HEWLETT PACKARD
Expanding Possibilities

HP | E2920 | Verification Tools, PCI Series

HP E2976A System Validation Package User's Guide

March 1999

Important Notice

Authors: Stephan Greisinger and Stefan Kunzi, t3 medien GmbH

Contents

Preface

Preface

The HP E2976A System Validation Package is the Option 310 to your testcard of the HP E2920 Computer Verification Tools PCI Series. In contrast to the other options available for the testcard, the System Validation Package is an independent program that is not part of the HP E2920 framework.

The System Validation Package makes full use of the features of HP's PCI Protocol Permutator & Randomizer (PPR) technology. This software enables you to employ the PPR capabilities without needing to understand the programming of the PPR via its C Application Programming Interface (C-API). The System Validation Package provides a whole range of automated tests for many test purposes.

However, some features of the Graphical User Interface (GUI) are not yet implemented with the current release of this tool. They will be made available with future versions. This manual, too, is not a complete documentation. It does not exhaustively describe all features and details of the System Validation Package, but giving a brief introduction to the different test configurations and data paths that can be evaluated with this tool. It also explains how to set up and run a test and how to interpret the results.

Introduction

Introduction

This chapter provides an introduction to the HP E2976A System Validation Package. It gives an idea of the general use of this software tool and shows how it is best included in your test environment. This chapter also gives an overview of the different data paths that can be tested in your PCI system and the types of tests that can be made with this tool.

Theory of Operation

The HP E2976A System Validation Package is a ready-to-use software package to perform system stress tests during the validation phase of PCs, servers, workstations, or other PCI-based systems. The tool programs and controls multiple HP PCI testcards to create application-realistic system traffic. This approach allows you to set up fully predictable traffic scenarios and provides measurable test coverage and test repeatability.

More specifically, with one single software tool you can

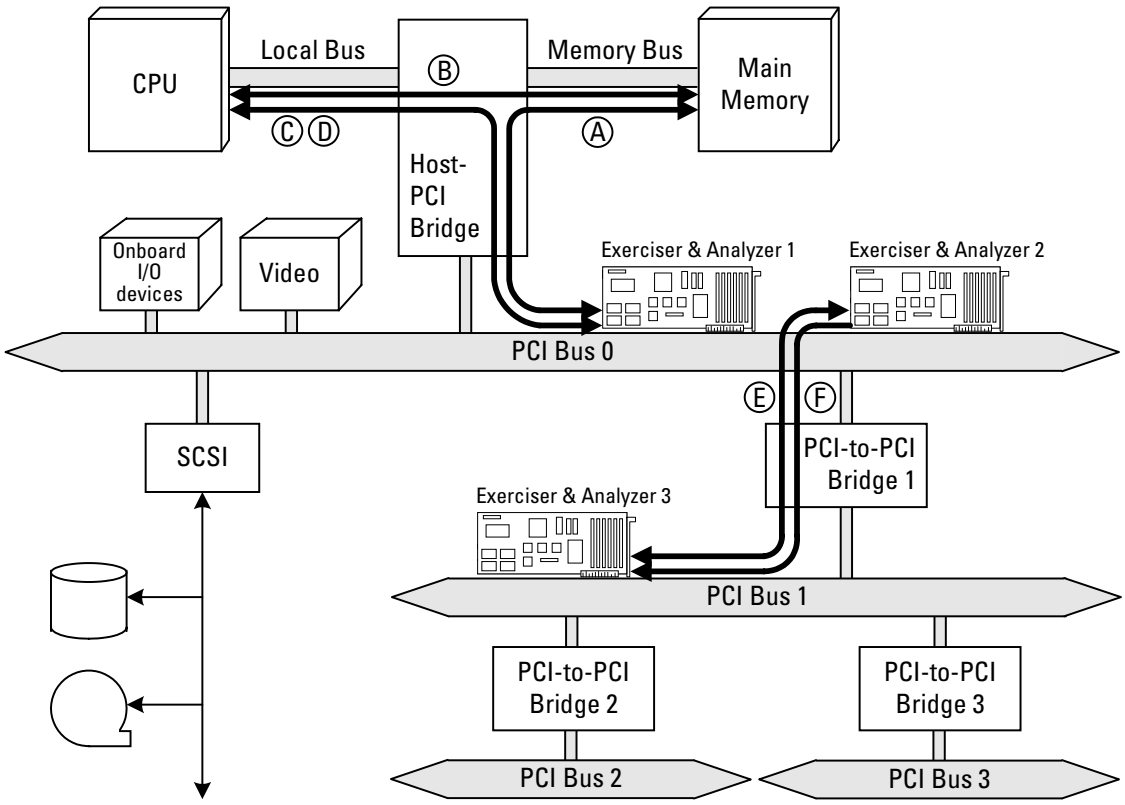
- run multiple tests on multiple testcards simultaneously,
- exercise multiple data paths on your test system simultaneously,
- generate protocol variations to cover all possible system conditions, even the most critical ones,
- provide capabilities for root cause analysis by creating test reports,
- combine these features by running different tests at the same time with several cards to generate realistic and stressful traffic.

Data Paths Overview

The HP PCI testcards can be plugged into any PCI bus of your system under test. Thus, a variety of data paths can be tested. The System Validation Package provides a whole range of test scenarios for testing the different data paths. The letters behind the items of the following list correspond to the data paths indicated in the figure below.

- PCI Exerciser to system memory (A)
- CPU and PCI Exerciser to system memory (A+B)
- CPU to PCI Exerciser memory space (C)
- CPU to PCI Exerciser I/O space (D)
- Peer to peer traffic between two PCI Exerciser cards (E)
- Master to target traffic between two PCI Exerciser cards (F)

The figure shows an example of a PCI machine to illustrate these data paths.



Additionally, the System Validation Package also provides tests that cause the testcard to generate traffic to itself via the PCI bus to increase the bus load, and tests that analyze the performance of the bus.

Test Coverage

Today's test methods for system validation typically take large amounts of time and lack the repeatability of critical system traffic scenarios. The approach usually made today is to simply plug standard PCI cards into the system under test, load it with traffic and wait until an error occurs. Even if running long tests, there still is a high probability that some situations have not occurred on the bus.

The tests available with the System Validation Package enable you to create system critical conditions purposely and in a repeatable way. Furthermore, you can let the test permutate through all thinkable variations of commands, block sizes, etc. to cover all situations that the system can possibly face.

