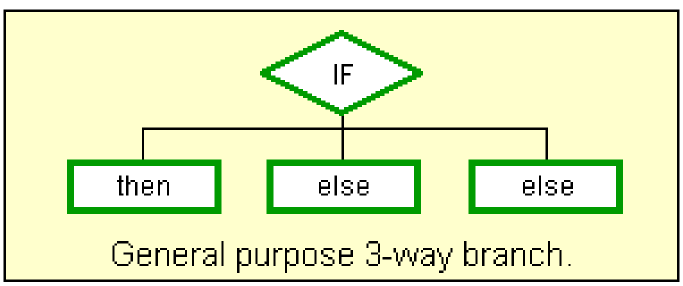
To modify trigger step display

To negate a function statement

Advanced 3-Way Branch (state)



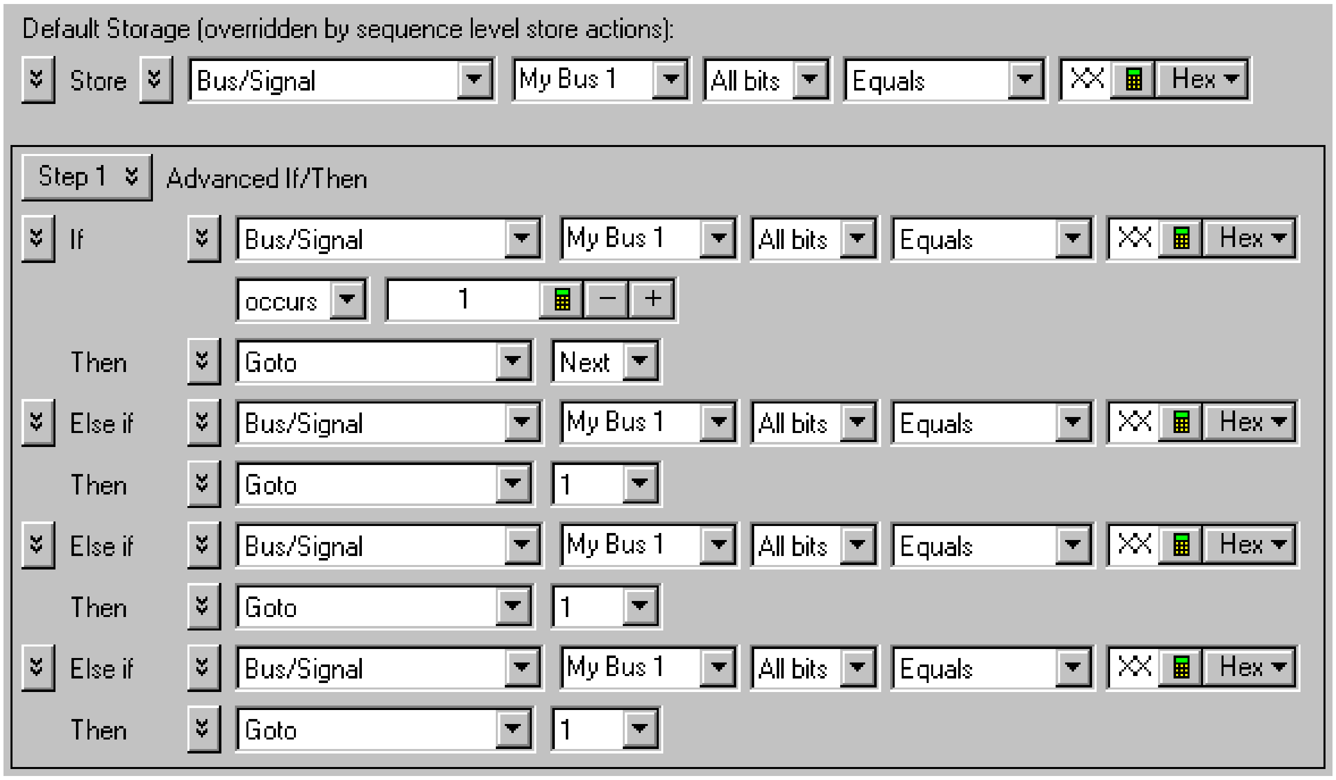
This trigger function is available when the acquisition mode is set to State - Synchronous. The three-way branch is evaluated true when either of three patterns (If or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



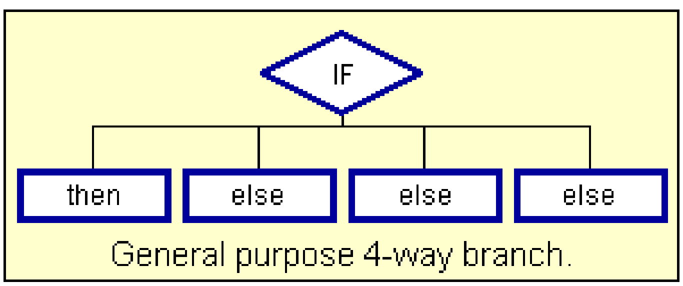
To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Advanced 4-Way Branch (state)



This trigger function is available when the acquisition mode is set to State - Synchronous. The four-way branch is evaluated true when either of four patterns (If or Else if) are found. Depending on which pattern is found true, the appropriate "Then" action is executed.



General purpose 4-way branch.

To edit this function

To set store qualification
To insert events and actions
To modify trigger step display
To negate a function statement

Specifications and Characteristics

Describes the specifications, characteristics, and requirements of the 1680A/AD-series and 1690A/AD-series logic analyzers.

- 1680A-Series Logic Analyzer Specifications
- 1680A-Series Logic Analyzer Characteristics
- 1690A-Series Logic Analyzer Specifications
- 1690A-Series Logic Analyzer Characteristics
- About the Probe Cable
- Signal Requirements

See Also

- What is a Specification
- What is a Characteristic

1680/1690-Series Logic Analyzer Characteristics

- General Information
- Probes
- State Analysis
- Timing Analysis
- Triggering
- Operating Environment Characteristics

General Information

State/timing channels:	34, 68, 102, and 136
Memory depth:	200 MHz State: A model = 256K; AD model = 1M 400 MHz Timing (full channel): A model = 512K; AD model = 2M 800 MHz Timing (half channel): A model = 1M; AD model = 4M 200 MHz Transitional Timing: A model = 256K; AD model = 1M
User interface:	Windows® 2000 Professional
Printers:	Can print to any local or network printer supported by Windows® 2000 Professional.
Dimensions:	1680 - 257 mm height (10.14 in), 443 mm width (17.45 in), 385 mm depth (15.15 in) 1690 - 153 mm height (6.05 in), 438 mm width (17.23 in), 335 mm depth (13.16 in)
Weight:	1680 - 13.2 kg (29.1 lbs) 1690 - 7.5 kg (16.5 lbs)

Probes

Input resistance:	100 kohm ±2%
Resistive tip capacitance:	1.5 pF
Minimum input voltage swing:	500 mV peak-to-peak
Maximum input voltage:	±40 V peak

State Analysis

Maximum State Speed	200 MHz
Minimum state clock pulse width:	1.2 ns
Time tag resolution:	4ns or ±0.1% (whichever is greater)
Maximum time count between states:	17 seconds
State clock/qualifiers	4 (2 on 34 channel models)

Timing Analysis

Maximum timing sample rate:	400 MHz/800 MHz

Sample period accuracy:	±0.01% of sample period ±100 ps
Channel-to-channel skew:	<1.5 ns typical
Timing interval accuracy:	±(sample period accuracy + channel-to-channel skew + 0.01% of time interval reading)

Triggering

Sequencer speed:	200 MHz
Occurrence counters:	1 per sequence level
Trigger sequence levels:	16

Operating Environment Characteristics

Temperature:	Instrument: 5°C to 50°C Disk media: 10°C to 40°C Probe lead sets and cables: 0°C to 65°C
Humidity:	Instrument: Up to 95% relative humidity at 40°C Disc media and hard drive: 8% to 85% relative humidity
Altitude:	4,572 m (15,000 ft) operating 15,300 m (50,000 ft) non-operating

1680/90A-Series Logic Analyzer Specifications

Threshold Accuracy	±(65 mV +1.5 % of setting)
Minimum Master-to-Master Clock Time	5.0 ns
Setup/Hold Time (Single Clock, Single Edge)	2.5 ns window adjustable from 4.5/-2.0 ns to -2.0/4.5 ns in 100 ps increments per channel
Setup/Hold Time (Multiple Clock, Multiple Edge)	3.0 ns window adjustable from 5.0/-2.0 ns to -1.5/4.5 ns in 100 ps increments per channel

About the Probe Cable

The probe cable contains 16 signal lines, 1 clock line, 20 ground lines, 1 serial line for communication with analysis probes, and 2 lines that carry minimal power for demo "device under test" systems.

