Agilent E2976A System Validation Package

User's Guide

Agilent Technologies

Important Notice

Copyright

Authors: Stephan Greisinger and Anja Schauer, t3 medien GmbH

Notice

The material contained in this document is subject to change without notice. Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

ATTENTION: Use of the Software is subject to the Agilent Software License Terms set forth below. Using the Software indicates your acceptance of these license terms. If you do not accept these licence terms, you may return the Software for a full refund. If the Software is bundled with another product, you may return the entire unused product for a full refund.

Agilent Software License Terms

The following License Terms govern your use of the accompanying Software unless you have a separate signed agreement with Agilent.

License Grant. Agilent grants you a license to use one copy of the Software. "Use" means storing, loading, installing, executing or displaying the Software. You may not modify the Software or disable any licensing or control features of the Software. If the Software is licensed for "concurrent use", you may not allow more than the maximum number of authorized users to Use the Software concurrently.

Ownership. The Software is owned and copyrighted by Agilent or its third party suppliers. Your license confers no title to, or ownership in, the Software and is not a sale of any rights in the Software. Agilent's third party suppliers may protect their rights in the event of any violation of these License Terms.

Copies and Adaptations. You may only make copies or adaptations of the Software for archival purposes or when copying or adaptation is an essential step in the authorized Use of the Software. You must reproduce all copyright notices in the original Software on all copies or adaptations. You may not copy the Software onto any public network.

No Disassembly or Decryption. You may not disassemble or decompile the Software unless Agilent's prior written consent is obtained. In some jurisdictions, Agilent's consent may not be required for limited disassembly or decompilation. Upon request, you will provide Agilent with reasonably detailed information regarding any disassembly or decompilation. You may not decrypt the Software unless decryption is a necessary part of the operation of the Software.

Transfer. Your license will automatically terminate upon any transfer of the Software. Upon transfer, you must deliver the Software, including any copies and related documentation, to the transferee. The transferee must accept these License Terms as a condition to the transfer.

Termination. Agilent may terminate your license upon notice for failure to comply with any of these License Terms. Upon termination, you must immediately destroy the Software, together with all copies, adaptations and merged portions in any form.

Export Requirements. You may not export or re-export the Software or any copy or adaptation in violation of any applicable laws or regulations.

U.S. Government Restricted Rights. The Software and any accompanying documentation have been developed entirely at private expense. They are delivered and licensed as "commercial computer software" as defined in DFARS 252.227-7013 (Oct 1988), DFARS 252.211-7015 (May 1991) or DFARS 252.227-7014 (Jun 1995), as a "commercial item" as defined in FAR 2.101(a), or as "Restricted computer software" as defined in FAR 52.227-19 (Jun 1987)(or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and any accompanying documentation by the applicable FAR or DFARS clause or the Agilent standard software agreement for the product involved.

Table of Contents

Introduction

This section introduces the E2976A System Validation Package (SVP). It gives an idea of the general use of this software tool and shows how it is best included in your test environment. This section 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 E2976A System Validation Package is a ready-to-use software package that performs system stress tests during the validation phase of PCs, servers, workstations, or other PCI-based systems. The tool sets up and controls several Agilent PCI testcards to create application-realistic system traffic. This allows you to set up fully predictable traffic scenarios and provides measurable test coverage and test repeatability.

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

- Putting the data paths in I/O systems under stress in a controlled and predictable way.
- Running several tests in parallel to increase system stress on multiple data paths.
- Independence of test and software architecture from the I/O system (PCI, System I/O).
- Providing the ability to control multiple testcards for different I/O systems (PCI 1x/2x/4x).
- Internal testcard controlling (controlling host on SUT), or external controlling, using a separate host computer (RS-232, Parallel/Fast Host).

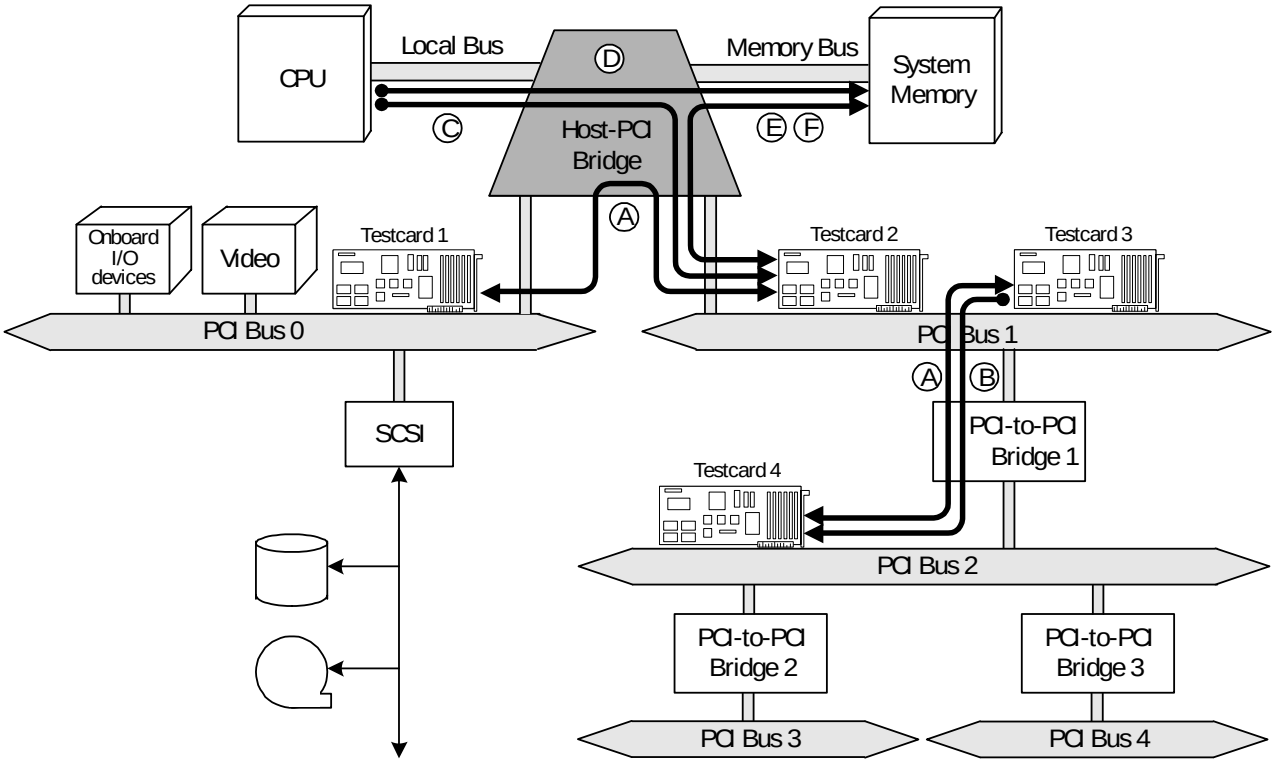
- Allowing full configuration; settings can be stored to and retrieved from disk.
- Providing extensive static reports (settings and configuration) and live reports (test progress and results).
- Providing ready-to-run tests to force the most critical conditions for the system.
- Providing repeatable tests for failure analysis and failure regression tasks.
- Supporting comparison of test results for performance evaluation and further system improvements.
- Making an easy link to R&D's debug environment.

Data Paths Overview

The Agilent PCI testcards can be plugged into any PCI bus of the system under test. This allows the testing of a variety of data paths. The System Validation Package provides a whole range of test actions for testing the different data paths. The letters in parentheses in the following list correspond to the data paths indicated in the figure below.

- Peer-to-peer traffic between two PCI testcards (A)
- Master-to-target traffic between two PCI testcards (B)
- CPU to testcard (C)
- CPU and testcard to system memory (D and E)
- Testcard to system memory (E)
- Testcard read from system memory (F)

These data paths are illustrated in the figure.



Additionally, the System Validation Package provides the Protocol Checker test and the PCI Configuration Scan test, which passively observe devices on the PCI bus.

Test Coverage

Two of the main problems of typical test methods for system validation are that they take a long time and are difficult to repeat. The usual approach is to simply plug standard PCI cards into the system under test, load them with traffic and wait until an error occurs. Even with very long tests, there is a high probability that not all possible scenarios will be tested.

The advantage of the System Validation Package is that you can design tests specifically for certain system-critical conditions. These tests can then be repeated as necessary. Furthermore, you can also let the SVP go through all possible variations of parameters, such as commands and block sizes, so that all situations that the system may face are covered.

Thus, you can cover all traffic situations for the system to be tested within minutes. The technology providing this coverage is called the PPR technology.

PPR, the Key Technology

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

PPR permutates the PCI protocol parameters and data traffic in a pseudo-random way. More specifically, all memory accesses are varied through all possible combinations of their attributes. This applies for varying block sizes and the use of the different memory commands, such as 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.

Permutations also include protocol attributes, which ensures that the transactions are performed with

- all possible wait states inserted by both the Exerciser's master and target,
- all possible 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 *Agilent E2975A PCI Protocol Permutator & Randomizer Software User's Guide*, which is delivered with the testcard.

Hints for programming on 64 bit systems

If you plan to run the SVP software under 64 bit Itanium systems, you should read the following.

Targeted are currently the 64 bit Microsoft .NET Server OSes.

To install, you need a separate installation file, named setup64.exe, located in the CD's ia64 directory. Do not install the 32bit setup.exe.

On 64bit Itanium systems the following is true:

- Kernel mode:

  Drivers always need to be 64 bit drivers; 32 bit drivers wont work. Especially, this means that you can't use the existing 32 bit drivers. Our 64 bit drivers are named b_2kpci_64.sys, b_2khif_64.sys, b_usb_64.sys and b_usbgen_64.sys.

- User mode:

  If you are starting an application, the .exe (and all needed dlls) need to be either all 32 bit files or all need to be 64 bit files, i.e. you cannot mix them. For example a 64 bit .exe cannot use a 32 bit dll.

  Our 64-bit dlls always have the suffix "xp64", e.g. capixp64.dll (instead of capikk.dll in 32 bit mode).

- For the SVP GUI, there is a 64 bit executable installed (together with its 64 bit dlls).

NOTE You should not run the 32 bit SVP GUI on IA64, because the memory driver (contained in b_2kpci_64.sys) needed by SVP works with native 64-bit addresses.

Getting Started

To set up a system test with the Agilent E2976A SVP Graphical User Interface (GUI), several steps are needed. This guided tour shows these steps by means of an example.

The recommended approach is:

1. Test preparation

   Insert testcards and start the software.

2. Test configuration

   Define the scenarios.

3. Definition of test functions

   Set test parameters and select testcards for each test.

4. Definition of testcard properties

   Check and modify master, target and protocol checker settings.

5. Test execution

   Run the test and get the test report.

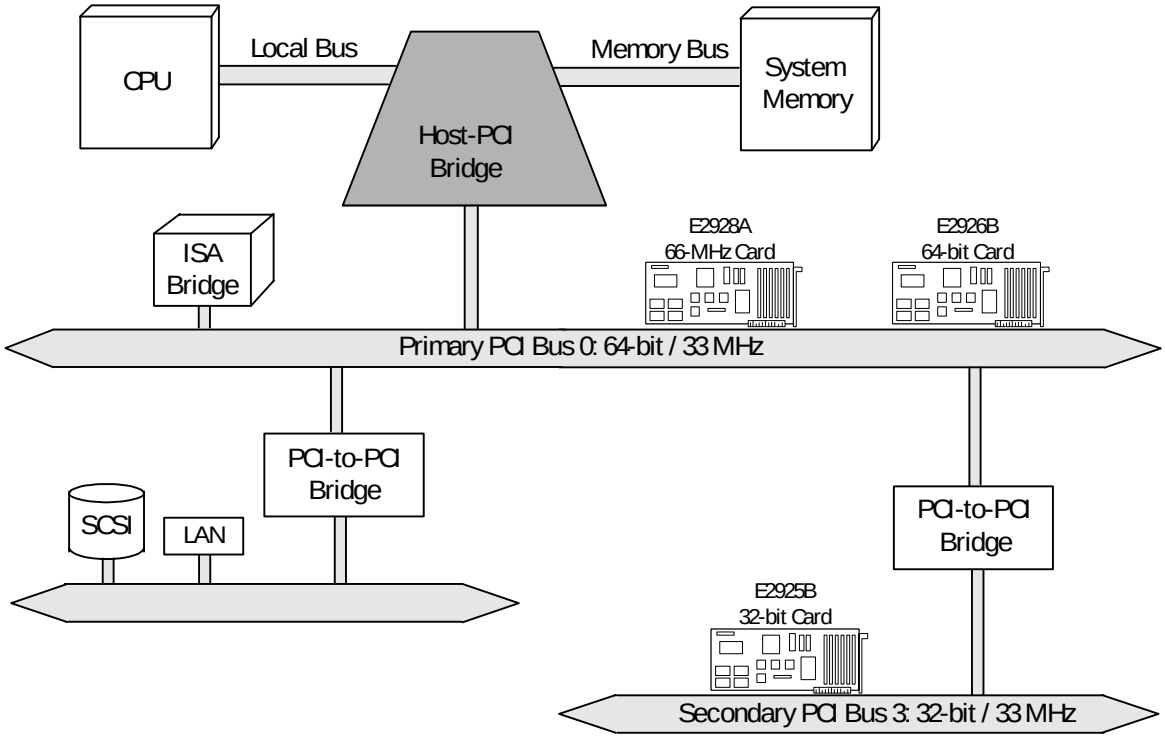
In this guided tour, you are shown how to set up a system test by means of a possible test configuration and some selected tests that are provided with the Agilent E2976A SVP software. These tests perform configuration space scanning, CPU interaction and stressing one PCI-to-PCI bridge, for example.

The online mode is used as a basis. If it is not possible for you to insert the required testcards, you can set up a system test in offline mode, but you cannot execute it. Reference is made to the differences between online and offline mode.

The test example is also available as a VPS file, which is delivered with the Agilent E2976A software. To view this example, switch to offline mode and open guided_tour.vps.

Test Example

Task A new system has been configured using several PCI busses, PCI-to-PCI bridges and other devices. The system looks like this:



The task is to check the overall system stability, focussing especially on the data path between Bus 0 and Bus 3 (the PCI-to-PCI bridge) and the data paths to the system memory. These data paths are stressed to detect protocol errors and data errors.

Solution Predefined tests provided by the Agilent E2976A SVP software will be used for:

- transferring data from the CPU to a testcard
- accessing the system memory from a testcard
- generating traffic in both directions between two testcards
- generating additional bus load for one bus
- scanning the configuration space of the bus system

These tests require three testcards inserted into the system under test as shown in the picture above.

