Data Sheet

Vanguard PCI/PCI-X

Bus Analyzer & Exerciser for PCI, PMC & CompactPCI

The Vanguard family of Bus Analyzers for PCI is a complete solution for analysis, exercising, protocol checking and compliance testing in a PCI, PCI-X, PMC and CompactPCI environment. The product family includes a 10/100 Mbit/s Ethernet interface, taking the power and flexibility found only in Curtiss-Wright Controls Defense Solutions' range of Bus Analyzers to unprecedented levels of productivity.

Networked PCI Bus Analyzer (built-in Ethernet port)

Allows the user to connect to the Vanguard analyzer at any location where a network is available

Supports PCI 0-66MHz, PCI-X 0-133MHz

Same analyzer for current and future projects

2M Sample Trace Buffer @ 256-bits

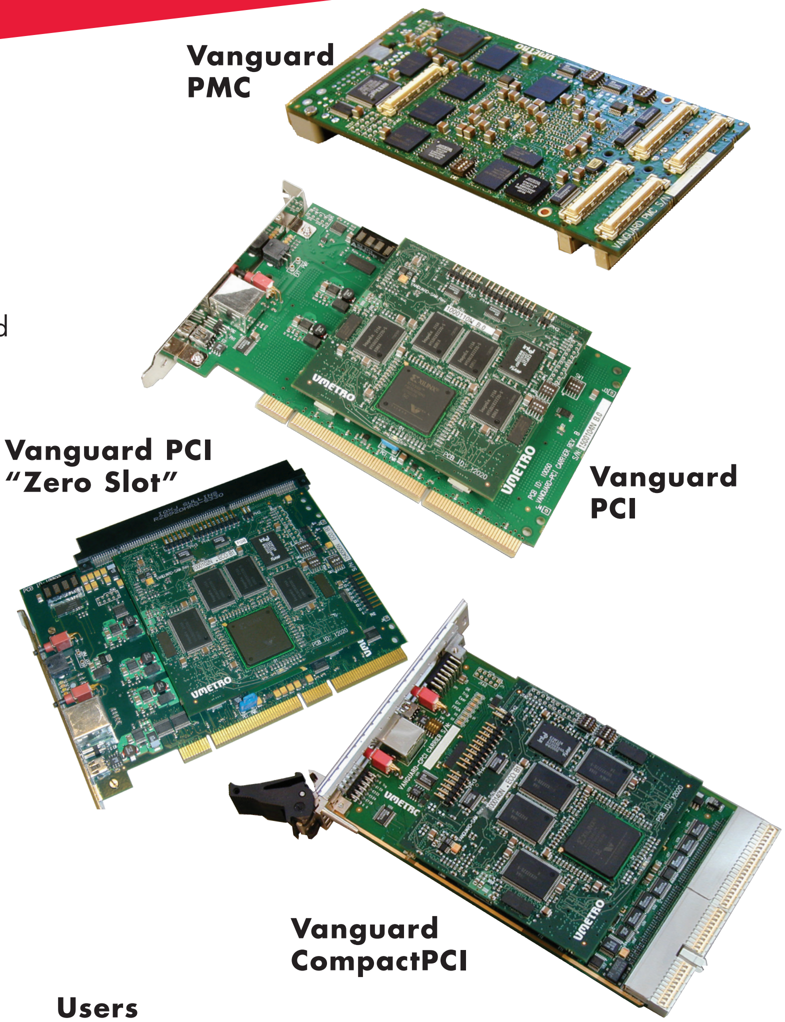
Simplifies error location by giving access to a large set of sample data

All functions can be used concurrently and independently

For maximum flexibility and productivity

Multi-level Trace Viewer

Makes it possible for software and hardware engineers to view the data in their individual preferred views based on the same captured data.



Users

- Software developers involved in I/O drivers, operating systems, etc.
- System integrators putting together equipment from various vendors
- Hardware designers of interface chips, motherboards and add-in cards

Learn More
Web / cwcdefense.com/sales
Email / defensesales@curtisswright.com

ABOVE & BEYOND

CURTISS WRIGHT Controls
Defense Solutions
cwcdefense.com

Introduction

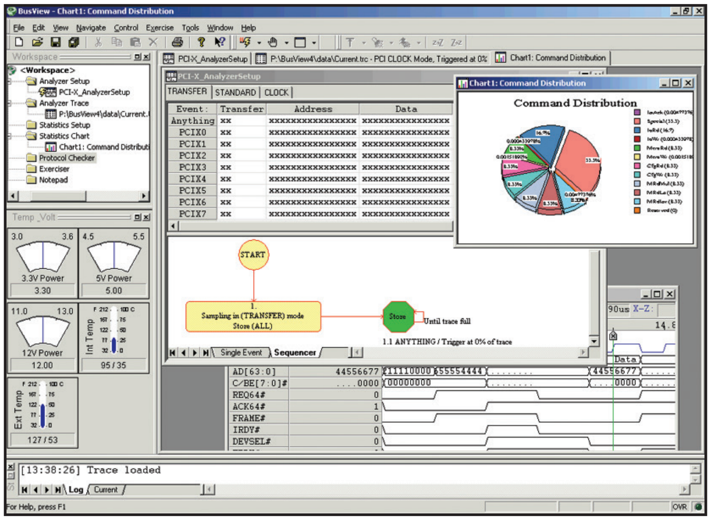
The Most Advanced Tool for PCI-X & PCI Debugging

The Vanguard PCI family of PCI, PMC and CompactPCI (cPCI) Bus Analyzers is designed for debugging, testing and validation of next-generation PCI-X and PCI boards and systems. The analyzer is able to capture and display all bus activity in PCI-X / PCI based systems up to 133MHz with powerful trigger and store qualifiers, and offers extensive statistics functions to measure system performance. At the same time, the optional exerciser unit may act as a PCI-X or PCI master or target controlled through the user-interface or with a built-in script capability, for generation of bus traffic and emulation of not yet available devices. Furthermore, the optional Protocol Checker automatically detects 71 PCI-X or 45 PCI protocol errors, and can operate concurrently with the state analyzer, exerciser and other functions. Optional functionality is unlocked via software keys. Utilizing all of the capabilities of the Vanguard, the Compliance Test Suite performs automated testing relative to the PCI/PCI-X specification.

The unit is controlled via USB or Ethernet from a PC running Windows® and Curtiss-Wright's BusView™ Graphical User Interface (GUI). The Ethernet connection adds a world of possibilities to the Bus Analyzer tool, by allowing the user to connect to the Vanguard analyzers anywhere a network is available.

As with all Curtiss-Wright analyzers, the unit may be powered from the target system or from an external power supply (except the Vanguard cPCI), and extensive on-line help is available.

Figure 1: The BusView GUI



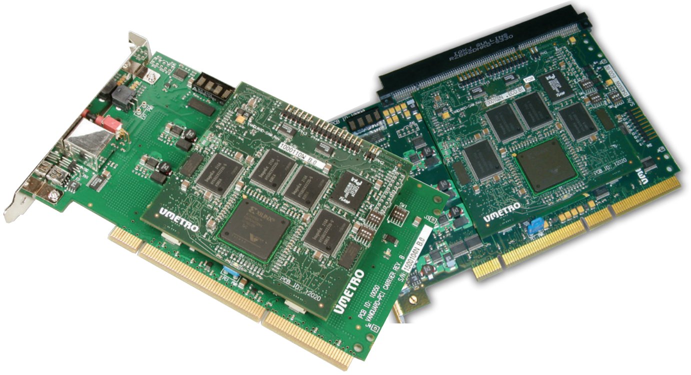
Event:	Transfer	Address	Data
Anything	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX0	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX1	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX2	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX3	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX4	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX5	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX6	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx
PCIX7	xx	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx

```
AD[63:0]        44556677  11110000  55554444  ........  44556677
C/BE[7:0]#      ....0000  00000000  ........  ....0000
REQ64#                 0
ACK64#                 1
FRAME#                 0
IRDY#                  0
DEVSEL#                0
```

```
[13:38:26] Trace loaded
```

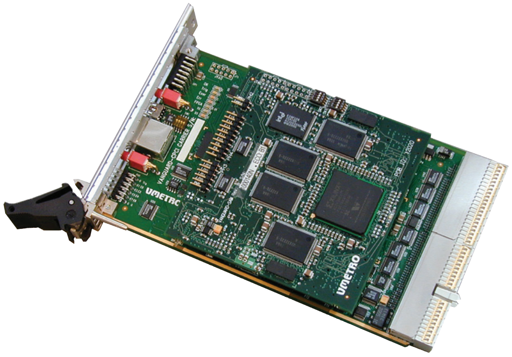
PCI

- PCI/PCI-X compliant design
- PCI Short Card form factor
- Dedicated version for single slot systems



CompactPCI

- Functions as System Slot Controller or Peripheral card
- Includes arbitration and clock generation capabilities
- Optional 6U adapter (CPCI-3U6U-ADA)