The cable ground lines are chassis (earth) grounds and not "floating" grounds. All the lines are woven into a flat ribbon that is 4.5 feet long. See Probing in the Quick Start/Installation Guide for the cable pinout and equivalent load.

Caution: Do not exceed 0.33 amps per cable, or the cable will be damaged.

Signal Requirements

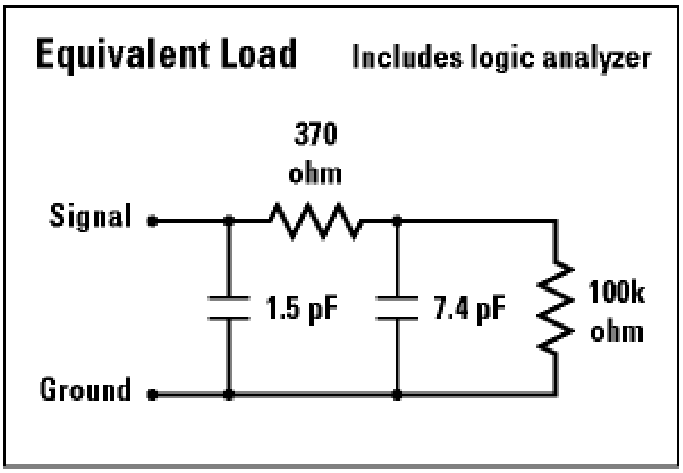
- Minimum Signal Amplitude
- Signal Loading
- Maximum Probe Input Voltage
- Overdrive

Minimum Signal Amplitude

Any signal line you intend to probe with the logic analyzer probes must supply a minimum voltage swing of 500 mV to the probe tip. If you measure signal lines with a smaller voltage swing, you may not get reliable results. The minimum input overdrive is the greater of 250 mV or 30% of signal amplitude.

Signal Loading

Any probed signal line must be able to handle the following load:



If the signal cannot handle this load, the target system may malfunction.

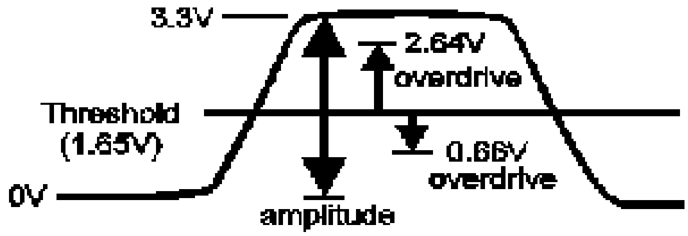
Maximum Probe Input Voltage

The maximum probe input voltage of each logic analyzer probe is 40 volts peak.

Overdrive

Overdrive is the amount a signal must exceed the threshold voltage for the logic analyzer to detect a change in logic level. For the 1680-series logic analyzer, overdrive is 250 mV or 30% of signal peak-to-peak amplitude, whichever is greater.

For example, given a 3.3 volt CMOS signal (low = 0V, high = 3.3 V) the optimal threshold is 1.65 V (50%). If the threshold is set less than 1.0 V or greater than 2.3 V, then a timing acquisition might show excessive channel-to-channel skew. For a state acquisition, the analyzer's setup and hold requirements might not be met.



The overdrive amount is specified as the greater of 250 mV or 30% of the signal amplitude because it has two purposes. The 250 mV ensures reliable switching or state detection. The 30% of amplitude ensures the threshold is reasonably centered within the waveform in order to minimize channel-to-channel skew (t_{PHL} vs t_{PLH}).

What is a Specification?

A specification is a numeric value, or range of values, that bounds the performance of a product parameter. The product warranty covers the performance of parameters described by specifications. Products shipped from the factory meet all specifications. Additionally, products sent to Agilent Customer Service Centers for calibration, and returned, meet all specifications. Specifications are verified by calibration procedures.

What is a Calibration Procedure?

Calibration procedures verify that products or systems operate within the specifications. Parameters covered by specifications have a corresponding calibration procedure. Calibration procedures include both performance tests and system verification procedure. Calibration procedures are traceable and must specify adequate calibration standards.

Calibration procedures verify products meet the specifications by comparing measured parameters against a pass-fail limit. The pass-fail limit is the specification less any required guardband.

The term "calibration" refers to the process of measuring parameters and referencing the measurement to a calibration standard rather than the process of adjusting products for optimal performance.

Note: Self-tests are not a substitute for calibration.

See Also

What is a Characteristic

What is a Characteristic?

Characteristics describe product performance that is useful in the application of the product, but that is not covered by the product warranty. Characteristics describe performance that is typical of the majority of a given product, but not subject to the same rigor associated with specifications. Characteristics are verified by *function tests*.

What is a Function Test?

Function tests are quick tests designed to verify basic operation of a product. Function tests include operator's checks and operation verification procedures. An operator's check is normally a fast test used to verify basic operation of a product. An operation verification procedure verifies some, but not all, specifications, and often at a lower confidence level than a calibration procedure.

See Also

What is a Specification

Data Formats

ALA Format

The default file format for saving configuration files is ALA. The Agilent Logic Analyzer (ALA) file format is used only in the Agilent 1680 and 1690-series logic analyzers. The ALA format is proprietary; the configuration file is not intended to be read by programs other than the 1680 or 1690 software. The ALA configuration file contains the information necessary to reconstruct the display appearance, instrument settings, and trace data (optional) that were present when the configuration file was created. The configuration file is saved through the file menu. The configuration file is also opened though the file menu.

The following is a general overview of what gets saved in the ALA configuration file.

General

Trigger specifications

Analyzer characteristics

Bus/Signal folders

Bus/Signal information (per bus/signal)

Marker information

Find value

Search event parameters

Symbols

Tools (e.g. inverse assemblers)

Filters

Listing display attributes

Waveform display attributes

General

- configuration File Owner, Project and Description (information as specified in the **File>Save** dialog edit boxes)
- date and time the configuration file was saved
- 16xx software version that saved the configuration file
- size and screen position of the active viewer in the main window
- main window toolbar and menu items
- tools (i.e., measurement, filter and viewer subsystems) that are active
- connection graph of plug-in connectivity

Trigger specifications

- history
- occurrence counter
- storage qualifiers
- timer values
- functions and function libraries and current function state
- external arming events

- named trigger events and event lists
- setup comments

Analyzer characteristics

- analyzer name
- number of pods
- acquisition depth
- state
  - setup and hold specifications
  - clock setup model
  - clock settings (qualifiers and mode)
  - sampling mode
- timing
  - sample period
  - sampling mode
- trigger position
- per-channel threshold settings
- threshold voltages
- setup dialog
  - column widths, column order, column on/off
  - horizontal scroll position
  - vertical scroll position
  - zoom factor
  - selected item
- poststore specification

Bus/Signal folders

- folder name
- expanded or collapsed
- contents

Bus/Signal information (per bus/signal)

- name
- locks
- channel assignments
- number base
- polarity

Marker information

- name
- position
- measurements
- comments

- locked or not
- sample offset
- color
- hidden or not

Find value

- marker name
- from marker ID

Search event parameters

- begin time
- end time
- event label
- event list
- occurrence count

Symbols

- symbol name
- symbol operator
- symbol context
- patterns or range values
- symbol base

Tools (e.g., inverse assemblers)

- name
- menus
- test mode
- internal labels

Note: If the configuration file is loaded into a logic analyzer that does not have a licensed copy of the tool, the tool information is not displayed.