Implementation

To use these features, three scenarios must be set up. The scenarios will be executed one after the other, the tests within each scenario will be executed concurrently.

- Scenario 1

  To check and report the system configuration, the whole configuration space of the bus can be scanned by using the PCI Configuration Scan test. To set up this test, the following settings are used:

Test Function to be used:	configscan
Testcard to be used:	E2928A (Bus 0)
Duration of the tests:	60 seconds
Start Delay:	0
Bandwidth:	100 %

  The test tries to occupy the bus with the whole bus bandwidth of 64 bit. This target bandwidth cannot always be achieved.

- Scenario 2

  This scenario includes two tests that run concurrently. The CPU to Testcard test transfers data from the CPU to a testcard on Bus 0. The Testcard to System Memory test accesses the system memory from a testcard on Bus 3. The following settings are used:

Test Functions to be used:	cpu2card	cardtosysmem
Testcards to be used:	E2926B (Bus 0)	E2925B (Bus 3)
Duration of the tests:	75 seconds	75 seconds
Start Delay:	0	0
Bandwidth:	100 %	40 %

  The CPU to Testcard test tries to occupy the whole Bus 0 bandwidth, the Testcard to System Memory test tries concurrently to occupy 40 % of the Bus 0 and Bus 3 bandwidth.

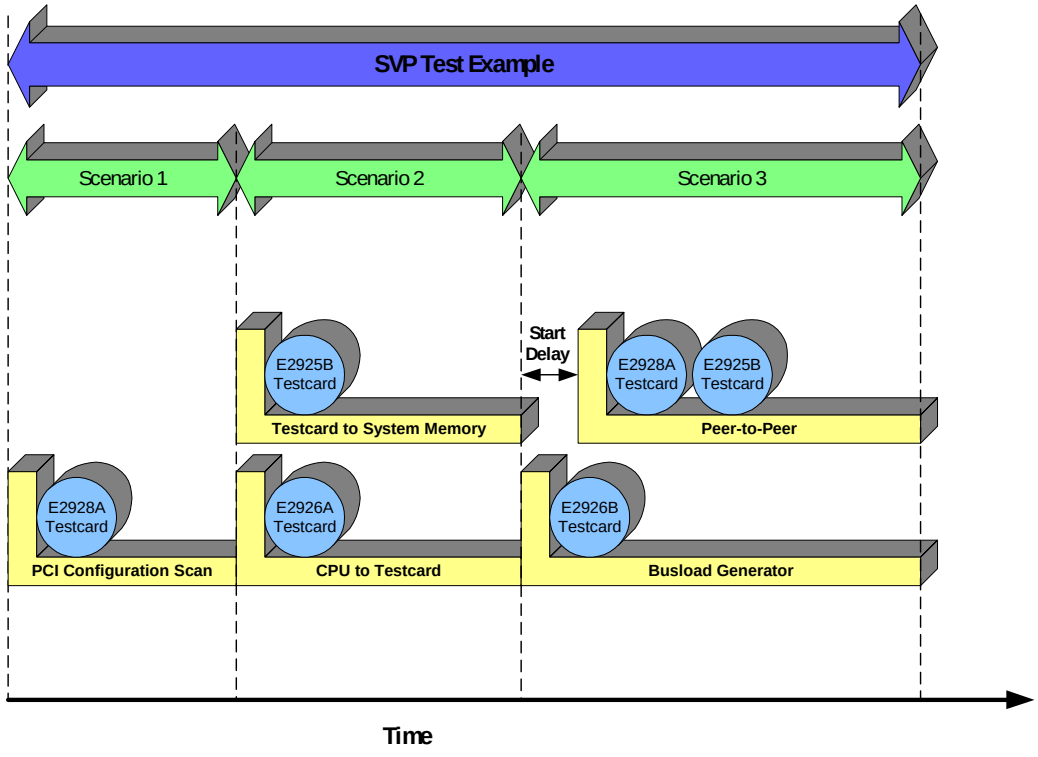
- Scenario 3

  This scenario includes two tests that run concurrently. The Peer-to-Peer test generates traffic between testcards on Bus 0 and Bus 3 in both directions. The bus load test generates additional load on Bus 0. The following settings are used:

Test Functions to be used:	peer2peer	busload
Testcards to be used:	E2928A (Bus 0), E2925B (Bus 3)	E2926B (Bus 0)
Duration of the tests:	200 seconds	215 seconds
Start Delay:	15 seconds	0 seconds
Bandwidth:	100 %	100 %

  The SVP software allows you to delay the start of a test and to use tests with different durations.

Timing The following figure shows the whole test configuration and the timing.



Preparing for Setting Up the Test Example

To prepare for the test example:

1 Before you use the software, consider which busses in your system under test are to be checked. You need to insert at least one testcard per tested bus.

2 Start the Agilent E2976A System Validation Package software.

For this test, we assume that the software is running on the system under test.

The software is in online mode by default and automatically scans the testcards connected to the system under test.

All available testcards are listed in the *Cards Available* list in the SVP object window which is always visible when you start the software.

By default, the testcards are named with Testcard 1, Testcard 2 and Testcard 3.

Test Setup in Offline Mode

If your system differs from that shown in the test example and if it is not possible for you to insert the required testcards, you can set up the test example in offline mode. To set up tests in offline mode:

1 Switch the software to the offline mode by clicking the icon  in the tool bar.

You can now define the testcards by inserting testcards into the *Available Cards* item in the navigator.

2 Click the *Available Cards* item and select *Insert New Card* from the *Edit* menu.

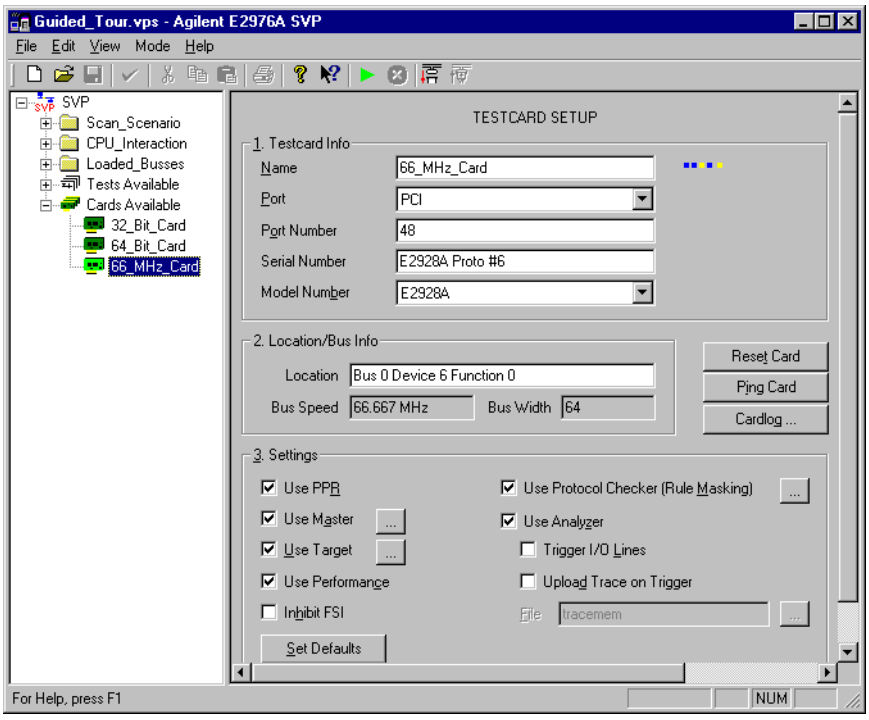
This inserts a new testcard in the navigator.

3 Repeat step 2 as necessary.

Renaming Testcards

To rename the testcards:

1. Click the Testcard 1 item in the navigator and enter **`32_Bit_Card`** in the *Name* field in the Testcard Setup window.
2. Rename Testcard 2 and Testcard 3 to **`64_Bit_Card`** and **`66_MHz_Card`** in the same way.



Setting Up the Example Test Configuration

The Agilent E2976A SVP software provides one scenario by default. For the test example, you need to insert another two scenarios and assign the test functions to the scenarios.

Create the Scenarios

To create all scenarios:

1. Click the SVP object in the navigator and select *Insert New Scenario* in the *Edit* menu.

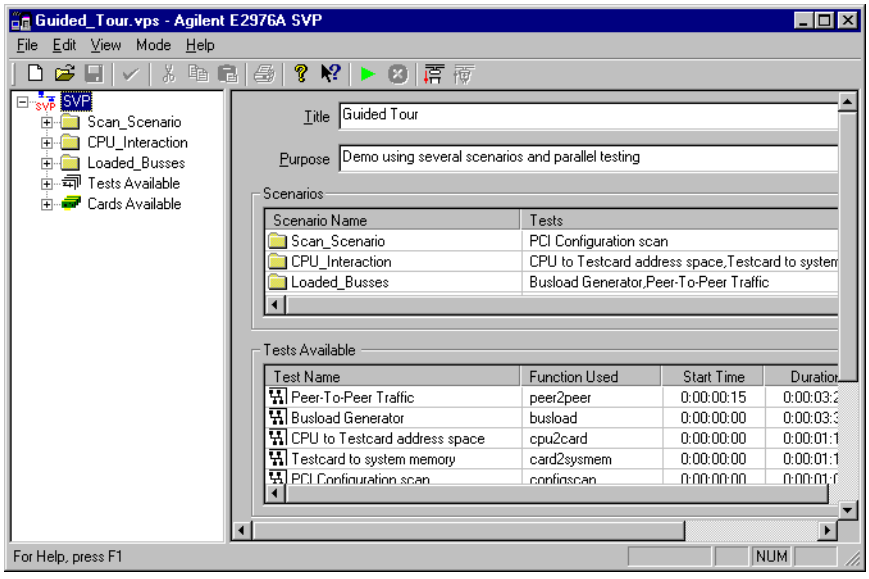
   This inserts one scenario into the navigator.

2. Repeat step 1 for the third scenario.

3. Click the Scenario_1 item in the navigator and enter the new name **`Scan_Scenario`** in the *Name* field in the Scenario Details window.

   The new name will appear in the navigator.

4. Rename Scenario_2 and Scenario_3 to **`CPU_Interaction`** and **`Loaded_Busses`**.



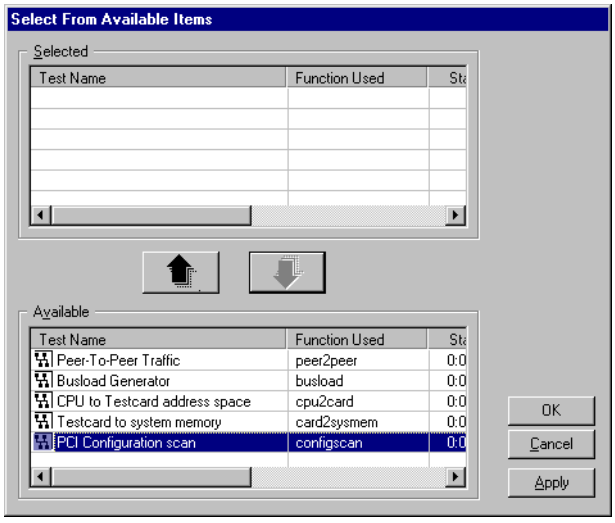
Insert Tests in Scenarios

To insert the PCI Configuration scan test in the Scan_Scenario scenario:

1 Click the Scan_Scenario in the navigator and click the *Select Test(s)* button in the Scenario Details window.

This opens the *Select From Available Items* dialog box.

2 Select the *PCI Configuration scan* test from the *Available* drop down list and click the upturned distribution arrow.



The PCI Configuration scan test will then appear in the *Selected* list.

3 Repeat steps 1 and 2 to insert

– the CPU to Testcard test and

– the Testcard to System Memory test

into the CPU_Interaction scenario.

4 Repeat steps 1 and 2 to insert

– the Busload Generator test and

– the Peer-to-Peer Traffic test

into the Loaded_Busses scenario.

NOTE It is possible to insert new tests in the *Available* list and remove available tests from the *Available* list. For further information, see *"Defining Test Functions" on page 53*.

Defining Test Functions

In the previous section, several test functions have been merged into different scenarios. The Agilent E2976A SVP software provides default settings for these tests. For our task, some test settings are to be modified. How this takes place is shown in this section.

Because the main steps are the same for all tests, only the setup for the PCI Configuration Scan test is described in all details.

PCI Configuration Scan Test Setup

To set up the PCI configuration scan test:

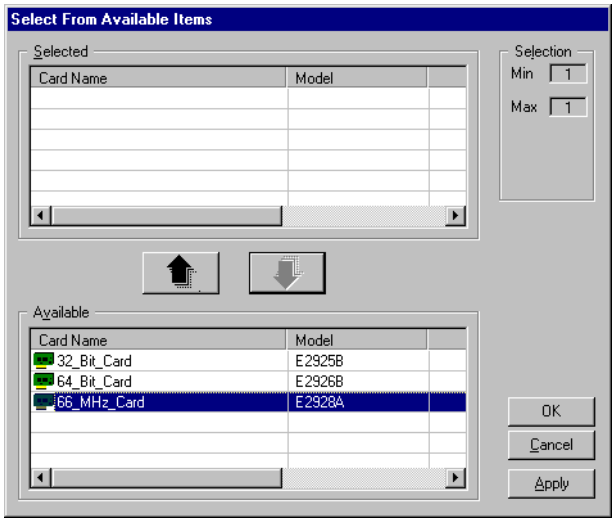
1. Click the PCI Configuration Scan test in the navigator to open the respective Test Setup window.

   The default parameter settings can be used in this PCI configuration scan test.

2. Click the *Select Card(s)* button.

   This opens the *Select From Available Items* dialog box.