PMC

- Top spacer card allows testing in single slot systems
- May also be used in PCI and cPCI systems with adapters

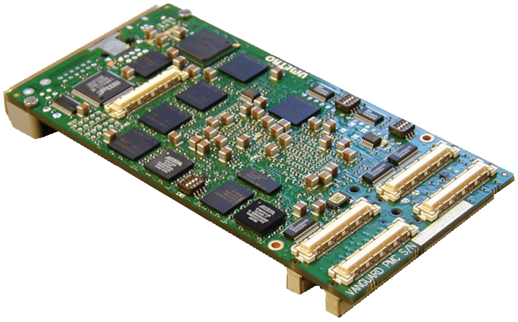


Figure 2: PCI

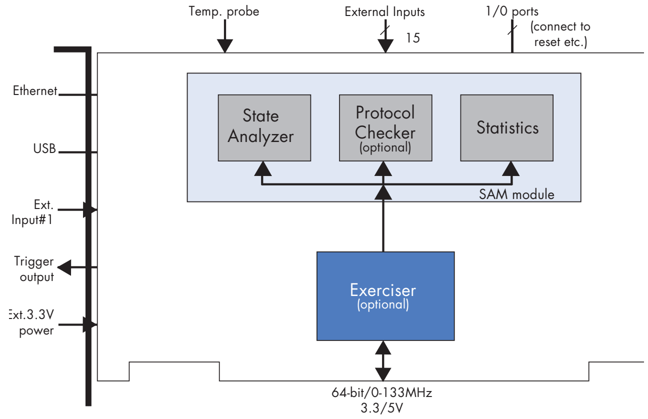


Figure 4: cPCI

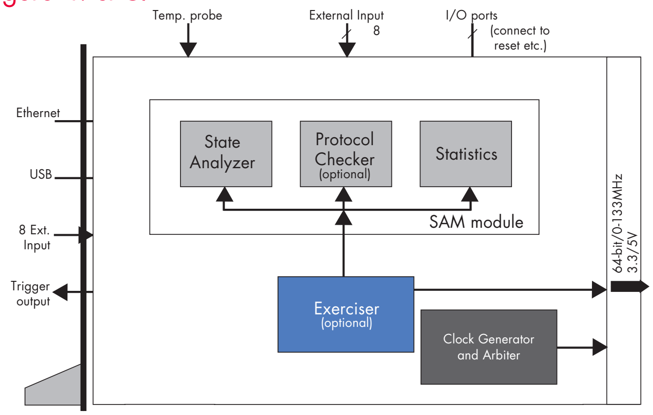


Figure 3: PMC

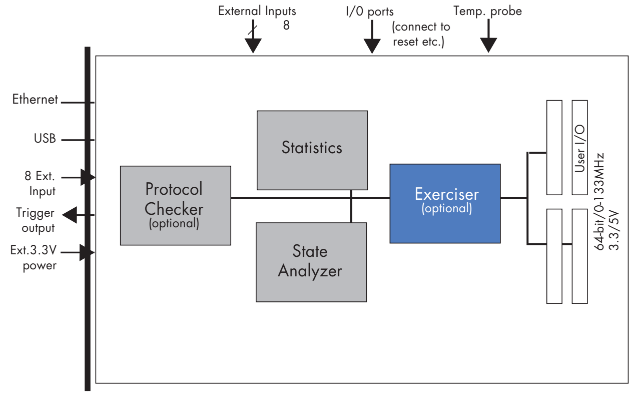
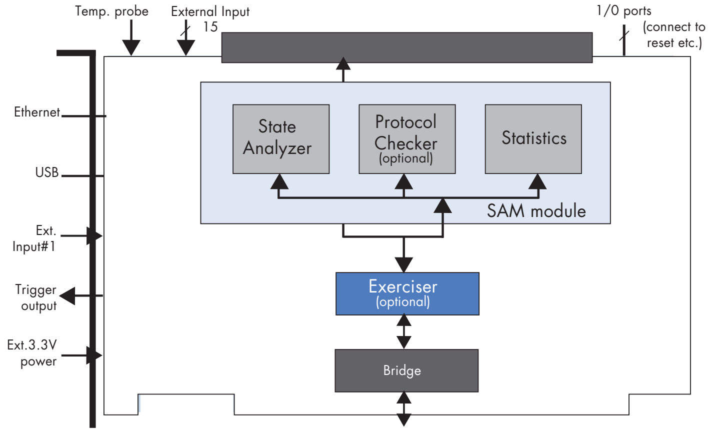


Figure 5: PCI0SL - "Zero Slot"



Modular Concept

The Vanguard product line consists of three different form factors: PCI, PMC and cPCI. All three form factors support the PCI and PCI-X protocol and signaling. The PCI and cPCI form factors share a common hardware module called the State Analyzer Module (SAM), which is interchangeable between the PCI, PCI0SL and cPCI carrier boards. For the PMC form factor, a single fully integrated module has been designed. The unique design allows the Vanguard PMC Networked Bus Analyzer to carry another PMC module on top which enables analysis of the PCI/PCI-X bus without having to remove a PMC module. The PMC P4 connector is also passed through, to accommodate PMC modules with rear I/O (e.g. to P2ac on VME boards). The Vanguard cPCI can function both as a peripheral card and as a system slot controller card. The VG-PCI0SL Carrier is specifically designed for testing in 133MHz PCI-X systems and other PCI-X systems where only a single plug-in slot is available. The Carrier uses a bridge to isolate the host bus from the secondary bus available for the device under test (DUT).

Network Connection

With Ethernet connectivity, the Vanguard products are able to connect a Bus Analyzer to the intranet or internet making the tool more productive, and introduce new applications for Bus Analyzers as described later.

The Vanguard will automatically request an IP address from the network's DHCP server if present (if not, an APIPA service will be tried as described on page 4). The Vanguard will be available for connection. The user will be able to detect all analyzers present on the local network from a host PC running Windows where the Networked Bus Analyzer GUI is installed (as shown in the figure above).

In most if not all network architectures (including a point-to-point cross-over Ethernet cable connection between a host PC and a networked Bus Analyzer), simply connecting

Figure 6: Network Connection

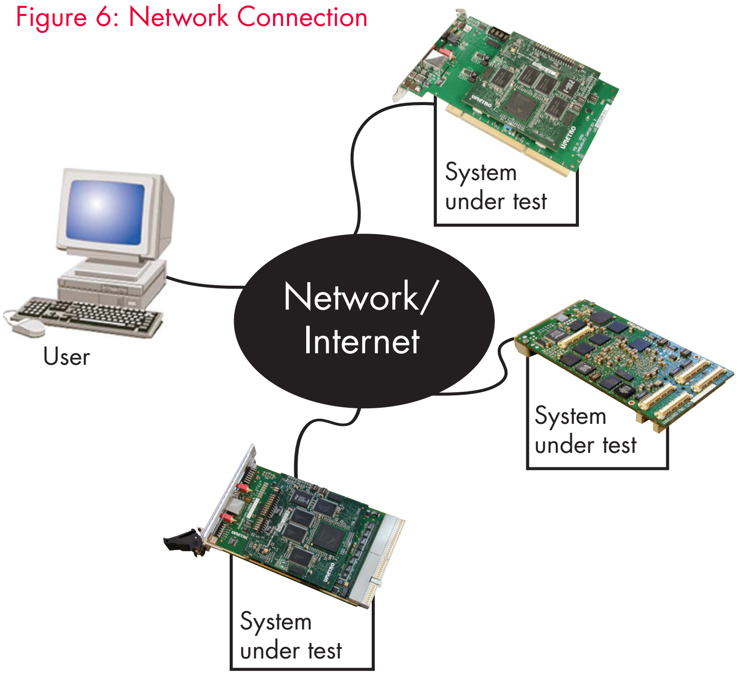
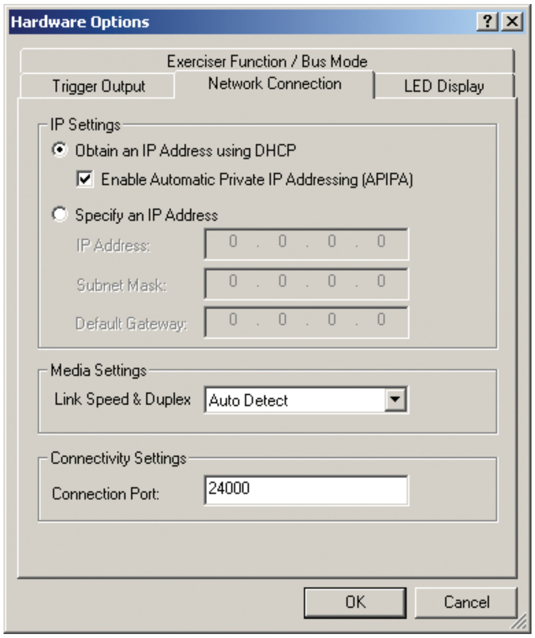


Figure 7: Network Connection Settings



Hardware Options

IP Settings

● Obtain an IP Address using DHCP

☑ Enable Automatic Private IP Addressing (APIPA)

○ Specify an IP Address

IP Address: 0 . 0 . 0 . 0

Subnet Mask: 0 . 0 . 0 . 0

Default Gateway: 0 . 0 . 0 . 0

Media Settings

Link Speed & Duplex: Auto Detect

Connectivity Settings

Connection Port: 24000

an analyzer to the network, inserting it into a system and powering up that system are all that is needed to make it available to all users present on the local network.