Filters

- filter name
- enabled or disabled
- filter events
- show filter name or not

Listing display attributes

- data area background color
- signal color
- tracker color
- show markers

- show activity indicators
- font size (bus/signal and data)
- column width (minimum and user set widths)

Waveform display attributes

- data begin position
- data end position
- row height
- time per division
- time delay
- color
    - overlay and axis
    - show values
    - waveform magnitude axis
    - soft glitch
- show values
- waveform style (bus or magnitude)
- show soft glitch
- bus order
- expanded
- snap marker to edge
- move marker on screen
- place marker on edge
- show value tool tip
- show transition width tool tip
- show waveform magnitude axis

CSV Format

You can export captured data to CSV (Comma-Separated Values) format files which can then be imported by other applications like Excel. Only captured (digital) data is exported. Data generated by tools (like, for example, integral and string data generated by inverse assemblers) is not included in the output.

Output is standard CSV format where the first row is column headings for time, sample, trigger and then channel data and each successive row contains data for those columns separated by commas. An example exporting five samples of timing mode acquisition is shown below.

```
Time,Sample,Trigger,allData,Pod2:Ch1,Pod2:Ch0,Pod1:Ch15,Pod1:Ch14,Pod1:Ch13,Pod1
:Ch12,Pod1:Ch11,Pod1:Ch10,Pod1:Ch9,Pod1:Ch8,Pod1:Ch7,Pod1:Ch6,Pod1:Ch5,Pod1:C
h4,Pod1:Ch3,Pod1:Ch2,Pod1:Ch1,Pod1:Ch0,
0,0,0,1 0000,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0.000000050000000000000000000,1,0,2 0001,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
0.000000050000000000000000000,1,0,2 0001,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
0.000000010000000000000000000,2,0,3 0002,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
0.000000015000000000000000000,3,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1
0.000000015000000000000000000,3,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1
0.000000020000000000000000000,4,0,1 0004,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
```

XML Format

When you save logic analyzer setups and data to generic configuration files, the information is stored in standard XML format (<Tag> body </Tag>). You can edit and create XML files using any text editor.

Each file contains:

- A File element indicating the *Agilent Logic Analyzer* version that created the file, the date the file was created, and whether the file has setup information only. The File element begins with the <File> start tag and ends with the </File> end tag.
- A Configuration child element that begins with the <Configuration> start tag and ends with the </Configuration> end tag.

In the following examples, explanatory comments are indicated like this.

Example:

```
 <File Owner="" Project="" Desc="" Version="01.20.0000" Date="Feb 20, 2002
10:58:45" SetupOnly="No">

 <Configuration>
      child elements
 </Configuration>

 </File>
```

In XML, elements can have attributes, which are*name="value"* pairs within the element's start tag. In the previous example, the Version attribute has the value "01.20.0000" and the Date attribute has the value "Feb 20, 2002 10:58:45". Values must be contained in quotes.

XML elements can have child elements, and those elements can have children, and so on. In logic analyzer configuration files, the <Configuration> element can have the child elements <Setup> and <Data>.

<Setup> Element

The <Setup> element contains all the configuration information for the logic analyzer and any tools, such as inverse assemblers. Logic analyzer information is contained within the <Module> and </Module> tags. Tool elements are indicated with the <Tool> and </Tool> start and end tags. Each tool must have a unique Name attribute. The <Setup> element is terminated by the </Setup> end tag.

<Module> Element

The <Module> element is further divided into <Sampling> and one or more <BusSignal> elements.

Example:

```
      comment and <Configuration> up here
 <Setup>

 <Module Name="200MHz State/800MHz Timing Analyzer 1">

 <Sampling Acquisition="Timing" Period="0.00000000125000000000000">
 </Sampling>

 <BusSignal Name="\\Address">
 <Symbol Name="start" Context="None" Operator="Equals" LowValue="FF00"
HighValue="" Base="Hex"></Symbol>
```

```
 <Symbol Name="display_io" Context="None" Operator="In Range" LowValue="8000"
HighValue="801F" Base="Hex"></Symbol>
 <Channels>0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15</Channels>
 </BusSignal>

       more BusSignal elements

 </Module>
```

<Sampling> Element

The <Sampling> element denotes how digital data was acquired.

<BusSignal> Element

<BusSignal> elements have zero or more <Symbol> child elements and one <Channels> child element.

<Symbol> Element

The <Symbol> element has the following attributes:

- Name.
- Context (always "None").
- Operator (can be "Equals", "Not Equal To", "Less Than", "Greater Than", "Less Than Or Equal To", "Greater Than Or Equal To", "In Range", or "Not In Range").
- LowValue.
- HighValue (used in range values).
- Base (can be "Bin" for binary, "Hex" for hexadecimal, "Oct" for octal, "Dec" for decimal, or "Sign" for signed decimal).

<Channels> Element

The <Channels> element lists the channels assigned to the particular bus or signal.

<Tool> Element

Tool elements have a Name attribute, and one or more <Label> child elements. A label will always have a name and type. Valid types are StringLabel and Integral Label. Filters are not saved.

Example:

```
    <Setup> and <Module></Module> up here

 <Tool Name="Arm Inverse Assembler 1">

 <Label Name="Arm Cycle information" Type="StringLabel">
 </Label>

 <Label Name="Arm Address" Type="IntegralLabel" Width="32">
 </Label>

     more Label elements

 </Tool>

 </Setup>
```

<Data> Element

The <Data> element contains all the data for both logic analyzer and any saved tools. It also has <Module> and <Tool> child elements.

<Module> Element

The <Module> element has Name, NumberOfSamples, and TriggerIndex attributes. The TriggerIndex attribute indicates which sample was the trigger point; the first sample is counted as 0.

<TimeData> Element

The <Module> element may contain a <TimeData> child element. The <TimeData> element exists only for non-periodic sampling (that is, transitional timing mode or state mode). Any other method of sampling is periodic, and as such the <Sampling> element will simply contain a Period attribute and no <TimeData> element will be generated.

<DigitalData> Element

Logic analyzer data consists of a single <DigitalData> element containing only digital data samples. There is one sample per line. Each sample is contained between a pair of curly braces and is formatted to appear (left to right) in the same order that the corresponding <BusSignal> elements appear in <Setup>, delimited by commas. For example, { Bus1, Bus2, Signal1 } where Bus1 was the first <BusSignal> element in <Setup>, Bus2 was second, and Signal1 was third.

Example:

```
<Module Name="200MHz State/800MHz Timing Analyzer 1" NumberOfSamples="60"
TriggerIndex="30">
      Notice there is no <TimeData> element because the running example is periodic
<DigitalData>
{7FE2,0,0},
{7FE3,0,0},
{7FE4,0,0},
     54 more samples here
{1B,0,0},
{1C,0,0},
{1D,0,0}
</DigitalData>

</Module>
```

<Tool> Element

The <Tool> element's only attribute is Name. Each tool must have a unique name. Tool data consists of two child elements, <IntegralData> and <StringData>.

<IntegralData> and <StringData> Elements

The <IntegralData> element contains only generated data for labels in the <Label> element of <Setup> that are Integral type.

The <StringData> element contains only generated data for labels of String type. As in <DigitalData>, there is one sample per line and each sample is contained within curly braces.

The one difference between the format of digital data and generated data is that generated data must account for subrows. This is accomplished through the use of square brackets that denote all subrows for a label. For example, {[label1_0,label1_1],[label2_0],[label3_0,label3_1,label3_2]} where label1 has one subrow, label2 has no subrows and label3 has three subrows. The first value is always the state value and the other remaining values are subrows.

Example:

```
<Tool Name="Arm Inverse Assembler 1">

<IntegralData>
{[0000 7FE2],[0000],[F001 0023]},
{[0000 7FE2],[0000],[F001 0023]},
{[0000 7FE4],[0000],[F001 0023]},
        54 more Integral Data samples
{[0000 001A],[0000],[F001 0023]},
{[0000 001C],[0000],[F001 0023]},
{[0000 001C],[0000],[F001 0023]}
</IntegralData>

<StringData>
{[* LSL     R0,R0,#0x0          T-[31:16]]},
{[* LSL     R0,R0,#0x0          T-[31:16]]},
{[* LSL     R0,R0,#0x0          T-[15:00]]},
        54 more StringData samples
{[* LSL     R0,R0,#0x0          T-[31:16]]},
{[* LSL     R0,R0,#0x0          T-[15:00]]},
{[* LSL     R0,R0,#0x0          T-[15:00]]}
</StringData>

</Tool>

</Data>
```

Fast Binary Data Format

The Fast Binary data format lets you export analyzer data for post-processing. The Fast Binary data format saves faster than ASCII, and can be parsed using programming tools.

