

Agilent E2929/E2930 Opt. 100 PCI-X Analyzer

## **User's Guide**



**Agilent Technologies**

## Important Notice

All information in this document is valid for both Agilent E2929 and Agilent E2930 testcards unless otherwise noted.

© Agilent Technologies, Inc. 2003

### Revision

June 2003

Printed in Germany

Agilent Technologies  
Herrenberger Straße 130  
D-71034 Böblingen  
Germany

Authors: t3 medien GmbH

### Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

### Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

### Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

### Safety Notices

#### CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

#### WARNING/DANGER

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

### Trademarks

Windows NT ® and MS Windows ® are U.S. registered trademarks of Microsoft Corporation.

# Contents

<b>Documentation Overview</b>	<b>5</b>
<b>Running A Sample PCI-X Analyzer Session</b>	<b>7</b>
PCI-X Analyzer Scenarios	7
Preparing for the Guided Tour	9
Guided Tour: Analyzing PCI-X Traffic from System Memory to the Testcard	10
Setting Up the Trigger	11
Setting Up the Storage Qualifier	13
Running the PCI-X Analyzer	14
Analyzing the Captured Waveforms	14
Analyzing the Captured Bus Cycles	16
Analyzing the Captured Transactions	17
<b>Capturing Data in the Trace Memory</b>	<b>19</b>
Data Stored in the Trace Memory	20
Setting Up the Data Capture	22
How to Set Up the Trigger	23
How to Specify a Trigger Pattern	25
How to Set Up the Storage Qualifier	27
How to Specify Transactions and States	28
Setting Up the Trigger Sequencer	29
Sample Sequencer Setup	31
How to Set Up the Trigger Sequencer	33
How to Run the Data Capture	34

<b>Viewing And Processing the Trace Memory Capture</b>	<b>37</b>
Using the Waveform Viewer	38
How to Arrange the Signal Display	40
How to Adjust Range and Resolution	41
How to Use the Markers	42
How to Synchronize the Listers	43
Using the Bus Cycle Lister	44
Browsing Through the Cycles	45
Using the Transaction Lister	45
Processing the Captured Data	48
<b>Measuring Real-Time Performance</b>	<b>49</b>
Predefined Performance Measures	49
Running a Real-Time Performance Measurement	50
How to Select a Testcard for the Measurement	52
How to Set Up the Testcard for the Measurement	52
How to Select Predefined Performance Measures	55
How to Set up Run Options	56
How to Run a Performance Measurement	57

# Documentation Overview

This section shows you the different types of documents offered by Agilent Technologies and gives you an overview of which documents are available when you work with the Agilent E2929/E2930 PCI-X Exerciser and Analyzer.

All information is valid for both Agilent E2929 and Agilent E2930 testcards unless otherwise noted. The following documents are available:

## Getting Started Guide

- **Getting Started Guide**

Introduces standard analysis features and provides an example of how to set up the protocol observer.

Provides an overview of how the Agilent E2930 testcard supports the PCI-X 2.0 features.

Gives detailed information about hardware and interfaces.

## User's Guides

- **Agilent E2929/E2930 Opt. 300 PCI-X Exerciser User's Guide**

Provides information on programming the testcard as an initiator and/or target device. It shows you how to actively stimulate the PCI-X bus.

This guide shows how to:

- Initiate data transfers on the PCI-X bus (act as requester-initiator).
- Act as completer-target.
- Handle split-completion transactions (act as completer-initiator).
- Handle open requests (act as requester-target).

- **Agilent E2929/E2930 PCI-X Analyzer User's Guide**

Provides information on how to examine the behavior and performance of a PCI-X device on the bus and shows how to perform functional tests such as data compares.

- **E2929 Opt. 200 PCI-X Performance Optimizer User's Guide**

Provides all features that are needed to evaluate and optimize any device under test in terms of the performance (post-processed performance analysis and optimization). Option 200 is available for Agilent E2929 testcards only.

- **Agilent E2920 PCI-X Series Opt. 320 C-API/PPR Programmer's Guide**

Provides information on how to set up test programs using the C functions described in the corresponding C-API/PPR Reference.

#### GUI and C-API/PPR References

- **Agilent E2929/E2930 Windows and Dialog Boxes Reference**

Provides reference information on all windows and dialog boxes of the Agilent E2920 graphical user interface (GUI).

- **Agilent E2929/E2930 Opt. 320 C-API/PPR Reference**

Describes all C functions, types and definitions of the application programming interface of the Agilent E2929/E2930 PCI-X testcard.

This reference also provides the commands and abbreviations that are used in the command line interface (CLI) of the graphical user interface.

- **Agilent E2922/E2923 Opt. 320 C-API/PPR Reference**

Describes all C functions, types and definitions of the application programming interface of the Agilent E2922/E2923 PCI-X testcard.

This reference also provides the commands and abbreviations that are used in the command line interface (CLI) of the graphical user interface.

# Running A Sample PCI-X Analyzer Session

The following application example explains how the testcard can be used in various analyzing tasks. After introducing the major scenarios for the PCI-X Analyzer and showing how to prepare for the sample session, you will find a guided tour:

- Guided Tour: Analyzing PCI-X Traffic from System Memory to the Testcard.

**NOTE** For E2930 testcards, the PCI-X Analyzer is available with the standard hardware (4M state trace memory). For E2929 testcards, the PCI-X Analyzer is only available with the option 100 hardware extension (2M state trace memory).

## PCI-X Analyzer Scenarios

The PCI-X Analyzer helps you, if you are:

- designing a PCI-X chip and you need to do bring-up or debugging,
- using a third party PCI-X chip on your motherboard or adapter card that you need to evaluate,
- trying to find the root cause of a failure that occurred during your chip or system level validation,
- writing and debugging low-level software (for example, BIOS code, device drivers).

It gives you the possibility to monitor the PCI-X bus to find out whether your software generates the correct PCI-X transactions, and also whether your device under test reacts correctly, both at the protocol and the data level.

The PCI-X Analyzer allows you to capture PCI-X traffic and view it as a state waveform, a bus cycle listing or as a transaction listing. The following example show you how to set up the PCI-X Analyzer and how to interpret the results.



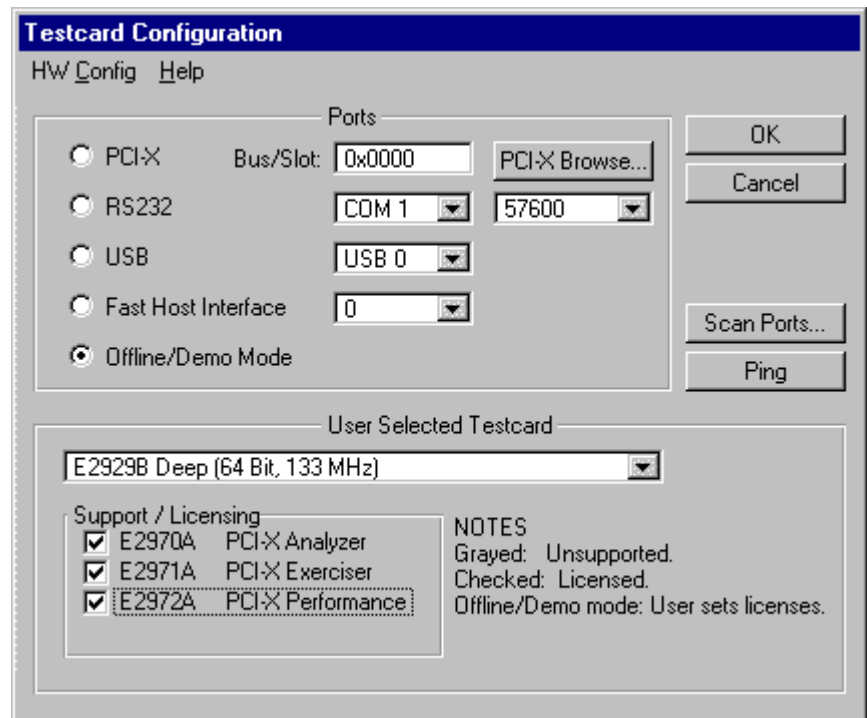
# Preparing for the Guided Tour

The example described in the guided tour is designed to be performed in Offline/Demo Mode—without hardware.

All the setup files (\*.bst) and logic analyzer trace files (\*.wfm) that are mentioned in the following text can be found under <your\_installation\_directory>\samples\demo. If you did not change the default setting during installation, <your\_installation\_directory> will be c:\Program Files\Agilent\E2920 PCI-X Series.

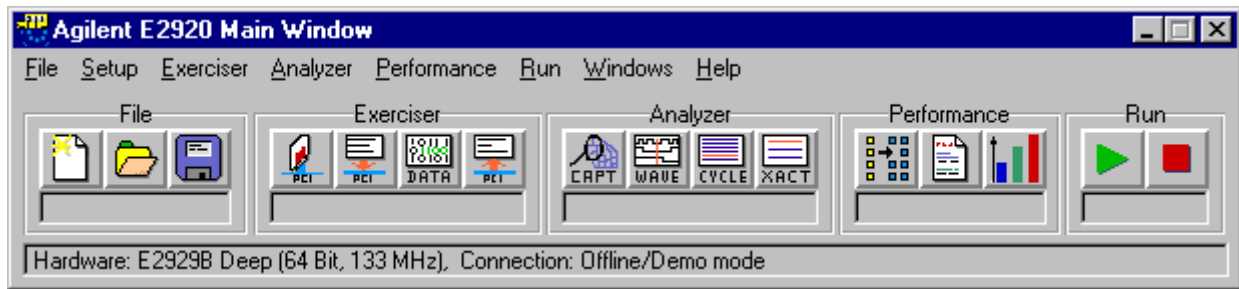
To prepare for the guided tour:

- 1 Launch the Agilent E2920 software.
- 2 From the *Setup* menu, choose *Testcard Configuration*.
- 3 In the Testcard Configuration window, select *Offline/Demo Mode*.



- 4 Now choose *E2929B Deep (64 bit, 133 MHz)* from the *User Selected Testcard* list, and select all licenses in the *Support/Licensing* group. Your display should look like the window shown above.

5 Click *OK* and the main window should look like this.



You are now ready to start the guided tour.

## Guided Tour: Analyzing PCI-X Traffic from System Memory to the Testcard


This example shows how to set up the testcard to trigger on a particular address range and capture PCI-X traffic that occurs around this triggerpoint. The traffic is generated by a testcard read from the system memory.

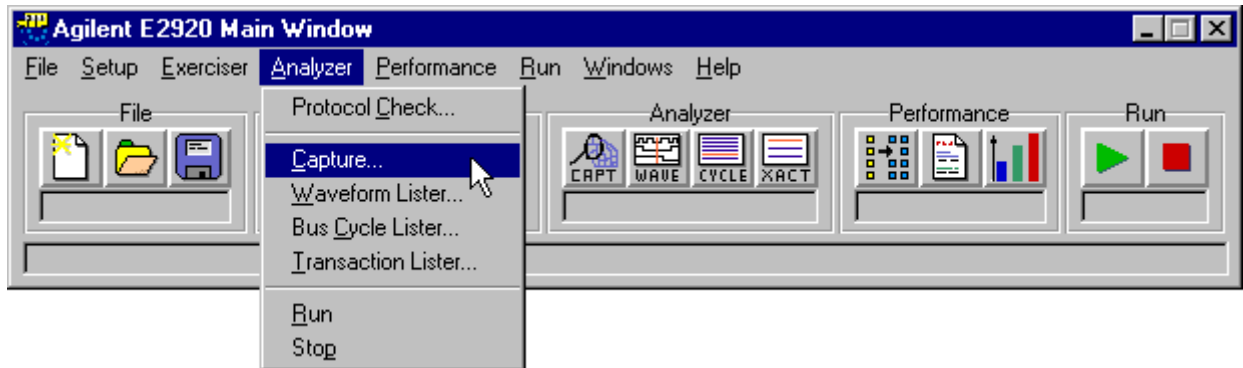
Afterwards, the captured data can be viewed at various levels of abstraction to analyze the PCI-X behavior of the participating devices.

For this example, the logic analyzer will be set up to trigger on a PCI-X address phase with an address of 0x10003000, which corresponds to the system memory. All PCI-X cycles will be stored, including idle cycles.

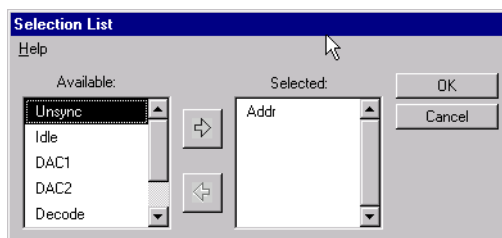
## Setting Up the Trigger

To set up a trigger and storage qualifier for the logic analyzer:

- 1 Use the Capture button  in the icon bar of the main window, or choose *Capture* from the *Analyzer* menu.