It may also be desirable for the user to configure the Vanguard to have a fixed IP address in the network. This is especially useful if the user is accessing the Vanguard via the Internet, and has to go through a Firewall. In order to access the Vanguard inside a Firewall, the Firewall must be configured to route accesses to a specific port number to an IP address inside the network.

APIPA

Automatic Private IP Addressing (APIPA) is a service offered by Windows operating systems that allows a computer to automatically assign itself an IP address in networks where no Dynamic Host Configuration Protocol (DHCP) server is present. APIPA also allows a networked Bus Analyzer to be connected directly to a host computer using a cross-over Ethernet cable with no configuration necessary.

USB

In addition to an Ethernet port, the Vanguard products are equipped with a USB (Universal Serial Bus) port.

Table 1: Network Connection Settings

Usage	Comment / Benefit
Delocalized connection	The system under test and the user are connected via a network, so the user is free to choose a location that better suits his tasks than being situated in a lab. A software engineer would typically choose to be in his own office.
Sharing of equipment	Since no physical cables need to be swapped, all that is required for one user to give another user access to a Networked Bus Analyzer is for him to close down his GUI connection.
Deployed equipment	A Networked Bus Analyzer can be installed in a deployed system (for example a base station or a radar processing unit), offering an engineer or service technician access to direct measurements on the running system as well as temperature and voltage readings.
Testing in inaccessible	Some systems that are being tested or environments monitored are located in specific areas where they are inaccessible (most military vehicles, airplanes and vessels have these areas).
Remote connection	Being able to connect to remote systems allows collaboration work to take place without artificial limitations. A user in Europe can access equipment in a lab in California just as easily as that for local equipment.

The addition of network connectivity to Bus Analyzers opens up powerful possibilities.

Advanced Trigger & Store Capabilities

The Vanguard products use a text or graphical trigger sequencer to specify triggers, store and count and delay qualifiers in an "If-then-else…" fashion. Together with the intuitive demultiplexed sampling method (which puts Command, Address, Attribute, Data and Status as separate items also in the trigger event specifiers) and true inside/outside range specifiers on address, attribute and data fields, the user can quickly and easily create sophisticated triggers and store qualifiers. The user can store trigger setups on the host PC. In sum, all these features allow the user to solve problems quickly, instead of spending precious time on trying to figure out how to set up and understand the analyzer.

Independent Sub-functions for Concurrent Operation

As can be seen from the block diagrams on page 3, the Vanguard Bus Analyzer products contain a comprehensive set of independent functional units. This means that there are separate hardware resources for the various functions so that the user can operate the Exerciser, Protocol Checker, Statistics and State Analyzer concurrently. This allows for the monitoring of bus traffic results generated by the Vanguard product and other bus agents simultaneously. This extensive use of hardware resources enables maximum debugging efficiency for the user.

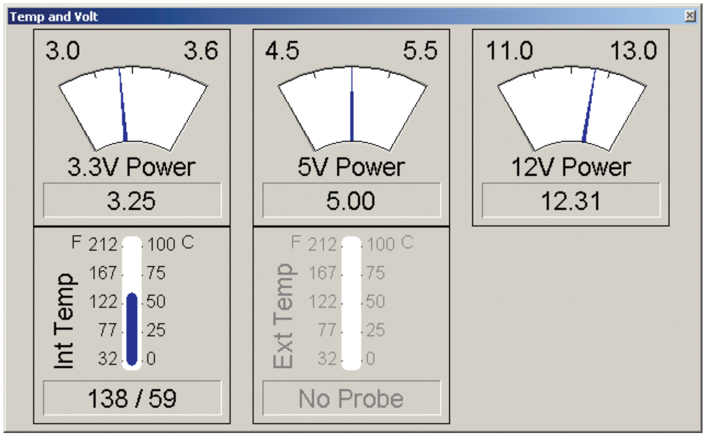
Additional Optional Functionality

The Vanguard Product Range is based on a field programmable architecture. A user can purchase a Vanguard basic model (including the State Analyzer and Statistics) and later add the Exerciser (-E or -E2) by purchasing a license key for the BusView software. Additionally, the field programmable architecture allows Curtiss-Wright to add features via software updates and a license key. BusView can automatically check for and download updates.

Monitoring Temperature & Voltages

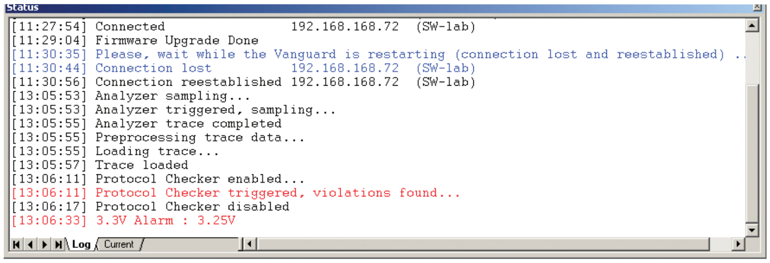
When debugging systems both locally and remotely, some hard-to-debug problems may be related to voltage drops and/or temperature variations. The Vanguard family of Bus Analyzers offers the ability both to monitor voltages and temperature, and to generate alarms if either voltages or temperatures fall outside a specified range.

Figure 8: Temperature & Voltage Monitoring



The Vanguard Analyzer family is often used for complex monitoring and debugging applications. In order to track events such as temperature alarms, PCI resets, protocol violations and triggers, a status log window is available.

Figure 9: Status Window



```
[11:27:54] Connected               192.168.168.72  (SW-lab)
[11:29:04] Firmware Upgrade Done
[11:30:35] Please, wait while the Vanguard is restarting (connection lost and reestablished) ..
[11:30:44] Connection lost         192.168.168.72  (SW-lab)
[11:30:56] Connection reestablished 192.168.168.72  (SW-lab)
[13:05:53] Analyzer sampling...
[13:05:53] Analyzer triggered, sampling...
[13:05:55] Analyzer trace completed
[13:05:55] Preprocessing trace data...
[13:05:55] Loading trace...
[13:05:57] Trace loaded
[13:06:11] Protocol Checker enabled...
[13:06:11] Protocol Checker triggered, violations found...
[13:06:17] Protocol Checker disabled
[13:06:33] 3.3V Alarm : 3.25V
```

State Analyzer

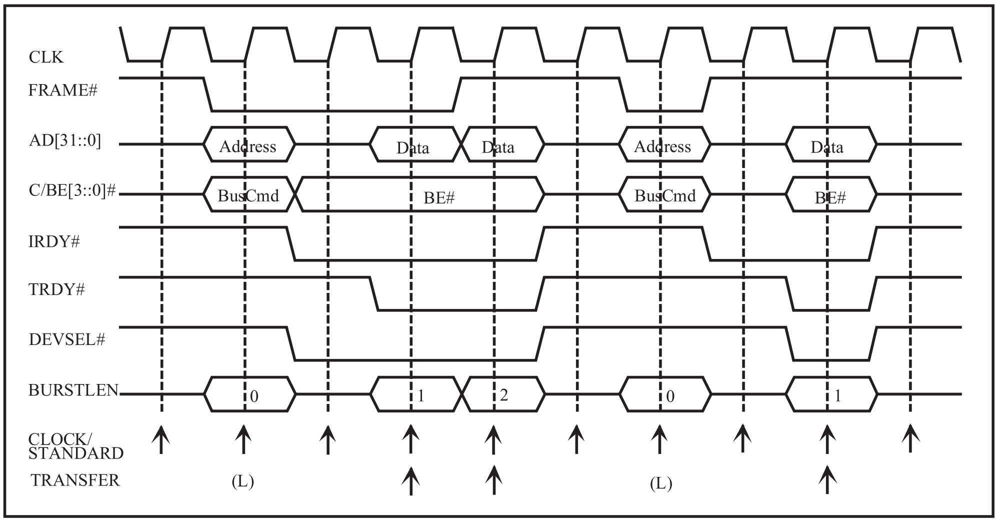
- 2M Sample Trace Buffer @ 256-bits
- Extensive Trace Decoding with Mnemonics
- Time Tag and Latency Tag per sample
- Protocol Sensitive Sampling modes for optimum trace usage and readability
- Demultiplexed Address, Attribute, Data, Command and BE# fields for maximum trigger and storage flexibility
- Multi-level Trace Viewer

The State Analyzer of the Vanguard captures and displays 64-bit or 32-bit PCI-X/PCI bus activity up to 133MHz with highly advanced (yet simple to use) triggering, filtering and counting capabilities. To provide optimum bus analysis for a given problem, the state analyzer offers a choice of three sampling modes, i.e. Clock Mode, Transfer Mode and Standard Mode.