Data Organization

The data is organized in a hierarchy of several individual objects representing different aspects of the acquired data. The analyzer data is basically organized by your defined 'labels' and channels called a LabelEntry. Multiple labels/channels from each analyzer are then grouped into a DataSet. All the labels in a DataSet have the same number of samples and have the same X-axis(Abscissa) values. Then one or more DataSets are grouped into a DataGroup. These are discussed in more detail below. The FileOut tool generates or writes out one DataGroup object.

Object IDs

There are several object types that have an object ID associated with them. When an object is referenced more than once within the DataGroup, the first occurrence of the object is written out, and subsequent objects have only their ID written out.

The Fast Binary File Format is a mixture of these binary and ascii objects:

DataGroup

DataSet

Label Entry

Label Data

IntegralData

Vertical Header

Abscissa Data Type (x-axis information)

Time Correlation Info

State Correlation Info

DataGroup

The top level object is the DataGroup. This object contains a general header information and a list of DataSet objects:

<DataGroup> ::=

<DataGroup Header>

Number of DataSets => "%d\n"

<DataSet #1>

<DataSet #2>

...

DataGroup Header

The DataGroup header is an object that identifies the file as a fast binary data file. It contains the format version and the information that describes how the DataSets are related to each other:

<DataGroup Header> ::=

```
FileId => "HPLogic_Fast_Binary_Format_Data_File\n"
FileVersion => "%d %d\n"
Correlation Bits  => "%d [%d]*\n"
Cross Correlation IDs
    => "%d %d\n"
```

- FileVersion - The file version fields represent the major and minor version numbers. These are used to identify different versions of the fast data format. Where the format has changed will be noted with the corresponding version numbers.
- Correlation Bits - The cross correlation IDs are used when correlating multiple DataGroups. DataGroups that are time correlatable will have the same Time Correlation Id. In the same way, DataGroups that are state correlatable will have the same State Correlation Id. An ID of -1 indicates cross correlation information is not available.

Bit#	Description
0	Time Correlatable
1	State Correlatable

Time correlatable indicates that the data within all the DataSets are time aligned. State correlatable indicates that each sample is state aligned across all DataSets, i.e. the DataSets are sample synchronized. A DataGroup with only one DataSet that has timing information will be both time and state correlatable.

- Cross Correlation ID's - The cross correlation IDs are used when correlating multiple DataGroups. DataGroups that are time correlatable will have the same Time Correlation Id. In the same way, DataGroups that are state correlatable will have the same State Correlation Id. An ID of -1 indicates cross correlation information is not available.

DataSet

The DataSet is a collection of label or channels. All of the labels in a dataset have a common X-axis(Abscissa) information.

```
<DataSet> ::=
   Number of Label Entries  => "%d\n"
   <Label Entry #1>
     . . .
   <Label Entry #n>
   <Abscissa Data Type>
   <Time Correlation Info>
   <State Correlation Info>
   Origin Path  => "`%s'\n"
   DataSet ID, Run ID
             => "%d %d\n"
   Begin and End Time
             => "%d %d\n"
   Start Sample
             => "%d\n"
   Last Sample
             => "%d\n"
```

- Origin Path - This is a string that describes the "path" that the data was processed through, i.e. what tools were used to process this data.
- DataSet ID, Run ID - These IDs are used to cross correlate and integrate multiple data sets. The DataSet ID identifies this data set. The Run ID is an ID used to correlate with other data sets. The way it is used is that if two or more data sets have the same Run ID, then they have the possibility of being correlated in some way. To determine if the data sets are truly correlated, you must examine their Time Correlation Info and State Correlation Info.
- Begin and End Time - These values show the time at which the data was acquired, both the time when the analysis began, and the time at which the trace ends. The time value is the number of seconds since midnight, January 1st, 1970.
- Start Sample - This value is the first sample number in the acquisition.
- Last Sample - This value is the last sample number in the acquisition.

Label Entry

The Label Entry structure is the information associated with one data label. This structure contains the label name, label data, label attributes, and symbol information.

<Label Entry> ::=
Label ID => "%d\n"
(if Label ID has not already been processed)
Label Name => "`%s'\n"
<Label Data>
new line => "\n"
Label Attribute Bits => "%d [%d]*\n"

- Label ID - There is a unique identifier for each label. When the same label is used in multiple DataSets, the label data is written only once. If this label's data has already been written to the file then the label name, label data, and label attributes are not included in this record.
- Label Attribute Bitset - These attributes are used by various tools within the instrument to identify certain labels. Most of these are for inverse assembler. The only one that might be of interest is StateCount which is set when doing state with state-tags type of acquisitions.

Bit#	Description
0	TypeAddress
1	TypeAddressB
2	TypeAddressC
3	TypeAddressD
4	TypeData
5	TypeDataB
6	TypeDataC
7	TypeDataD
8	TypeStatus
9	TypeStatusB
10	TypeStatusC

11	TypeStatusD
12	TypeClock
13	TypeClockPos
14	TypeClockNeg
15	TypeClockBoth
16	TypeSampleEdge
17	TypeSampleEdgePos
18	TypeSampleEdgeNeg
19	TypeSampleEdgeBoth
20	TypeControl
21	StateCount
22	Sequencer

Label Data

The label data can be one of several objects. The object is determined by the first string of the label data object. Following the 'type' string is the raw binary data of the label. Then comes a bitset that describes the attributes of this label. The currently defined label data objects are:

"NoData"
"States"
"StateCount"
"Glitch"
"Analog"
"TextLines"

- NoData - This is the label data object that is empty!

```
<NoData> ::=
  Label Data Type   => "NoData\n"
  Label Attribute bitset
                => "%d [%d]* \n"
```

- States - This is the label data object that is most common. This is used to represent sampled data of n-bit width. The <IntegralData> will contain the actual sample data and is accessed with a zero based index.

```
<States> ::=
  Label Data Type   => "States\n"
  <IntegralData>
  new line          => "\n"
  Label Attributes bitset
                => "%d [%d]* \n"
```

- StateCount - This label data object is used to store state counts when acquiring data with state tagging turned on.

```
<StateCount> ::=
  Label Data Type   => "StateCount\n"
```

<IntegralData>
new line => "\n"
Label Attributes bitset
=> "%d [%d]* \n"

- Glitch - This object is used for labels that have data that was acquired with glitch mode turned on. The sample <IntegralData> contains the sample data just like the State object. The glitch <IntegralData> contains the glitch data. There is a one-to-one correspondence between the sample and glitch data. When the glitch data is true, a 1 in any bit position, indicates that a glitch occurred on the corresponding sample.

<Glitch> ::=
Label Data Type => "Glitch\n"
Sample <IntegralData>
new line => "\n"
Glitch <IntegralData>
new line => "\n"
Label Attributes bitset
=> "%d [%d]* \n"

- Analog - The object is for analog type data. The integral array contains the digital quantization-levels of the sampled signal. The vertical header contains the information to convert the digital data to voltage levels.

<Analog> ::=
Label Data Type => "Analog\n"
<IntegralData>
new line => "\n"
<VerticalHeader>
Analog units => "%s\n"
Label Attributes bitset
=> "%d [%d]* \n"

- TextLines - The object is for text type data. The integral array contains text strings of data.