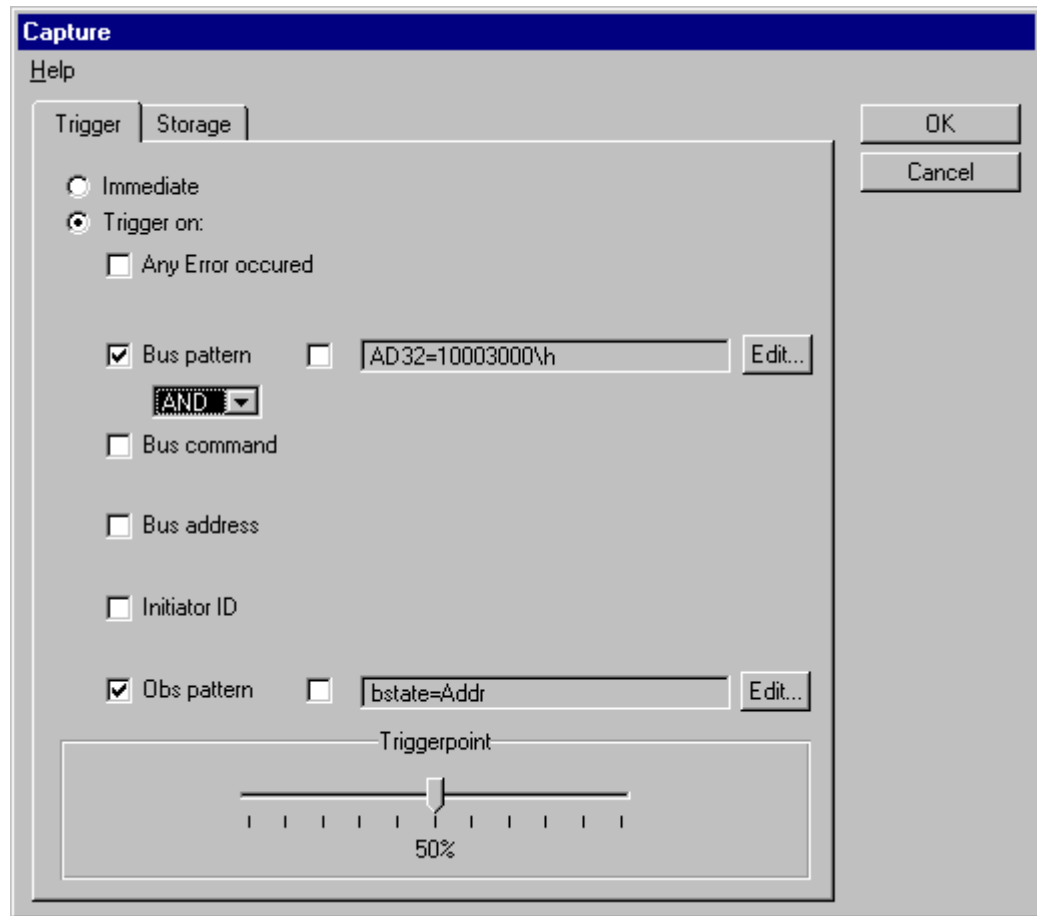


- 2 In the Capture dialog box, select the *Trigger* tab.
- 3 Choose *Trigger on:* and select *Bus pattern*.
- 4 Click the *Edit* button.  
This opens the Pattern Editor dialog box.
- 5 In the Pattern Editor:
  - Enter `10003000\h` for *AD32*.
  - Select *x* for *FRAME*.
- 6 In the Capture window, select *Obs pattern*.
- 7 Open the Pattern Editor dialog box and click on the text field to the right of the signal *bstate*.  
This opens the Selection List dialog box.
- 8 Highlight *Addr* and click the right-arrow button to place it in the right (*Selected*) box and click *OK*. You can use the left-arrow button to remove the *DONT CARE* entry from the *Selected* list.



- 9 In the Capture window, select *AND* from the list below *Bus pattern*.  
This builds a conjunction between bus and observer pattern terms.

The Capture window should now look like this.



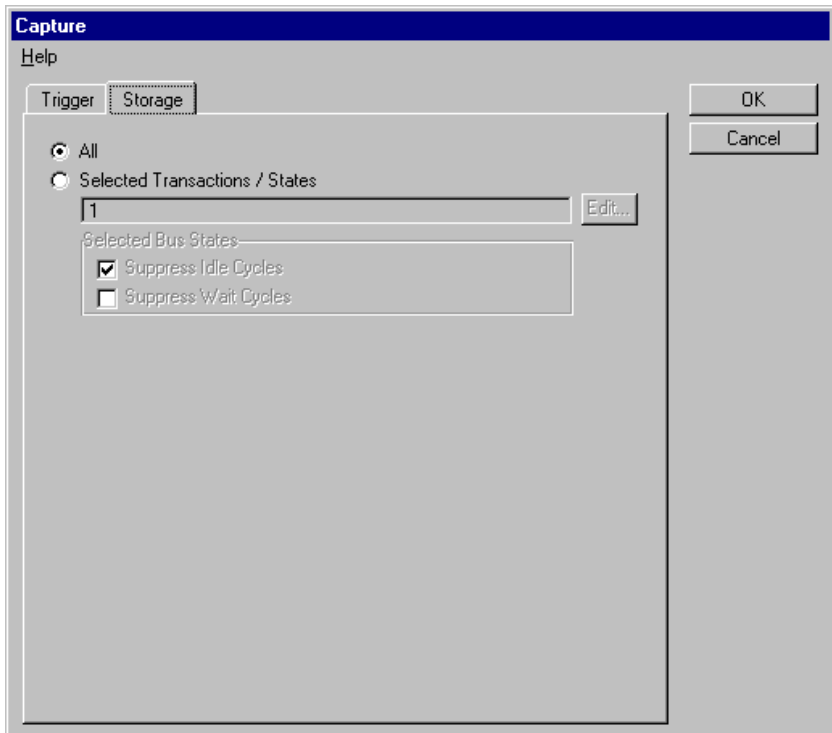
## Setting Up the Storage Qualifier

Now, take a look at the *Storage* tab of the Capture window. On this tab, you can define the storage qualifier. The default is *All*, which instructs the logic analyzer to unconditionally capture one sample per clock. Alternatively, you can choose *Selected Transactions/States*, which allows you, for example, to suppress idle cycles between transactions during a transaction.

With the pattern fields, you can further restrict what is captured in the trace memory by storing only particular transaction types (for example, only memory writes) or storing only transactions where the testcard is participating as a initiator or target.

- 1 In the Capture dialog box, select the *Storage* tab.


For this example, the default *All* can be used.



- 2 Press *OK* in the Capture window, and the logic analyzer is ready to run.


## Running the PCI-X Analyzer

Starting the PCI-X Analyzer in offline mode results in an error, but this would be the next step in our procedure if we were connected to a testcard.


This is normally done by pressing the Run button  in the main window, which also starts the PCI-X Exerciser if installed or by selecting *Run* from the *Analyzer* menu. When connected to a testcard, the status bar of the Analyzer group in the main window changes to *Running...* to indicate that the Analyzer is running and waiting for a trigger signal.

## Analyzing the Captured Waveforms

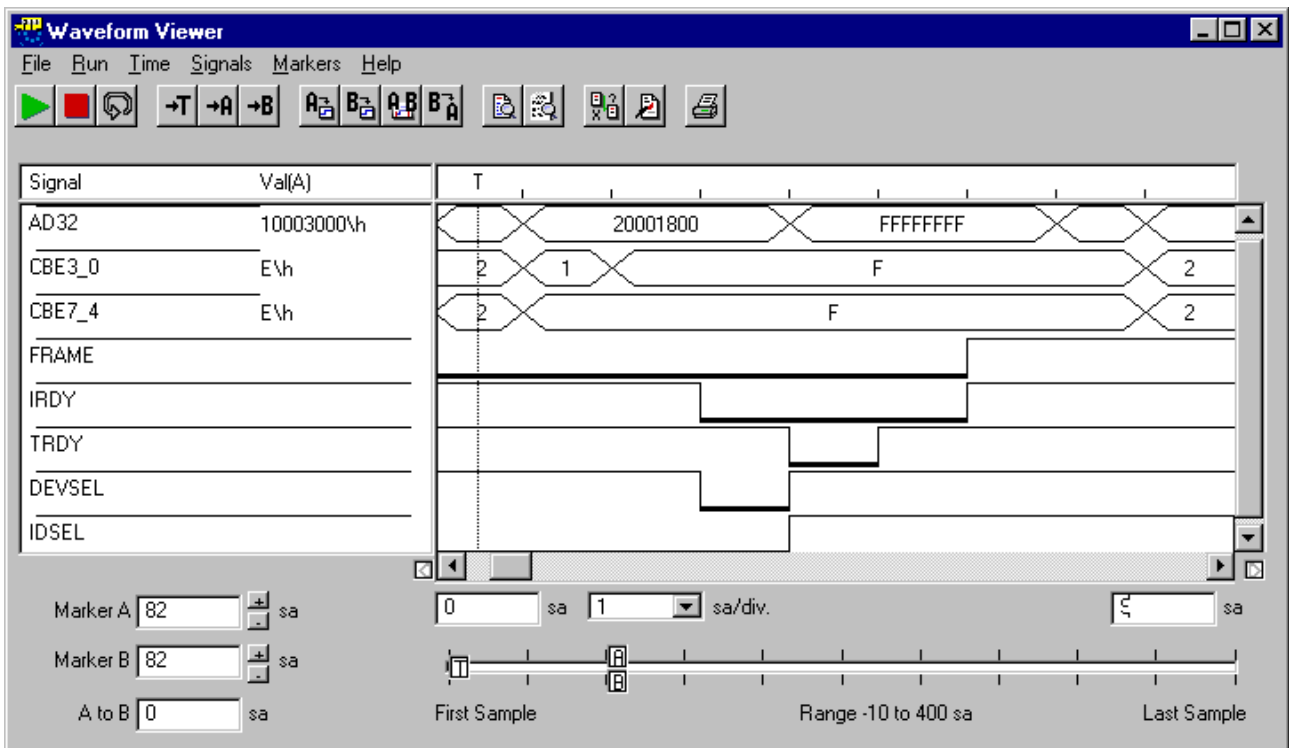
The captured data can be analyzed at different levels of abstraction. We start by using the waveform viewer:

- 1 Click the Waveform Viewer button  in the main window (or use the *Waveform Lister* item from the Analyzer menu) to open the waveform viewer.
- 2 From the *File* menu in the Waveform Viewer window select *Load from file* and load the trace file *rtbeh.wfm*.

**3** To navigate within the waveform viewer,

- press the Goto Trigger button  to view data at the trigger,
- enter a sample number in the left or right fields below each corner of the waveform display, or
- use the scroll bar to move the viewed data.



Markers A and B can also be moved with the mouse or by entering a position.

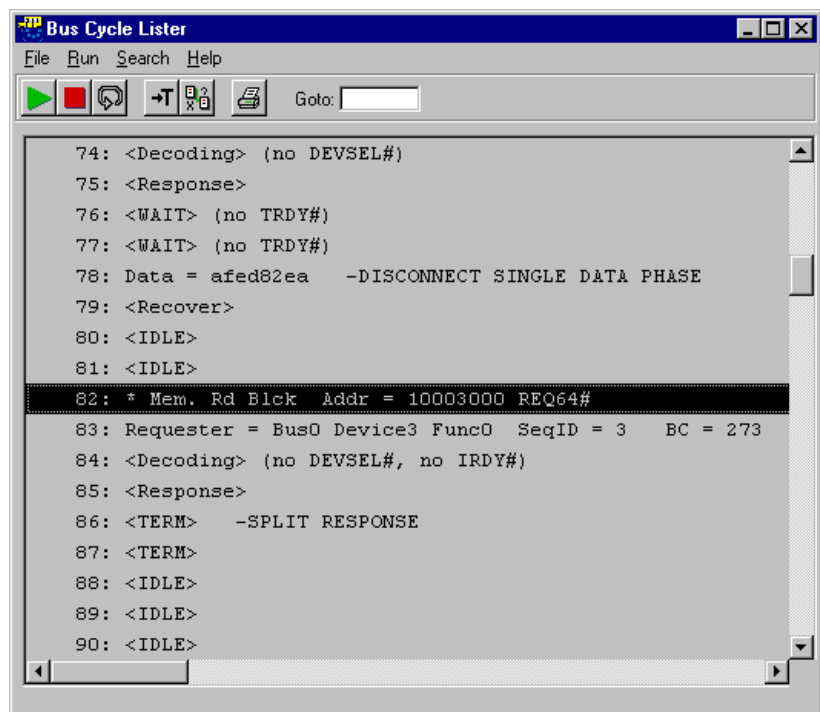



If you need more information about the buttons and other controls in the waveform viewer, drag your mouse over the control icons to view the tool tips.

## Analyzing the Captured Bus Cycles

Although the waveform viewer is appropriate for analyzing single transactions or when you need to check the state of individual control signals, it is tedious to “read” PCI-X transactions by looking at the waveform viewer. This is where the bus cycle lister helps.

- 1 Click the Bus Cycle Lister button  in the main window (or use the *Bus Cycle Lister* item from the Analyzer menu) to open the bus cycle lister.
- 2 In the Bus Cycle Lister window, press the Goto Trigger button  to go to the triggerpoint.




- 3 If you want to view the waveform for a given set of lines in the bus cycle lister, highlight the desired lines in the bus cycle lister and press the Cross Reference button  (make sure that the waveform viewer is still open or minimized).