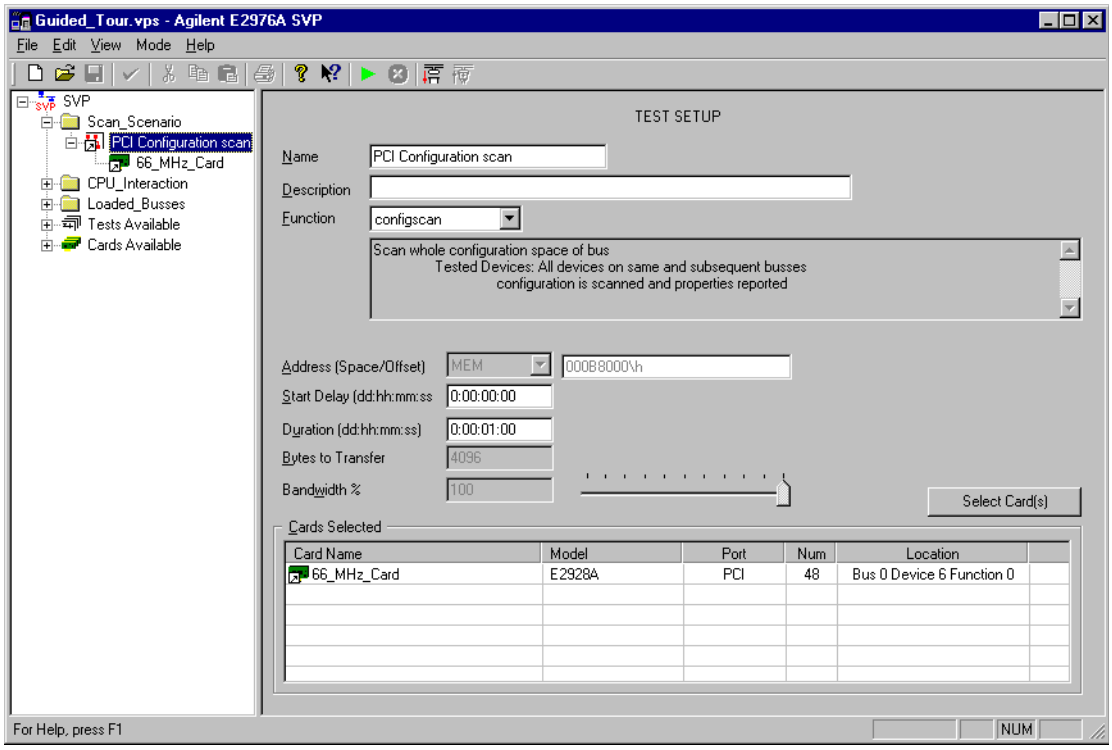
3. Select from the *Available* list the *66_MHz_Card* testcard and click the upturned distribution arrow.



   The *66_MHz_Card* testcard will then appear in the *Selected* list.

4. Click *OK* to close the dialog box.

The resulting Test Setup window is:



You can get a short description of the current test below the *Function* drop down list.

CPU to Testcard Test Setup

To set up the CPU to Testcard test:

1. Open the Test Setup window for the CPU to Testcard Address Space test.

   The testcard memory address space (*MEM*) is selected by default.

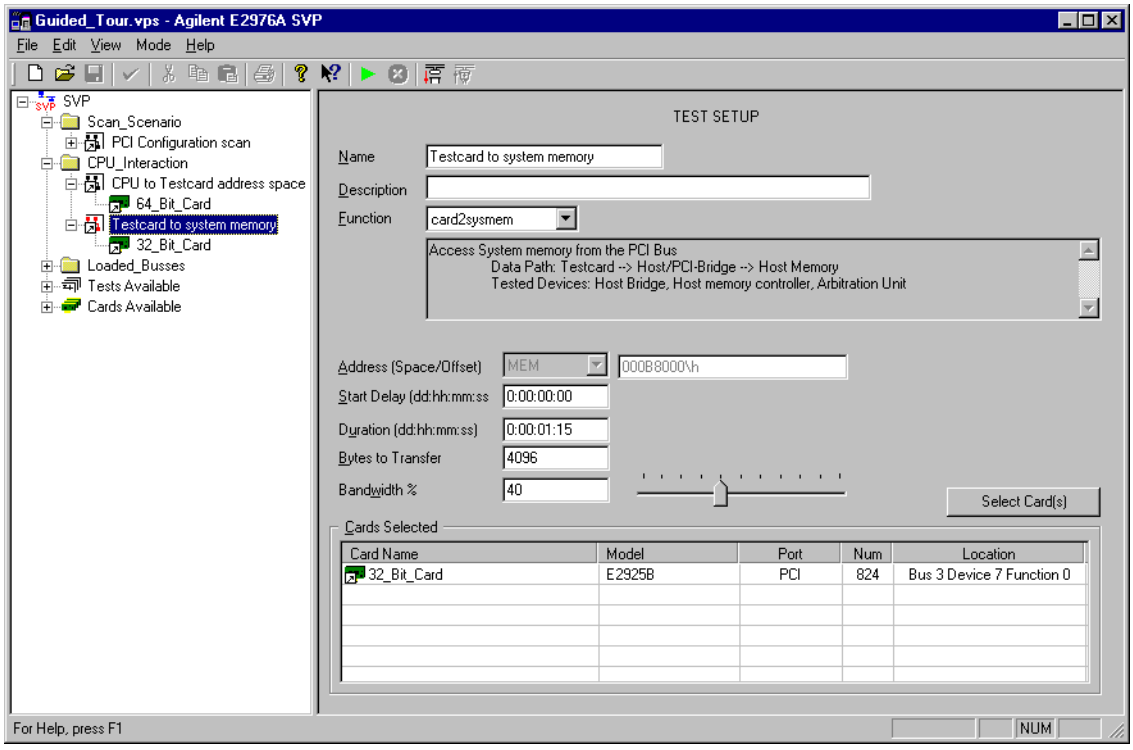
2. Enter a value of `1` minutes and `15` seconds in the *Duration* field.
3. Select the *64_Bit_Card* testcard for this test.

Testcard to System Memory Test Setup

To set up the Testcard to System Memory test:

1. Open the respective Test Setup window for the Testcard to System Memory test.
2. Enter a value of `1` minutes and `15` seconds in the *Duration* field.
3. Enter a value of `40` % in the *Bandwidth* field .
4. Select the *32_Bit_Card* testcard for this test.

The resulting Test Setup window is:



Busload Generator Test Setup

To set up the Busload Generator test:

1. Click the Busload Generator test in the navigator to open the respective Test Setup window.

   The testcard memory address space (*MEM*) is selected by default.

2. Enter a value of `3` minutes and `35` seconds in the *Duration* field.

3. Select the *64_Bit_Card* testcard for this test.

Peer-To-Peer Traffic Test Setup

To set up the Peer-To-Peer Traffic test:

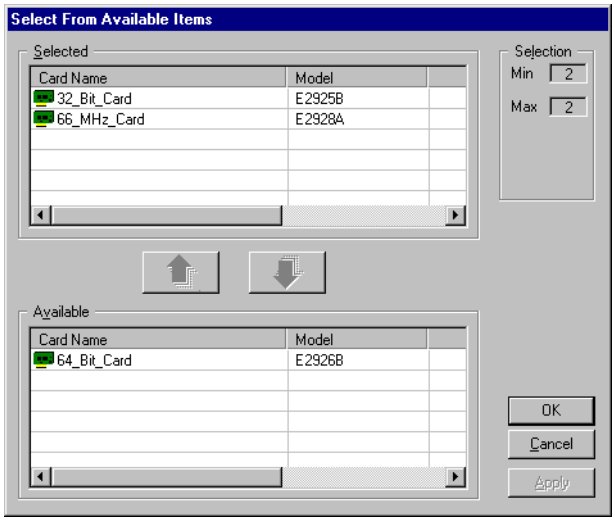
1. Open the Test Setup window for the Peer-To-Peer Traffic test.

2. Enter a value of `15` seconds in the *Start Delay* field.

3. Enter a value of `3` minutes and `20` seconds in the *Duration* field.

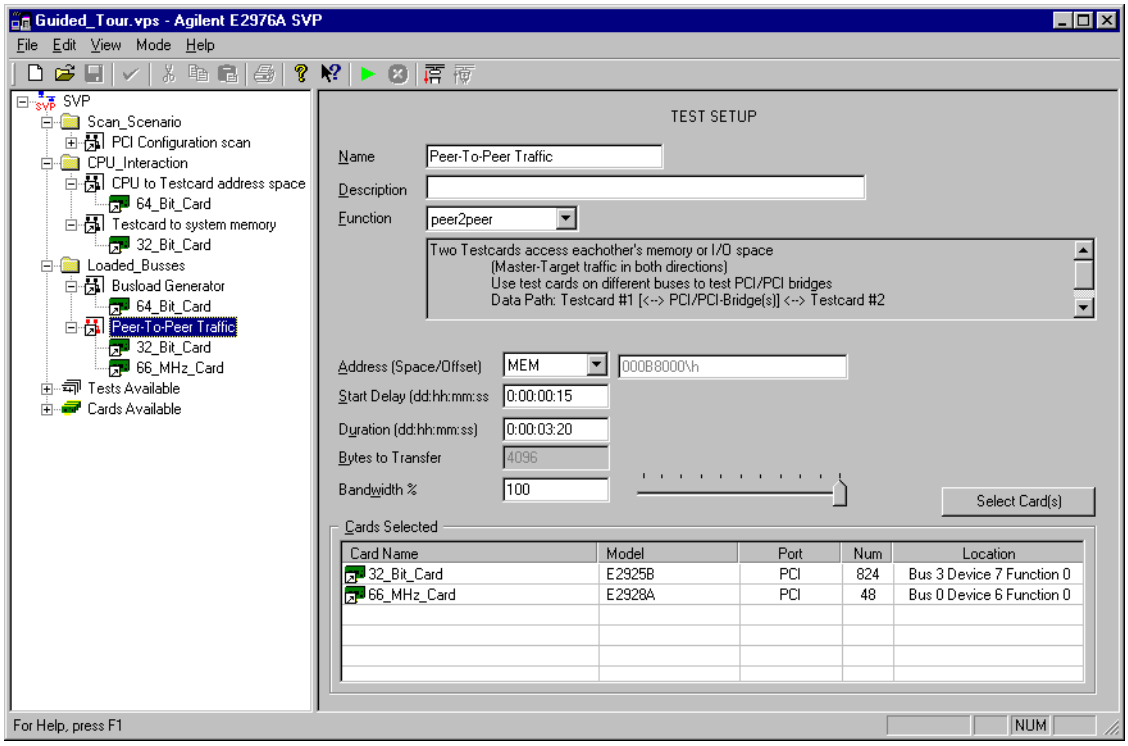
4 Select the *32_Bit_Card* testcard and the *64_Bit_Card* testcard for this test.

The resulting Select From Available Items dialog box is:



For this test, two testcards are required. The SVP software prevents the selection of more or less testcards. The limits are shown under *Selection*.

The resulting Test Setup window is:



Setting Testcard Properties

The Agilent E2976A SVP software provides testcard settings for all available testcards by default. The settings are available via the Testcard Setup window and can be adapted to the actual tests.

The settings determine which testcard features are active. If the master, target, PPR and protocol checker features are active, you can select further testcard properties, for example, various read/write commands. These properties are available via the details buttons next to the respective check boxes.

In the test example, the identical master and target settings for all testcards are used. The 32_Bit_Card testcard is shown as an example.

The protocol checker properties of the 66_MHz_Card differ from the 32_Bit_Card. Therefore, the protocol checker properties for the 66_MHz_Card are listed separately.

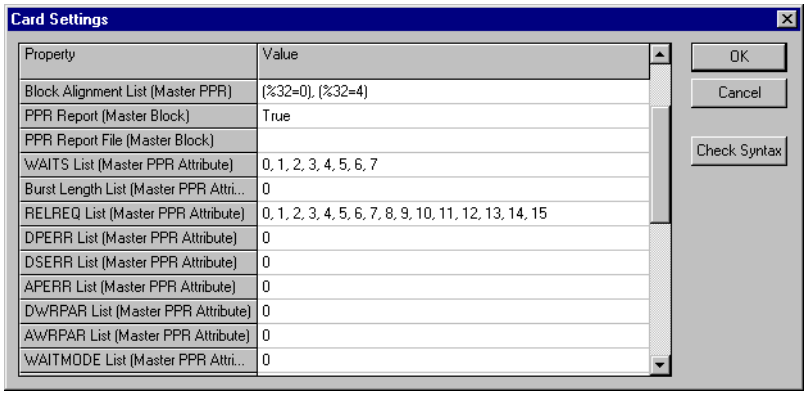
To set master, target and protocol checker properties of the 32_Bit_Card testcard:

1 Open *Cards Available* in the navigator and select the *32_Bit_Card* testcard.

  This opens the Testcard Setup window of this testcard.

2 Click the details button next to the *Use Master* check box.

  This opens the master *Card Settings* dialog box.



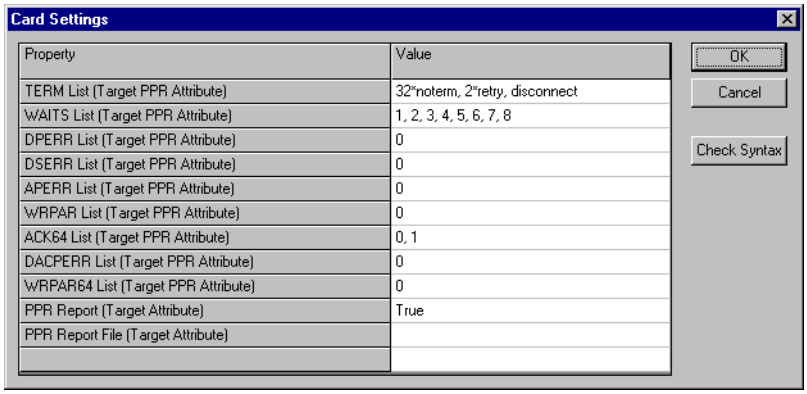
3 For the example the following master properties can be set:

Property	Value	Description
PPR Report File (Master Block)	mblock.rpt	File name for the master block report.
WAITMODE List (Master PPR Attribute)	0, 1	List of values to keep the address constant during the WAITS phases or not: 0: Address is stable 1: Address toggles
STEPMODE List (Master PPR Attribute)	0, 1	List of values to keep the address constant during the STEPS phases or not: 0: Address is stable 1: Address toggles
TRYBACK List (Master PPR Attribute)	0, 1	List of Fast Back-to-Back cycle tries: 0: Does not try Fast Back-to-Back cycle. 1: Tries Fast Back-to-Back cycle.
DELAY List (Master PPR Attribute)	0	No master transaction delay.

4 Click OK to verify the settings.

5 Click the details button next to the *Use Target* check box.

This opens the target *Card Settings* dialog box.