Sampling Modes

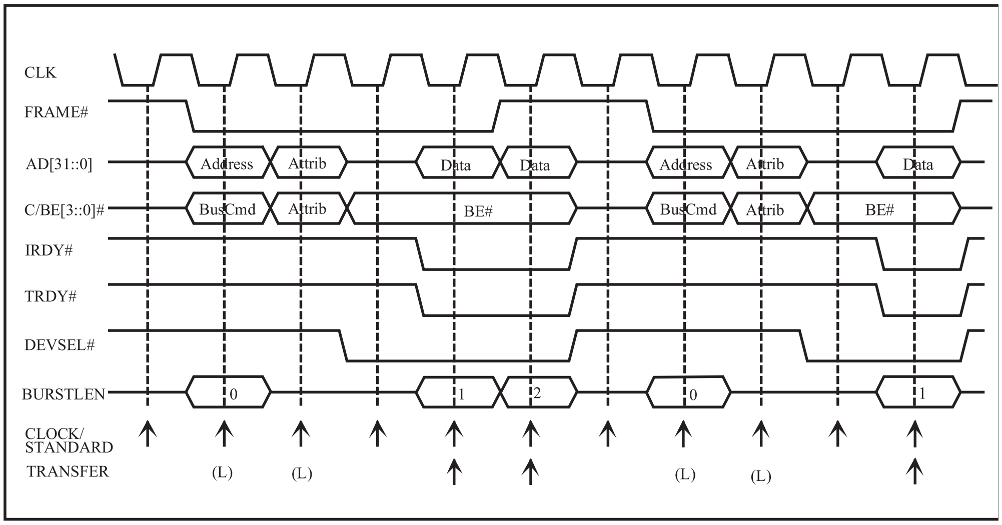
When sampling the PCI-X/PCI bus, the Vanguard captures over 90 bus signals, 16 external signals, and internally generated time tags and utility bits (256-bits in total) into a 2M Sample deep circular Trace Memory. This is a vast amount of information. To provide the user with the most suitable display for different applications, the Vanguard offers three different ways to capture the bus activity.

Figure 10: PCI



- PCI bus transactions sampled in Clock, Standard and Transfer Modes.
- Clock and Standard Modes sample every rising edge of the PCI clock.
- Transfer Mode latches the Address to the data.

Figure 11: PCI-X



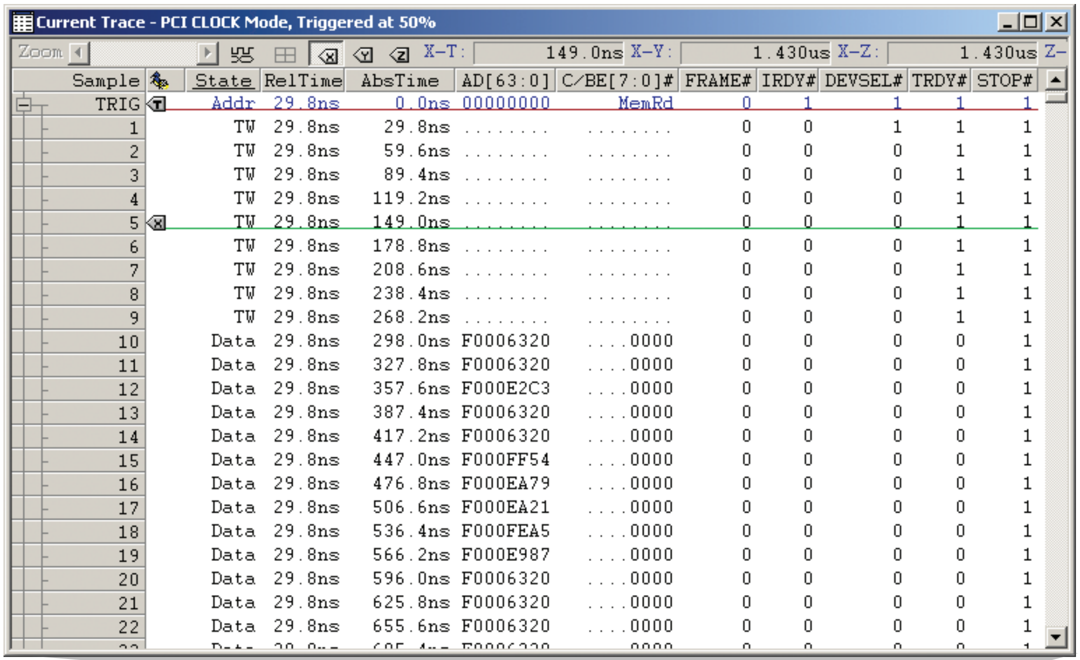
- PCI-X bus transactions sampled in Clock, Standard and Transfer Modes.
- Clock and Standard Modes sample every rising edge of the PCI clock.
- Transfer Mode latches the Address and Attribute phases to the Data.

Clock Mode (Low-level Bus Sampling)

Clock Mode sampling is a “clock-by-clock” view of the bus. This sampling mode acquires samples regardless of whether the protocol of the bus is working or faulty. It shows the signals as they appear. The Vanguard uses the “state” field to display mnemonics for the address, attribute, data, command and wait states identified in the transaction. The traces are displayed as alphanumeric or waveform views. This sampling mode is suitable for low-level hardware analysis of the bus.

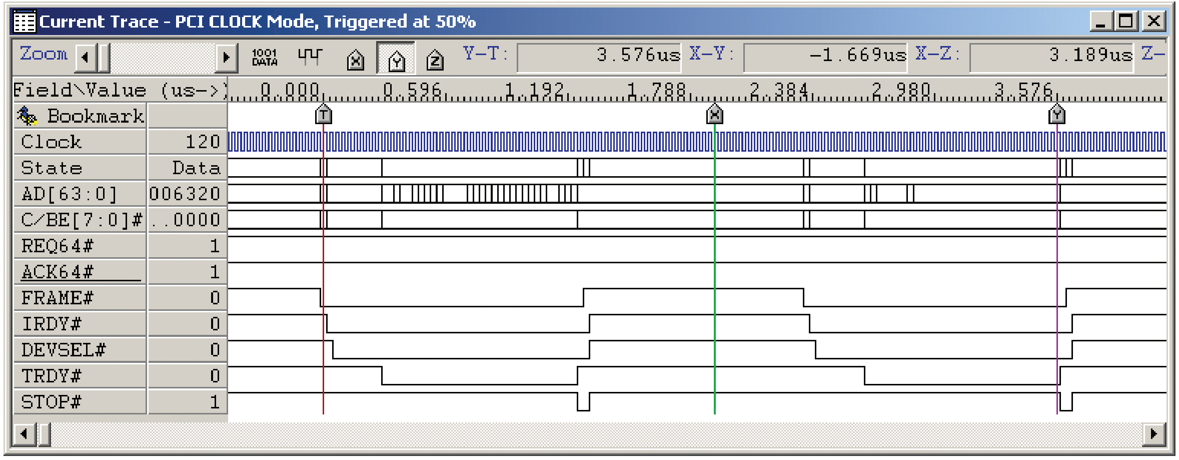
For software development and system integration issues, debugging the PCI or PCI-X bus on a “clock-by-clock” basis is not useful. In these cases, the user is not concerned with the bus protocol. Therefore, Curtiss-Wright offers two additional unique sampling modes which focus on bus traffic rather than bus protocol.