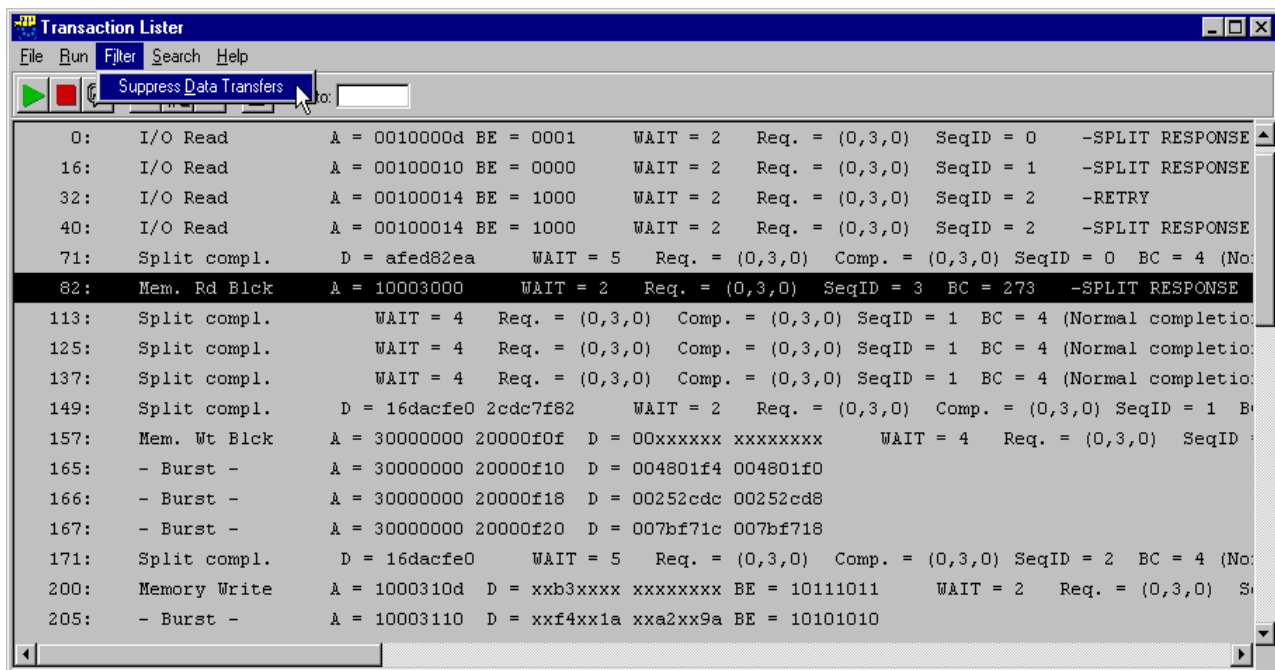


## Analyzing the Captured Transactions

To get a more compressed overview of the transactions that occurred on the bus:

- ◆ Click the Transaction Lister button  in the main window (or use the *Transaction Lister* item from the Analyzer menu).

The transaction lister removes idles from the display and summarizes the number of waits for each data phase, just showing useful information such as address and data phases. Address reconstruction is also done during bursts.



**Suppressing Data Phases** To view the address phases only, check *Suppress Data Transfers* in the *Filter* menu.



# Capturing Data in the Trace Memory

The PCI-X Analyzer records information such as PCI-X signals and bus states in its trace memory. The information is stored as one trace memory line per clock cycle.

The size of the trace memory is 4M states for E2930 testcards, 2M states for E2929 testcards with option 100.

## Controlling the Data Capture

The PCI-X Analyzer provides all features required to make optimal use of the available memory depth:

- The trigger allows you to start capturing data when a programmable trigger event has occurred. Furthermore, by specifying the triggerpoint, you can additionally control the number of bus cycles stored before and after the trigger event.
- Storage qualification allows you, during data capture, to selectively filter certain phases or clocks. For example, you can focus on accesses to a particular device, or on address phases only. Only selected phases and clocks are captured.

Another possibility to focus on address phases is to suppress data transfer in the *Transaction Lister*. In this case, both address and data phases are captured, only the display changes. See “*Using the Transaction Lister*” on page 45.

## Using the Data Capture

Data stored in the trace memory can be displayed at different levels of abstraction: as waveforms, bus cycles, or transactions.

Not only can you analyze the data in terms of PCI-X behavior, but you can also implement functional tests such as data compares. For example, if you are debugging a LAN interface, you could capture all of the blocks of data going into and coming out of the card. Once this data has been captured, you could analyze it to isolate the bugs on the LAN interface.

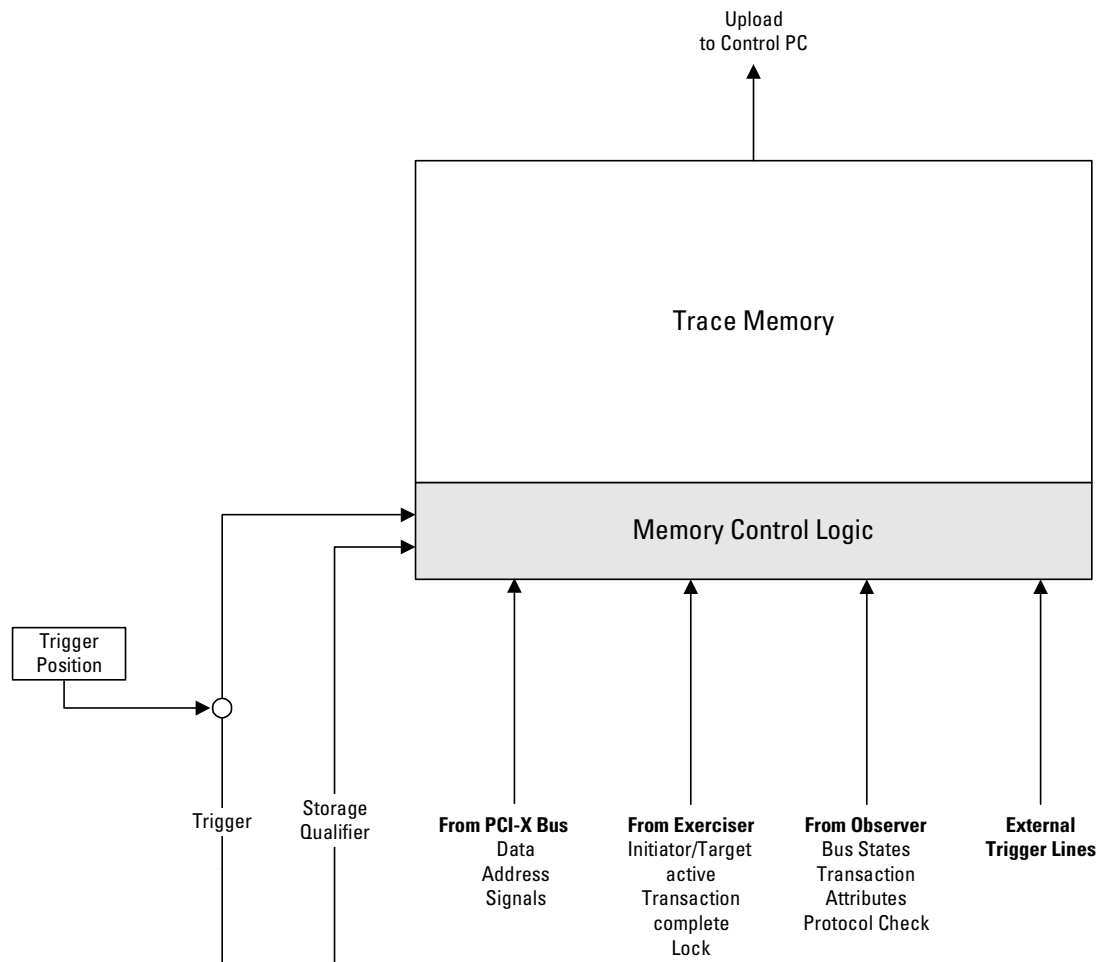
# Data Stored in the Trace Memory

The trace memory on the Agilent E2929/E2930 testcard stores all PCI-X signals and bus states, the Exerciser states, and additional information. The trace data is stored as one trace memory line per PCI-X clock cycle and can be used for low-level debugging and analysis of bus traffic.

## Trace Memory Components

The capture is controlled by the trace memory *trigger* and the *storage qualifier*.

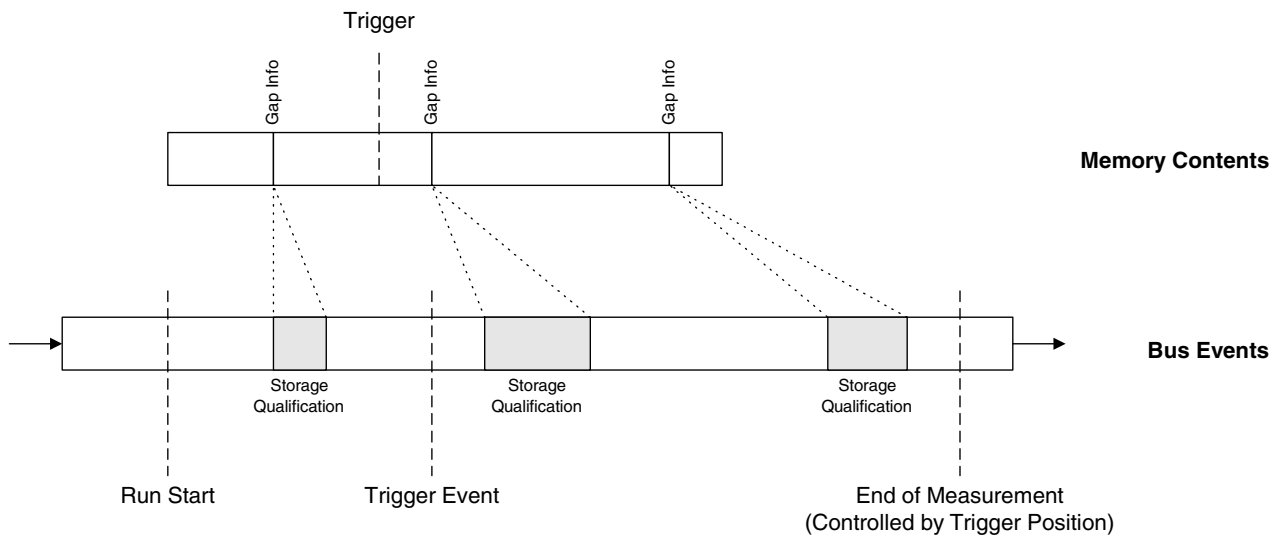
The figure below gives an overview of how the trace memory is built up.



## Recording Data

The trace memory is a circular memory that is filled continuously while the Analyzer is running. The storage qualifier controls which bus states are recorded. If one or more lines are filtered, a gap information is stored instead.

Recording is stopped after the trigger event has occurred. The trigger position specifies the amount of pre-trigger and post-trigger data to be kept in the data capture.



The lower bar in the figure represents events on the bus. Some of them do not meet the storage qualifier condition and are therefore filtered out (gray areas). Gap information is stored instead of the events.

After the Analyzer has been started, the memory is filled. Because the trace memory is a circular memory, previously captured states will be overwritten until the trigger event occurs.

Stored data is represented by the upper bar in the figure: not suppressed bus events and gap information. When the trigger event occurs, the trace memory continues to be filled until the specified amount of post-trigger information is stored.

**Uploading Data** Once the measurement is complete, the trace memory contents are uploaded. They can be displayed in the waveform viewer, the bus cycle lister, and the transaction lister.

The waveform viewer, the bus cycle lister and the transaction lister allows you to upload this data to a file (wfm file).

To reduce the upload time, you can specify the size of the trace memory to be uploaded. This does not reduce the amount of data that is recorded.

**NOTE** If the Analyzer is set up to fill the trace memory after a trigger event and the event does not occur, stopping the Analyzer sets an artificial triggerpoint. The trace memory will then contain 100 % pre-trigger history. The last captured state is the state before the Analyzer was stopped.

Nevertheless, the trace memory will still be empty if no samples were taken because of the storage qualification.

## Setting Up the Data Capture

Setting up the data capture consists of the following steps:

- Setting up the trigger.  
If required, specify a trigger pattern.
- Setting up the storage qualifier.