With these tests you can easily cover all traffic conditions for your test system within minutes. The technology providing this coverage is called the PPR technology.

PPR, the Key Technology

Hewlett Packard's Protocol Permutator & Randomizer (PPR) technology allows you to overcome the lack of repeatable test conditions with very high and predictable test coverage.

PPR is a technology which allows the permutation of the PCI protocols and data traffic in a pseudo random way. More specifically, all memory accesses are permutated through all possible combinations of their attributes. This applies for varying block sizes, the use of the different memory commands write, read, write invalidate, read line, and read multiple. Furthermore, permutations are made in terms of the alignments and byte enables. This means, that all variations of byte, word, and dword accesses are used.

The protocol variations also test with different protocol attributes. This includes that the transactions are performed with all possible

- wait states inserted by both the Exerciser's master and target
- transaction terminations by the target (except for target abort)
- both 64-bit and 32-bit accesses attempted by the master
- both acceptance and non-acceptance of 64-bit accesses by the target.

Thus, not only critical test patterns can be transferred between different system components, they are also automatically permutated to emulate all thinkable traffic scenarios.

For more information on how the PCI Protocol Permutator & Randomizer works, please refer to the *HP E2975A PCI Protocol Permutator & Randomizer Software User's Guide*.

Installation

Installation

This chapter covers all information needed to properly install and run the HP E2976A System Validation Package. Furthermore, it provides a summary of the possible hardware configurations. Also, some recommendations are included for these configurations to make best use of the System Validation Package.

The software must be installed on the system under test. Tests cannot be run remotely as with the HP E2920 Graphical User Interface.

The minimum requirements to your test system for running the program are:

- Windows NT 4.0 or higher.
- At least 10 MB of free hard disk space.
- One System Validation Package license per system under test.
- At least one HP PCI Exerciser and Analyzer card.

Installing the Software

There are two different ways that you may have received the HP E2976A System Validation Package. If you installed it from the CD together with the HP E2920 software, it is already installed on your computer and you can find the executable file SystemValidate.exe in the same program folder. If you want to solely install the System Validation Package, you can do so by using the self extracting zip file named InstallSVP.exe in the directory SystemValidate on the installation CD.

To unzip this file and install the software:

1. Open the Windows Explorer, locate the zip file InstallSVP.exe in the directory SystemValidate and double-click it.

   You are then asked in which directory you want to store the software.

2. Select a directory and click *Unzip*.

   If this directory does not yet exist, the software will create it for you and save all unpacked files into this directory. The WinZip Self-Extractor window closes and the batch file installntdrv.bat starts running (unless you unchecked this option in WinZip).

3. When the driver installation is finished, press any key and close the DOS-Window.

4. Start the HP E2976A System Validation Package software by double-clicking SystemValidate.exe in the Windows Explorer.

   The first time you are running the program, a message box appears telling that the software was unable to open the license file. The three options are to start the program in demo mode, to enter the license, or to quit by clicking the *Cancel* button.

5. Click *Enter License* and then enter the license key that was shipped with the software.

6. For quick and easy access to the System Validation Package, you can create a shortcut to it and place it on your desktop. For this purpose, click on the file SystemValidate.exe (in the Windows Explorer) with the right mouse button and select *Create Shortcut*. Then drag the shortcut that you just created in this directory to your desktop and drop it there.

Possible and Recommended Hardware Configurations

The HP E2976A System Validation Package supports the following PCI Exerciser and Analyzer cards:

- HP E2925B
- HP E2926B
- HP E2928A
- HP E2940A
- HP E2926A
- HP E2927A

The precondition for the use of the HP E2976A System Validation Package with any of these testcards is that the Exerciser option is enabled (option #300 or #600).

Several testcards can be placed on the same PCI bus to increase traffic from different devices or to test different data paths or devices at the same time. You can also plug several testcards on different buses in the system under test, for example, to test the bridges between these buses.

Recommended Configuration

For every system under test, the recommended configuration is to have:

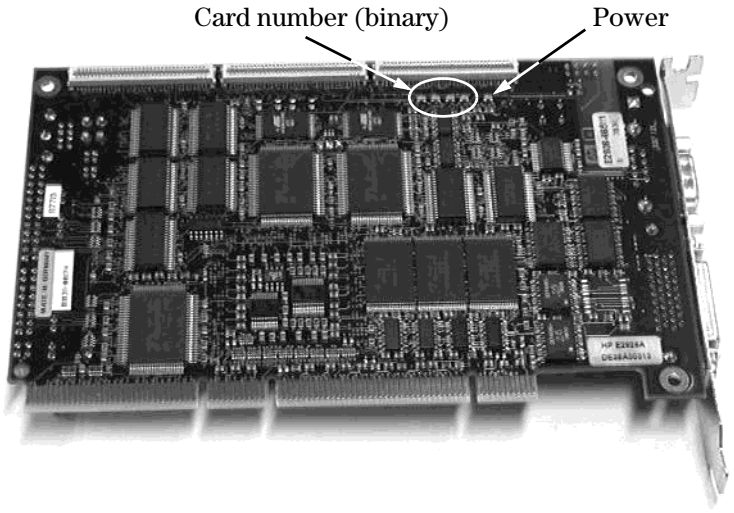
- One HP E2976A System Validation Package license.
- One PCI Exerciser and Analyzer card for each PCI bus.
- One additional PCI Exerciser and Analyzer card for a peer-to-peer test on one bus.
- One HP E2971A PCI Exerciser Graphical User Interface.

It is recommended that you run the Analyzer and Exerciser GUI on a remote machine. For example, if the tested bus hangs, you still can access the trace memory of the testcard from the external GUI.

Identification of Card Numbers

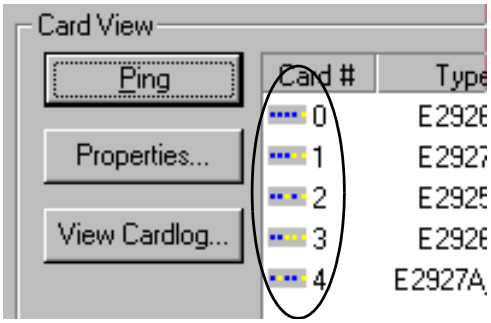
In case you have several HP testcards plugged into your system under test, the System Validation Package assigns numbers to them in the GUI to tell

them apart. These numbers can be identified on the cards as well, coded in a row of five green LEDs along the top edge of the back of the card.