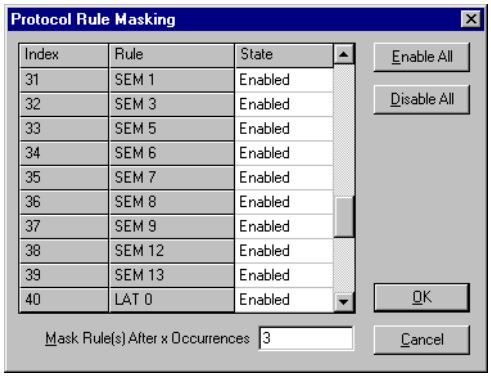
6 For the example the following target properties can be set:

Property	Value	Description
Termination List (Target PPR Attribute)	0	No termination.
Waits List (Target PPR Attribute)	0, 1, 2, 3, 4, 5, 6, 7, 8	List of number of waits that are permutated.
PPR Report File (Target Attribute)	tattr.rpt	File name for the target attribute report.

7 Click OK to verify the settings.

8 Click the details button next to the *Use Protocol Checker (Rule Masking)* check box.

This opens the Protocol Rule Masking dialog box.



9 Disable SEM8, SEM9 and LAT0 by clicking into the state column of the specified rules.

Violations of these rules will not be detected.

10 Click OK to verify the settings.

For the protocol checker properties of the *66_MHz_Card* testcard, the default settings (all rules are enabled) can be used.

Running the Test Example

NOTE You can only run the test if all testcards are available.

To run the test example:

1. Ensure that you are in online mode.

   If not, click the go online icon  in the toolbar.

2. Click the run icon  in the toolbar.

   The software automatically opens the *SVP Reporting* dialog box where the test report is shown.

The resulting test report is displayed in the following section. Because new test status information is added to this report every few seconds, the report can be very large. Hence, only a part of the report can be shown here.

Analyzing the Test Report

Report Start

The report starts with the start date and time of the test session. The first scenario (Scan_Scenario) with start date and time and its test (PCI Configuration scan) then follows. The report shows the initialization, start date and time of the PCI Configuration scan test and the expired time of the scenario.

```
*************** Starting test at 08-Mar-2000, 16:20:04 h *************


---------------------- Scenario <Scan_Scenario> ---------------------


++++++++++++++++++++++ Scenario <Scan_Scenario> ++++++++++++++++++++


******** Starting scenario at 08-Mar-2000, 16:20:04 h
Starting Scenario <Scan_Scenario> at 08-Mar-2000, 16:20:04 h


-------------------- Test <PCI Configuration scan> -------------------
Initializing test PCI Configuration scan with function PCI Configuration
scan
Starting Test <PCI Configuration scan> at 08-Mar-2000, 16:20:09 h


---------------------- Scenario <Scan_Scenario> ---------------------
time into scenario is 15 s.
```

The expired time of the PCI Configuration scan test and the status of the testcard used in this test are now listed. Here you can see the performance for the whole bus and for the 66-MHz testcard.

```
******** Report at 08-Mar-2000, 16:20:19 h


------------------- Test <PCI Configuration scan> -------------------
time into test is 1 s.


----------------------- Testcard <66_MHz_Card> ----------------------
Starting Testcard <66_MHz_Card> at 08-Mar-2000, 16:20:09 h
Performance Status
 whole bus: Utilization 44.96% / Throughput 1.51% / Efficiency   3.37%
 this card: Utilization  40.61% / Throughput 0.48% / Efficiency   1.18%
```

The results of scanning the configuration spaces of all devices in the whole bus configuration are now listed. The report begins with the devices on the primary bus.

The results of scanning the ISA Bridge on the primary bus 0 are shown below:

```
Bus has 5 devices

Device at location Bus 0 Device 4 Function 0:

Intel Corporation:
82371AB: PIIX4 ISA Bridge
               Vendor Id: 8086
               Device Id: 7110
                 command: 000f
                          +IO +MEM +MASTER +SP.CYCLE -MWI ENABLE -
PALSNOOP ENABLE -PERRRESP -WAITCTRL -SERREN -FB2B ENABLE
                  status: 0280
                          FB2BCAP Medium Devsel Speed
             revision Id: 0001
               classCode: 060100 (Bridge Device, PCI/ISA)
          cacheline Size: 00
            latencyTimer: 00
              headerType: 80
                    BIST: 00
Base Address Registers:
                   BAR 0:          00000000 Space32
                   BAR 1:          00000000 Space32
                   BAR 2:          00000000 Space32
                   BAR 3:          00000000 Space32
                   BAR 4:          00000000 Space32
                   BAR 5:          00000000 Space32
           cardbusCISPtr: 00000000
     subsystem vendor Id: 0000
            subsystem Id: 0000
  expansion ROM BaseAddr: 00000000
          capability Ptr: 00
               reserved1: 00000000
               reserved2: 00000000
          interrupt Line: 00
           interrupt Pin: 00
                 min GNT: 00
                 max LAT: 00
```

After all devices have been scanned, the status of the trace memory trigger and the observer is shown. The observer status gives information about the first occurring protocol errors and the accumulating subsequent errors.

The detected protocol errors are listed by specifying the violated protocol rule and a short description.

```
Tracememory trigger occurred

Observer Status:

  PROTOCOL ERROR:

  57: LAT 0:Targets are required to complete the initial data phase of a
            transaction within 16 clocks, subsequent data phases within 8
            clocks. (PCI Spec. Appendix C, Rules 25 and 26)

  ACCUMULATED PROTOCOL ERRORS:

  34: W64 2:REQ64# must not be used with special cycle or interrupt
            acknowledge command. Only memory commands support 64 bit data
            transfers (no config commands, no IO commands). (PCI Spec.
            Sect. 3.10. 64-Bit Bus Extension)

  52: SEM 9:A delayed transaction hasn't terminated within 2^15 clocks.
            (Agilent rule to detect potential deadlocks)
```

Each scenario report ends with a final report. This report contains the test results (maximum performance for the whole bus and the used testcard) and the protocol checker results. That means, the number of rule violations per rule and testcard and the number of occurred errors in this scenario are listed.

```
++++++++++++++++++++++ Scenario <Scan_Scenario> ++++++++++++++++++++

******** Final Report at 08-Mar-2000, 16:20:22 h

----------------------- Testcard <66_MHz_Card> ----------------------
Test results:
  Maximum Performance during test:
  whole bus: Utilization  44.96% / Throughput   1.51% / Efficiency   3.37%
  this card: Utilization  40.61% / Throughput   0.48% / Efficiency   1.18%
Protocol checker results:
  34: violated 1 times
  52: violated 1 times
  57: violated 1 times
  Protocol checker counted total 3 violations

Scenario finished, 3 errors found.

++++++++++++++++++++++++ End of final report ++++++++++++++++++++++++
Total elapsed time of Scenario is 18 s.
```

The reporting continues with all further scenarios in the same way described above. In defined intervals (3 seconds), the performance of the whole bus and testcards used in this actual scenario and all errors that can be found with a testcard are displayed.

Below is an example of a statement of the CPU_Interaction scenario report.

```
++++++++++++++++++++ Scenario <CPU_Interaction> +++++++++++++++++++

******** Starting scenario at 08-Mar-2000, 16:20:22 h
Starting Scenario <CPU_Interaction> at 08-Mar-2000, 16:20:22 h
time into scenario is 28 s.

******** Report at 08-Mar-2000, 16:20:50 h

---------------- Test <CPU to Testcard address space> --------------
Initializing test CPU to Testcard address space with function CPU to
Testcard address space
Setting up card 64_Bit_Card for CPU to Testcard address space
Starting Test <CPU to Testcard address space> at 08-Mar-2000, 16:20:27 h
time into test is 23 s.

----------------------- Testcard <64_Bit_Card> ----------------------
Starting Testcard <64_Bit_Card> at 08-Mar-2000, 16:20:27 h
Performance Status
 whole bus: Utilization  31.24% / Throughput   2.09% / Efficiency   6.70%
 this card: Utilization  17.60% / Throughput   0.98% / Efficiency   5.55%
Tracememory trigger occurred
Starting CPU Test
Address is <MEM:FED40000\h>
Virtual address is 1570000
Size is 262144
Starting CPUTest <CPU to Testcard address space> at 08-Mar-2000, 16:20:27 h
Entered Run Function
Starting Test (cpu2card_run)

------------------ Test <Testcard to system memory> ----------------
Initializing test Testcard to system memory with function Testcard to
system  memory
Starting Test <Testcard to system memory> at 08-Mar-2000, 16:20:50 h

----------------------- Testcard <32_Bit_Card> ----------------------
Overriding user setting for parameter <delay> to limit bandwidth to 40%


---------------------- Scenario <CPU_Interaction> -------------------
time into scenario is 31 s.
```

The Test Configurations

This section covers all information needed to run the E2976A System Validation Package (SVP), including a summary of the possible hardware configurations and some recommendations for these configurations to make best use of the SVP.

Internal and external testcard control are also decribed in this section.

Possible Hardware Configurations

The Agilent E2976A System Validation Package supports the following Agilent PCI testcards:

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

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

Several testcards can be plugged into 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 into different busses in the system under test, for example, to test the bridges between these busses. You can plug several testcards on busses into the SUT, to test the PCI host bridge and the PCI host bridge configuration.

Recommended Configuration

For every test configuration, the recommended configuration is to have

- one Agilent E2976A System Validation Package license,
- one PCI testcard for each PCI bus,
- one additional PCI testcard for a peer-to-peer test on one bus.

Testcard Control

System validation makes it necessary to control many testcards at a time with one system validation tool. This tool allows two methods of controlling testcards: internal and external control.

Internal Control

Internal control of testcards means that the SVP software must run on the system under test (SUT).

External Control

External control of testcards means that the SVP software runs on a controlling host, which is connected via standard RS-232 or via the Fast Host Interface card, which is delivered with the testcard. External control of testcards is used whenever either the SUT's operating system does not support direct access to the I/O busses, or when the SUT's state is not stable enough to sustain a test software running. For test functions that require extra SUT actions, a Front Side Interface Executable is used.These test functions are:

- CPU to testcard
- Testcard to system memory
- CPU and testcard to system memory

The Front Side Interface Executable

The Front Side Interface is used whenever internal control of testcards is not desired or cannot be achieved. Communication to FSI takes place using a defined protocol, which communicates via the testcard's mailboxing interface. The software needed to communicate via the mailboxing interface must be running on the system under test. This stand-alone executable is the Front Side Interface Executable (FSI-E) .

Running the FSI-E

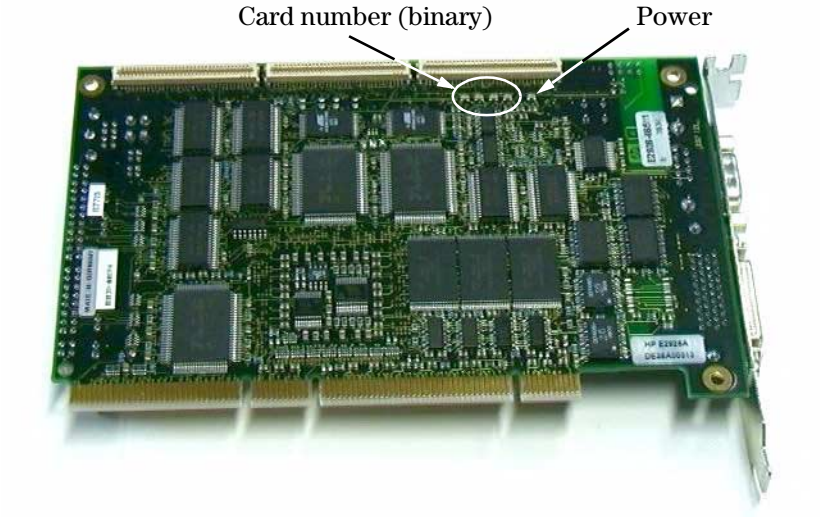
The FSI-E is executed on the SUT either from a boot floppy, or directly by clicking the *FSI Executable* in the Windows *Start* menu.

Installation for DOS operating systems After you have installed the Agilent E2976A System Validation Package software, you can find the file fsidos.exe in the <Installation Directory>\fsidos directory. This is the DOS-Version of the FSI-Executable. You can copy this file via a floppy disk to the destination DOS operating system.

Installation for Windows NT After you have installed the Agilent E2976A System Validation Package software on the destination system, you can start the FSI-E by clicking the *FSI Executable* in the Windows *Start* menu.

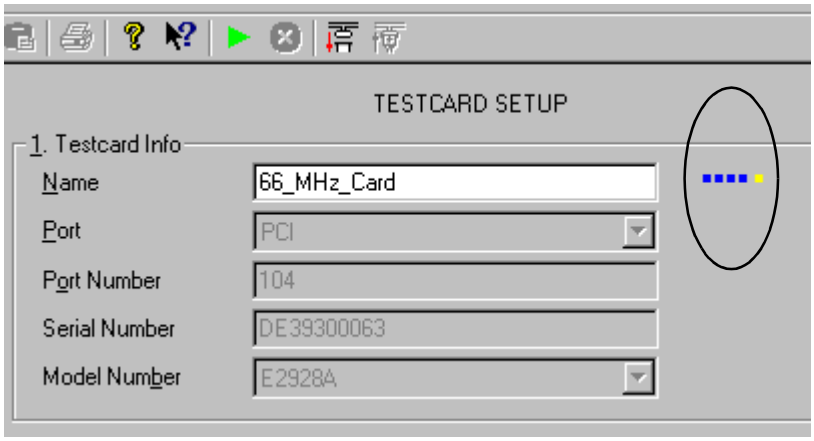
Identification of Card Numbers

In case you have several PCI testcards plugged into the 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, which are 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 separated slightly further than the others. It is always lit when the card is powered and SVP is connected. Closing SVP re-enables heartbeat.

Card Number LEDs The other four LEDs show a binary coding of the card number, where the LED farthest to the right represents the least significant bit. The pattern of the LEDs is displayed in the GUI next to the card name in the Testcard Setup window for easier identification.



NOTE This applies for all PCI testcards, except the Agilent E2940A CompactPCI testcard. On this testcard, the five LEDs are in a vertical row.

The Available Tests

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

General Test Description

Making full use of the testcards' features, the SVP can be used to put data paths within PCI-based computer systems under stress or to test individual chips such as host bridges or PCI-to-PCI bridges. The testcards' analyzer capabilities can be used in parallel to monitor the traffic and to track protocol errors.