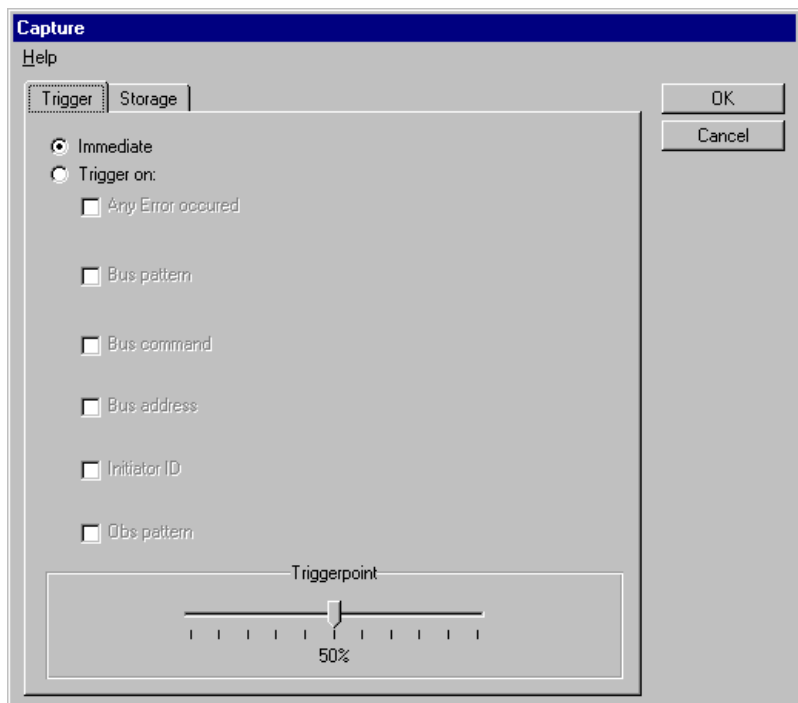
## How to Set Up the Trigger

To set up the trigger, you must program the trigger event for the trace memory trigger.

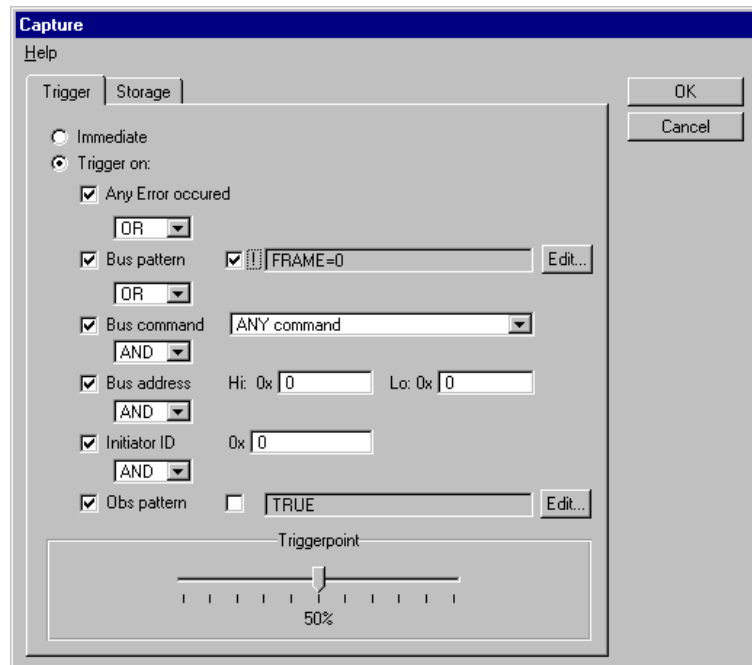
By default, the trigger is set to immediate. The data capture will be started as soon as the Analyzer is started.

To specify a trigger event:

- 1 In the main window, click the Capture button .



- 2 Select *Trigger on:* to set up a trigger event.
- 3 Select one or more events to trigger on. The selected events can be OR- or AND-combined.



- 4 Specify the selected events.

To specify the bus and observer patterns, click the corresponding *Edit* button.

This opens the Pattern Editor dialog box, where you can set up the pattern (as described in “*How to Specify a Trigger Pattern*” on page 25). When you are finished, the pattern term is displayed.

For more information, please refer to the *Agilent E2929/E2930 Windows and Dialog Boxes Reference*.



- 5 Move the slider to specify a *Triggerpoint*:
  - 0 % means: no pre-trigger history is stored
  - 100 % means: only pre-trigger history is stored
- 6 Now the trigger is set up. Click *OK* to store your settings or change to the *Storage* tab to set up the storage qualifier.

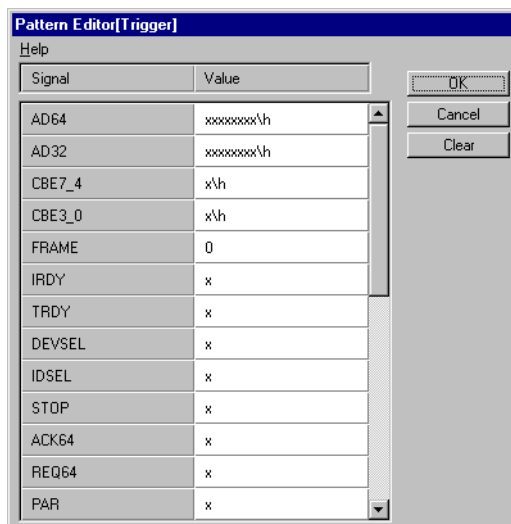
## How to Specify a Trigger Pattern

The Pattern Editor helps you in setting up the conditions that describe the trigger event, among other events.

Specifying a pattern means selecting the signals and values of interest. Your selections are AND-combined to build the pattern term.

To specify the trigger pattern:

- 1 Click the *Edit* button on the *Trigger* tab in the Capture dialog box to open the Pattern Editor.

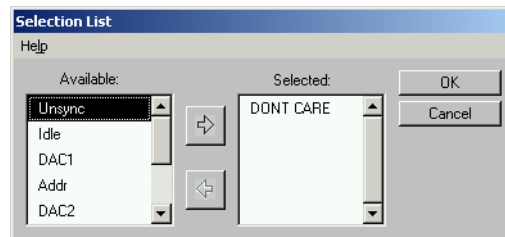


- 2 To assemble the pattern, click in the *Value* column of the required signals.

The method of entering the values depends on the type of the signal:

- If the signal is a bit vector, for example, the address and data bus signal *AD32*, you can enter the value directly into the value field. Possible formats are hexadecimal (entry ends with **\h**), decimal (**\d**) and binary (**\b**).
- If the signal is a single bit, for example, the initiator ready signal *IRDY*, you can select a value from the list in the value field. The possible values are 0, 1, and x (don't care).

- If the signal is a type that can take several different values, for example, the bus state *bstate* or the transaction command *xact\_cmd*, you can select a value. When clicking the value field, the Selection List dialog box opens to present the possible states.



Select from the list of available states. The software OR-combines the *Selected* (font) states.

To return to the pattern editor, click *OK*.

- 3 After you have included all required signals, click *OK*.

#### Conjunction of Signals

The signals of the pattern are AND-combined automatically to build the conditional expression.

##### Example:

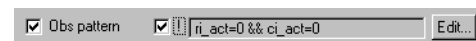
Setting the FRAME and IRDY values to 1, will result in this logical expression:

```
"FRAME=1 && IRDY=1"
```

#### Disjunction of Signals

If you need OR-combined signals, use the check box in front of this pattern. This negates the specified pattern. Using the rule of deMorgan turns the conjunction into a disjunction and reverse the signals.

##### Example:



This results in:

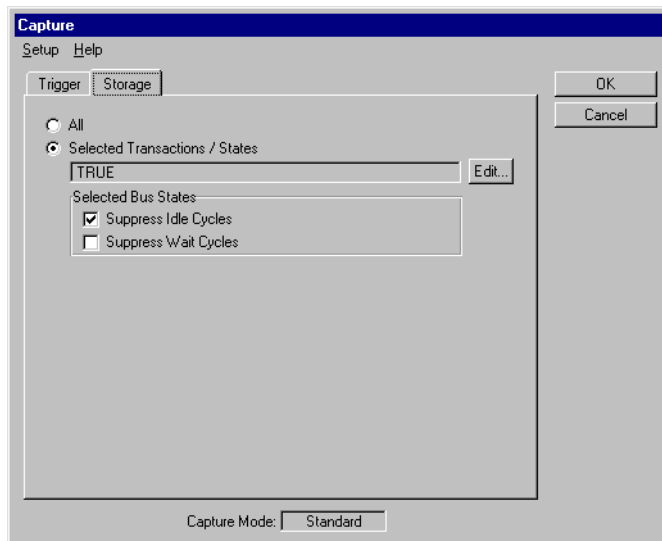
```
"ri_act=1 || ci_act=1"
```

## How to Set Up the Storage Qualifier

To set up the storage qualifier in standard capture mode, program a pattern to specify the transactions and states to be stored. Furthermore, you can select to suppress certain cycles if they are irrelevant for your test. By default, all cycles will be captured.

To specify the storage qualifier:

- 1 From the *Analyzer* menu, select *Capture* and change to the *Storage* tab.



- 2 Select *Selected Transactions/States* to start setting up the storage qualifier.

- 3 Click the *Edit* button to specify the pattern term that describes the storage qualifier condition.

This opens the Pattern Editor dialog box, where you can set up the pattern as described in “*How to Specify Transactions and States*” on page 28. On return, the pattern term will be displayed in the text field.

- 4 In the *Selected Bus States* section, check the bus states you wish to suppress from being stored (idle and/or wait cycles).
- 5 Click *OK*.

## How to Specify Transactions and States

If you wish to exclude certain transactions or bus states from being stored in the trace memory, you need to specify a condition for the storage qualifier. The pattern editor supports you in setting up this expression.

**NOTE** The storage qualifier condition identifies the transactions and states to be stored and, thus suppresses all others.

To specify the condition, select transactions and states:

- 1 Click the *Edit* button on the *Storage* tab in the Capture dialog box to open the Pattern editor.



- 2 To define the pattern, click in the *Value* column of the required signals. The method of entering the values depends on the type of the signal; if you keep the mouse button pressed, a box appears listing suitable values for selection.

- **xact\_cmd**

Only transactions that use the selected commands are stored.

- **ct\_act**

Set this to 1 to store only transactions in which the *completer-target* of the Agilent E2929/E2930 testcard is involved, or set it to 0 to suppress these transactions.

- **ri\_act**

Set this to 1 to store only transactions in which the *requester-initiator* of the Agilent E2929/E2930 testcard is involved, or set it to 0 to suppress these transactions.

- **xact\_tran64**

Set this to 1 to store only transactions with a 64-bit request, or set it to 0 to suppress these transactions

**3** After you have included all the required signals, click *OK*.

The assembled transactions and state settings are automatically AND-combined to build the storage qualifier condition.

## Setting Up the Trigger Sequencer

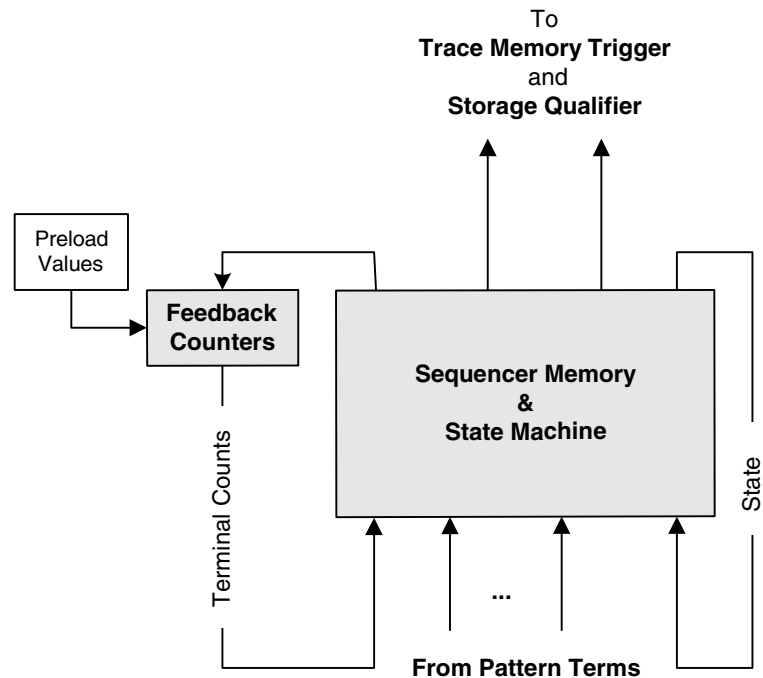
For more advanced measurements, you can program the Agilent E2929/E2930 testcard's trigger sequencer to implement sophisticated trigger and storage qualifier conditions.

### Goals of the Sequencer

The sequencer is designed to detect sequences of bus states. It is implemented with a state machine that compares bus states with programmable pattern terms to recognize these sequences.

The trigger sequencer controls the *trigger* and the *storage qualifier* for the trace memory. The trigger is fired after a specified sequence of bus states has been detected. After the trigger has been fired, the trace memory is filled with sampled states that meet the conditions of the storage qualifier.

The following figure details the trace memory trigger sequencer.



**Sequencer Internals** The sequencer provides an internal memory, a state machine, and two 32-bit feedback counters (A and B). The state machine controls the operation of the sequencer.

For each state, *transition conditions* specify when to switch to the next state. The transition conditions can be built from pattern terms and the terminal count of the sequencer's feedback counters.

**Feedback Counter** A feedback condition is used to decrement a loop variable that starts counting at the respective preload value (feedback counter A or feedback counter B). The preload value is set if a specified preload condition is met. If the loop variable reaches zero, the respective terminal count signal (tc\_fba or tc\_fbb) is set, which can be used within a pattern term.

**Sequencer Description Table** The sequencer is controlled by a *sequencer description table* consisting of a number of transitions. Each transition is defined by a table row, holding a state number, the number of the next state, and the condition defining when to switch to the next state.