Power LED The LED at the rear end of the row is slightly further separated than the others. It is always lit, when the card is powered.

Card Number LEDs The other four LEDs code the card number with a binary number, where the last LED represents the least significant bit. The pattern of the LEDs is also displayed in the GUI next to the card number for easier identification.



NOTE This applies for all PCI testcards, except the HP E2940A CompactPCI testcard. On the CompactPCI testcard the five LEDs are located equally spaced in a vertical row.

The Available Tests

The Available Tests

This chapter contains information on the different test actions that are provided with the HP E2976A System Validation Package. It briefly explains, how the tests work, which data paths are involved and what requirements must be fulfilled.

General Test Descriptions and Requirements

When running a test, the System Validation Package emulates typical traffic scenarios on a PCI system. For example, data transfer from the CPU to the SCSI card, LAN to LAN card traffic, concurrent system memory access from LAN card and CPU. So far, these are typical scenarios that are today generated in so-called hot mockup tests (plugging all cards into the system and generate random traffic).

With the System Validation Package, the system validation process is significantly enhanced by:

- Increased test coverage by increasing the number of variations of system traffic.
- Providing programmable tests to force the most critical conditions for the system.
- Providing repeatable tests for failure analysis and failure regression tasks.
- Providing comparability of test results for performance evaluation and further system improvements.
- Producing report files for the detection of problems on the bus immediately before the system hangs.
- Creating test reports to document system quality.
- Making an easy link to R&D's debug environment.

Basic Test Structure

All tests that use memory accesses to emulate data traffic work with the same structure:

1. Initializing an internal Exerciser data memory with random data.
2. Writing this data to the destination memory.
3. Reading this data back from the destination memory.
4. Comparing the read data with the initial data field
5. Reporting a compare error if detected.

Besides compare errors, the tests will also be terminated on the detection of protocol errors, timing errors, and bus hangs.

Furthermore, you can use cross-triggering with these tests. Cross-triggering means, that you connect the trigger I/O ports of two testcards. If a trigger event occurs on one testcard, the other is triggered, too. This is very useful, for example, to capture the data traffic on two buses at the

same time, because problems on one bus may have their root cause on another bus.

This basic test structure applies for all tests available with the System Validation Package except for the tests “Analyzer Only” and “Configuration Scan”. For details on the different tests, see the respective section.

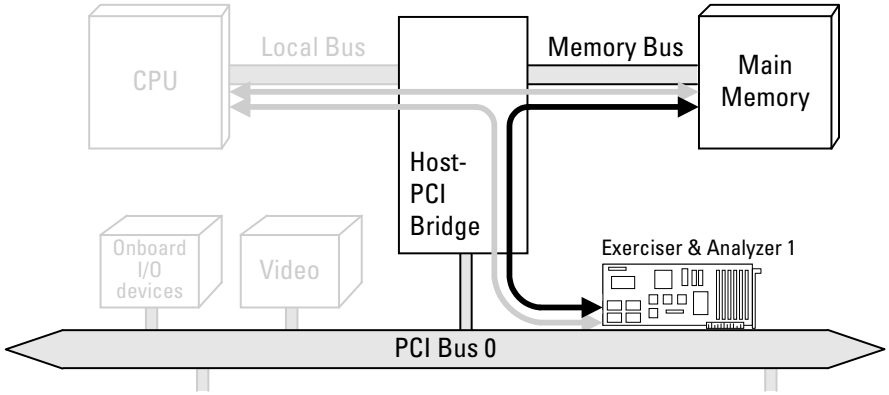
NOTE The following descriptions of the tests are also found in the Add/Edit System Action dialog box in the GUI where you select your test actions.

PCI Exerciser to System Memory

This test accesses the system memory from the PCI bus. This is implemented by the testcard’s Exerciser being set up as a master and sending different write and read commands.

Tested Data Path The PCI bus from the PCI Exerciser to the host/PCI bridge and the system memory bus from the host/PCI bridge to the system memory.

Tested Devices The host/PCI bridge, the host/PCI bridge configuration, the host memory controller, and the arbitration unit.



CPU and PCI Exerciser to System Memory

This test accesses the system memory space on two data paths at the same time:

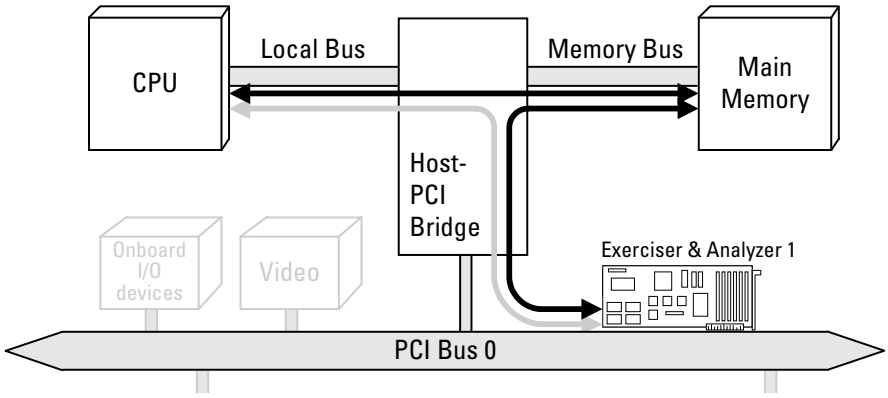
- The CPU accesses system memory through the memory controller.
- The PCI Exerciser master accesses system memory through the Host/PCI bridge.

Both devices perform different read and write commands to system memory. They access the same 4-KB memory page with each device allocated half the page in order to stress the cache controller.

This test can also be made with harder constraints (see the test below).

Tested Data Path The CPU local bus, the PCI bus from the PCI Exerciser to the host/PCI bridge and the system memory bus from the host/PCI bridge to the system memory.

Tested Devices The host/PCI bridge, the host/PCI bridge configuration, the host memory controller, and the arbitration unit.



CPU and PCI Exerciser to System Memory (hard)

This test accesses the system memory space on two data paths at the same time:

- The CPU accesses system memory through the memory controller.
- The PCI Exerciser master accesses system memory through the Host/PCI bridge.

Both devices perform different read and write commands to system memory. They access the same 4-KB memory page with each device allocated half the page in order to stress the cache controller.