Basic Test Structure

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

1. Writing a block of data to the destination memory.
2. Reading this data back from the destination memory.
3. Comparing the read data with the initial data field.
4. Writing a block of different data to the same location to make sure that data patterns always change.
5. Reading this data back from the destination memory.
6. Comparing the read data with the data most recently read.
7. Repeating the whole sequence with a different protocol behavior to cover more test cases.
8. Reporting detected errors in data comparison.

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 also triggered. This is very useful, for example, to capture the data traffic on two busses at the same time, because problems on one bus may have their root cause on another bus.

The following table gives an overview of the tests defined in the test library and their requirements.

Test	Number of Cards	Master	Target	Other
Testcard Read from System Memory	1	X	System main memory	
Peer-To-Peer Traffic	2	X	Testcard's memory or I/O	
Testcard to System Memory	1	X	System main memory	Locked memory
CPU to Testcard	1		Testcard's memory or I/O	
CPU and Testcard to System Memory	1			
Busload Generator		X	Self	
Master-To-Target Traffic	2	X	Testcard's memory or I/O	
PCI Configuration Scan		(X)	All devices on the same bus or on subsequent busses. Read-only.	
Protocol Checker	1 max			

NOTE Short descriptions of the tests are also found in the Test Setup window in the GUI where you select your test functions.

Busload Generator

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

Tested Data Path

The tested data path is the PCI bus into which the testcard is plugged.

Tested Devices

The tested devices are the arbitration unit and other devices on the same bus.

Recommendations for Testcard Settings

The accuracy of the bandwidth setting depends on the use of the PPR feature. To make best use of the busload generator test, there are some recommendations for setting the testcard properties used for this test:

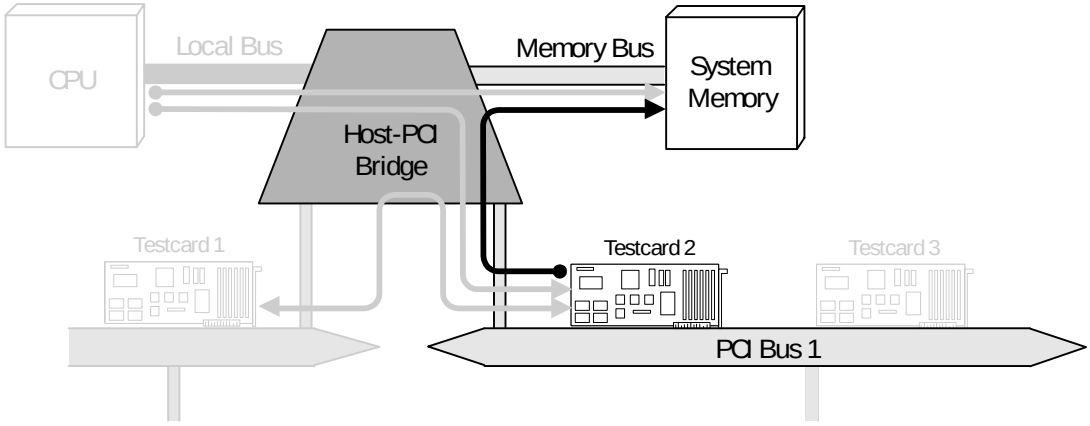
- If you do not need the PPR feature, disable the *Use PPR* check box in the Testcard Setup window.
- If you want to use the PPR feature for the master, select the *Use PPR* check box and set the following target properties:

Property	Value	Description
Termination List (Target PPR Attribute)	0 (noterm)	No termination
Waits List (Target PPR Attribute)	0: for all testcards except E2928/28_Deep 1: for testcard E2928/28_Deep	List of number of waits

For further information on setting testcard properties, refer to *Testcard Setup Window* in the *Agilent E2976A System Validation Package GUI Reference* (pdf-file).

Testcard to System Memory

This test accesses the system memory from the PCI bus. To do this, the testcard is defined as a master and sends different write and read commands.

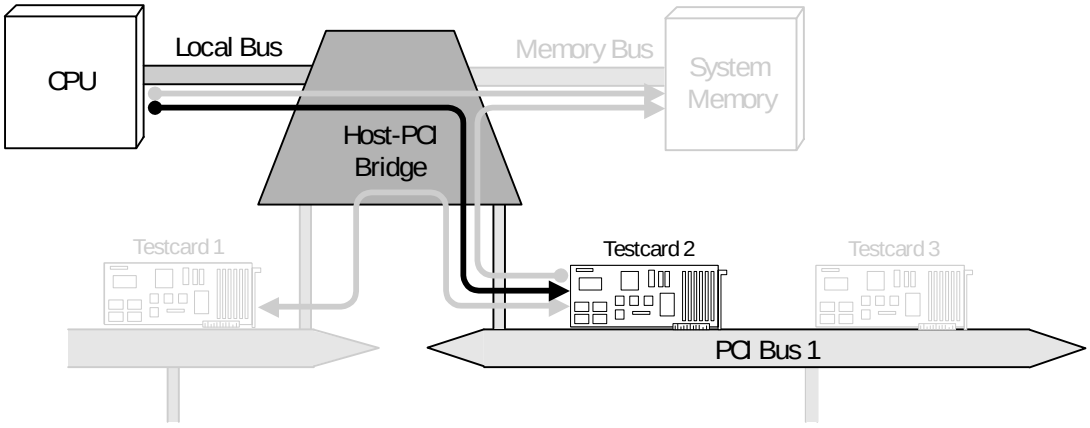


Tested Data Path The tested data path is the PCI bus from the PCI testcard to the host-PCI bridge and the system memory bus from the host-PCI bridge to the system memory.

Tested Devices The tested devices are the host-PCI bridge, the host-PCI bridge configuration, the host memory controller and the arbitration unit.

CPU to Testcard

This test accesses either the memory space or the I/O space of the testcard from the CPU. To do this, the test card is defined as a target.



CPU to Testcard Memory Space

Tested Data Path The tested data path is the CPU local bus and the PCI bus from the host-PCI bridge to the PCI testcard.

Tested Devices The tested devices are the host-PCI bridge, the host-PCI bridge configuration and the host memory controller.

CPU to PCI Testcard I/O Space

The access to the I/O space takes place via a virtual memory buffer and uses the I/O read and I/O write commands only.

Tested Data Path The tested data path is the CPU local bus (I/O access) and the PCI bus from the host-PCI bridge to the PCI testcard I/O port.

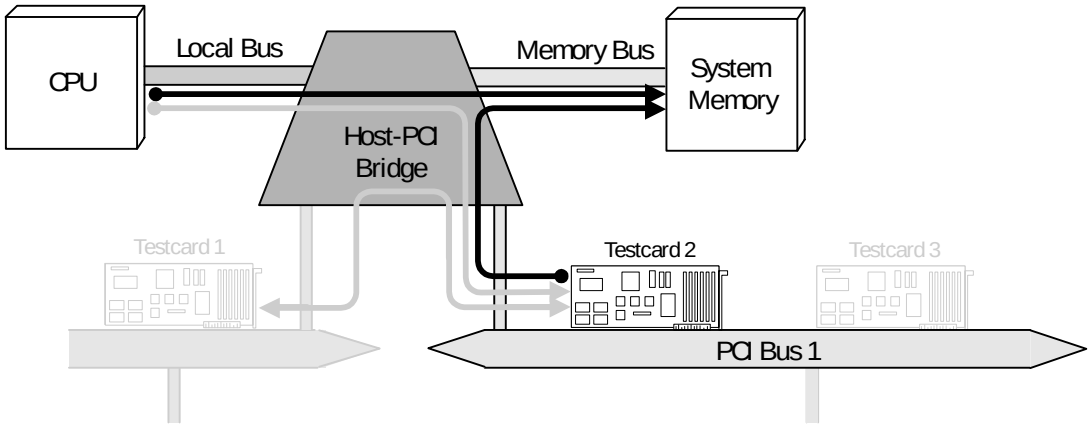
Tested Devices The tested devices are the host-PCI bridge, host-PCI bridge configuration and the host memory controller.

CPU and Testcard to System Memory

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

- The CPU accesses system memory space via a virtual memory.
- The PCI master of the testcard accesses system memory through the host-PCI bridge.

CPU and testcard 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.

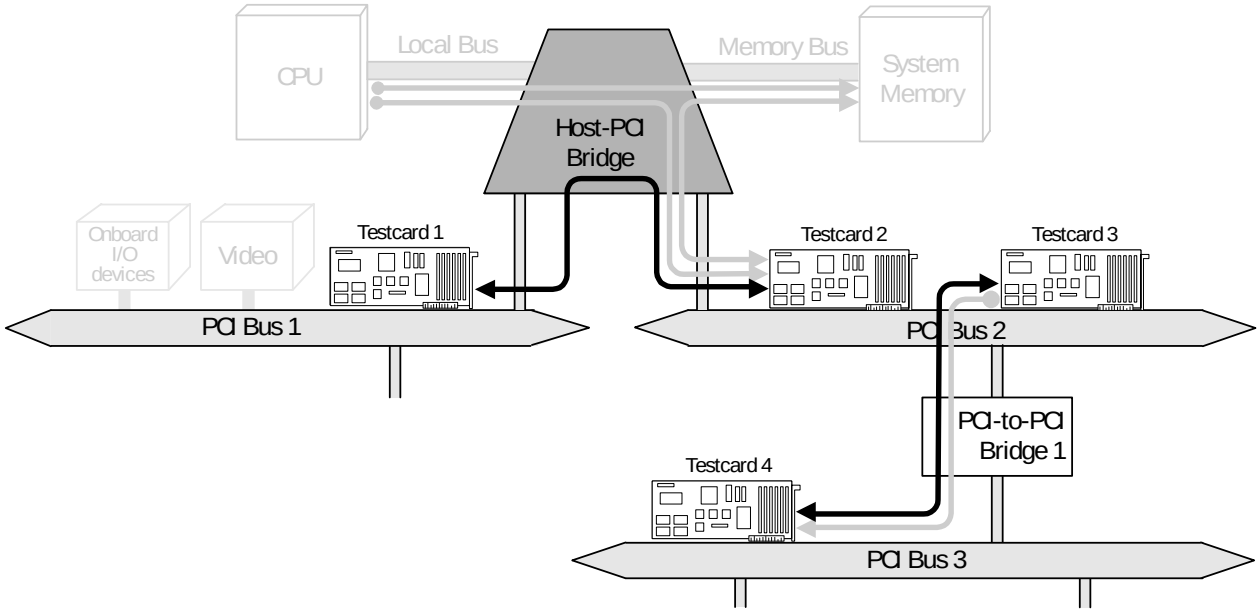


Tested Data Path The tested data path is the CPU local bus, the PCI bus from the PCI testcard to the host-PCI bridge, and the system memory bus from the host-PCI bridge to the system memory.

Tested Devices The tested devices are the host-PCI bridge, the host-PCI bridge configuration, the host memory controller, and the arbitration unit.

Peer-To-Peer Traffic

The peer-to-peer test requires two PCI testcards, that are set up to access each other's memory space. This is implemented with master to target traffic in both directions. The testcards on different busses are used to test the PCI-to-PCI bridge(s) between them.



Tested Data Path The tested data path is the PCI bus(es) from testcard #1 through the PCI-to-Host-PCI bridge to testcard #2 and the PCI bus(es) from testcard #3 through the PCI-to-PCI bridge to testcard #4.

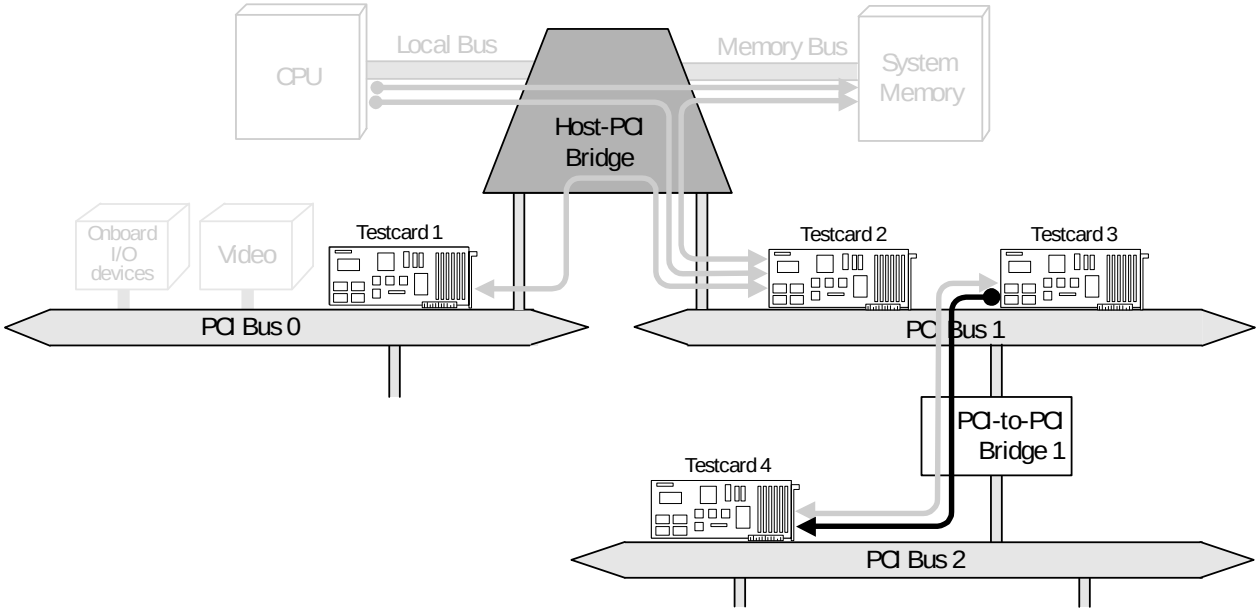
Tested Devices The tested devices are the PCI-to-PCI bridge(s), the PCI-to-PCI bridge configuration(s), and the arbitration unit(s).

NOTE If the selected address space is *MEM* and if PPR is activated, different memory commands (mem_read, mem_readline, mem_readmultiple, mem_writeinvalidate, and mem_write) are permutated.

Master-To-Target Traffic

This test requires two PCI testcards. One testcard accesses the other testcard's memory space. This is implemented with master-to-target traffic in one direction only (in contrast to the Peer-to-Peer test). The testcards on different buses are used to test the PCI-to-PCI bridges in between.