Figure 12: Clock Mode - For Hardware Analysis



Sample	State	RelTime	AbsTime	AD[63:0]	C/BE[7:0]#	FRAME#	IRDY#	DEVSEL#	TRDY#	STOP#
TRIG	Addr	29.8ns	0.0ns	00000000	MemRd	0	1	1	1	1
1	TW	29.8ns	29.8ns	....	....	0	0	1	1	1
2	TW	29.8ns	59.6ns	....	....	0	0	0	1	1
3	TW	29.8ns	89.4ns	....	....	0	0	0	1	1
4	TW	29.8ns	119.2ns	....	....	0	0	0	1	1
5	TW	29.8ns	149.0ns	....	....	0	0	0	1	1
6	TW	29.8ns	178.8ns	....	....	0	0	0	1	1
7	TW	29.8ns	208.6ns	....	....	0	0	0	1	1
8	TW	29.8ns	238.4ns	....	....	0	0	0	1	1
9	TW	29.8ns	268.2ns	....	....	0	0	0	1	1
10	Data	29.8ns	298.0ns	F0006320	0000	0	0	0	0	1
11	Data	29.8ns	327.8ns	F0006320	0000	0	0	0	0	1
12	Data	29.8ns	357.6ns	F000E2C3	0000	0	0	0	0	1
13	Data	29.8ns	387.4ns	F0006320	0000	0	0	0	0	1
14	Data	29.8ns	417.2ns	F0006320	0000	0	0	0	0	1
15	Data	29.8ns	447.0ns	F000FF54	0000	0	0	0	0	1
16	Data	29.8ns	476.8ns	F000EA79	0000	0	0	0	0	1
17	Data	29.8ns	506.6ns	F000EA21	0000	0	0	0	0	1
18	Data	29.8ns	536.4ns	F000FEA5	0000	0	0	0	0	1
19	Data	29.8ns	566.2ns	F000E987	0000	0	0	0	0	1
20	Data	29.8ns	596.0ns	F0006320	0000	0	0	0	0	1
21	Data	29.8ns	625.8ns	F0006320	0000	0	0	0	0	1
22	Data	29.8ns	655.6ns	F0006320	0000	0	0	0	0	1

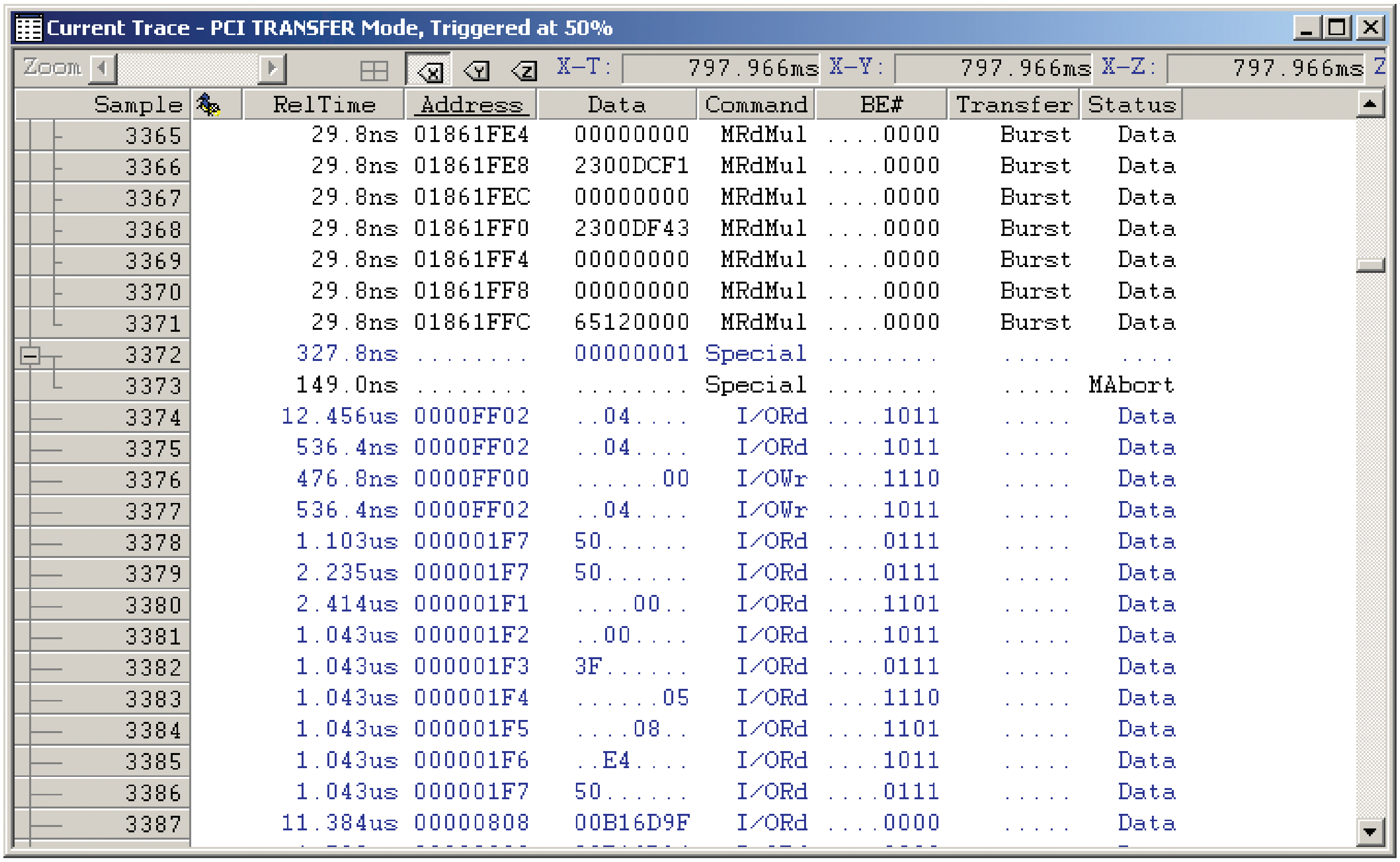
Stores one sample per PCI/PCI-X clock-cycle. This mode captures the details of how the PCI/PCI-X bus is exercised, clock-cycle by clock-cycle. This is useful to verify the behavior of hardware, such as bus interface state machines. In this mode, no demultiplexing takes place, all signals are captured straight from the bus and displayed as an alphanumeric list and/or as waveforms.



Transfer Mode (High-level Bus Sampling)

Transfer Mode sampling uses protocol-sensitive bus sampling, demultiplexing of Address/Attribute/Data/Command/BE#, internally generated utility signals for burst detection and extensive mnemonics. This hides protocol details from the user and conserves trace buffer space by not sampling during idle/wait states. Thus, the user can easily set trigger specifiers and interpret trace data easier. In addition, the user is given information about time from trigger, time between transactions and transfers, the latency or number of wait states per transaction and whether a transfer is part of a burst or at the start of a burst.

Figure 13: Transfer Mode - For Software Analysis



Current Trace - PCI TRANSFER Mode, Triggered at 50%

X-T: 797.966ms X-Y: 797.966ms X-Z: 797.966ms Z

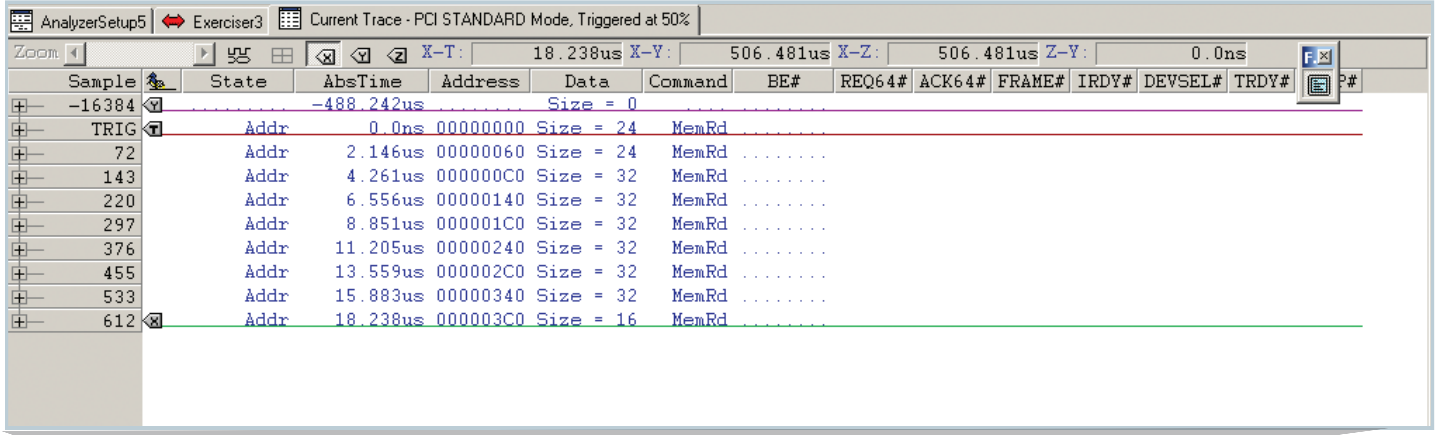
Sample	RelTime	Address	Data	Command	BE#	Transfer	Status
3365	29.8ns	01861FE4	00000000	MRdMul	0000	Burst	Data
3366	29.8ns	01861FE8	2300DCF1	MRdMul	0000	Burst	Data
3367	29.8ns	01861FEC	00000000	MRdMul	0000	Burst	Data
3368	29.8ns	01861FF0	2300DF43	MRdMul	0000	Burst	Data
3369	29.8ns	01861FF4	00000000	MRdMul	0000	Burst	Data
3370	29.8ns	01861FF8	00000000	MRdMul	0000	Burst	Data
3371	29.8ns	01861FFC	65120000	MRdMul	0000	Burst	Data
3372	327.8ns	....	00000001	Special	....	.....	....
3373	149.0ns	....	....	Special	....	.....	MAbort
3374	12.456us	0000FF02	..04....	I/ORd	1011	.....	Data
3375	536.4ns	0000FF02	..04....	I/ORd	1011	.....	Data
3376	476.8ns	0000FF00	00	I/OWr	1110	.....	Data
3377	536.4ns	0000FF02	..04....	I/OWr	1011	.....	Data
3378	1.103us	000001F7	50......	I/ORd	0111	.....	Data
3379	2.235us	000001F7	50......	I/ORd	0111	.....	Data
3380	2.414us	000001F1	....00..	I/ORd	1101	.....	Data
3381	1.043us	000001F2	..00....	I/ORd	1011	.....	Data
3382	1.043us	000001F3	3F......	I/ORd	0111	.....	Data
3383	1.043us	000001F4	05	I/ORd	1110	.....	Data
3384	1.043us	000001F5	....08..	I/ORd	1101	.....	Data
3385	1.043us	000001F6	..E4....	I/ORd	1011	.....	Data
3386	1.043us	000001F7	50......	I/ORd	0111	.....	Data
3387	11.384us	00000808	00B16D9F	I/ORd	0000	.....	Data

Stores one sample per valid Data Phase. Each sample includes the Address, Bus Command and Attribute information which is latched from the address phase (demultiplexing). Signals are grouped for easy interpretation of the bus activity. This is the optimum way to analyze bus transactions during software development.