In addition, each transition defines output conditions:

- the feedback counter decrement conditions
- the feedback counter preload conditions
- the trigger signal
- the storage qualifier

**NOTE** All actions take place on a state transition. There are no inherent actions by being in any state.

## Sample Sequencer Setup

**Example Task** The following example shows the basic principles of programming the trigger sequencer. The following sequence is to be detected:

1. Wait for the address phase of an access to video memory.
2. When the address phase is detected, trigger and store all the transfers.
3. Stop storing if an idle cycle occurs.
4. Wait for the next access to video memory.

**Pattern Terms** For this sequence, the following patterns need to be detected and are therefore assigned to pattern terms:

- bus0: `addr_phase==1 && AD32==b8xxx\h`

This programs bus pattern term bus0 to detect an address phase that addresses the range b8000 ... b8fff.

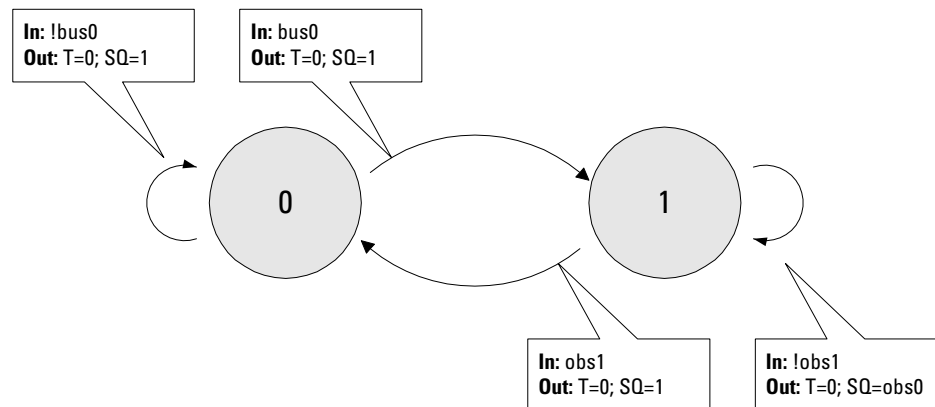
- obs0: `xact_cmd==Mem_ReadDW && ri_act==1`

This programs observer pattern term obs0 to detect Memory Read DWord transfers executed by the requester-initiator.

- obs1: `bstate==Idle`

This programs pattern term obs1 to detect idle cycles.

**Building a State Diagram** The next step is to determine the sequence in which the patterns are to be detected and what is to happen to the trigger and storage qualifier. Especially when planning for long and difficult sequences, it is recommended that you use a state diagram like the following:



This state diagram can easily be translated into a sequencer description table.

Each transition (arrow) in the state diagram requires a row in the table. The sequencer description table for the example is as follows:

**Table 1 Example Sequencer Description Table**

Cur State	Next State	XAct Cond.	Trig Cond.	SQ Cond.
0	0	!bus0	0	0
0	1	bus0	1	1
1	1	!obs1	0	obs0
1	0	obs1	0	1

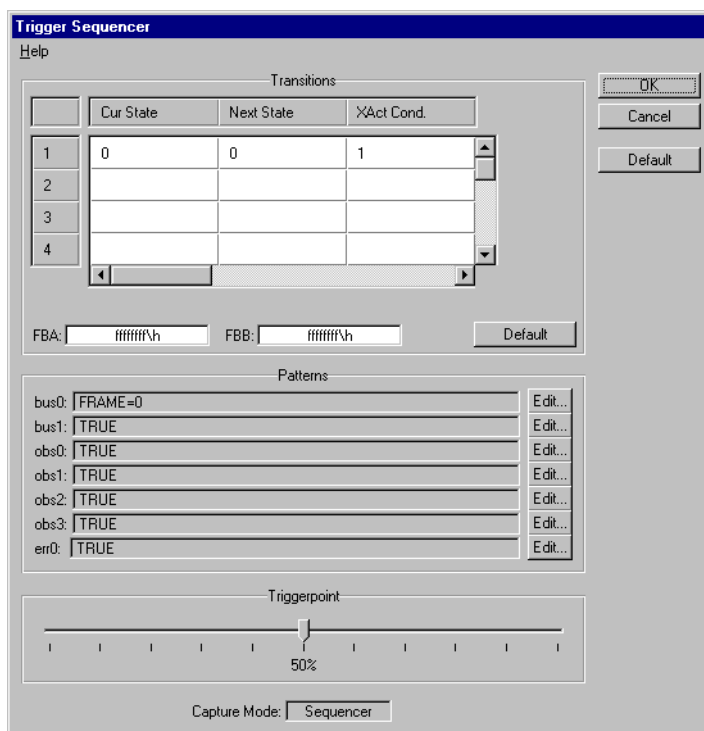


## How to Set Up the Trigger Sequencer

Setting up the trigger sequencer can be quite complicated, depending on the test requirements. The following instructions only outline the basic steps. Use this sample sequencer setup to make yourself familiar with the principles of sequencer programming.

To set up the trigger sequencer:

- 1 From the *Setup* menu, check *Trigger Sequencer* under *Options*.
- 2 From the *Analyzer* menu, select *Capture...* to open the *Trigger Sequencer* dialog box.



Proceed as follows to enter the sequencer setup you have developed on your own:

- 1 Enter your sequencer description table into the *Transitions* table.
- 2 Set the *Preload Value* (*FBA* or *FBB*) of the sequencer's feedback counter (not used in the example).

The preload values of the sequencer's feedback counters determines how often a sequence must occur before an output signal is set or a transition is made.

- 3 Assemble the patterns *bus0*, *obs0* and *obs1* as described for the trigger patterns in "How to Specify a Trigger Pattern" on page 25.

- 4 Move the slider to specify a *Triggerpoint*:
  - 0 % means: no pre-trigger history is stored
  - 100 % means: only pre-trigger history is stored
- 5 Now the trigger sequencer is set up. Click *OK* to store your settings.

**NOTE** All transition conditions of one state must be exclusive. This means, that one—and only one—transition condition of a state must turn true at a time. Otherwise, the software will not accept the table because the table does not uniquely define the sequencer's behavior.


## How to Run the Data Capture

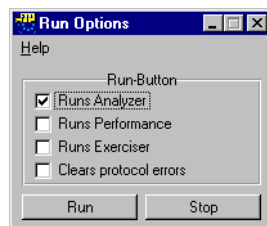
After setting up the trigger and storage qualifier, the testcard is ready to capture data.

Starting the PCI-X Analyzer enables the pattern terms and the trigger.

The analysis requires traffic on the PCI-X bus under examination. Therefore, you must load traffic onto the bus when running the PCI-X Analyzer. This traffic can be generated by application-level test programs (benchmarks, and so forth) or by means of one or more Agilent PCI-X Exercisers.



**Running the Analyzer** There is more than one way to run the PCI-X Analyzer:

- The Run button  and the *Start* item from the *Run* menu start the testcard. The components to be started can be specified in the Run Options window (select *Options* from the *Run* menu):

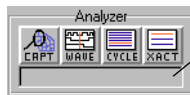


The number of available options depends on the installed software and hardware options.

Note that you can direct the software to clear all pending protocol errors when the testcard is started.

- Clicking the Run button  starts the testcard (*Analyzer and Exerciser*).
- To start the PCI-X Analyzer only:
  - select *Run* from the *Analyzer* menu, or
  - click the Run button  in the waveform viewer, the bus cycle lister or the transaction lister.

**Monitoring the Test** While the test is running, messages appear in the PCI-X Analyzer status bar displaying its current status.




Analyzer Status Bar

**Stopping the Analyzer** When the trigger event occurs, the Analyzer continues to write post-trigger history to the trace memory as specified by the triggerpoint and then stops.

You can stop the PCI-X Analyzer manually as well, for example, if the trigger pattern does not occur or if the test runs over an unexpectedly long time.

There is more than one way to start the PCI-X Analyzer:

- Clicking the Stop button  stops both Exerciser and Analyzer.
- To start the PCI-X Analyzer only:
  - select *Stop* from the *Analyzer* menu, or
  - click the Stop button in the waveform viewer, the bus cycle lister or the transaction lister.



# Viewing And Processing the Trace Memory Capture

**The Three Listers** The PCI-X Analyzer provides three tools to view and evaluate the captured PCI-X traffic at different levels of abstraction:

- **Waveform Viewer**

At the lowest abstraction level, the waveform viewer displays the state of each signal (0 or 1) and the bus conditions (addresses and data on address/data lines, byte enables, and commands on the C/BE lines) at each clock cycle of the capture.

- **Bus Cycle Lister**


The bus cycle lister derives information on the type of each bus cycle and displays a list of the detected signals and states per bus cycle. For example, the state of the C/BE lines during an address phase is evaluated to display the referring command.

- **Transaction Lister**

At the highest abstraction level, the transaction lister shows a list of the transactions found in the captured data. It summarizes the clocks of each transaction in one line and lists certain attributes or parameters that were detected during the transaction (for example, address, waits, retries).

**Synchronizing the Listers** The listers can be used in parallel and be synchronized to view the same range of samples of the capture at the same time (see “*How to Synchronize the Listers*” on page 43).


**Processing the Information** The information shown in the listers can be saved and restored for later analysis (see “*Processing the Captured Data*” on page 48).

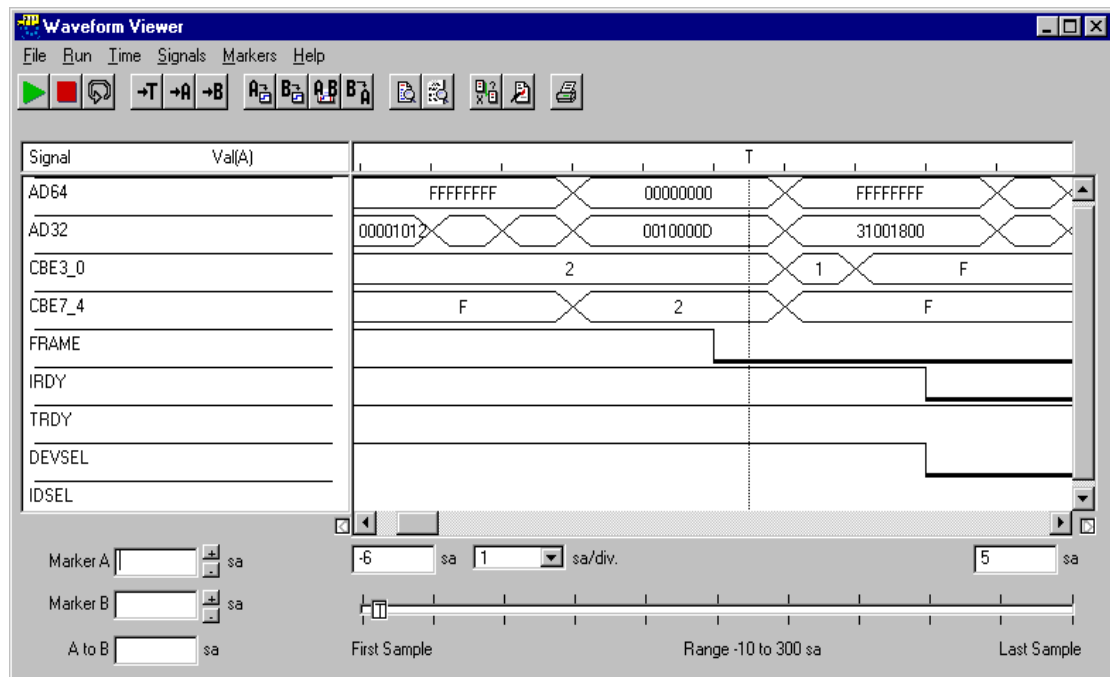
**NOTE** If the captured data is not uploaded from trace memory after the trigger event has occurred (for example, if the testcard has been run in stand-alone mode), click the Reload button  to upload the current capture.