The difference to the test above is that this test uses harder constraints. It does not insert any wait states and uses longer bursts.

Tested Data Path The CPU local bus, the PCI bus from the PCI Exerciser to the host/PCI bridge and the system memory bus from the host/PCI bridge to the system memory.

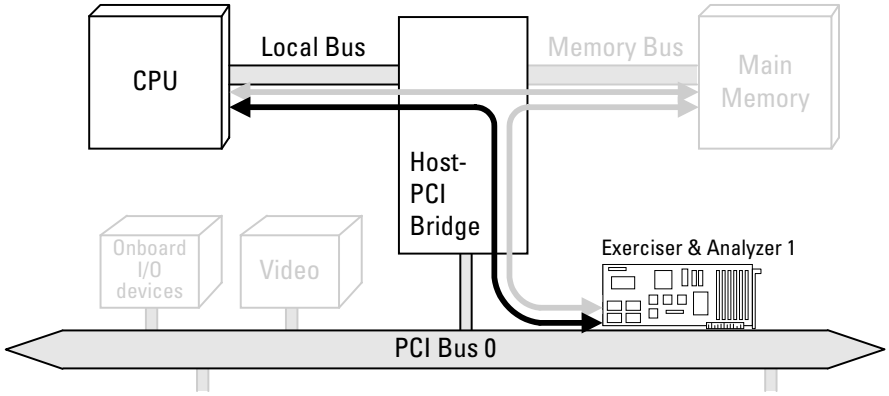
Tested Devices The host/PCI bridge, the host/PCI bridge configuration, the host memory controller, and the arbitration unit.

CPU to PCI Exerciser Memory Space

This test accesses the memory space of the PCI Exerciser from the CPU. It uses different memory read and memory write commands.

Tested Data Path The CPU local bus and the PCI bus from the host/PCI bridge to the PCI Exerciser.

Tested Devices The host/PCI bridge and the host memory controller.



CPU to PCI Exerciser I/O Space

This test accesses the I/O space of the PCI Exerciser from the CPU via a virtual memory buffer. It uses the same data path as the test above, but employs the I/O read and I/O write commands only.

Tested Data Path The CPU (I/O access) local bus and the PCI bus from the host/PCI bridge to the PCI Exerciser I/O port.

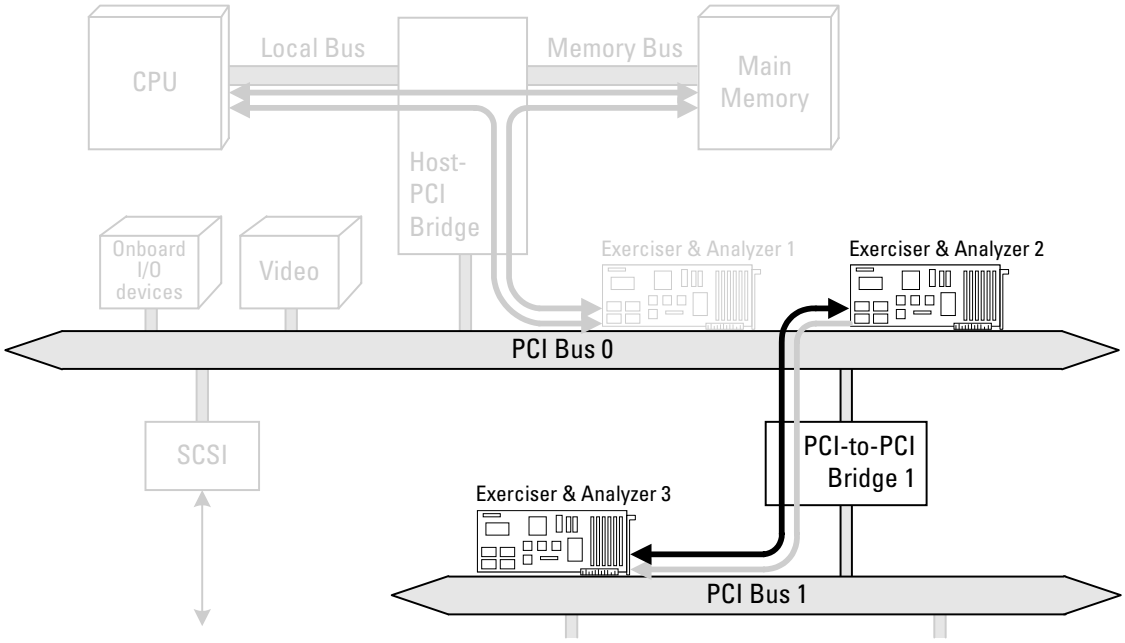
Tested Devices The host/PCI bridge and the host memory controller.

Peer to Peer Test

This test requires two PCI Exerciser cards, that are set up to access each other's memory space. This is implemented with master to target traffic in both directions. Use the testcards on different buses to test the PCI/PCI bridge(s) between them.

Tested Data Path The PCI bus(es) from Exerciser #1 through the PCI/PCI bridge(s) to the Exerciser #2.

Tested Devices The PCI/PCI bridge(s), the PCI/PCI bridge configuration(s), and the arbitration unit(s).



Master to Target Test

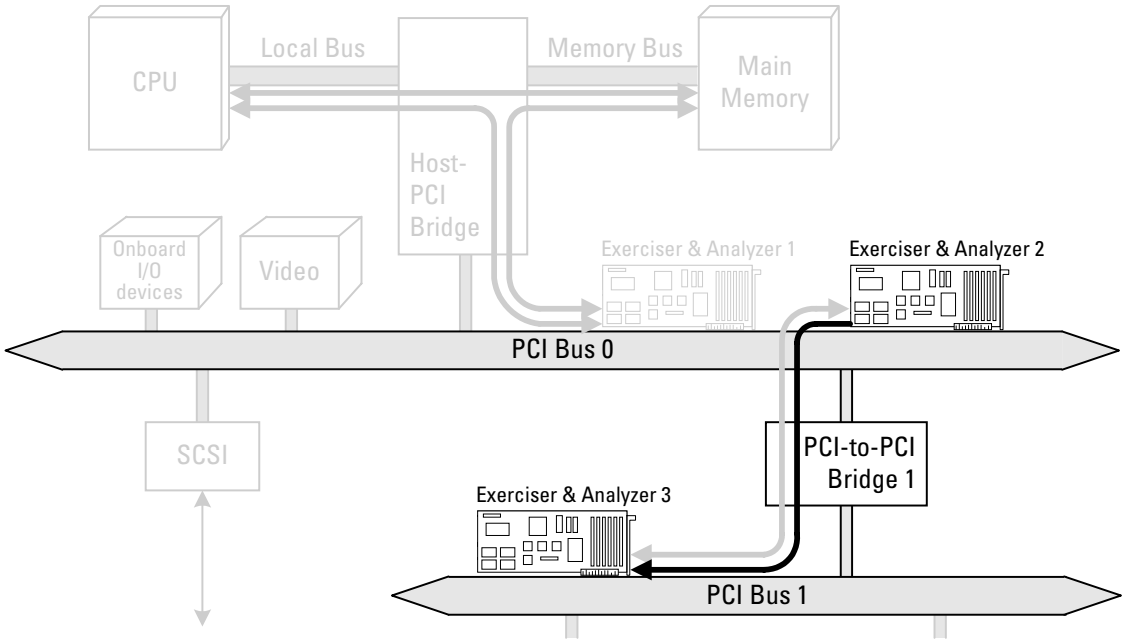
This test requires two PCI Exerciser testcards. One testcard accesses the other card's memory space. This means master to target traffic in one direction only. Use the testcards on different buses to test the PCI/PCI bridges in between.