<Text> ::=
Label Data Type => "TextLines\n"
Lines <Strings>
new line => "\n"
Label Attributes bitset
=> "%d [%d]* \n"

Label Data Attributes Bitset

The label data attributes is a bitset that represents the different attributes this data has. At this time there are 22 label data attributes:

Bit#	Description
1	Label has width
2	Label has length(samples)

3	Label has Glitch Data
4	Label has a Vertical Header
5	Label Data has trigger
6	Range
7	Label Data has Statistical data
8	Label Data has Integral data
9	Label Data is in floating point
10	Label Data is signed
11	Label Data is unsigned
12	Label Data is Periodic
13	Label has symbols
14	ValueItor
15	GlitchItor
16	SampledData
17	VersusTime
18	VersusFreq
19	Analog
20	Digital
21	StateCounts
22	TextLines

IntegralData

The IntegralData structure is used to represent signed and unsigned n-bit data. n is the bit width of the label, or the width of this object when shared among several labels. Like label data, there are several objects of IntegralData which is determined by the first line of this object. The objects defined so far are:

"IntegralArray<signed8>"
"IntegralArray<signed16>"
"IntegralArray<signed32>"
"IntegralArray<signed64>"
"IntegralArray<unsigned8>"
"IntegralArray<unsigned16>"
"IntegralArray<unsigned32>"
"IntegralArray<unsigned64>"
"BitPackedData"
"BitBlockData"
"BitBlock"
"PagedBitBlock"
"PagedIntegralData<type size>"
"StringData"

The above list can be broken down into four basic formats, IntegralArray, BitPackedData, BitBlockData, and Strings. Strings are used for textual data. The IntegralArray is used for labels that are exactly 8, 16, 32, or 64 bits wide. The BitPackedData is used for all other label widths. The BitBlockData is similar to BitPackedData in that it contains the raw data for multiple labels. The difference is that the bits associated with the label may not be consecutive or in order.

Paged objects are the same as their non-paged counter parts except that they may contain a filename instead of data. This indicates that the data is located in the named file instead of the current file being read. The additional files will be formatted with a header describing that the file is a data file, and then the actual data will follow with exactly the same format as if it were in the original file. This is being done because a fair number of computer systems have a 2GB file size limit. There are situations where the original Fast Binary Data File would be greater than 2GB, hence these paged objects.

IntegralArray<type size>

```
IntegralArray<type size> ::=
    Integral Data Type       => "IntegralArray<type size>\n"
    Integral Data ID         => "%d\n"
    (if ID has not already been processed)
    Length                   => "%d\n"
    StuckOne, StuckZero      => "%s %s\n"
    Raw Bytes                => n bytes
```

- Type and Size - Type is the sign-ness of the label data values, i.e. "signed" or "unsigned". Size is the width in bits of the label.
- Length - Length is the number of samples that is associated with this label.
- StuckOne and StuckZero - StuckOne indicates which bits, if any, of the label have a value of 1 for all of the samples. This value has the same width as the label does, which can be larger than 32 bits, so therefore is formatted as a string(%s). StuckZero indicates bits with a value of 0.
- Raw Bytes - A raw section of bytes that is n bytes long. n = length * size in bytes. The block of data can be viewed as an array of size length with each element be size bits wide. Then access the data using a zero based array index.

BitPackedData

```
BitPackedData ::=
    Integral Data Type       => "BitPackedData\n"
    Integral Data ID         => "%d\n"
    (if ID has not already been processed)
    Start Bit, Width, Inverted
                             => "%d %d %d\n"
    StuckOne, StuckZero      => "%s %s\n"
    BytesPerLine, IntegralPerLine
                             => "%d %d\n"
    DataBlock                => &<IntegralArray&<type size>>
```

- Start Bit, Width, Inverted - Start Bit is starting bit position within the bitblock for this label. Width is the number of bits used by this label. Inverted is a flag indicated whether the data is inverted.
- StuckOne and StuckZero - StuckOne indicates which bits, if any, of the label have a value of 1 for all of the samples. This value has the same width as the label does, which can be larger than 32 bits, so therefore is formatted as a string(%s). StuckZero indicates bits with a value of 0.
- BytesPerLine and IntegralPerLine - These two fields indicate the width of the bitblock

data in bytes and in terms of the base type of the IntegralArray.

- DataBlock - DataBlock is an IntegralArray that can be shared by multiple labels. Each label that occupies this DataBlock has a starting position and it's width, so that each line/sample within the datablock is wide enough to support all labels sharing this DataBlock.

BitBlockData

BitBlockData ::=

Integral Data Type => "BitBlockData\n"

Integral Data ID => "%d\n"

(if ID has not already been processed)

Extractor => <Label Extractor>

StuckOne, StuckZero => "%s %s\n"

DataBlock => <BitBlock> or <PagedBitBlock>

- Extractor - The Label extractor is used to map the bits of a label definition to specific bits within a bitblock.
- StuckOne and StuckZero - StuckOne indicates which bits, if any, of the label have a value of 1 for all of the samples. This value has the same width as the label does, which can be larger than 32 bits, so therefore is formatted as a string(%s). StuckZero indicates bits with a value of 0.
- DataBlock - DataBlock is either a <BitBlock> or a <PagedBitBlock> that is shared by multiple labels. For each label that uses this bitblock, a label extractor is used to access the bitblock to retrieve the appropriate data associated with that particular label.

BitBlock

BitBlock ::=

Integral Data Type => "BitBlock\n"

reserved => "%d\n%d\n"

Integral Data ID => "%d\n"

(if ID has not already been processed)

Length, BytesPerLine => "%u %u\n"

Raw Bytes => n bytes

- Integral Data ID - This is the Objects ID. An ID value of 0 indicated that there is no data associated with this object. If the ID has not been seen earlier in the file, the object information follows. If the ID has been seen, then the next object follows.
- Length and BytesPerLine - Length is the number of samples (rows) of data. BytesPerLine is the number of bytes is each sample or row.
- Raw Bytes - A raw section of bytes that is n bytes long. n = length * bytesPerLine. The block of data can be viewed as an array of samples, with each sample being bytesPerLine wide. This array is then accessed using a zero based array index.

PagedBitBlock

PagedBitBlock ::=

Integral Data Type => "PagedBitBlock\n"

reserved => "%d\n%d\n"

Integral Data ID => "%d\n"

(if ID has not already been processed)
UseFile, Filename => "%d `%s`\n"
reserved => "%d %d\n"
Length, BytesPerLine => "%u %u\n"
Raw Bytes => n bytes

- Integral Data ID - This is the Objects ID. An ID value of 0 indicated that there is no data associated with this object. If the ID has not been seen earlier in the file, the object information follows. If the ID has been seen, then the next object follows.
- UseFile and Filename - UseFile is a boolean that indicates whether to use a file or not. The filename is a full path name. A value of 0 means that the data continues on the next line. A value of 1 indicates that the data is located in the associated file. When reading the additional file, filename, the first section is a file comment. The second section starts with the string "HPLogic_Additional_Data_File n", then the rest of this Integral Data Object follows, i.e. Length and BytesPerLine and Raw Bytes.
- Length and BytesPerLine - Length is the number of samples (rows) of data. BytesPerLine is the number of bytes for each sample or row.
- Raw Bytes - A raw section of bytes that is n bytes long. n = length * bytesPerLine. The block of data can be viewed as an array of samples, with each sample being bytesPerLine wide. This array is then accessed using a zero based array index.

PagedIntegralData<type size>

PagedIntegralData<type size> ::=
Integral Data Type => "PagedIntegralData<type size>\n"
Integral Data ID => "%d\n"
(if ID has not already been processed)
UseFile, Filename => "%d `%s`\n"
reserved => "%d %d\n"
Length, BytesPerLine => "%u %u\n"
Raw Bytes => n bytes

- Type and Size - Type is the sign-ness of the label data values, i.e. "signed" or "unsigned". Size is the width in bits of the label. Valid bit widths are 8, 16, 32, 64.
- Integral Data ID - This is the Objects ID. An ID value of 0 indicated that there is no data associated with this object. If the ID has not been seen earlier in the file, the object information follows. If the ID has been seen, then the next object follows.
- UseFile and Filename - UseFile is a boolean that indicates whether to use a file or not. The filename is a full path name. A value of 0 means that the data continues on the next line. A value of 1 indicates that the data is located in the associated file. When reading the additional file, filename, the first section is a file comment. The second section starts with the string "HPLogic_Additional_Data_File n", then the rest of this Inegral Data Object follows, i.e. Length and BytesPerLine and Raw Bytes.
- Length and BytesPerLine - Length is the number of samples (rows) of data. BytesPerLine is the number of bytes for each sample or row. BytesPerLine will be either 1, 2, 4, 8.
- Raw Bytes - A raw section of bytes that is n bytes long. n = length * bytesPerLine. The block of data can be viewed as an array of samples, with each sample being bytesPerLine wide. This array is then accessed using a zero based array index.