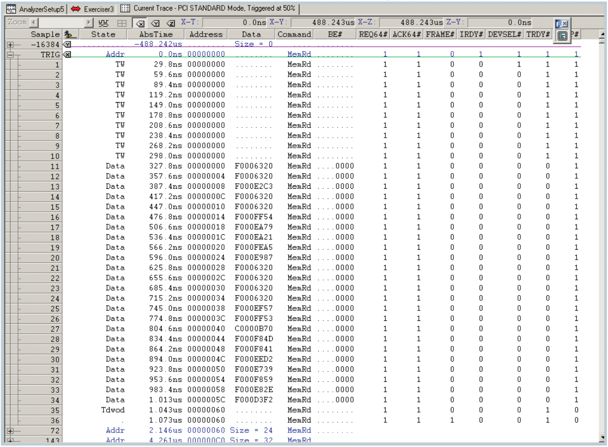
Standard Mode

Standard Mode sampling allows the user to view both a transaction summary showing address and transfer size information as well as “clock-by-clock” status of all signals from the same captured data! Standard Mode allows the detail level of a captured trace to be adjusted dynamically using the Multi-level Trace viewer. The ability to expand and collapse the captured trace data increases the productivity of the tool, since the decoding and demultiplexing of the data is done automatically.

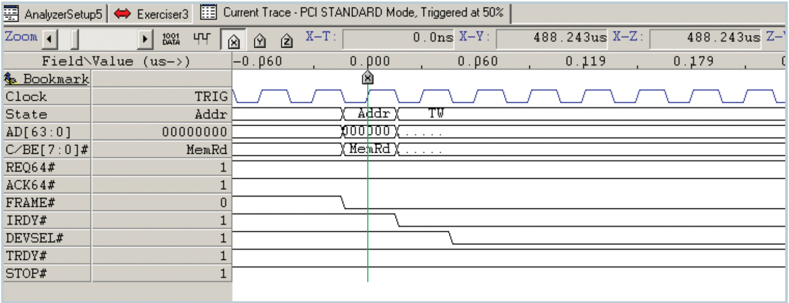
Figure 14: Standard Mode - For Software & Hardware Analysis



```
AnalyzerSetup5 | Exerciser3 | Current Trace - PCI STANDARD Mode, Triggered at 50%
Zoom   X-T: 18.238us X-Y: 506.481us X-Z: 506.481us Z-Y: 0.0ns
  Sample     State     AbsTime    Address    Data     Command   BE#    REQ64# ACK64# FRAME# IRDY# DEVSEL# TRDY# P#
  -16384   ........   -488.242us  ........  Size = 0  ....  ........
  TRIG       Addr       0.0ns    00000000  Size = 24  MemRd ........
      72     Addr       2.146us  00000060  Size = 24  MemRd ........
     143     Addr       4.261us  000000C0  Size = 32  MemRd ........
     220     Addr       6.556us  00000140  Size = 32  MemRd ........
     297     Addr       8.851us  000001C0  Size = 32  MemRd ........
     376     Addr      11.205us  00000240  Size = 32  MemRd ........
     455     Addr      13.559us  000002C0  Size = 32  MemRd ........
     533     Addr      15.883us  00000340  Size = 32  MemRd ........
     612     Addr      18.238us  000003C0  Size = 16  MemRd ........
```



```
AnalyzerSetup5 | Exerciser3 | Current Trace - PCI STANDARD Mode, Triggered at 50%
Zoom   X-T: 0.0ns X-Y: 488.243us X-Z: 488.243us Z-Y: 0.0ns
  Sample   State    AbsTime   Address   Data      Command  BE#       REQ64# ACK64# FRAME# IRDY# DEVSEL# TRDY# P#
  -16384  ........ -488.242us ........  Size = 0   ....  ........
  TRIG     Addr      0.0ns   00000000  ........  MemRd  ........     1      1      0      1      1      1     1
     1     TW       29.8ns   00000000  ........  MemRd  ........     1      1      0      0      1      1     1
     2     TW       59.6ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     3     TW       89.4ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     4     TW      119.2ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     5     TW      149.0ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     6     TW      178.8ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     7     TW      208.6ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     8     TW      238.4ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
     9     TW      268.2ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
    10     TW      298.0ns   00000000  ........  MemRd  ........     1      1      0      0      0      1     1
    11     Data    327.8ns   00000000  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    12     Data    357.6ns   00000004  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    13     Data    387.4ns   00000008  F000E2C3  MemRd  ....0000     1      1      0      0      0      0     1
    14     Data    417.2ns   0000000C  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    15     Data    447.0ns   00000010  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    16     Data    476.8ns   00000014  F000FF54  MemRd  ....0000     1      1      0      0      0      0     1
    17     Data    506.6ns   00000018  F000EA79  MemRd  ....0000     1      1      0      0      0      0     1
    18     Data    536.4ns   0000001C  F000EA21  MemRd  ....0000     1      1      0      0      0      0     1
    19     Data    566.2ns   00000020  F000FEA5  MemRd  ....0000     1      1      0      0      0      0     1
    20     Data    596.0ns   00000024  F000E987  MemRd  ....0000     1      1      0      0      0      0     1
    21     Data    625.8ns   00000028  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    22     Data    655.6ns   0000002C  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    23     Data    685.4ns   00000030  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    24     Data    715.2ns   00000034  F0006320  MemRd  ....0000     1      1      0      0      0      0     1
    25     Data    745.0ns   00000038  F000EF57  MemRd  ....0000     1      1      0      0      0      0     1
    26     Data    774.8ns   0000003C  F000FF53  MemRd  ....0000     1      1      0      0      0      0     1
    27     Data    804.6ns   00000040  C0000B70  MemRd  ....0000     1      1      0      0      0      0     1
    28     Data    834.4ns   00000044  F000F84D  MemRd  ....0000     1      1      0      0      0      0     1
    29     Data    864.2ns   00000048  F000F841  MemRd  ....0000     1      1      0      0      0      0     1
    30     Data    894.0ns   0000004C  F000EED2  MemRd  ....0000     1      1      0      0      0      0     1
    31     Data    923.8ns   00000050  F000E739  MemRd  ....0000     1      1      0      0      0      0     1
    32     Data    953.6ns   00000054  F000F859  MemRd  ....0000     1      1      0      0      0      0     1
    33     Data    983.4ns   00000058  F000E82E  MemRd  ....0000     1      1      0      0      0      0     1
    34     Data    1.013us   0000005C  F000D3F2  MemRd  ....0000     1      1      0      0      0      0     1
    35     Tdwod   1.043us   00000060  ........  MemRd  ........     1      1      0      0      0      1     0
    36     .       1.073us   00000060  ........  MemRd  ........     1      1      1      0      0      1     0
    72     Addr    2.146us   00000060  Size = 24 MemRd  ........
   143     Addr    4.261us   000000C0  Size = 32 MemRd
```



```
AnalyzerSetup5 | Exerciser3 | Current Trace - PCI STANDARD Mode, Triggered at 50%
Zoom   X-T: 0.0ns X-Y: 488.243us X-Z: 488.243us Z-
Field\Value (us->)   -0.060    0.000    0.060    0.119    0.179
Bookmark
Clock          TRIG
State          Addr          Addr   TW
AD[63:0]       00000000      000000 .....
C/BE[7:0]#     MemRd         MemRd  .....
REQ64#         1
ACK64#         1
FRAME#         0
IRDY#          1
DEVSEL#        1
TRDY#          1
STOP#          1
```

Multi-level Trace Viewer
The Vanguard product’s powerful Standard Mode offers an innovative way of visualizing data captured in multiple views. This means the data can be viewed as summarized transactions when initially looking through the data, and then one can zoom in to look at individual clock cycles in a detailed waveform view when the point of interest is identified. This mode makes it possible for software and hardware engineers to view the data in their individual preferred views based on the same captured data.