This test can be very helpful, for example, if you have problems with a PCI/PCI bridge, and the peer-to-peer test did not pass. Then you can use this test to check both directions separately.

Tested Data Path The PCI bus(es) from Exerciser #1 (master) through the PCI/PCI bridge(s) to the Exerciser #2 (target).

Tested Devices The PCI/PCI bridge(s), the PCI/PCI bridge configuration(s), and the arbitration unit(s).

NOTE The first testcard in the list in the Add/Edit System Action dialog box will be the master, the second will be the target. Use the manual mode to assign the testcards appropriately for your test.



Bus Load Generator

With this test the PCI Exerciser simply generates traffic from its master to its own target via the PCI bus. This self-traffic is used to stress the bus with additional bus load.

Tested Data Path The PCI bus into which the Exerciser is plugged.

Tested Devices The arbitration unit and other devices on the same bus.

Analyzer Only

This test is not driving any transactions on the PCI bus. Instead, all PCI devices on the bus are observed passively, not only a single device. For this purpose the PCI Analyzer is set up to

- check for PCI protocol violations,
- check for PCI timing violations,
- check for Data transfer errors (compare errors),
- check for Parity errors,
- check for Bus hang-ups and bus locks,
- and to perform bus load measurements.

The detected problems are logged in the log file. Optionally, a PCI trace memory waveform file can be generated for in-depth root cause analysis.

PCI Configuration Scan

This test is not driving any transactions on the PCI bus. Instead, it is scanning the whole configuration space of the bus (using the command config_read). This configuration space report—stored in the log file—allows a proper documentation of the test conditions during the test run.

Because the configuration space may change with each system reboot, this is an incredible help when trying to identify errors that only occur sporadically.

Recommendations on Test Duration

The recommended duration of a system stress test strongly depends on the type of the test and even more on the other devices that communicate simultaneously on the system. Hence, it is very difficult to do exact estimations on test durations.

Only System Validation Package Traffic

If you are running a single test with the System Validation Package without any other traffic being on the PCI bus, the PPR permutates through all possible combinations within about one minute on 33 MHz buses and about 30 seconds on 66 MHz systems.

If you are running two or more tests of this software on the same bus, the completion of the tests is delayed by the respective factor.

Traffic Caused by Multiple Devices

In contrast, if you are testing a system that has other traffic on the bus at the same time, you cannot predict when all possible test conditions have occurred. Although you know the behavior of the System Validation Package, you probably do not know about the other devices' behavior. These devices might behave "friendly" most of the time. And in many cases you cannot even make this device behave the most critical way for testing purposes.

If you want to find out whether your system under test can stand these worst-case conditions, or if you want to measure the performance for this case, you have to run much longer tests to reach a high probability that all cases occurred.

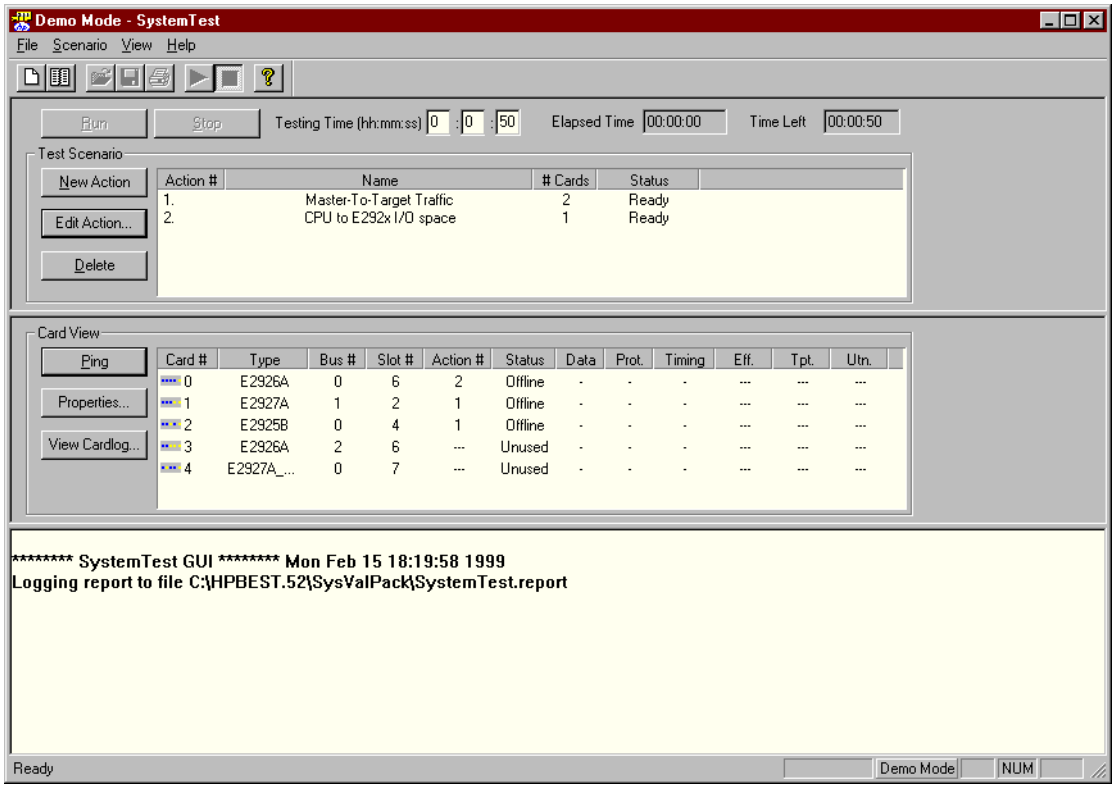
Test Guidelines

Test Guidelines

This chapter contains the guidelines for setting up and running a system test with the HP E2976A System Validation Package. Within these steps you also learn about the major features of the GUI. The information included in this chapter should then enable you to set up your own tests that match your personal needs.