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



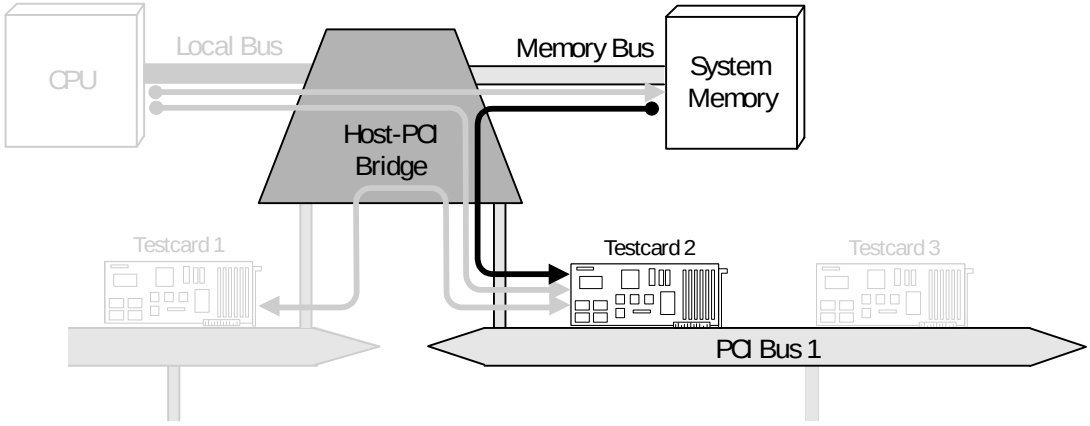
Tested Data Path The tested data path is the PCI bus(es) from testcard #3 (master) through the PCI-to-PCI bridge(s) to the testcard #4 (target).

Tested Devices The tested devices are the PCI-to-PCI bridge(s), the PCI-to-PCI bridge configuration(s), and the arbitration unit(s).

NOTE If the selected address space is *MEM* and if PPR is activated, different memory commands (mem_read, mem_readline, mem_readmultiple, mem_writeinvalidate, and mem_write) are permutated.

Testcard Read from System Memory

The testcard read test is used to check availability/readability of certain memory areas (system memory, device-mapped memory space and I/O space).



Tested Data Path The tested data path is the PCI bus from the host-PCI bridge to the PCI testcard and the system memory bus from the system memory to the host-PCI bridge.

Tested Devices The tested devices are the host-PCI bridge, the host-PCI bridge configuration, the host memory controller, and the arbitration unit.

Protocol Checker

This test does not drive any transactions on the PCI bus. The testcard only observes all PCI devices on the bus by checking PCI protocol violations. The detected problems are logged in the test report.

PCI Configuration Scan

In this test the testcard actively scans the whole configuration space of the bus. The configuration space report, which is stored in the test report, documents the test conditions during the test run.

Because the configuration space may change with each system reboot, this can be a big 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. It is therefore very difficult to estimate test durations.

Estimating Test Durations

To estimate test durations, you can check the PPR reports after setting the testcard properties. You can view the PPR reports when you check the syntax of the PPR reports attributes. For further details, refer to *"Check Testcard Settings" on page 56*.

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 busses 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 will have occurred. The behavior of the System Validation Package is predictable, but the behavior of the other devices is not. These devices might behave "friendly" most of the time. But in many cases you cannot make these devices behave most critically for testing purposes.

If you want to find out whether the 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.

Testing with the User Interface

This section gives information about the test architecture and shows how to set up and define the desired test configuration.

- Testing Principles

  Overview of the process used to perform a system test.

- Setting Up the Test Configuration

  Defining scenarios.

- Defining Test Functions

  Setting test parameters and selecting testcards for each test.

- Check Testcard Settings

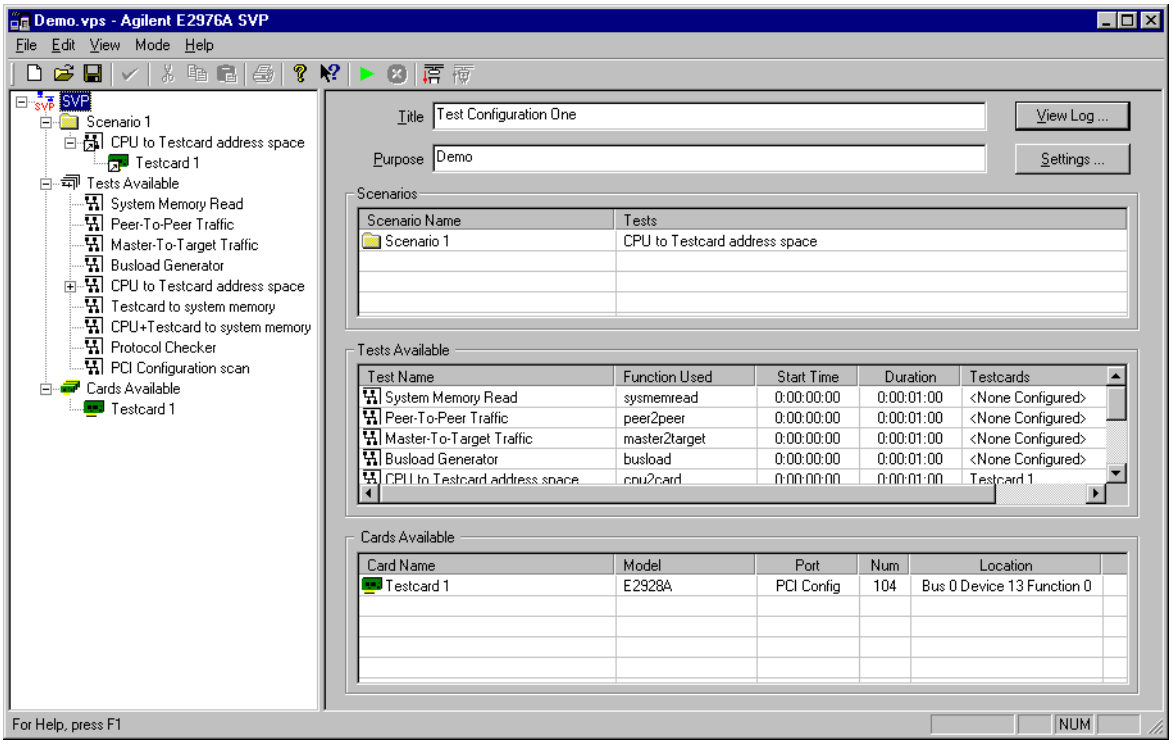
  Opening the Testcard Setup window and checking testcard properties.

- Running the Test

  Information on the test execution and the test report.

Testing Principles

This section shows the major steps necessary for setting up PCI system tests with the Agilent E2976A System Validation Package. Additionally, you will find information on the different components of the Graphical User Interface (GUI).



Testing with the Graphical User Interface means:

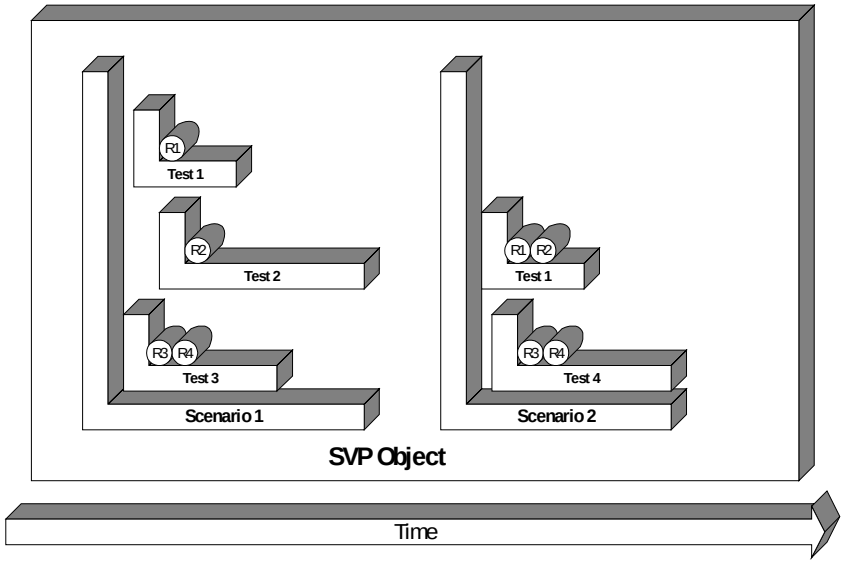
1. Running the System Validation Package

   The software automatically scans all connected PCI busses and other control interfaces (RS-232, Fast Host Interface) for Agilent testcards, and tries to initialize the detected testcards. Testcards that could not be initialized are not available for tests. This applies, for example, for testcards that are currently connected to the Agilent E2920 GUI.

   If you want to set up a test without any available testcards, you can switch the software into offline mode.

2. Setting up the test configuration

   The following figure shows an overview of the test architecture.



   The System Validation Package uses a top-down approach to assure testing flexibility. The smallest unit is called a test. It performs one single task, for example, a memory read. The test can use resources such as testcards or processor units (R1, R2, …).

   One step up in the hierarchy is the scenario, which combines several tests for parallel running. Each test and each resource can only be used once per scenario. The scenarios are surrounded by the SVP object, which provides control interfaces for all sub-levels.

   There are certain settings at each level of the hierarchy that are also valid for all sub-levels.

   See *"Setting Up the Test Configuration" on page 52* for details.

3. Checking or modifying the properties of the available tests

   This step is more or less optional. Basically you can run any test with the default settings. In the case of defining the delay and test duration it is needed to adapt the values. See *"Defining Test Functions" on page 53* for details.

4. Checking or modifying the properties of the available cards

   See *"Check Testcard Settings" on page 56* for details.

5. Running the test

   Click the Run button to start the test. See *"Running the Test" on page 59* for details.

Setting Up the Test Configuration

When starting the SVP software, you can find one scenario (Scenario 1) in the SVP object in the navigator (on the left side of the screen).

Add Scenarios to the SVP Object

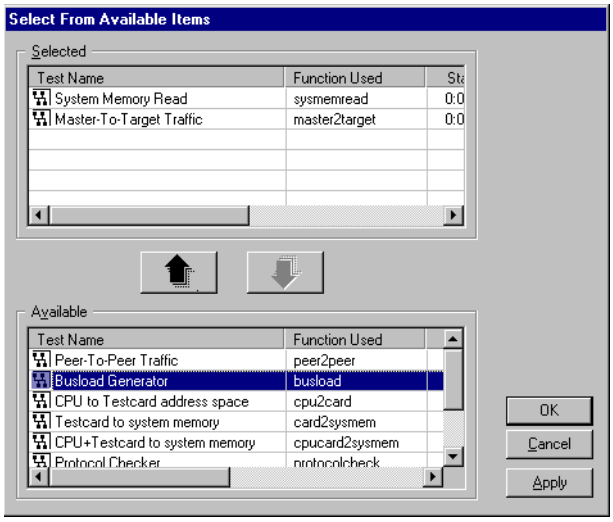
To include further scenarios in the navigator, select the SVP item and click either

- *Insert Scenario* in the shortcut menu, or
- *Insert New Scenario* in the *Edit* menu.

Select Tests in Scenarios

To select tests in a Scenario, select the scenario in the navigator and click *Select Test(s)* either from the shortcut menu or from the *Edit* menu.

In both cases, the *Select From Available Items* dialog box is opened. You can now move *Available* tests to the *Selected* tests container and vice versa.



View Scenario Settings

You can view settings of one scenario by clicking on the respective scenario in the navigator. This opens the Scenario Details window. This window gives information about the total duration of the current scenario and shows all selected tests.

View the Entire SVP Object

You can get an overview of the entire configuration by clicking on the SVP item in the navigator. This opens the SVP Object window.

Defining Test Functions

When starting the SVP software, you will find predefined tests in the *Tests Available* item of the navigator. You can either use these tests for your test configuration or you can create new tests. The new tests can use the same functions as the predefined tests.

To insert new tests:

1 Click the Tests Available item in the navigator.

2 Select *Insert Test* in the shortcut menu, or

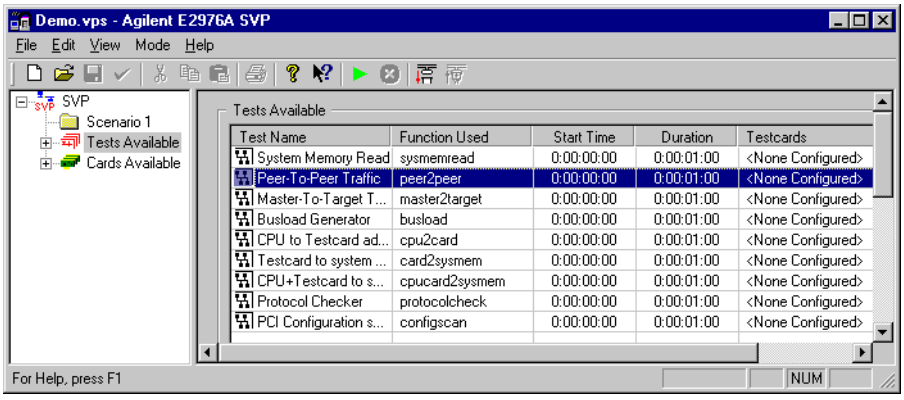
select *Insert New Test* in the Edit menu.

A new test item (for example, Test 1) with predefined settings appears in the navigator and the Tests Available window. The settings can be modified.