# Using the Waveform Viewer

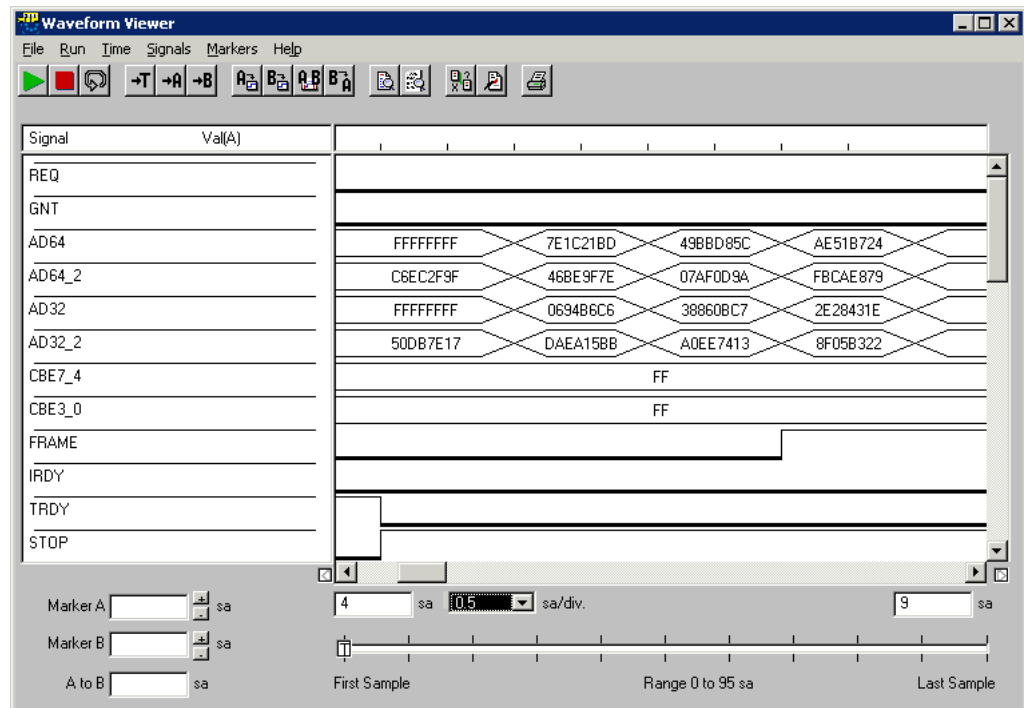
The waveform viewer displays the state of each signal (0 or 1) and the bus conditions (addresses and data on address/data lines, byte enables, and commands on the C/BE lines) at each clock cycle of the capture.

To call the waveform viewer:

- ◆ In the main window, click the Waveform Viewer button .



**PCI-X Mode 2** For traffic captured in PCI-X Mode 2 systems, the waveform viewer looks different in order to reflect the data transferred at double data rate (DDR).



For Agilent E2930 testcards, the extra “signals” AD32\_2 and AD64\_2 are available. These can be used to display the values captured in the second half of a DDR cycle. AD32 and AD64 only display the values in the first half of the cycle.

**NOTE** When the testcard is switched from Mode 2 to Mode 1 (or vice versa) while the Waveform Viewer is displayed, the display might be corrupted. In this case, restart the PCIX GUI.

In both modes, the waveform viewer supports you in analyzing the captured data by allowing you to

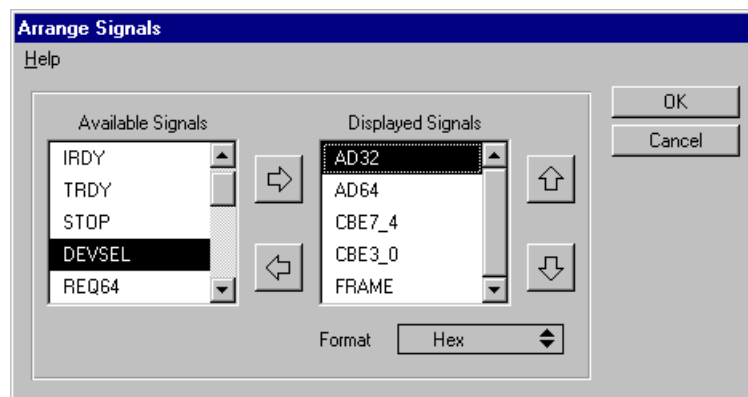
- restrict the display to show only the signals and bus states relevant for your test (see “*How to Arrange the Signal Display*” on page 40),
- improve the view by specifying the range and resolution of the displayed clock cycles (see “*How to Adjust Range and Resolution*” on page 41),
- set markers, for example, to toggle between two locations within the sample in order to compare them (see “*How to Use the Markers*” on page 42).

To analyze the captured data at the different abstraction levels at the same time, you can synchronize the currently displayed listers (see “*How to Synchronize the Listers*” on page 43).

## How to Arrange the Signal Display

To optimize the signal display in the waveform viewer to show only signals relevant for your test, you can exclude all irrelevant signals and arrange the order in which the signals are displayed. Proceed as follows:

- 1 From the *Signals* menu of the waveform viewer, select *Arrange*.



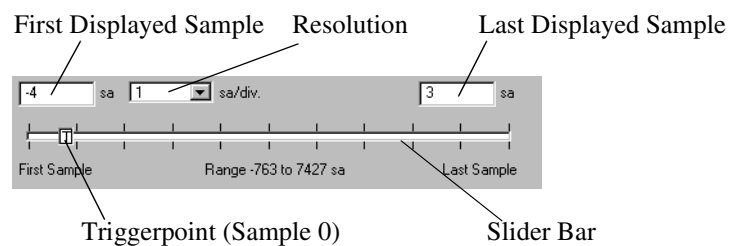
- 2 Click on the horizontal arrow buttons to exclude or include signals (the signals shown in the left area will not be displayed in the lister).
- 3 Click on the vertical arrow buttons to arrange the signals to arrange them as they should appear in the lister.
- 4 For the address/data and byte enable signals, determine whether they are displayed in decimal or hexadecimal format.
- 5 Click *OK*.

**NOTE** Using the *Signals* menu of the waveform viewer, you can also adjust the height of the displayed signals.









## How to Adjust Range and Resolution

Below the signal display the waveform viewer provides information on the currently selected view: first and last displayed sample and zoom factor (resolution). The slider bar represents the complete capture and shows the location of the triggerpoint and of the markers (if set). The samples are always numbered so that the triggerpoint is located at sample 0.



The waveform viewer provides full flexibility to adjust the view to your needs:

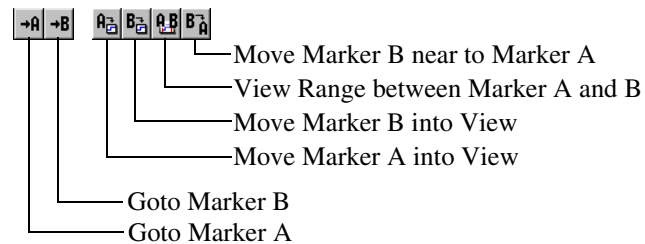
- To enlarge the display, resize the Waveform Viewer window.
- Scroll horizontally so that the area you wish to focus on is in the center of the view.
- Click the Goto Trigger button  to move the view to the triggerpoint.
- Click the Goto Marker A and B buttons   to move the view to the respective marker.
- Click the Zoom buttons   to zoom in or out.
- The resolution shows the number of samples between two division marks (top row of diagram).
- Click the Redraw button  to refresh the display.

## How to Use the Markers

The waveform viewer provides two markers, A and B. With these markers you can

- mark two particular samples of your interest,
- set the visible range in the lister's display,
- set a begin and end marker when using the cross reference function.

The following buttons allow you to control the markers:



**Placing the Markers** To place the markers, proceed as follows:

- 1 Scroll horizontally so that the position for marker A is in the center of the view.
- 2 Click the Move Marker A into View button.  
Marker A is set in the center of the lister's display.
- 3 Repeat these steps for marker B.

**Moving the Markers** To move the markers, you can

- use the sliders A and B.
- drag and drop the marker symbols in the header of the lister's display.
- enter values in *Marker A*, *Marker B*, or *A to B*.
- click the *Move Marker B near to Marker A* button.

**Using the Markers** When you have finished placing the markers A and B, you can

- move the view to A or B by clicking the Goto Marker A or Goto Marker B buttons.
- set the visible range between A and B by clicking the View Range between Marker A and B buttons.


The markers A and B of the waveform viewer are also used by the cross-reference function (see “*How to Synchronize the Listers*” on page 43).

**NOTE** The waveform viewer will display the values of bus signals at Marker A (on the left hand side).

## How to Synchronize the Listers

The displays of all three listers—waveform viewer, bus cycle lister, and transaction lister—can be synchronized by means of the *cross-reference* function. This function is available in any lister. It allows you to view the same samples in all open listers for examination on different abstraction levels.

To synchronize the listers:

- 1 Open the desired listers.
- 2 If no capture is loaded, upload the trace memory (after an Analyzer run) or load a file.
- 3 In one of the listers, select the range to be viewed:
  - in the **transaction lister** and **bus cycle lister**: keep the left mouse button pressed while moving the mouse cursor over the samples
  - in the **waveform viewer**: set markers A and B. The samples between them are considered as “selected”.
- 4 Click the Cross Reference button  to synchronize all open listers.


# Using the Bus Cycle Lister

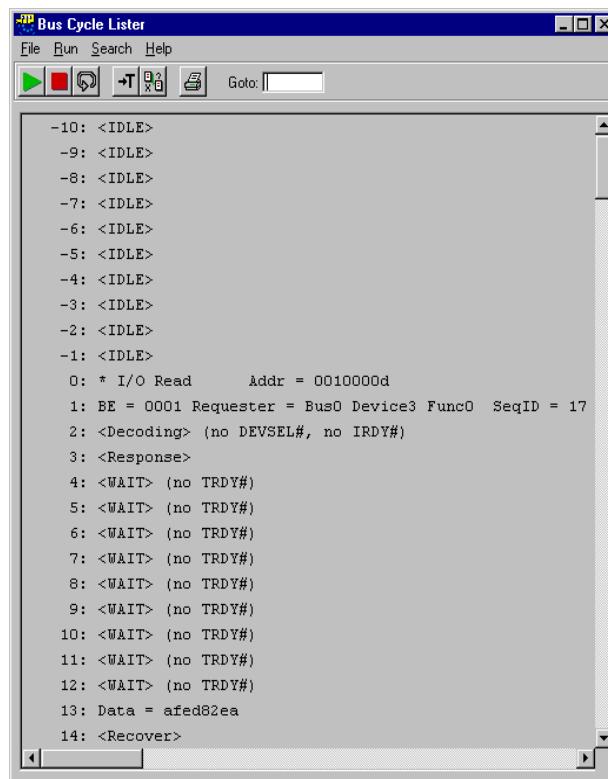
The bus cycle lister derives and displays information according to the type of each bus cycle, for example:

- If it is an address cycle: which is the transferred address and which command is in use?
- If it is a cycle of a data phase within a burst: what is the data and which byte enable signals are set?
- Is it a wait cycle?
- Is it an idle cycle?

The lister provides a search feature allowing you to search for errors and strings in the list and an export feature to store the textual list or parts of it as a text file.

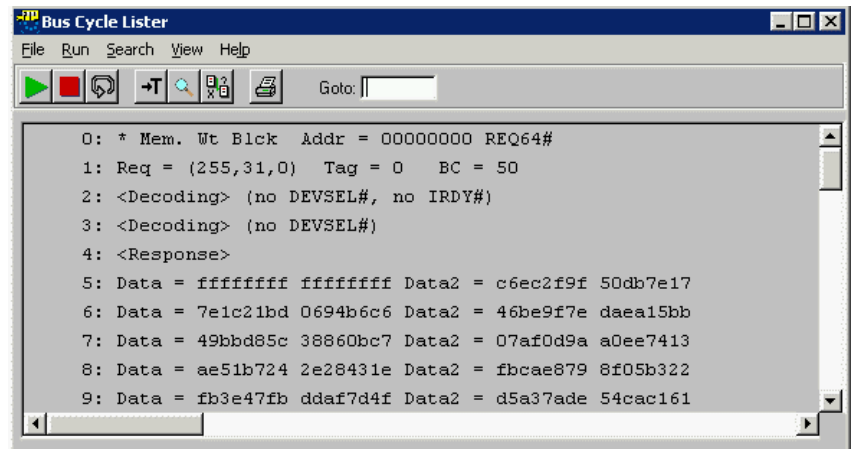
To call the bus cycle lister:

- ◆ In the main window, click the Bus Cycle Lister button .




Each line in the list represents one bus cycle. For each cycle, its type (transaction, idle, etc.) and detected attributes are stated.

**PCI-X Mode 2** For traffic captured in PCI-X Mode 2 systems, the bus cycle lister shows data transferred at double data rate (DDR) in two columns. For each data phase, *Data* shows the data transferred in the first half cycle, *Data2* shows the second half cycle.



## Browsing Through the Cycles

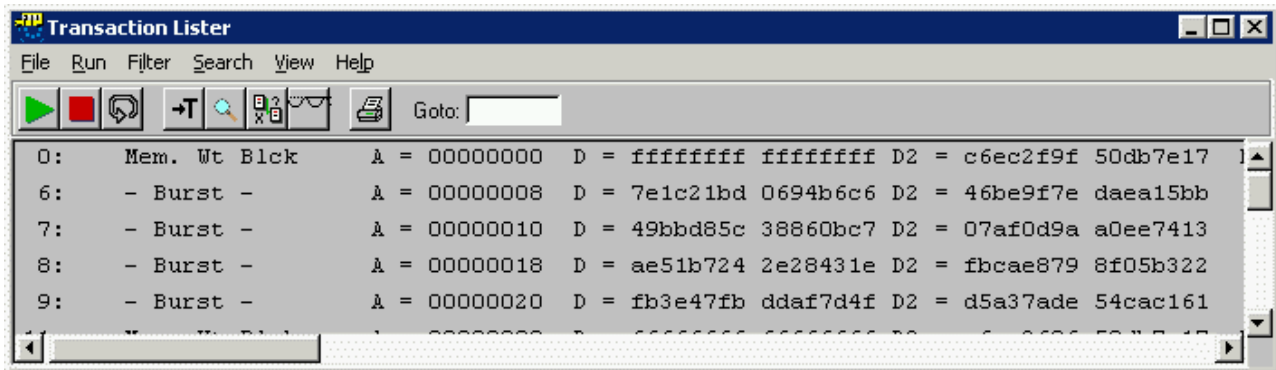
The bus cycle lister supports several methods of browsing through the cycles:

- You can move in the capture using the scroll bars.
- You can move to the triggerpoint by clicking the Goto Trigger button .
- You can move to a particular sample by entering a sample number in the *Goto* text field and pressing return.
- You can use the *Search* menu items to search for strings or errors.