Strings

Strings ::=

Integral Data Type => "StringData\n"

Integral Data ID => "%d\n"

(if ID has not already been processed)

Length => "%d\n"

String Block => "%d %s [%d %s]*"

- Length - Length is the number of strings contained in the data.
- String Block - The block of data can be viewed as two sections. First, a value is read which represents the number of characters in the current string, say x. The following x bytes are the actual string itself. Then immediately following is a character count of the next string, and so on.

Label Extractor

<Label Extractor> ::=

Bytes, Width, Inverted => "%u %u %d\n"

Mask => "%x [%x]* \n"

reserved => "%d\n%d\n"

HaveReorder => "%d\n"

Reorder => <Reorder>

- Bytes, Width, and Inverted - Bytes is the size of the mask. Width is the number of bits that are set in the mask, also the number of bits of the associated label. Inverted indicates if the label has negative polarity, i.e. does the data need to be complemented after extraction.
- Mask - The mask is an array of bytes that specifies which bits of the bitblock are needed for this label. The bytes of this mask corresponds to the bytes of each sample. A bit set to a 1 in the mask indicates that the corresponding bit in the sample should be used. The data is stored in big Endian format, hence byte 0 is the most significant and continues till byte n. For example, bytes=2, width=7:

```
          Byte 0    Byte 1
         111111
       bit 54321098  76543210
 Mask:     10010011  00101010
 Sample:   11100000  01111000
           --------  --------
 Value:    1  0  00    1 1 0      =>  1000110  => 0x46
```

If inverted is true or 1, then the value would become 0111001(0x39)

- HaveReorder - HaveReorder is a flag that indicates whether the bits need to be reordered. Reordering occurs after the extraction takes place. If the label bits are configured with bit reordering this field will have a non-zero value, and the reorder object will follow. If haveReorder is zero, nothing will follow.

Reorder

Endian Flags => "Endian16: %d Endian32: %d Endian64: %d "

and Width => "Endian128: %d Width: %d\n"

Bit Order Map => ["%d %d\n" or " %s %s\n"](width times)

The reorder object describes the desired label bit ordering. The appropriate Endian flags are set true when little endian is specified. Width specifies the number of bits that are defined in this label, i.e. it should be the same as the width specified in the parent Extract object. When a custom bit ordering is specified, all the Endian flags will be false/0. If all Endian flags are zero then there will be a bit order map following the flags. The bit order map consists of two arrays that are width long, i.e. there is an array element for each bit that is set in the mask. The first array is the map from display bits to channel/bitblock bits, the second array is the map from channel bits to display bits.The arrays are of type int if the width is less than or equal to 32, otherwise the array is setup as strings to be processed with any method that you have for handling greater than 32 bit integers. The arrays are maps that map the bit(index of array) you are working with to the bit position(value of array) that it should be moved to. For example, a 4 bit label:

Channel bit	Display bit
0	2
1	0
2	1
3	3

(index)	Array1	Array2
0	0010	0100
1	0100	0001
2	0001	0010
3	1000	1000

Vertical Header

The vertical header is used to describe information about a signal that goes beyond just digital data. At this time, the only specific application is to describe the signal from an scope. As with other polymorphic objects, this object supports several types of objects which is specified by the first string/line. The possible types at this time are:

- "DefaultOrdinateHeader"
- "ScopeHeader"
- DefaultOrdinateHeader

  <DefaultOrdinateHeader> ::=

  Header Type => "DefaultOrdinateHeader\n"

- ScopeHeader

  <ScopeHeader> ::=

  Header Type => "ScopeHeader\n"

  YIncrement, YOrigin, YReference, NumBits

  => "%f %f %d %d\n"

  YOrigin - is the voltage value at center screen.

  YIncrement - the voltage difference between consecutive data values.

  YReference - the value that specifies the data value at center screen, where YOrigin occurs.

  NumBits - the number of bits for the width of the scope data.

Abscissa Data Type (x-axis information)

The Abscissa Data object contains the time and/or state information that describes how the data "flows" in this data set. The state portion of this object describes the number of samples and where the trigger is located within the samples. If there is time associated with this data set, then the time portion describes any timing information. This is a polymorphic object and as such, the first line determines which object is actually used. The current possible objects are:

- AbscissaData - The Abscissa Data object is a base object that describes the number of samples and the trigger position for all of the labels (LabelEntry) contained in this data set. The abscissa attributes give the characteristics of the abscissa data.

```
<AbscissaDataType> ::=
    Abscissa Data Type      => "AbscissaData\n"
    Number of Samples, Trigger Position
                            => "%d %d\n"
    Abscissa Attribute bitset
                            => "%d [%d]* \n"
```

- Periodic - The Periodic object is used when the timing information is based on periodic sampling.

```
<AbscissaDataType> ::=
    Abscissa Data Type      => "Periodic\n"
    Number of Samples, Trigger Position
                            => "%d %d\n"
    Origin, Increment       => "%s %s\n"
    Abscissa Attribute bitset
                            => "%d %d\n"
```

Origin is the time of the first sample.

Increment is the time between samples.

These two times are defined in terms of strings because these values are normally kept as 64-bit signed integers. A value of 1 represents 1 pico-second.

- TimeTags / PagedTimeTags - The TimeTags object is used for any timing information that is not periodic.

```
<AbscissaDataType> ::=
    Abscissa Data Type        => "TimeTags\n"
    Number of Samples, Trigger Position
                              => "%d %d\n"
    Time Values               => IntegralArray&<signed64>
    Abscissa Attribute bitset => "%d [%d]* \n"
```

The PagedTimeTags object is also used for any timing information, but it has one additional data element, Time Values Flag. If this flag is zero, then the Time Values element is skipped!

```
<AbscissaDataType> ::=
    Abscissa Data Type        => "PagedTimeTags\n"
    Number of Samples, Trigger Position
                              => "%d %d\n"
    Time Values Flag          => "%d\n"
    (if TimeValuesFlag is non-zero, then Time Values)
    Time Values               => IntegralArray<signed64>
    Abscissa Attribute bitset => "%d [%d]* \n"
```

- Time Values - The time is kept in 64 bit signed integers. These integers represent the number of pico-seconds. The time value is an array of times and is accessed with a zero based index. There is a one-to-one correspondence between the sample array and this time value array.

Abscissa Attributes bitset

Bit#	Description
1	Abscissa has Length
2	Abscissa has HorizontalHeader
3	Abscissa has TimeCorrelation
4	Abscissa has TriggerRow
5	Abscissa is TimeTag
6	Abscissa is StateTag
7	Abscissa is SamplePeriod
8	Range
9	Abscissa is Periodic
10	Abscissa has TimeItor
11	Abscissa has StateNumberItor
12	Abscissa consists of SampledData

Time Correlation Info

The Time Correlation info object describes how this data set is time correlated to other data sets.

<Time Correlation Info>::=

Correlation Type => "TimeCorrelationInfo\n"

TimeCorrelationType, Source, CorrelationTime

=> "%d %d %s\n"

- TimeCorrelationType - The type of time correlation for this data set. Correlation can be of 3 different types. No time correlation means there is no time correlation information for this data set. Conditional time correlation means that the data set can be time correlated under certain circumstances with data from another machine. Unconditional time correlation means that the data is unconditionally correlated with another data set as it is from the same machine.

  0 No Time Correlation

  1 Conditional Time Correlation

  2 Unconditional Time Correlation
- Source - The source is the machine id from which the data set was obtained.
- CorrelationTime - The correlation time is a zero-relative time based on the data set that is considered the trigger data set. If the time is non-zero, this is the amount of time offset to use to correlate this data from the trigger data's time.

State Correlation Info

The State Correlation info object describes how this data set is state correlated to other data sets.