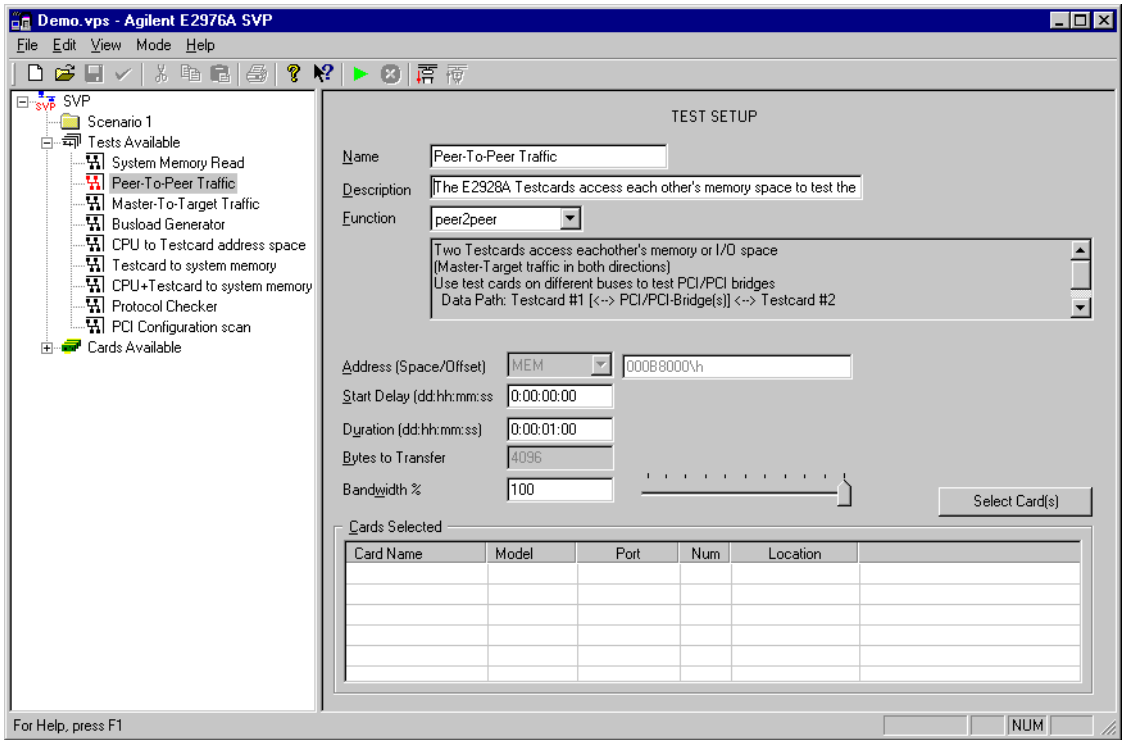
View Test Settings

To view the settings of one test, you can either

- open *Tests Available* in the navigator and click the respective test item, or
- click *Tests Available* in the navigator. This opens the Tests Available window where you can double-click the test you want to define.

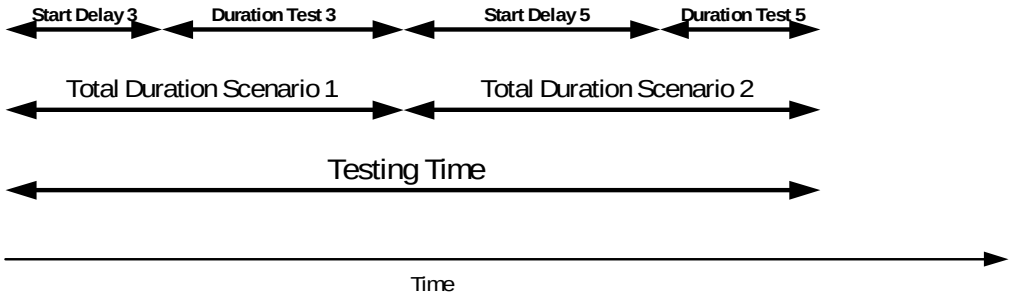


In both cases, the Test Setup window of the respective test is opened.



Here you can modify the current test settings (you can select another test function, for example) and specify the testing time.

The testing time is the sum of the total durations of all scenarios. The total duration is the maximum period of time of the start delay and duration defined for each test in this scenario. These values can be defined in the Test Setup window.



For detailed explanations of all test settings, refer to the *Agilent E2976A System Validation Package GUI Reference* (pdf-file).

Remove Available Tests

If there are tests in the Available Tests window that are not needed for the test configuration setup, you can remove them.

To remove tests from the Available Tests window:

1 Select the respective test in the *Tests Available* item in the navigator.

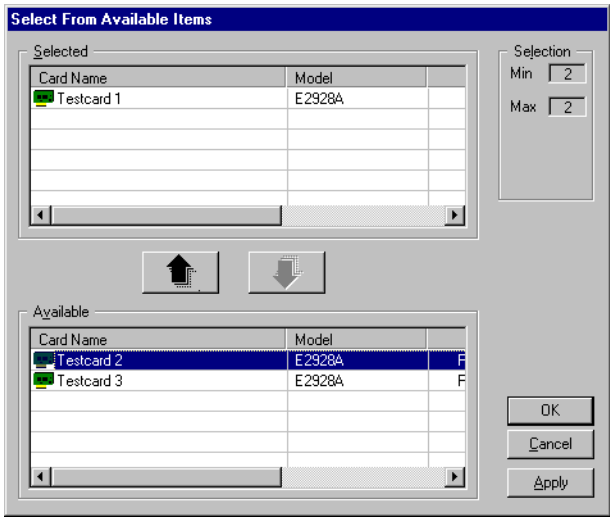
2 Select *Remove Test* in the shortcut menu or in the Edit menu.

Select Testcards

Each test needs a minimum number of testcards. To assign testcards to the currently selected test, you can either

- click *Select Card(s)* button in the current Test Setup window, or
- use the shortcut menu of the selected test and click *Select Card(s)*, or
- click *Select Card(s)* in the Edit menu.

In all cases, the *Select From Available Items* dialog box is opened. You can now move *Available* testcards to the *Selected* testcards container and vice versa.



For further information, refer to the *Agilent E2976A System Validation Package GUI Reference* (pdf-file).

For checking the testcard settings of all available testcards, refer to *"Check Testcard Settings" on page 56*.

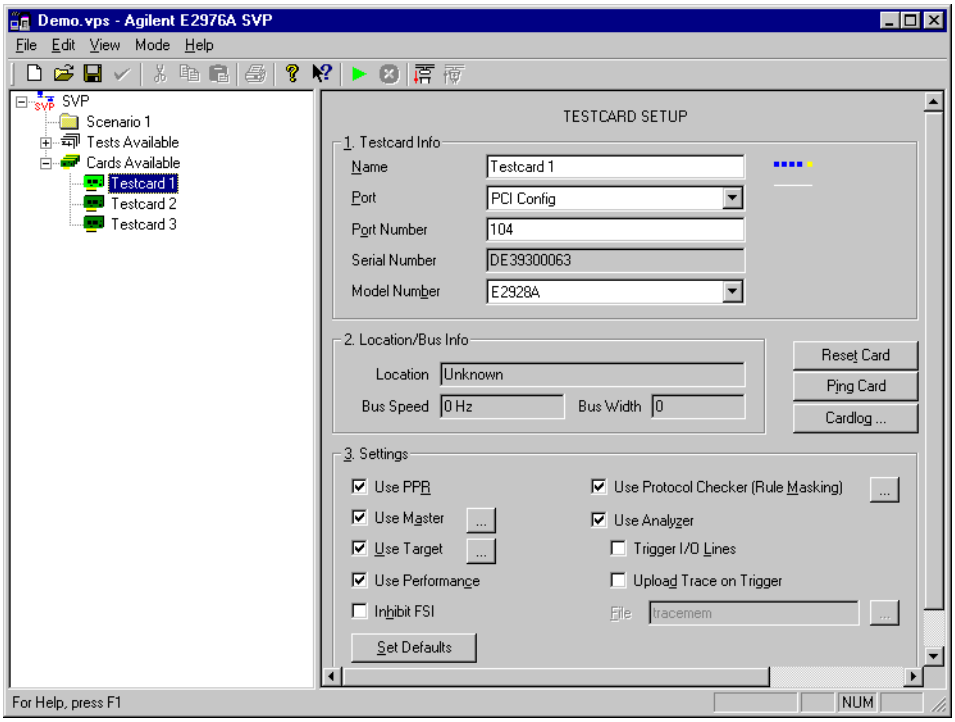
Check Testcard Settings

To view the settings of the available cards, you can either

- open *Cards Available* in the navigator and click the respective testcard, or
- click *Cards Available* in the navigator.

  This opens the Testcards Available window where you can double-click the testcard you want to check.

  In both cases, the Testcard Setup window of the respective testcard is opened.



The parameters in the *Testcard Info* group and the *Location/Bus Info* group can only be modified in offline mode.

You can modify all current testcard settings under *Settings*. Here you can enable and disable card features (PPR, master, target, performance, FSI, protocol checking and analyzer features). Clicking the details buttons next to the *Use Master* and *Use Target* check boxes shows the available testcard properties that can be modified.

Testcard Features

Master and Target

The testcard's exerciser consists of the master and target, and the data memory and data compare unit. Both master and target are programmed and set up automatically for the various tests (except protocol checker) and are not directly accessible as in the E2920 software. Behavior of the two parts can be specified by various property settings.

Analyzer

The testcard's analyzer part includes the protocol observer, trace memory, performance counters 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 specification and partly by Agilent. Each individual rule can be masked out and will then neither trigger the trace memory nor appear in any report. To mask rules, click the details button next to the *Use Protocol Checker (Rule Masking)* check box.

NOTE Because a testcard can be used in several scenarios, an automatic rule masking is performed on a per-test basis. That means that the user setting of the mask is restored prior to each test.

- Trace Memory

  Testcard's trace memory is set up to trigger on:

  – protocol violations

    Masked rules will not trigger the trace memory, and disabling of the observer disables triggering on any protocol rule.

  – data compare errors

  – bus hang

    Too many retries without transfers.

  – trigger-in event

    See external triggering, below.

- Performance Measurement

  To monitor system performance, each testcard measures two sets of performance metrics: one for the whole bus, and one for the performance of testcard transactions.

- External/Cross Triggering

  To facilitate triggering of external measurement devices, and to enable you 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. That is, 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; …
  ```

Estimating Test Durations

When the PPR feature is active (the *Use PPR* check box is selected), you can estimate test durations by using PPR reports for master and target testcard settings.

For that purpose, view the respective reports by opening the *Card Settings* dialog boxes and clicking the *Check Syntax* button. If the syntax is ok, the PPR report is displayed. Here you can find the parameters (for example, the estimated testing time for one attribute page) needed to estimate the duration of one test function. (Test durations are set in the Test Setup window.)

For further information on testcard settings, refer to the *Agilent E2976A System Validation Package GUI Reference* (pdf-file).

Running the Test

After you have successfully set up your test configuration, all information that is known about the software, the system, the test setup and the type of test hardware is listed in the static report. You can view this report by clicking *Static Report …* in the View menu.

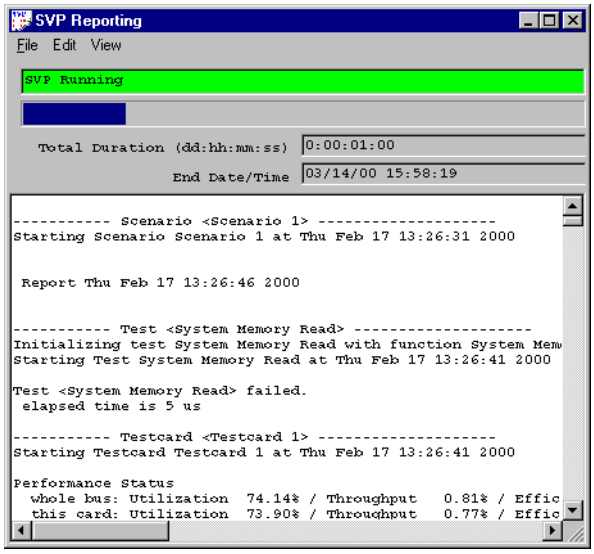
Before you run a test ensure that you are in online mode. If not, click *Go Online* in the *Mode* menu.

To run the test configuration, click either

- the *Run* icon in the toolbar, or
- *Run* in the shortcut menu, or
- *Run* in the File menu.

When the test is started, the SVP software automatically selects the SVP object in the navigator and opens the SVP Reporting dialog box. You can now view the status of the current test and the test report.

For details on the test report, refer to the *Agilent E2976A System Validation Package GUI Reference* (pdf-file).



Status of the Testing

The following messages can appear in the status line of the SVP Reporting dialog box:

SVP Initializing Test The specified test is currently being initialized.

SVP Running The test is running.

SVP Stopped The test has finished. If errors occurred, they are listed in the test report.

SVP Error! The test has stopped due to a detected error. The error is listed in the test report.

Error Types

The *CardLog* report provides additional information on the type of errors that have been detected. Possible errors are protocol violations, errors in data comparison or master abort conditions. These errors will also be reported in the test report.

Performance

You can also view the performance of the system under test during the test run. The performance is displayed in terms of utilization, data throughput and efficiency for the whole bus and for the current card. These values are displayed during run time, but are not stored in any report file. For detailed explanations of these measures, refer to *Predefined Performance Measures* in the *Analyzer User's Guide* (pdf-file), which is delivered with the testcard.

Test Results

This section briefly explains the results of the system tests provided with the Agilent E2976A System Validation Package. It covers the contents of the different output files as well as some error handling instructions.

For information on:

- the log file, refer to the *"Log File Description" on page 61*.
- the PPR report files, refer to the *"Report File Description" on page 62*.
- the error handling, refer to *"Error Handling" on page 63*.

Log File Description

The log file is opened when the System Validation Package is started. The default name is svp.log and the default location is the reports subdirectory in the SVP directory.

View the Log File

To view the log file, click the *View Log …* button in the SVP Object window.

Log File Contents

The log file includes:

- The start date and time of the test session.
- All tests that were run during the session including their
  - test start date and time,
  - type of test,
  - test termination with the results, either *passed* or *failed*,
  - the elapsed time of the test.

- All testcards that are used during the session including their
  - start date and time,
  - testcard status (performance, observer, trace memory trigger),
  - testcard configuration.
- Test results for each testcard after a scenario has been finished including
  - maximum performance,
  - protocol checker results.
- Total elapsed time for each scenario.

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 about all the PPR functions that were called during the tests.

Report Files

The PPR report files are:

- Master Block file

  This file contains the settings of all generic PPR properties and master block permutation properties.

- Master Attribute file

  This file contains the settings of all generic PPR properties and master attribute permutation properties.

- Target Attribute file

  This file contains the attribute settings for the target.

The PPR report file names can be defined in the testcard settings (see *Testcard Setup Window* in the *Agilent E2976A System Validation Package GUI Reference* (pdf-file)).

Report File Contents

The contents of the PPR report files are fairly complex and you should not need to open them to interpret 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 explanation of the report file contents, please refer to the *Agilent E2975A PCI Protocol Permutator & Randomizer Software User's Guide*.

Error Handling

This section is not meant to be a complete troubleshooting guide. It lists some of the errors that might occur on some systems and that are relatively easy to handle.

Test Function Errors

Every test function within the test scenarios uses the PCI Analyzer to trigger at certain error events. The different types of errors that may occur with this tool are:

- PCI protocol errors

  One of the PCI protocol rules is violated.

- Data compare errors

  An error has occurred in the transferred data.

- Master abort condition

The GUI allows you to define whether the tests continue after an error. This can be done in the SVP Test Settings dialog box, which can be accessed via the SVP Object window.

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

For example, if you have problems with the communication over a PCI/PCI bridge using the "Peer-to-Peer Traffic" test, you could examine both directions separately with two "Master-to-Target Traffic" tests to find out if the errors occur in one direction only.

Testcard Errors

One of the following testcard errors may occur:

- Testcards detection errors

  When the program is started, it scans the system under test for Agilent testcards.

  However, you can reset these testcards to factory defaults—including both memory and I/O space—in the Testcard Setup window. In both cases these testcards are excluded from tests until the system is rebooted.

  For more information on the handling of testcard detection errors, please refer to the C-API description in the user's guide (pdf-file), which is delivered with the respective testcard.

- Errors during test run

  If the error occurred during test run, the *Cardlog* in the program window displays the status *Error*. The function that caused the error is logged in the SVP Reporting dialog box where the Cardlog is displayed. In this case, too, a card reset and reboot may help.

  For more information on the handling of errors during test run, please refer to the C-API description in the user's guide (pdf-file), which is delivered with the respective testcard.

- PPR errors

  If the testcard error was detected by the PPR, its name starts with `B_E_PPR`. Please refer to the *Agilent E2975A PCI Protocol Permutator & Randomizer Software User's Guide* (pdf-file), which is delivered with the testcard.

Common Protocol Errors

The Analyzer of the Agilent 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).

In the system default configuration, these protocol errors will also terminate the test, even if there is no problem on the bus.

To keep these protocol errors from terminating a test for a particular protocol rule:

1. Determine the testcard(s) that detected the error and open the respective Testcard Setup window, for example by clicking on this testcard in the navigator.
2. Click the details button next to the selected *Use Protocol Checker (Rule Masking)* check box to open the Protocol Rule Masking dialog box.
3. Disable the respective rule by toggling its enabled/disabled field.
4. Repeat these steps for all testcards that detected this error.

Setup File Reference

All settings of the System Validation Package 2.0 can easily be set or modified by the user via the Graphical User Interface (GUI). The settings have reasonable default values.

The settings file has the following structure:

- Test configuration

  Information about the software version, inserted scenarios, operation mode, name of the log file, testcard identifier, title and purpose of the test configuration.

- Test sequence configuration

  Information about inserted scenarios and tests.

- Settings of the available tests

  Test settings are introduced by the name of the test in brackets (for example, `[Test System Memory Read]`).

- Settings of the available testcards

  Testcard settings are introduced by the testcard name in brackets (for example, `[Testcard Testcard 1]`).

You can save your GUI configuration settings to a VPS file. This file can be loaded by using the *Load* feature in the GUI.

The VPS file is in plain ASCII format and can be edited with any standard text editor.

For more information, especially on PPR settings, please refer to the *Agilent E2975A PCI Protocol Permutator & Randomizer Software User's Guide* (pdf-file), which is delivered with the testcard.

Settings File Formats

The settings file contains a set of properties, following the syntax below:

```
<settings file>     :    <header> <scenario info>+ <test info>+ <card info>*
<header>            :    <setting>+
<scenario info>     :    "[scenario" <scenario name> "]" <setting>+
<test info>         :    "[test" <test name> "]" <setting>+
<card info>         :    "[testcard" <card name> "]" <setting>+
<scenario name>     :    unquoted string
<test name>         :    unquoted string
<testcard name>     :    unquoted string
<setting>           :    <property name> "=" <property value>
<property name>     :    see tables below
<property value>    :    <DWORD value> | <string value> | <address value> |
                         <boolean value>
<DWORD value>       :    hex format (0x1234 or 1234\h), decimal format (1234),
                         binary format (011\b)
<string value>      :    quoted string
<address value>     :    "<" ( "MEM" | "IO" | "CONF" ) ":" [<hi-addr>] <hi-addr> ">"
```

Example

```
[Test System Memory Read]
starttimeoffset = 0\h
duration = 3c\h
address.space = mem
function = "sysmemread"
address = <MEM:000B8000\h>
bandwidth = 64\h
size = 1000\h
description = "This is the text"
items.list = "Testcard 1"
```

Scenario and Test Parameter

Scenario Property

Scenarios allow several tests to be run concurrently. Any testcard can only be used once per scenario. Scenarios have no special settings except for the list of tests that are used.

Scenario Property used in the GUI	Scenario Property used in the Settings File	Values	Description
Scenario	items.list	string list	List of tests (by names) that are used in this scenario

Test Properties

All tests share some or all of the following properties:

Test Property used in the GUI	Test Property used in the Settings File	Values	Description
Address Offset	address	address value	Physical address of the memory
Address Space	address.space	“mem” or “io”	Memory space or I/O space is used
Bytes to Transfer	size	DWORD	Size of the memory (in bytes) that is use.
n/a	address.prefetch	True or False	The pre-fetchable decoder is used if available
Bandwidth %	bandwidth	0 … 100	Value of the maximum bandwidth. **Note:** Using a bandwidth < 1.0 will cause that the PPR testcard setting B_M_DELAY is overridden.
Description	description	string	User-defined description
Function	function	string	Short name of test function
Start Delay	starttimeoffset	DWORD	Start Delay in seconds (from start of scenario)
Duration	duration	DWORD	Duration of the test (in seconds)

Testcard Parameters

Testcard parameters can be divided into:

- Testcard and Location Information
- Card Features Settings
- Master Settings
- Target Settings
- Protocol Checker (Rule Masking)

Testcard and Location Information

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
Serial Number	card.serialnumber	string	valid serial number	Serial number of the testcard
Port	connection.port	port string	rs232 | fhif | pci	Connection port of the testcard
Port Number	connection.portnum	DWORD	depends on port	Connection port number
Model Number	card.model	string	valid models	Model number of the testcard
Location	card.location	string	valid locations	Location of the testcard (for example, Bus 0 Device 13 Function 0)

Card Features Settings

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
Use Performance	use.performance	boolean	True or False	Performance counters of the testcard are enabled or disabled
Use Protocol Checker (Rule Masking)	use.observer	boolean	True or False	Protocol Observer of the testcard is enabled or disabled
Use Master	use.master	boolean	True or False	Master of the testcard is enabled or disabled
Use Target	use.target	boolean	True or False	Target of the testcard is enabled or disabled
Use PPR	use.ppr	boolean	True or False	Attribute permutation is enabled or disabled
Use Analyzer	use.tracememory	boolean	True or False	Analyzer of the testcard is enabled or disabled
Use Trigger I/O Lines	use.triggerio	boolean	True or False	Cross-triggering is enabled or disabled
Upload Trace on Trigger	tracememory.upload	boolean	True or False	Trace memory upload is enabled or disabled
Inhibit FSI	inhibit.fsi	boolean	True or False	Inhibit connection to FSI on host
File	tracememory. upload.file	string		Name of the file to which the card's trace memory is written (without extension)
n/a	performance.measure	DWORD	0 … 7	Measure used by performance
n/a	performance. cardmeasure	DWORD	0 … 7	Measure used by card's performance

Master Settings

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
n/a	master.blockpage	DWORD	0 … 16	Blockpage used by the master
Write Command	master.block.cmd.write	DWORD	mem_write, mem_ writeinvalidate	PCI bus command for the block transfer (mem)
Read Command	master.block.cmd.read	DWORD	mem_read, mem_readline, mem_ readmultiple	PCI bus command for the block transfer during address phase (mem)
Master Internal Address	master.address.internal	DWORD	0 … size of data memory	Internal address of the testcard's data memory; used by the master

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
Block Size List (Master PPR)	ppr.master.block.size.list	string	Multiple of 4 in the range of 4 … 128k	List of numeric values for block sizes, measured in bytes
Block Algorithm (Master PPR)	ppr.master.block.alg	DWORD	0 … 3	Algorithm used to pick values from the value list of master block properties
Block Byte Enable List (Master PPR)	ppr.master.block.byten.list	string	0 … 15	List of numeric values for C/BE byte enables
Block Commands List (Master PPR)	ppr.master.block.cmds.list	DWORD	0 … 15	List of PCI bus commands used for permutations
Block Alignment List (Master PPR)	ppr.master.block.align.list	string	*Granularity:* Power of 2 between cache-line size and 8192. *Offset:* Multiple of 4 between 0 and 8188	Granularity and offset within this granulariy that restrict the start of a block
PPR Report (Master Block)	ppr.master.block.report	boolean	True or False	Writing of the PPR report is enabled or not
PPR Block Report File (Master Block)	ppr.master.block.report.file	string		Name of the file used for the PPR report or master block permutations
Master Attribute Page (memory)	master.attrpage.memory	DWORD	0 … 63	Attribute page used for access to the memory space by testcard
Master Attribute Page (i/o)	master.attrpage.io	DWORD	0 … 63	Attribute page used for access to the I/O space by testcard
Waits List (Master PPR Attribute)	ppr.master.attr.waits.list	string	0 … 30	List of numbers of waits
Burst Length List (Master PPR Attribute)	ppr.master.attr.last.list	string	0 … 2^{32}	List of last phases of bursts (this is, burst lengths)
Release Request (Master PPR Attribute)	ppr.master.attr.rreq.list	string	0 … 15	List of number of cycles after which REQ# is released after assertion of FRAME#
DPERR List (Master PPR Attribute)	ppr.master.attr.dperr.list	string	0 or 1	List of parity errors, signaled or not signaled
SPERR List (Master PPR Attribute)	ppr.master.attr.dserr.list	string	0 or 1	List of system errors in the data phase, signaled or not signaled
APERR List (Master PPR Attribute)	ppr.master.attr.aperr.list	string	0 or 1	List of system errors in the address phase, signaled or not signaled
DWRPAR List (Master PPR Attribute)	ppr.master.attr.dwp.list	string	0 or 1	List of wrong parities set one clock after a write data transfer, inverted or not inverted
AWRPAR List (Master PPR Attribute)	ppr.master.attr.awp.list	string	0 or 1	List of wrong parities set one clock after the address phase, inverted or not inverted

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
WAITMODE List (Master PPR Attribute)	ppr.master.attr. waitmode.list	string	0 or 1	List of values to keep the address constant during the WAITS phases or not
STEPMODE List (Master PPR Attribute)	ppr.master.attr. stepmode.list	string	0 or 1	List of values to keep the address constant during the STEPS phases or not
STEPS List (Master PPR Attribute)	ppr.master.attr. steps.list	string	0 or 1	List of numbers of additional clocks during an address phase They are added between assertion of GNT# and assertion of FRAME#.
TRYBACK List (Master PPR Attribute)	ppr.master.attr.tryback. list	string	0 or 1	List of Fast Back-to-Back cycle tries
DELAY List (Master PPR Attribute)	ppr.master.attr.delay.list	string	2 … 2^{21}	List of numbers of clocks a master transaction is delayed before its start **Note:** Delay will be modified if bandwidth < 100 % (specified in test).
REQ64 List (Master PPR Attribute)	ppr.master.attr.req64.list	string	0 or 1	List of 64-bit transfer tries
AWRPAR64 List (Master PPR Attribute)	ppr.master.attr.awp64. list	string	0 or 1	List of wrong parities (PAR64) set one clock after the address phase, inverted or not inverted
DACWRPAR (Master PPR Attribute)	ppr.master.attr.dacwp. list	string	0 or 1	List of wrong parities signaled in the second cycle of a dual address cycle, inverted or not inverted
DACWRPAR64 (Master PPR Attribute)	ppr.master.attr. dacwp64.list	string	0 or 1	List of wrong parities (PAR64) signaled in the second cycle of a dual address cycle, inverted or not inverted
DACPERR List (Master PPR Attribute)	ppr.master.attr.dacperr. list	string	0 or 1	List of system errors in the second cycle of a dual address cycle, signaled or not signaled
DWRPAR64 List (Master PPR Attribute)	ppr.master.attr.dwp64. list	string	0 or 1	List of wrong parities (PAR64) set one clock after a write data transfer, inverted or not inverted
RESUMEDELAY List (Master PPR Attribute)	ppr.master.attr. resumedelay.list	string	2 … 127	List of clock numbers after which the master resumes after a target termination
PPR Report (Master Attribute)	ppr.master.attr.report	boolean	True or False	Writing of the PPR report is enabled or not
PPR Report File (Master Attribute)	ppr.master.attr.reportfile	string		Name of the file used for the PPR report of master attribute permutations

Target Settings

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
n/a	target.attrpage	DWORD	0 … 63	Attribute page used by the target
Termination List (Target PPR Attribute)	ppr.target.attr.term.list	string	0 … 3	List of termination modes, for example, “32*noterm, 2*retry, disconnect, abort”
WAITS List (Target PPR Attribute)	ppr.target.attr.waits.list	string	0 … 30	List of number of waits
DPERR List (Target PPR Attribute)	ppr.target.attr.dperr.list	string	0 or 1	List of parity errors, signaled or not signaled
SPERR List (Target PPR Attribute)	ppr.target.attr.dserr.list	string	0 or 1	List of system errors in the data phase, signaled or not signaled
APERR List (Target PPR Attribute)	ppr.target.attr.aperr.list	string	0 or 1	List of parity errors in the address phase, signaled or not signaled
WRPAR List (Target PPR Attribute)	ppr.target.attr.wp.list	string	0 or 1	List of wrong parities set one clock after a write data transfer, inverted or not inverted
ACK64 List (Target PPR Attribute)	ppr.target.attr.ack64.list	string	0 or 1	List of 64-bit requests, acknowledged or not acknowledged
Target Attribute DACPERR List	ppr.target.attr.dacperr.list	string	0 or 1	List of address parity errors, signaled or not signaled
Target Attribute WRPAR64 List	ppr.target.attr.wp64.list	string	0 or 1	List of wrong parities set one clock after a write data transfer, inverted or not inverted
Target Attribute Report	ppr.target.attr.report	Boolean	True or False	Writing of the PPR report is enabled or not
Target Attribute Report File	ppr.target.attr.reportfile	string		Name of the file used for the PPR report or target attribute permutations

Protocol Checker (Rule Masking)

Card Property used in the GUI	Testcard Property used in the Settings File	Type	Range	Description
Protocol Rule Masking State Disabled (lower bits)	protocolrule.mask.lo	DWORD	32	Masked protocol rules (bit 0 … 31).
Protocol Rule Masking State Disabled (higher bits)	protocolrule.mask.hi	DWORD	20	Masked protocol rules (bit 32 … 51).
Mask Rule(s) After x Occurrences	protocolrule.mask. count	DWORD	0 … $(2^{32} - 1)$	Number of occurrences the rule is masked.

Index

V

Publication Number: 5988-4906EN

Agilent Technologies