## Using the Transaction Lister

The transaction lister shows a list of the transactions found in the captured data. It summarizes the clocks of each transaction in one line and lists certain attributes or parameters that are detected during the transaction.

**PCI-X Mode 2** For traffic captured in PCI-X Mode 2 systems, the transaction lister shows data transferred at double data rate (DDR) in two columns. For each transaction, *D* shows the data transferred in the first half cycle, *D2* shows the second half cycle.



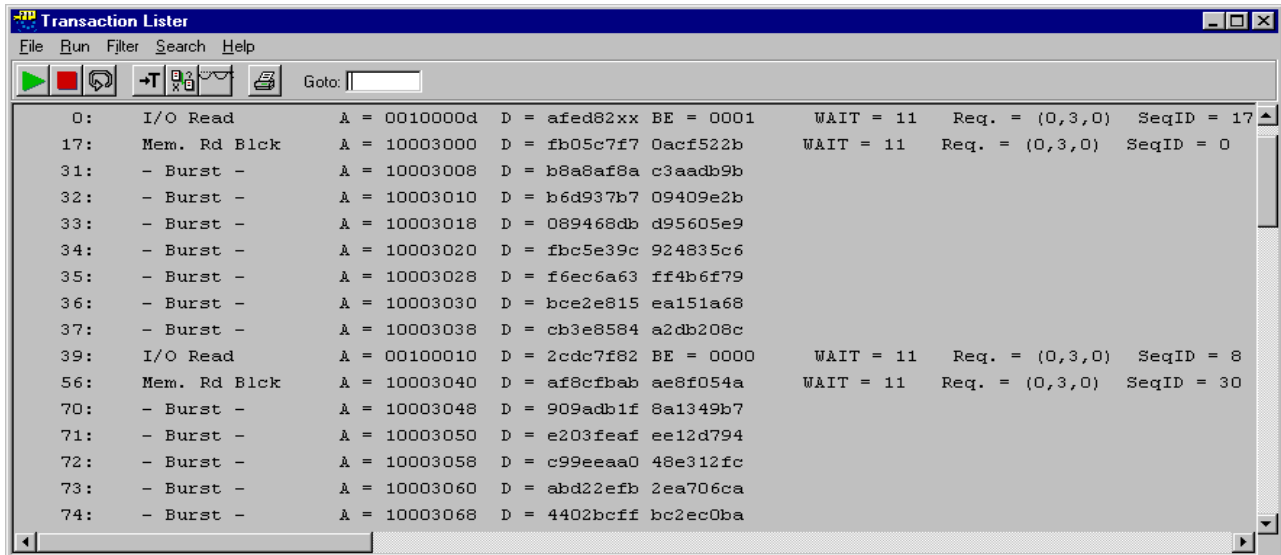
**Searching for Errors** The lister provides a search feature allowing you to search for errors and strings in the list and an export feature to store the textual list as a text file.

**Suppressing Data Phases** If you are not interested in data phases, you can suppress data transfers. In the *Filter* menu, check *Suppress Data Transfers* to horizontally filter out all data transfers. This is useful to remove long bursts.

To view both data and address phases, uncheck *Suppress Data Transfers*.

To call the transaction lister:

- ◆ In the main window, click the Transaction Lister button .



Each line in the list represents a complete transaction and states information about the transaction:

- start sample number
- bus command
- bus address
- data (if any)
- detected behaviors



The transaction lister provides the same scrolling, moving, searching, and storing features as the bus cycle lister. Please refer to “*Browsing Through the Cycles*” on page 45.

# Processing the Captured Data

Normally, the data captured in the trace memory is uploaded automatically after the trigger event has occurred so that you can view it directly in the lists. You can analyze the data, save it to disk and load it again for later analysis.

However, if the PCI-X Analyzer has been run in stand-alone mode or under control of a C program (without using the graphical user interface), you will have to upload the captured data.

The lists provide the following functions for processing the captured data:

- The Reload button  allows you to upload the current capture from the testcard's trace memory into the user interface software.
- The Print button  allows you to print the captured data as currently displayed.
- The *File* menu in each lister provides menu items to save the contents of the trace memory to a waveform file (.wfm) and to load data from a file. A range can be specified to save only a part of the captured data.
- The *File* menu in the Bus Cycle Lister and in the Transaction Lister provides the *Export to File* item, allowing you to save the lists as text files.
- The *File* menu in the Bus Cycle Lister provides the *Export selected Range* item, allowing you to save only the selected range of bus cycles as a text file.



# Measuring Real-Time Performance

In the bring-up and debug phase of a PCI-X device or a system (containing PCI-X bus and PCI-X devices), you need to evaluate the performance of the device or system under test.

The Agilent PCI-X E2920 software supports real-time performance measurement by providing predefined, standardized performance measurements, such as PCI-X efficiency and PCI-X utilization.

These measures can be set up easily. The User Interface for performing real-time performance analysis is available via a separate GUI: the **PCI-X Real Time Performance GUI**.

The performance measurement is based on counting certain events on the PCI-X bus. For the predefined measurements, the counters are set up automatically.

For E2929 testcards, the **PCI-X Performance Optimizer** (option 200) is available. This option expands the possibilities of real-time performance measurements by providing means for detailed post-processed analysis.

## Generating PCI-X Traffic

The PCI-X Analyzer can measure any kind of PCI-X traffic, regardless of how it was generated. However, it is useful to generate traffic in a controlled way for reproducibility in case of troubleshooting or root cause analysis.

Typically, you will use benchmark tests to generate traffic for this purpose.

## Predefined Performance Measures

For real-time performance measurements, the PCI-X Analyzer counts occurrences of predefined events or sequences of events on the PCI-X bus. The results are derived and displayed in real time.

**Available Measurements** The following predefined measures are provided:

- **Throughput**

Throughput is the amount of transferred data per time. It is measured in MByte per second. When running a real-time measurement, this value is displayed in percent of the maximum value.

The maximum value can vary between 133 MByte/s in a 33 MHz/32-bit system, 533 MByte/s in a 66 MHz/64-bit system, 1066 MByte/s in a 133 MHz/64-bit system and up to 2133 MByte/s in a PCI-X 266/64-bit system.

- **Utilization**

Utilization measures the relation between busy bus time and total bus time during a transfer.

- **Efficiency**

Efficiency is a measure of how well the bus is used. It is the most important value when considering PCI-X performance.

The efficiency of a transfer is the relation between the amount of data that was *really* transferred and the amount of data that *could* have been transferred by the used cycles of that transfer (busy clocks).

Efficiency is derived from throughput and utilization. An efficiency near 100 % means that a device made best use of the time it occupied the bus (utilization) by transferring as much data as possible during that time frame (high throughput).

- **Retry Rate**

This is the ratio between transactions terminated by retry and all terminations.

- **Split Rate**

This is the ratio between transactions terminated by split and all terminations.

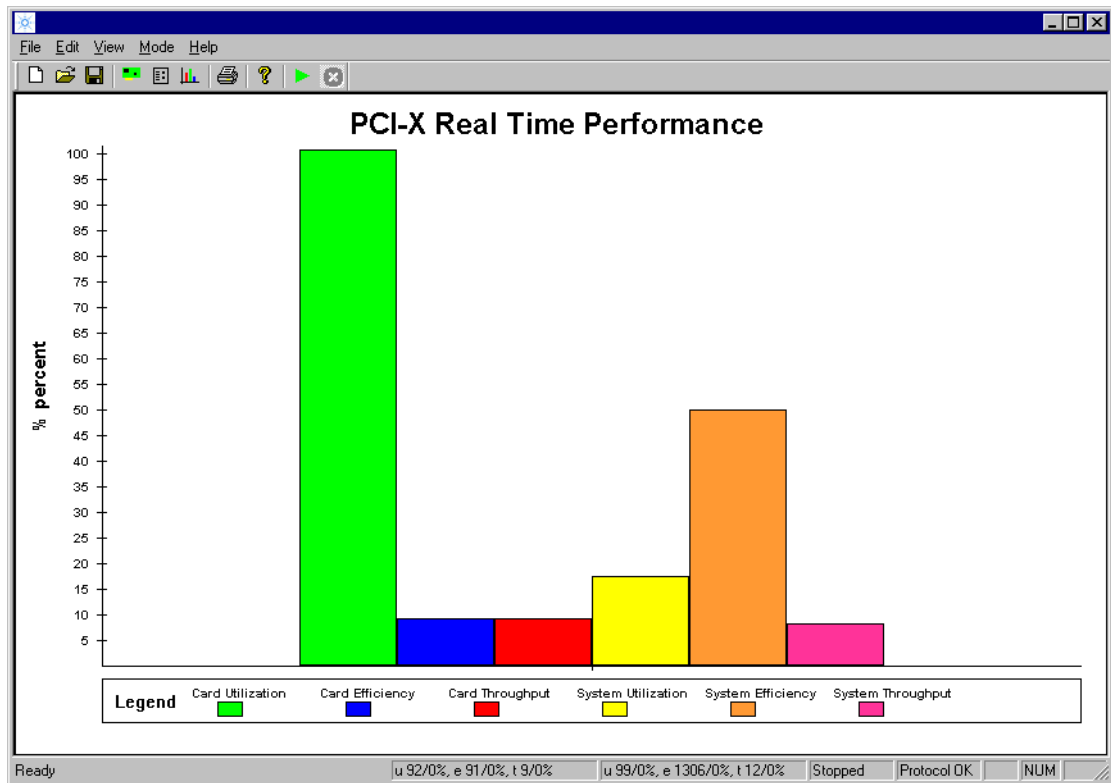
## Running a Real-Time Performance Measurement

The User Interface for performing real-time performance analysis is available via a separate GUI, the PCI-X Real Time Performance GUI.

This GUI can be accessed by selecting:

Start > Programs > Agilent E2920 PCI-X > PCI-X Real Time Performance GUI

The user interface for the real-time performance measurements looks as follows:



**Navigation** The most important features of the real-time performance measurement are available via the buttons in the tool bar. All these features and more are available via the menus, as well.

**Procedure** Running an RTP measurement requires the following steps:

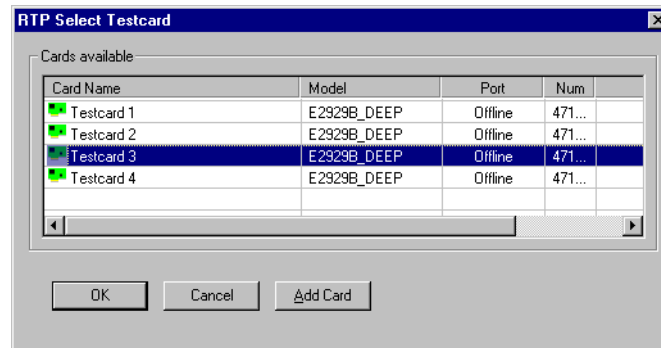
- Selecting a testcard for the measurement
- Setting up the testcard for the measurement
- Selecting the required performance measures
- Setting Run options like update intervals, report generation and chart types
- Running the measurement

## How to Select a Testcard for the Measurement

To select a testcard:

- 1 Select *Select Testcard...* from the *Edit* menu.

This opens the RTP Select Testcard window.



- 2 Click onto the testcard you need for the performance measurement.

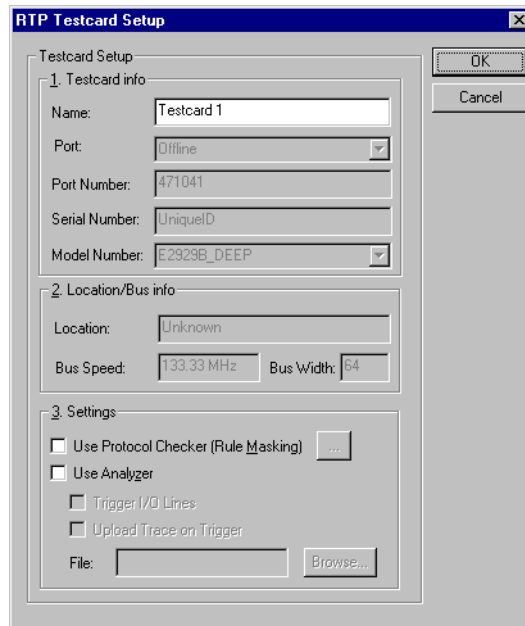
The selected testcard can be specified in the RTP Testcard Setup window.

## How to Set Up the Testcard for the Measurement

To view the settings of the available testcards:

- ◆ Select *Setup Testcard...* from the *Edit* menu.

This opens the RTP Testcard Setup window.



You can modify all current testcard settings under *Settings*. Here you can enable and disable card features (protocol checker and analyzer features).