Single Event Mode

In many cases a simple trigger like "If Event X then Trigger" is sufficient. For this purpose, the default "Single Event Mode" provides a trigger on the event pointed to in the Event Patterns window. When a more complex trigger is required, or a store or count qualifier is needed, the user may switch to the Sequencer.

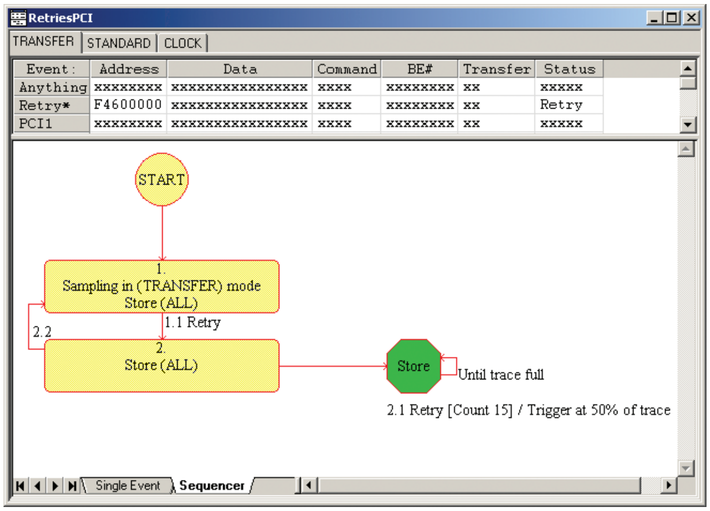
Powerful Trigger & Store Qualifiers

To focus in on the cycles of interest, Vanguard products are equipped with powerful triggers and store qualifiers, based on eight word recognizers and a flexible sequencer. Each of the word recognizers offers powerful operators like RANGE, NOT and BINARY for address and data fields, and the user may specify signal values as hex, binary or mnemonics like "MemRd", "IOWri", etc. This eliminates the need for the user to remember the actual signal values for the different cycle types.

Demultiplexed Address/Data

The PCI-X/PCI bus multiplexes Address and Data (and Attributes for PCI-X) into a common 32-bit or 64-bit bus. In a similar way, the bus Command signals are multiplexed with the data byte enables (BEx#). However, multiplexing makes it more difficult to analyze the bus using a regular logic analyzer, since a given sample does not contain all information about a bus transfer. To overcome this, the Vanguard products have the capability to demultiplex Address/Attribute/Data and COMMAND/BEx# into separate trace channels. This feature simplifies readability of the trace and allows powerful triggers and store qualifiers involving both address and data to be defined easily.

Figure 15: Graphical Sequencer

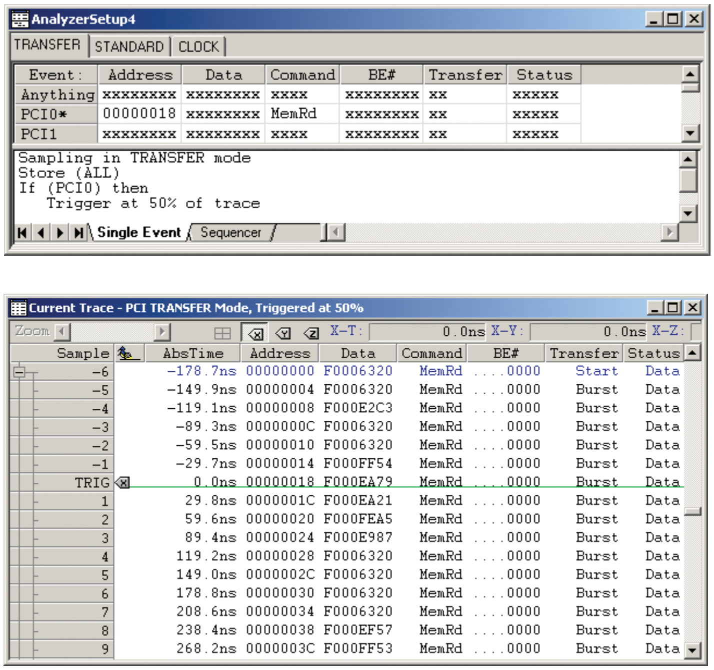


Event:	Address	Data	Command	BE#	Transfer	Status
Anything	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx
Retry*	F4600000	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	xx	Retry
PCI1	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx

Triggering on an Address Inside a Burst

Since a PCI or PCI-X burst transfer has a single address and multiple data phases, it is important to be able to trigger on an address inside a burst. The Vanguard products allow the user to trigger on addresses that were never present on the bus. If we start a burst at address 0x00000000, and the length of that burst was 256 bytes, the Vanguard would be able to trigger on address 0x00000018.

Figure 16: Triggering on an Address Inside a Burst



Event:	Address	Data	Command	BE#	Transfer	Status
Anything	xxxxxxxx	xxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx
PCI0*	00000018	xxxxxxxx	MemRd	xxxxxxxx	xx	xxxxx
PCI1	xxxxxxxx	xxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx

```
Sampling in TRANSFER mode
Store (ALL)
If (PCI0) then
   Trigger at 50% of trace
```

Sample	AbsTime	Address	Data	Command	BE#	Transfer	Status
-6	-178.7ns	00000000	F0006320	MemRd	0000	Start	Data
-5	-149.9ns	00000004	F0006320	MemRd	0000	Burst	Data
-4	-119.1ns	00000008	F000E2C3	MemRd	0000	Burst	Data
-3	-89.3ns	0000000C	F0006320	MemRd	0000	Burst	Data
-2	-59.5ns	00000010	F0006320	MemRd	0000	Burst	Data
-1	-29.7ns	00000014	F000FF54	MemRd	0000	Burst	Data
TRIG	0.0ns	00000018	F000EA79	MemRd	0000	Burst	Data
1	29.8ns	0000001C	F000EA21	MemRd	0000	Burst	Data
2	59.6ns	00000020	F000FEA5	MemRd	0000	Burst	Data
3	89.4ns	00000024	F000E987	MemRd	0000	Burst	Data
4	119.2ns	00000028	F0006320	MemRd	0000	Burst	Data
5	149.0ns	0000002C	F0006320	MemRd	0000	Burst	Data
6	178.8ns	00000030	F0006320	MemRd	0000	Burst	Data
7	208.6ns	00000034	F0006320	MemRd	0000	Burst	Data
8	238.4ns	00000038	F000EF57	MemRd	0000	Burst	Data
9	268.2ns	0000003C	F000FF53	MemRd	0000	Burst	Data

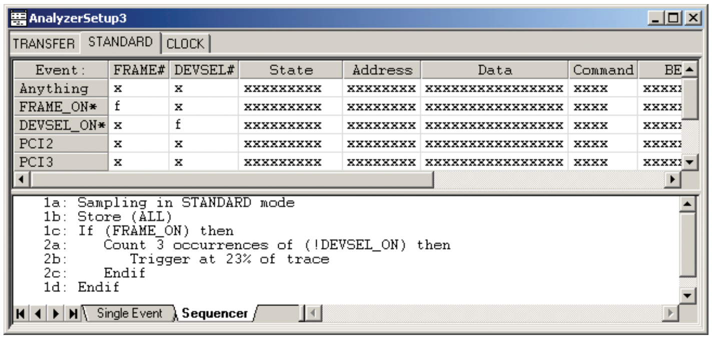
Address/Data Range

Each of the eight word recognizers allows for precise address and data ranges to be defined, with both 'inside' and 'outside' possibilities. This allows the user to trigger, store or count on accesses to a specific area, e.g. a particular data structure or a hardware device. Note that the range can be specified with any arbitrary value down to the last digit, while other analyzers only allow 2n size ranges to be defined (by setting "don't care" in the least significant bits).

Binary Details

Values for multi-bit fields like Address and Data are typically entered in hexadecimal format. But occasionally one may want to specify values of individual bits only, like setting bit 14 to 0 and/or bit 0 to 1, and so on. Setting triggers based on these Binary Details is possible by pressing the "(" when entering the desired number.

Figure 17: Trigger on the Falling Edges of Signals



Event:	FRAME#	DEVSEL#	State	Address	Data	Command	BE
Anything	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
FRAME_ON*	f	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
DEVSEL_ON*	x	f	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
PCI2	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
PCI3	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx

```
    1a: Sampling in STANDARD mode
    1b: Store (ALL)
    1c: If (FRAME_ON) then
    2a:    Count 3 occurrences of (!DEVSEL_ON) then
    2b:       Trigger at 23% of trace
    2c:    Endif
    1d: Endif
```

Edge/Level Triggers

In order to make each event as powerful as possible, Curtiss-Wright's Vanguard product family has the ability to trigger on edges (rising, falling or both) as well as levels. This makes it easier for the user to specify a trigger. As an example, triggering on the falling edge on the FRAME# signal would make sure the trigger is at the start of a transfer, whereas triggering on FRAME# being 0 could trigger in the middle of an ongoing transfer.