Test Setup

This section shows the major steps included when setting up PCI system tests with the HP E2976A System Validation Package. Additionally, there is also some information on the different components of the Graphical User Interface (GUI) found here.



Mainly, setting up a system test with the software's Graphical User Interface consists of the following parts:

1 Run the System Validation Package.

The software automatically scans all connected PCI buses for HP Exerciser and Analyzer cards and initializes them. Cards that could not be initialized are not available for tests. This applies for example for cards that are currently connected to the HP E2920 GUI, or cards that do not have both a memory space and an I/O space assigned.

In the latter case, the software offers to reset these testcards to factory default settings that include such memory and I/O spaces. However, these testcards will still not be available for tests until the system is rebooted.

If the program cannot find any testcard usable for its tests, it switches to demo mode and informs you with a message box.

2 Check or modify the properties of the available testcards.

This step is more or less optional. Basically you can run any test with the default settings. In some cases, however, it is quite helpful to make use of the features provided here. See "*Checking Card Properties" on page 32* for details.

3 Select the test actions.

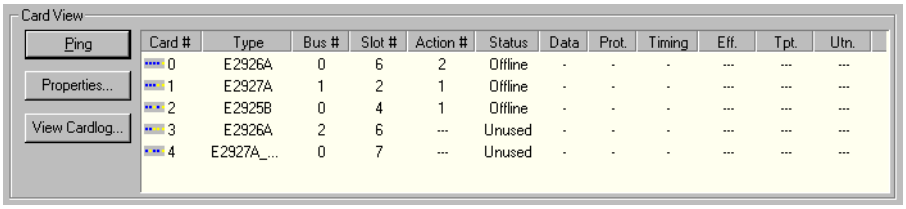
In the *Test Scenario* area—also called action view—you see all the currently specified test actions and their status. To run a test you need to set up at least one test action. See "*Selecting Test Actions" on page 34* for details.

4 Run the test scenario.

After you have set up your test scenario, you need to specify, how long your test should run. Then start the test by clicking the Run button. See "*Running a Test" on page 35* for details.

Checking Card Properties

The middle part of the System Validation Pack's GUI is titled *Card View*. Here you see a list of all testcards found in the system under test while the program got started.

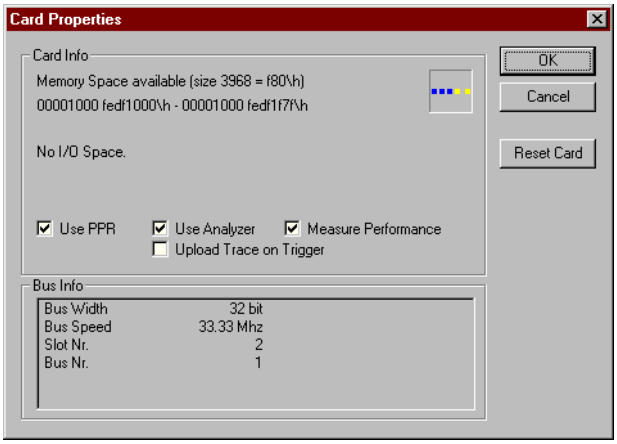


This list provides information about every card's number, its type, its location (bus and slot), and the test action, it is currently used for, along with status information. The explanation of the status information is found in "*Running a Test" on page 35*.

To check or modify additional properties of the testcards, there are three buttons to the left of the testcard list.

Ping Button Clicking this button merely checks whether the connection to the testcard works properly. It opens a connection to the card. Then it calls a ping function, which causes the red LED on the card to flash, and closes the connection again.

Properties Button Clicking this button opens the Card Properties dialog box.



This dialog box contains information both on the testcard and on the PCI bus it is plugged into. For the *Card Info* this includes the address range of the memory space and the I/O space together with the four check boxes:

- *Use PPR*. If unchecked, default attributes are used for transactions. This means no permutations are made. The PCI Exerciser uses always the same block size, does not switch through different bus commands and more.
- *Use Analyzer*. The Analyzer detects errors on the PCI bus and can capture the data traffic in its trace memory. If unchecked, no traffic is captured. The bus observer still will run, but it is not queried by the CPU.
- *Measure Performance*. If unchecked, the PCI performance counters do not run. As a result the performance values efficiency, throughput and utilization are not displayed in the card view during test run.

  It is not necessary to measure performance with more than one testcard per bus. Use of these PCI measurements causes a lot of non-permutated traffic on the bus. Hence, to reduce this overhead, choose the option “Measure Performance” for only one testcard per bus.
- *Upload trace on trigger*. If this option is checked the System Validation Package writes the trace memory from the testcard to a waveform file (wfm file) after the test is triggered. This option usually is not recommended, because it can take very long to complete.

Reset Button

You can also reset a testcard to factory default settings by clicking the *Reset* button. This reset will not come into effect until the test system is rebootet.

View Cardlog Button

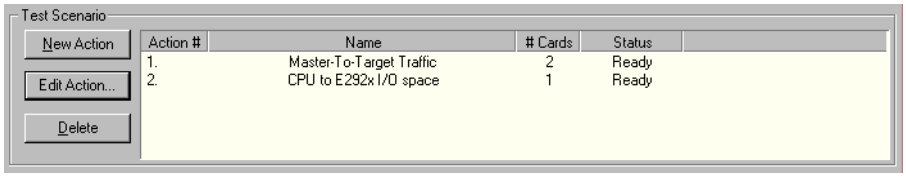
Clicking this button in the *Card View* opens the View Cardlog dialog box. This dialog box is used to inform about the connection established to the testcard. Especially if errors occur on the testcard the respective function that noticed the error will be listet here.