**Features** The testcard's Analyzer part includes the protocol observer and trigger in/out capabilities.

- **Protocol Checker**

The testcard's protocol checker continuously monitors the bus and checks for violations of predefined protocol rules, which are partly defined by the PCI-X specification and partly by Agilent. Each individual rule can be masked out. In this case, it neither triggers the trace memory, nor appears in the report. To mask rules, click the details button next to the *Use Protocol Checker (Rule Masking)* check box.

- External/Cross Triggering

To facilitate triggering of external measurement devices, and to trigger other testcards in the system for a *snapshot* whenever an error occurs, the testcards are set up to use the external trigger lines that must be connected to reflect their internal triggering state. Whenever the testcard's trace memory triggers, a trigger-out signal is generated. All trigger-in lines are monitored and used to trigger the card's trace memory.

Which trigger-out line is used for triggering is determined by the testcard's bus number. Therefore, only one testcard per bus needs to be used for cross-triggering.

To find out which trigger-out line is used, use the following formula:

`triggerline := bus number MOD 12`

**Example:**

`bus number is 16 -> trigger line is 4;`

`bus number is 5 -> trigger line is 5; ...`

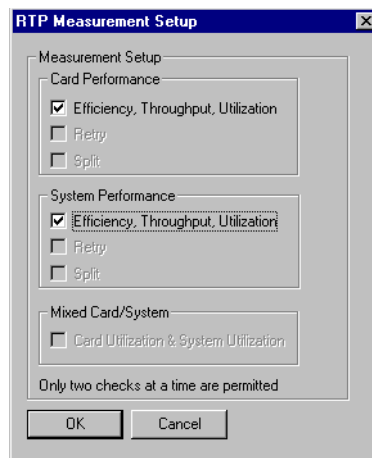
For further information on testcard settings, refer to *RTP Setup Testcard Window* in the *Agilent Agilent E2929/E2930 Windows and Dialog Boxes User Interface Reference*.

## How to Select Predefined Performance Measures

The Agilent E2920 Performance software can calculate two real-time measures simultaneously. The test results are displayed side by side on screen.

To select the predefined performance measures to be used:

- 1 From the *Edit* menu, select *Setup Measurement...*



- 2 Check the predefined measurements to be calculated and displayed for card performance, system performance or both.

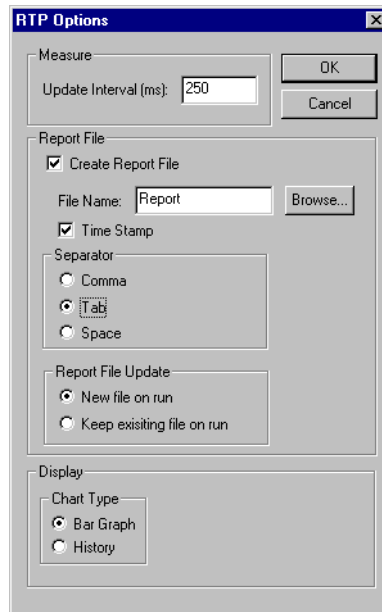
**NOTE** You can only run two measurements at one time.

The settings will be used the next time you start a performance measurement.

## How to Set up Run Options

To set up run options:

- 1 Open the RTP Options dialog box by selecting *Options...* from the *View* menu.



- 2 Set the time interval in which the display is to be updated. You can enter an *Update Interval* between 100 ... 7FFFFFFF\h milliseconds.
- 3 To get a file (\*.rtl) that contains all measurement results, check *Create Report File*. This allows you to specify further report options. For more information on report options, please refer to *RTP Options Dialog Box* in the *Agilent Agilent E2929/E2930 Windows and Dialog Boxes User Interface Reference*.
- 4 Specify how the results are displayed:  
Select the *Chart Type*:
  - *Bar Graph* shows the results of the last measurement time interval.
  - *History* shows the result history. This is useful to detect peaks.




## How to Run a Performance Measurement

After completing the setup, the real-time performance measurement can be started.

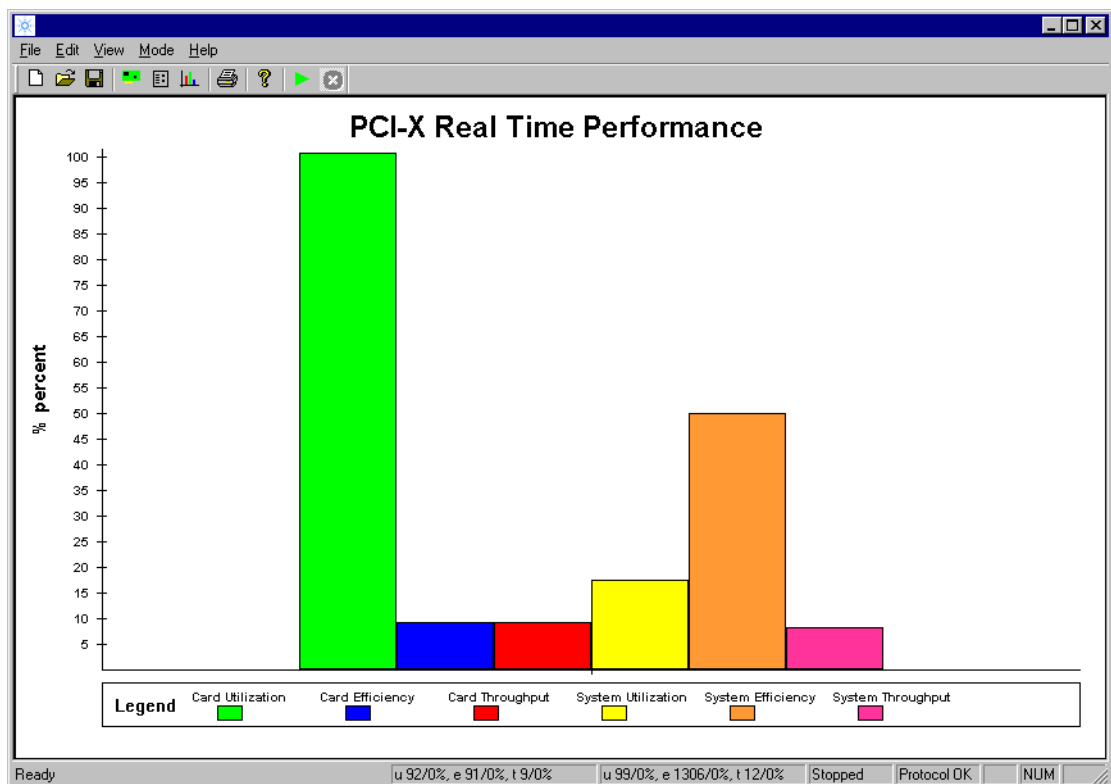
Performance measurements can only be run in online mode. Click *Go Online* in the *Mode* menu, if necessary.

**Running the Test** To run the test configuration, click either

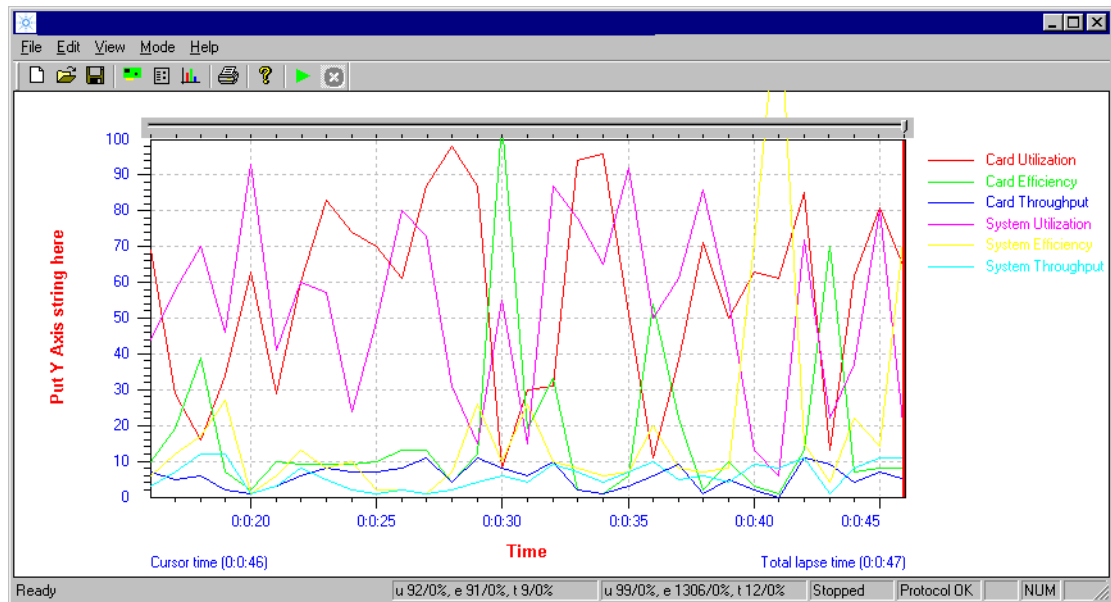
- the *Run* icon  in the tool bar, or
- *Run* in the File menu.

**Display Modes** This shows the results of the selected measurements per time interval.

- If you have selected *Bar Graph* in the RTP Options dialog box, the results are displayed as follows.



- If you have selected *History* in the RTP Options dialog box, the results are displayed as follows.



**Stop the Measurement** To stop the measurement, click the *Stop* button and close the window.

# Index

## A

---

AD32\_2 signal 39  
AD64\_2 signal 39  
Analyzer see PCI-X Analyzer  
Arrange Signals (window) 40

## B

---

Browsing through the Cycles 45  
Bus Cycle Lister  
    Browsing 45  
    Button 44  
    PCI-X Mode 2 45  
    Window 44

## C

---

Capture  
    Button 23  
    Window 12  
Captured Bus Cycles, analysis 16  
Captured Data, processing 48  
Captured Transactions, analysis 17  
Captured Waveforms, analysis 14  
Capturing Data, Setup 22  
Components  
    of the Trace Memory 20  
Cross Reference (button) 43  
ct\_act 28

## D

---

Data Capture  
    Processing 48  
    Running the test 34  
    Setup 22  
    Uploading 22  
Data Recording 20  
Documentation Overview 5

## E

---

Efficiency 50  
Examples See Guided Tour

## F

---

Feedback Counter 30

Format (hex/decimal) in Waveform Viewer 40

## G

---

Goto Trigger (button) 15  
Guided Tour  
    Analyzing PCI-X Traffic from System Memory to the Testcard 10

## L

---

Listers  
    Bus Cycle 44  
    Synchronization 43  
    Transaction 45  
    Types 37

## M

---

Markers  
    Waveform Viewer 42

## N

---

Navigation  
    RTP Main Window 51

## O

---

Offline/Demo Mode (radio button) 9  
Overview  
    Documentation 5

## P

---

Pattern Editor (window) 25  
PCI-X Analyzer  
    Status Bar 35  
PCI-X Mode 2  
    Bus Cycle Lister 45  
    Transaction Lister 46  
    Waveform Viewer 39  
PCI-X Performance  
    Running a Measurement 57  
Performance Measures  
    Predefined 49  
Preload Value 33  
Protocol Checker (RTP) 53

## R

---

Range 41  
Real-Time Performance Measurement  
    Running 57  
    Setup (window) 55  
Recording Data 20  
Reload (button) 48  
Resolution 41  
Retry Rate 50  
ri\_act 29  
RTP  
    see Real-Time Performance Measurement  
Run (button) 34  
Running a Real-Time Performance Measurement 57  
Running the Analyzer 34

## S

---

Selected Transactions/States (Storage Qualifier Setup) 27  
Selecting  
    Predefined Performance Measures 55  
Sequencer  
    Description Table 30  
    Internal 30  
Settings  
    Testcard (RTP) 52  
Setup  
    of the Sample Sequencer 31  
    of the Storage Qualifier 27  
    of the Trigger 23  
Signal Display 40  
Specifying  
    Transactions and States 28  
    Trigger Bus Patterns 25  
Split Rate 50  
State Diagram 32  
States, Specifying 28  
Status Bar 35  
Stop (button) 35  
Stopping the Analyzer 35  
Storage (tab) 27  
Storage Qualifier Setup 27  
Synchronizing the Listers 43

**T**

---

## Testcard

- Cross Triggering (RTP) 54
- External Triggering (RTP) 54
- Protocol Checker (RTP) 53
- Settings (RTP) 52

## Throughput 50

## Trace Memory

- Components 20
- Stored Data 20
- Trigger Sequencer 30

## Transaction Lister

- Button 47
- PCI-X Mode 2 46
- Window 45

## Transactions, Specifying 28

## Transitions 33

## Trigger

- Bus Pattern Specification 25
- Point 25
- Setup 23

## Trigger (tab) 11

## Trigger Out Line 54

## Triggering

- Cross (RTP) 54
- External (RTP) 54

**U**

---

## Update Interval (performance measure) 56

## Uploading

- Captured Data 22

## Utilization 50

**V**

---

## Viewer

- Waveform 38

**W**

---

## Waveform Viewer 38

- Button 38
- PCI-X Mode 2 39
- Window 14

**X**

---

## xact\_cmd 28

## xact\_tran64 29

**Z**

---

## Zoom 41