<State Correlation Info>::=

Correlation Type => "StateCorrelationInfo\n"

Offset => "%s\n"

- Offset - The correlation offset is a zero-relative number of states based on the data set that is considered the `trigger' data set. This value represents how many states to offset from the `trigger' data set to correlate the data. (Note: this value may be invalid. It is recommended that no dependence is placed on this field.)

Glossary

A

acquisition

Denotes one complete cycle of data gathering by a measurement module. For example, if you are using an analyzer with 200K memory depth, one complete acquisition will capture and store 200K states in acquisition memory.

acquisition depth

The acquisition depth is the amount of memory that is filled with data on an acquisition. The choices available depend on the maximum memory depth available in the analyzer that is being used.

action

Actions are things that the analyzer does as a part of triggering, for example "Then Trigger and Fill Memory" or "Start Timer."

activity indicator

Symbols next to logic analyzer channels that indicate whether a signal is a logic-high, or logic-low, or whether the signal is changing between highs and lows.

advanced trigger

Advanced triggers provide more power than simple triggers, but are more complex.

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.

arming

Typically, instruments are armed immediately when Run or Run Repetitive is selected. For example, logic analyzers are commonly used to arm oscilloscopes.

asynchronous sampling

When the logic analyzer acquires samples from the device under test asynchronously, that is, at regular intervals, such as every 100 ns. Also known as timing mode.

B

beginning of acquisition

The beginning of the acquisition is the point in time where the collection of data begins.

bits

A bit is a single signal in a bus. Numbering of bits begins with 0.

bus

A bus is a group of associated signals, such as ADDR or DATA.

C

captured data

Signal values that have been sampled by the logic analyzer and stored in its memory.

channel

A single line of input to the logic analyzer. Each channel corresponds to a lead that is connected to the device under test. Each channel is used to acquire one and only one signal from the device under test.

clock channel

A special logic analyzer input channel that can be used to determine the analyzer's sampling. Clock channels are identified on a pod by CLK.

D

data channel

A channel that carries data. Data channels cannot be used to clock logic analyzers. Data channels are numbered as opposed to clock channels which are labeled CLK.

default storage

Default storage means "unless sequence step storage specifies otherwise, this is what should be stored". Sequence step storage always overrides default storage.

delay

Delay is the horizontal position of the waveform on the screen for the timing analyzer. Delay time is measured from the trigger point in seconds.

device under test

The system under development whose digital signals are captured by the logic analyzer.

don't care

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 bus/signal.

double-click

When using a mouse as a 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

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.

E

edge

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 or glitch.

event

Events are the things you are looking for in your device under test, for example ADDR=0 or ADDR=5.

external trigger

A signal outside the logic analyzer that is used to synchronize measurements between instruments. For example, the logic analyzer can be armed (activated) by a signal that comes from another instrument. Logic analyzers are commonly used to trigger oscilloscopes through a BNC connection.

G

glitch

A glitch occurs when two or more transitions cross the logic threshold between consecutive timing analyzer samples.

I

inverse assembler

A tool that displays the assembly language instructions for captured machine code.

L

logic analyzer

An instrument that captures and displays digital signal values. A logic analyzer is like an oscilloscope, except that it only displays two voltage levels (a logic high or 1, and a logic low or 0) instead of many voltage levels. Because a logic analyzer only captures 1s and 0s, its sample rate can be slower than an oscilloscope that needs to capture move voltage detail. Consequently, a logic analyzer can capture a greater amount of overall execution time.

M

marker

A relocatable reference point in the data display. Markers can be used to measure time intervals or sample intervals. Markers are assigned to patterns in order to find patterns or track sequences of state in the data.

menu bar

The menu bar is located at the top of all windows. Use it to select drop down menus that contain tool or system options.

O

occurrence

Occurrence is used in triggering to define how many times something happens during the acquisition.

offline analysis

Analyzing previously captured and saved logic analysis data without data acquisition hardware. In other words, you can use the *Agilent Logic Analyzer* application by itself on a Windows NT/2000 computer to analyze data in the waveform, listing, and compare windows.

P

pattern

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.

pod

A collection of 16 data channels and 1 clock channel. Pods are used to physically connect data and clock signals from the device under test to the analyzer.

pod pair

A group of two pods containing 16 data channels and 1 clock channel each. Pod pairs are used to physically connect data and clock signals from the device 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.

point

To point to an item, move the mouse cursor over the item.

polarity

Positive polarity is when an incoming low voltage is shown with a high waveform and a logical value of 1. Negative polarity is when an incoming high voltage is shown with a low waveform and a logical value of 0. Polarity affects the display of values and waveforms, and does not affect the trigger.

preprocessor

See analysis probe.

probe

A device to connect the various instruments of the logic analysis system to the device under test. 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 device under test.

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 recieved a message.

Q

quick trigger

Quck trigger allows you to quickly set up a simple trigger within the waveform and listing displays, by drawing a rectangle in the display area with the mouse. After a simple trigger has been defined, and the analyzer is run, the trigger is stored and can be recalled at any time.

R

repetitive measurement

A measurement in which the logic analyzer's trigger condition is searched for, and data storage is filled, repetitively.

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.

run

The single run measurement will save captured data to trace memory one time. The amount of data stored during a single run is equal to the amount of trace memory allotted.

run repetitive

The run repetitive measurement will save the captured data to trace memory repetitively. The amount of data stored in a repetitive run is the same as a single run. During a repetitive run once the trace memory is full the system clears the trace memory and begins to refill with new data. This cycle will continue until the run is stopped.

S

sample

A data sample is a single measurement. When an instrument samples the device under test, it takes a single measurement as part of its data acquistion cycle. The number of samples acquired is equal to the logic analyzers memory depth.

sample period

The sample period is the period of time between samples. The sample period can be based on an internal sampling clock (also known as timing analysis or asynchronous sampling). Or, the sampling can be based on a signal in the device under test (also known as state analysis, or synchronous sampling).

sampled data

Signal values that are sampled by the logic analyzer (not necessarily stored).

sampling

The process by which the logic analyzer looks at digital signals.

search

Searches through the acquired data for specified data pattern or value, time value, sample number, or marker. Search criteria can range from specific bits to multiple events, depending on which search option you choose.

simple trigger

Simple triggers include triggers such as edges and bus patterns.

single measurement

A measurement in which the logic analyzer's trigger condition is searched for, and data storage is filled, once.

skew

Skew is the difference in channel delays between measurement channels.

snap to edge markers

Snap to edge markers enable easy placement of markers on waveform edges. When a marker is moved in the data display area, the cursor changes to a green "direction arrow" indicating the direction of the next valid edge. A red "valid edge" bar is placed on the next edge that the marker will be placed on.

state analyzer

A logic analyzer that samples based on a clock signal in the device under test.

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.

state mode

When the logic analyzer acquires samples from the device under test synchronously, in other words, when a signal or signals from the device under test indicates when to acquire a sample. For example, the logic analyzer might take a sample whenever there is a rising edge on a signal from the device under test. Typically, the signal used to set up the sampling is a state machine clock signal or microprocessor clock signal. Also know as synchronous sampling.

stop

Stops the measurement currently in progress.

storage qualification

Storage qualification is only available in a state measurement, not timing measurements. Store qualification allows you to specify the type of data (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 wait-loops. Storage qualification lets you filter out specific types of data as the acquisition is running, which saves memory. In contrast filters can hide data after it has been collected.

symbols

Names assigned to particular bus or signal values. Symbols in a display of captured data values are easy to read. Also, symbols make it easy to set up triggers on particular values. For example, in a communication protocol you could display the value FF as "end of file."

synchronous sampling

When the logic analyzer acquires samples from the device under test synchronously, in other words, when a signal or signals from the device under test indcates when to acquire a sample. For example, the logic analyzer might take a sample whenever there is a rising edge on a signal from the device under test. Typically, the signal used to set up the sampling is a state machine clock signal or microprocessor clock signal. Also know as state mode.

T

target system

See device under test.

threshold voltage

The voltage level that the signal must cross before the logic analyzer recognizes a change in voltage levels. A high voltage level is indcated by a "1" and a low voltage level is indicated by a "0." TTL and ECL are two examples of voltage levels that the signal must cross.

time/division

Time/division controls the "zooming" of a waveform display. Increasing the time/division zooms out, while decreasing the time/division zooms in.

timer

Timers are used to create either a user-defined delay or a time standard which valid data duration is evaluated against.

timing analyzer

A logic analyzer that samples at regular intervals based on an internal clock signal.