For more information on the handling of errornous testcards please refer to the C-API description in the user’s guide of the respective card.

If the card error was detected by a function of the PPR—the function's name starts with B_E_PPR—then the reference is found in the *HP E2975A PCI Protocol Permutator & Randomizer Software User's Guide*.

Selecting Test Actions

The upper part of the System Validation Pack's GUI is called the *Test Scenario* view and includes the action table containing all specified test actions within the test scenario.



In this table you can find information on the action number, its name, the number of included testcards and the status. The meaning of the different items in the status column is explained in "*Running a Test*" *on page 35*.

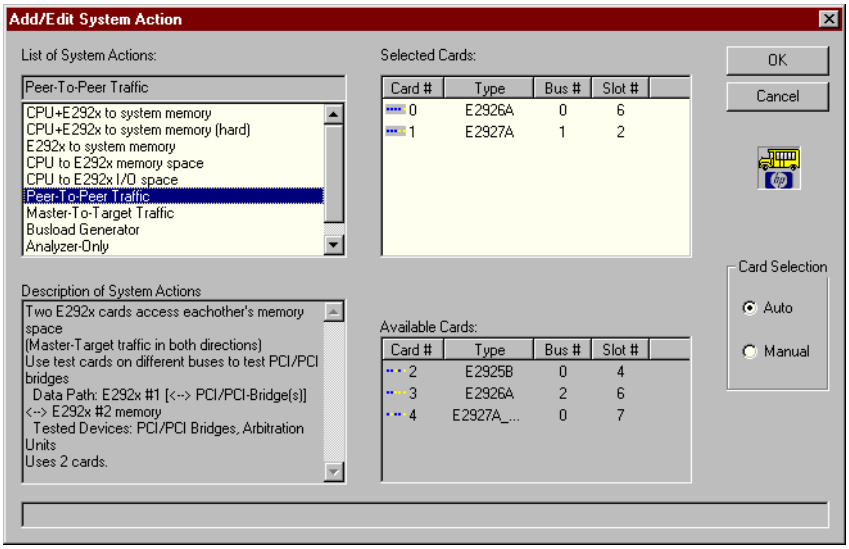
New Action New Action button. Clicking this button opens the Add/Edit System Action dialog box, where you can specify a new action.

Edit Action To edit/modify an existing action, select it by clicking on its action number and then click the *Edit Action* button. This opens the Add/Edit System Action dialog box.

Delete Action To delete a selected action from the action table, click the *Delete* button.

Add/Edit System Action Dialog Box

This dialog box is used to add or modify a test action in your scenario.



In order to add a new test action or to modify an existing one, do the following steps:

1 Select a test from the *List of System Actions* (top left) by clicking on it.

2 Check the *Description of System Actions* (bottom left) if this test meets your requirements.

3 Use the radio buttons to the right to select whether you want your *Card Selection* being done automatically (*Auto*) or manually (*manual*).

   – If you chose the testcards to be selected automatically, you see the suggestions in the *Selected Cards* list (top right).

   – If you chose to select the testcards yourself instead, use the arrow buttons to move highlighted cards from the *Available Cards* list (bottom right) to the *Selected Cards* list and back. To highlight a card click on its card number.

4 Click *OK* to close the dialog. If the *OK* button is grayed out, the number of selected cards does not meet the selected test's requirements. Check the *Description of System Actions* again.

Running a Test

After you have successfully set up your test scenario, start your test by doing the following:

1 Specify test time.

   In the center of the top row of the GUI you see the specified testing time in the format (hh:mm:ss). The default is 50 seconds.

   Change this value to your preferred test duration.

2 Click Run

   Click the large *Run* button in the upper left corner of the GUI, or the Run button in the tool bar. Alternatively, you can also select *Run* from the *Scenario* menu or press *F5* on your keyboard.

When the test scenario is started, you can view the status of the test actions and the testcards in the *Status* column of the respective list.

Test Action Status

The different items that can occur in the *Status* column of the actions list are:

- *INITIALIZING*. The specified test is currently being initialized.
- *READY*. The test is initialized and waiting for the start signal.
- *RUNNING*. The test is running.
- *TERMINATING*. The test is about to finish, for example, while saving the trace memory to a file.
- *FINISHED*. The test has finished without any errors.
- *ERROR*. The test has stopped due to a detected error.

Testcard Status

The different items that can occur in the testcard's *Status* column are:

- ERROR. A problem with the testcard itself occurred. This card is not available for any test scenarios. See details in the Card Log dialog box.
- EXCLUDED. This testcard is not available for any test scenarios. Possible reasons are lacking resources on the testcard or a connection between this testcard and the Best GUI.
- UNUSED. This testcard is available for test selections, but currently not selected for a test.
- OFFLINE. This tescard is selected for a test scenario, but the test has not yet started (waiting for a call to function BestOpen).
- IDLE. The test on this testcard has started (BestOpen called), the test is currently initialized, but no action has been performed yet.
- BUSY. This testcard is busy with the selected test.

Error Types

The *Card View* in the GUI additionally provides information on the type of errors that are detected. The columns *Data*, *Prot.*, and *Timing* indicate if a data compare error, a protocol error, or a timing error respectively caused the trigger of the Analyzer to terminate the test action. A minus sign (-) indicates, that no error of this type occurred, while an error is indicated by an exclamation mark (*!*). Errors are also reported in the log file.

If the abbreviation *n/a* ("not available") is displayed in the *Timing* column, this testcard is not calibrated. Its timing checker is not available, and no timing errors will be detected by this testcard. To calibrate the testcard, you need to send it to the manufacturer.

Performance

You can also view the performance of the testcards during the test run. The performance is displayed in terms of the efficiency (Eff.), the data throughput (Tpt.), and the bus utilization (Utn.). All these values are presented for information during runtime, but are not stored in any report file. Definitions of these measures are found in the section "Predefined Performance Measures" in the user's guide of any of the testcards.

Test Results

Test Results

This chapter briefly explains the results of the system tests performed with the HP E2976A System Validation Package. It covers the contents of the different output files as well as some basic error handling instructions.

For information on

- the log file, refer to the "*Log File Description" on page 39*.
- the PPR report files, refer to the "*Report File Description" on page 40*.
- the error handling, refer to "*Error Handling" on page 40*.

If you checked the *Upload trace on trigger* check box during setup, the System Validation Package also created a waveform file containing the trace memory contents. The name of this file is also listed in the log file.