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 device under test, timing measurements capture traces of electrical activity over time. These measurements are asynchronous with the device under test.

timing mode

When the logic analyzer samples from the device under test asynchronously, that is, at regular intervals, such as every 100 ns. Also known as asynchronous sampling.

trace

See acquisition.

transitional timing

When the logic analyzer is in transitional timing mode, the timing analyzer samples data at regular intervals, but only stores data when there is a threshold level transition (high-to-low transition, or low-to-high transition). Each time a level transition occurs on any of the bits, data on all channels is stored. A time tag is stored with each stored data sample so the measurement can be reconstructed and displayed later.

trigger

The event about which acquired data is stored; in other words, the event that you are looking for. For example, you may want to trigger on an edge in order to see the events that lead up to it and the events that happen after it. The event that triggers the logic analyzer becomes a reference point in the data display.

trigger function

Trigger functions are preprogrammed components that are used to build trigger sequences.

trigger history

Each time you set up a new trigger and run the measurement, the trigger setup is saved in the configuration file. Each saved trigger can be retrieved and reused. The default number of triggers saved is 10.

trigger position

The location of the trigger event in trace memory. If you want to view data after, about, or before the trigger event, you set the trigger position to the start, center, or end of trace memory, respectively.

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.

V

value at measurement

The value at measurement measures the value of a bus or a single signal at a specified marker location in data. Measurement results are displayed in the marker measurement display bar.

Z

zooming

To expand and contract the waveform along the time base by varying the value in the time/div field. This action allows you to view specific portions of a particular waveform.

16700-series logic analyzers
offline analysis 17
accessories................................ 222
acquiring data 81
acquisition 347
acquisition depth............................ 70
acquisition speed 226
action 347
actions...................................... 101
activity indicator 347
activity indicators 152
adding a bus or signal.................... 49
advanced clocking 75
advanced trigger 89
ala 324
alias 61
all reference data 162
analysis probe 347
analysis tool development API ... 17
AND 101
API
tool development..................... 17
arming 347
asynchronous sampling............ 347
See timing mode.................... 347
autosize columns 152
background 152
beginning of acquisition 347
bits... 347
bus 347
bus width 47
bus/signal names
sorting 62
bus/signal setup............................ 47
buttons 229
cable 319
captured data 347
capturing data 81
center about 176
change display scale 140
change time/division 140
channel 347
channel width 77
characteristics 323
clock... 214
clock channel 347
collapse a bus 140
color... 134
difference foreground/background
.... 162
reference data background 162
column autosize 152
column properties 152
commands 233
comma-separated value files 328
comments 59
compare 17
compare differences
run until a number.................... 161
Compare display options 162
Compare Overview bar 158
comparing a range of samples 159
comparing captured data 156
config screen
See sampling 67
configuration 118
configuration translator................ 200
configure................................... 102
connect..................................... 189
connecting to target
See probing 316
connectivity 189
contact information 222
convert 200
counter 102
create marker 173
CSV format 328
cursors
see markers............................ 243
data channel 347
data valid 57
default base 54
default storage 92
delay 347
delete 50
demultiplex 72
development (tool) API.............. 17
device under test 347
dialogs...................................... 259
difference foreground/background
color 162
difference pair reference data 162
differences in compared data
finding 158
display...................................... 131
don't care 347
double-click.............................. 347
drag and drop........................... 347
draw rectangle in data 140
duration.................................... 279
edge 347
edge (multiple) trigger 17
edge trigger 85
edges

multiple....................................41
edit 238
edit menu 238
e-mail
send on compare differences 161
error messages 191
event 347
events 101
expand a bus 140
exporting data 114
external trigger 347
external triggering....................... 107
fast binary data format 333
favorite 109
file 237
file format 328
file formats................................ 130
file menu................................... 237
filter 259
filtering 259
find 163
finding differences in compared data
.... 158
folder..60
font size.................................... 152
format screen
see bus/signal setup 47
front panel 229
getting started 20
glitch 347
glitch (multiple) trigger 17
glitches
multiple....................................41
glossary 347
go to 167
group signals 143
help..20
help menu 246
hold...57
importing 167xx fast binary data... 121
input only reference data 162
install 188
inverse assembler 347
inverse assembler development API 17
Japanese help 18
keyboard commands 233
keypad 229
knobs 229
label
See bus/signal setup...................47
LAN 189
listing window 147
loading saved data and setups 121
loading/saving user-defined symbols 17
local area network 189
locate
See search 347
logic analyzer 347
logic analyzer basics......................20
macro
See trigger function 280
marker 347
marker menu............................. 243
markers 243
master clock 75
master/slave clock 72
maximum input 320
measure time
See markers 243
measurement examples 40
memory depth 226
menu bar 347
menus 236
minimum amplitude.................... 320
moving markers 182
multiple edge/glitch trigger 17
multiple edges/glitches 41
negate 98
negative logic 56
network 189
numeric base 54
occurrence 102
offline 191
offline analysis 123
offline analysis for 16700-series logic analyzers 17
offset
reference data 160
open... 118
operating environment................. 316
OR 101
overlay 142
overview 20
pattern 347
place on edge markers................. 134
pod 347
pod pair 347
point 347
polarity 347
poststore 71
preprocessor 347
prestore 71
printer 188
printing..................................... 186

probe ... 347
probe cable ... 319
probing ... 316
probing options ... 218
product overview ... 222
properties ... 134
properties dialog ... 264
protocol ... 347
pv tests ... 202
qualifier ... 74
quick trigger ... 347
range of samples
  comparing ... 159
read marker ... 183
recall trigger ... 263
reference ... 236
reference buffer
  copying data to ... 157
reference data
  comparing to ... 156
  display options ... 162
reference data background color ... 162
reference data offset ... 160
rename ... 178
repetitive measurement ... 347
repetitive run
  stop on compare differences ... 161
right-click ... 347
row height ... 134
run ... 347
run repetitive ... 347
run until a number of compare
  differences ... 161
run/stop menu ... 244
running the analyzer ... 110
sample ... 347
sample period ... 347
sampled data ... 347
sampling ... 347
sampling options ... 77
sampling period ... 316
sampling setup ... 67
save ... 112
saving captured data ... 111
saving/loading user-defined
  symbols ... 17
scale
  See change time/division ... 140
search ... 163
self tests ... 202
sequence level storage ... 92
setup menu ... 241
setup time ... 57
signal ... 143
signal loading ... 320
signal requirements ... 320
simple compare ... 17
simple trigger ... 85
single measurement ... 347
skew ... 347
slave clock ... 75
snap to edge markers ... 347
sorting bus/signal names ... 62
specifications ... 318
state analyzer ... 347
state clocking ... 72
state measurement ... 347
state mode ... 347
stop ... 347
stopping the analyzer ... 110
storage qualification ... 347
store qualification ... 92
store trigger ... 109
symbols ... 265
symbols (user-defined)
  saving/loading ... 17
synchronous sampling ... 347
  See state mode ... 347
system options ... 261
tabbed windows ... 145
target system ... 347
testing the logic analyzer ... 202
threshold voltage ... 347
time/division ... 140
timer ... 104
timing analyzer ... 347
timing measurement ... 347
timing mode ... 347
tool development API ... 17
tool tips ... 146
toolbars ... 247
tools ... 242
tools menu ... 242
trace ... 347
training ... 20
transitional timing ... 347
trigger ... 85
  multiple edge/glitch ... 17
trigger function ... 280
trigger history ... 347
trigger in ... 107
trigger out ... 107
trigger position ... 71
trigger sequence ... 347
triggering ... 82
tutorial ... 20

user interface 229
user-defined symbols
saving/loading 17
value at measurement.............. 347
view menu 240
waveform window 131
web sites 222
window menu 245
windows.................................... 257
working with markers 171
xml 329
zoom in on data 140
zooming 347

Publication Number: 5988-7525EN

Agilent Technologies