Log File Description

The log file is always opened when starting the System Validation Package. The default name of that file is SystemValidate.report and the default location is your current directory, where the program was started from.

Change the Log File

You may change the log file at any time during the test session, but this is not recommended while a test is still running. To change the log file

- choose the *Logfile* item from the *File* menu,

or, alternatively,

- click the Open Logfile button in the tool bar.

If you select a file that already exists, you get a warning message. You can either choose to overwrite this file or to select another one.

NOTE If you use the default log file SystemValidate.report, it will not be overwritten in a new session. All new information will be appended to the existing file. Thus, this standard log file can contain the loggings of more than one test session.

Log File Contents

The log file contains a copy of the contents of the output panel at the bottom of the GUI. They contain basic information about the tests run during the test session. This includes:

- The start date and time of the test session
- The path and file name of the log file
- All tests that were run during the session including their
  - test start date and time and scheduled test duration,
  - type of test with target address range where applicable,
  - the names of the PPR report files and the trace memory file, if applicable,
  - the test start signal,
  - for CPU-to-I/O-tests the amount of transferred data,
  - test termination with the results, either *PASSED* or *FAILED*,
  - and the elapsed time for the test.
- The date and time when closing the test session

Report File Description

The output functions of the PCI Protocol Permutator & Randomizer create several report files for each test action in your test scenario. They contain detailed information on all the PPR functions that were called during the tests.

Report File Names

The names of the PPR report files are

- <testname>_<cardnumber>_<read/write>.pprrpt

  For every testcard acting as a master two block report files are created, one for the read commands and one for the write commands. These files contain the information on all the transactions and permutations on the bus during the test. The names of these files are listed in the log file.

- masterattr_<cardnumber>.pprrpt

  This file contains the attribute settings for the master on testcard number <cardnumber>.

- targetattr_<cardnumber>.pprrpt

  This file contains the attribute settings for the target on testcard number <cardnumber>.

Report File Contents

The contents of the PPR report files are fairly complex and it usually is not intended that you need to open them to interprete your test results. However, for a detailed analysis of your test specification and the results, they are provided with the System Validation Package.

For a complete description of the report file contents, please refer to the *HP E2975A PCI Protocol Permutator & Randomizer Software User's Guide*.

Error Handling

This section is not supposed to be a complete troubleshooting guide. However, there are some errors that might occur on some systems and that are relatively easy to handle.

Test Action Errors

Every test action within the test scenarios is using the PCI Analyzer to trigger on certain error events. The different types of errors that can be detected with this tool are:

- PCI protocol errors. One of the PCI protocol rules is violated.

- Timing errors. A violation of the signal setup or hold time rules has occurred.
- Data compare errors. An error has occurred in the transferred data.
- Bus hangs. No more transactions can be completed on this bus.

On the occurrence of any of these errors, the current test action is terminated. The *Action View* of the program window displays the action status *Error*, and in the *Card View* an exclamation mark appears to indicate the error type. Additionally, the error is reported both in the output area and the log file.

To narrow down the problem with the tested devices, it can help to test the same data path with another test action.

For example, if you tested the host/PCI bridge with the “CPU and PCI Exerciser to System Memory (hard)” test and encountered errors, you can try to test with less hard constraints. Alternatively, you can test the system memory accesses from the CPU and the PCI Exerciser with serarate tests.

Another example is, if you have problems with the communication over a PCI/PCI bridge using the “Peer to Peer Test”. Instead, you can examine both directions separately with two “Master to Target Tests”, to find out if the errors occur in one direction only.

Testcard Errors

A HP Exerciser and Analyzer testcard may encounter an error on some occasions:

- When the program is started, it scans the system under test for HP Exerciser and Analyzer testcards. Any testcard that does not have both a memory space and an I/O space assigned cannot be used for tests with this software. An error message comes up, informing you that these cards are excluded from all tests.

  However, you can reset these testcards to factory defaults—including both memory and I/O space—either already during program start or in the Card Properties dialog box later on. In both cases these testcards are still excluded from tests until the system is rebootet.
- If the error occurred during test run, the *Card View* in the program window displays the status *Error*. Then the function that caused the error is logged in the Card Log dialog box. In this case, too, a card reset and reboot may help.

For more information on the handling of erroneous testcards please refer to the C-API description in the user’s guide of the respective card.

If the card error was detected by a function of the PPR, its name starts with `B_E_PPR`. Then the reference is found in the *HP E2975A PCI Protocol Permutator & Randomizer Software User’s Guide*.

Common Protocol Errors

The Analyzer of the HP testcard observes 53 different protocol rules on the PCI bus and triggers the trace memory if any violation of these rules occurs. A violation of some of these rules, however, does not always cause

problems. In fact, on some machines protocol errors occur regularly. An example is the rule LAT0 (the Target Ready signal is not asserted within 16 clock cycles after Initiator Ready was asserted).

If a protocol error like this occurs on your system under test, the test action terminates even if there is no problem on the bus. If you want to make your Analyzer insensitive for a particular protocol rule:

1. Determine the testcard(s) that detected the error in the *Card View* in the GUI.
2. If the Best GUI is external (preferably), open its Testcard Configuration dialog box from the *Setup* menu and select the port of the erroneous card.

   If the Best GUI is located on your system under test, open its Testcard Configuration dialog box from the *Setup* menu and scan the PCI bus for the port of the erroneous card.

   Step 3 and step 4 can be skipped, if the *Upload Trace on Trigger* option was selected for this test.
3. Upload the trace memory to the Best GUI by opening any of the Analyzer windows, for example, the Waveform Viewer window (from the *Analyzer* menu). This is not recommended if the GUI is running on the system under test. It is a very slow process and might even cause the system to become unstable.
4. Save the trace memory contents from the waveform viewer as a waveform file (wfm file) for later analysis.
5. Open the protocol checker from the *Analyzer* menu and mask the respective rule. A rule is masked and unmasked by toggling its enabled/disabled field.
6. Save this protocol mask as power up behavior by selecting the item *Save As PU Defaults* from the *File* menu in the Best GUI's main window and then clicking *OK*.
7. Repeat these steps for all testcards that detected this error.
8. Reboot your system under test to make all the settings come into effect.

NOTE Remember to undo these changes later. You will not be informed about the settings of the protocol mask when using the System Validation Package. If you forget to reset them, you will leave these errors undetected in future test sessions.