Slot-specific or User-defined Signals

The PCI bus has slot-specific signals, such as the Request (REQx#) and Grant (GNTx#) signals used for arbitration. These signals can be brought in through 16 external inputs on the pin headers. The external inputs can be used to set trigger conditions or be viewed in the trace display.

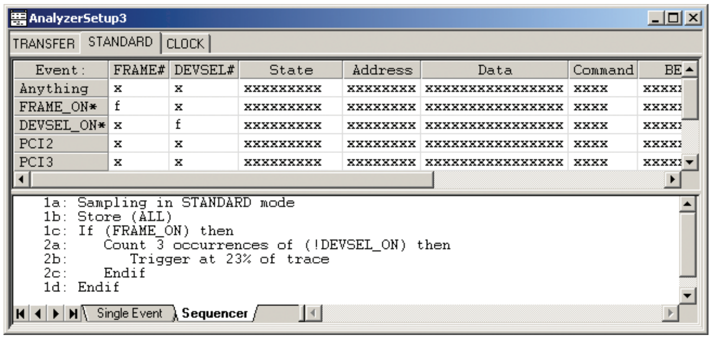
Grant Latching

When the slot-specific GNTx# signals are connected, a special latch can be activated to hold the value of the active grant during all data phases. The latched GNTx# value is available in the trace, triggers, store conditions and statistics.

Examples Setups

Example setups are provided to assist the novice user in defining trigger setups and to provide a "shortcut" to a number of typical trigger scenarios. BusView offers a function called "Predefined Setups". This function programs the event patterns and sequencer with the appropriate values and commands for the selected task (see dialog box below).

Figure 18: Trigger Example



Event:	FRAME#	DEVSEL#	State	Address	Data	Command	BE
Anything	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
FRAME_ON*	f	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
DEVSEL_ON*	x	f	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
PCI2	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx
PCI3	x	x	xxxxxxxxx	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxx

```
    1a: Sampling in STANDARD mode
    1b: Store (ALL)
    1c: If (FRAME_ON) then
    2a:    Count 3 occurrences of (!DEVSEL_ON) then
    2b:       Trigger at 23% of trace
    2c:    Endif
    1d: Endif
```

16-level Sequences of 8 Events/256-bits

The Vanguard provides an advanced 16-level Trigger Sequencer with powerful operators like IF, ELSEIF, ELSE, STORE, COUNT, DELAY and GOTO. The sequencer is shown in a separate window on the main status screen, and may be used to define nested trigger conditions, define store qualifiers, count and delay statements. This sequencer is extremely powerful, since each event used contains 256-bits of information.

Adjustable Trigger Positions

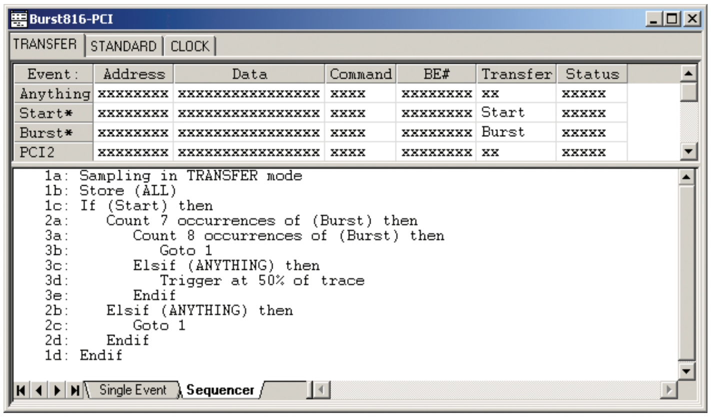
For maximum flexibility, the trigger can be placed at any position in the trace buffer with a 1% resolution. This allows the user to utilize the 2 MSamples (64MB) of trace memory in the best possible way.

Trigger Examples

Example 1: Triggering on a PCI burst transfer that contains between 8 and 16 data phases. This illustrates the power of Curtiss-Wright's unique Transfer Mode sampling, since each sample contains a transfer (or in some cases a failed transfer). We start by selecting Transfer Mode and defining two events, START (indicating the first data phase in a

transfer) and BURST (indicating the second or subsequent data phases in a transfer). We begin by looking for the start of a burst, and when found we count 7 more data transfers. Notice that we do not have to filter out wait states, since Transfer Mode automatically skips these (but counts them). If anything other than a burst data transfer is detected, we restart the sequencer. Next, since we have already counted the minimum number of data phases required, we start another counter counting 8 more data phases. If this count reaches 8, we have had too many data phases and restart the sequencer. If not, we tell the analyzer to trigger, and we are done!

Figure 19: Text Mode Sequencer



Event:	Address	Data	Command	BE#	Transfer	Status
Anything	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx
Start*	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	Start	xxxxx
Burst*	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	Burst	xxxxx
PCI2	xxxxxxxx	xxxxxxxxxxxxxxxx	xxxx	xxxxxxxx	xx	xxxxx

```
1a: Sampling in TRANSFER mode
1b: Store (ALL)
1c: If (Start) then
2a:    Count 7 occurrences of (Burst) then
3a:       Count 8 occurrences of (Burst) then
3b:          Goto 1
3c:       Elsif (ANYTHING) then
3d:          Trigger at 50% of trace
3e:       Endif
2b:    Elsif (ANYTHING) then
2c:       Goto 1
2d:    Endif
1d: Endif
```

Time & Wait State Tags

The Vanguard family provides a high-resolution Time Tag for each sample in the trace list. The time tag may either show relative time between each sample or absolute time from the trigger point.

Another tag in the trace display shows the latency for each PCI transfer, which is defined as the number of wait states from FRAME# asserted to TRDY# asserted.

Search & Extract Trace Data

Powerful Search and Extract functions are provided for easy location of particular samples in the trace memory. After a trace is collected, any combination of signals can be searched for, extracted or hidden in the trace buffer.

Trace Compare

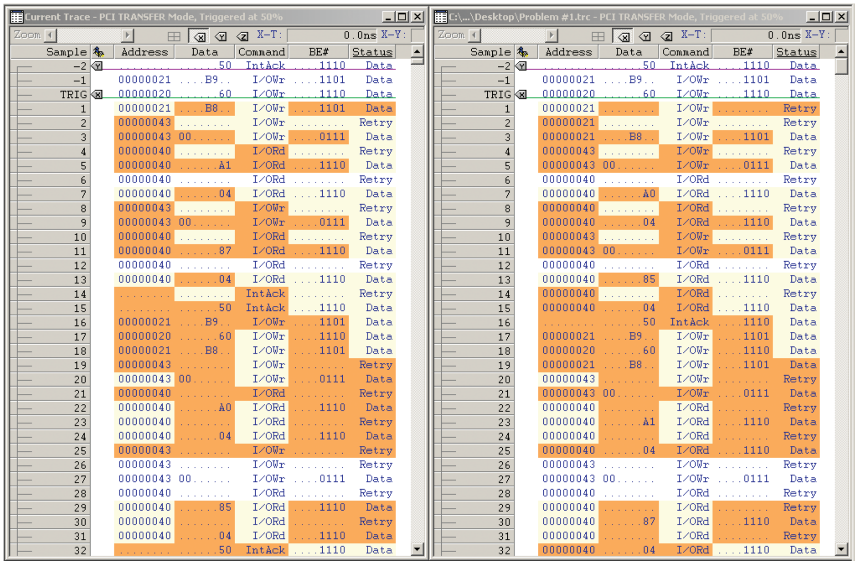
A powerful feature of BusView is the ability to compare the contents of the trace buffer with a trace stored on file on the host PC. This greatly simplifies error location by rapidly spotting differences between failing and functioning systems (see figure below). For example, a system works fine under some circumstances but not under others, or when one out of several presumably identical systems fails.

Example 2: Triggering the analyzer if it takes more than 3 clocks from the master asserting FRAME# until the target asserts DEVSEL#. We define two events, one indicating FRAME# being asserted (the falling edge of FRAME#) and one indicating DEVSEL# being asserted (the falling edge of DEVSEL#). In the sequencer, we start looking for FRAME# being asserted. Once this occurs, we count three consecutive clocks of DEVSEL# not being asserted, and we trigger the analyzer.

Trigger Output

A Trigger Output signal is available on a pin header on the front panel. The LVTTL compatible signal has selectable polarity and mode, i.e. the signal may simply change logic level on trigger, or it may be selected to pulse when a trigger or a valid store condition occurs. The Trigger Output signal is useful for triggering external instruments such as high-speed oscilloscopes, counters, etc.

Figure 20: Trace Compare



Current Trace - PCI TRANSFER Mode, Triggered at 50%
C:\...\Desktop\Problem #1.trc - PCI TRANSFER Mode, Triggered at 50%
Zoom
X-T:
0.0ns
X-Y:

Current Trace - PCI TRANSFER Mode, Triggered at 50%

Sample	Address	Data	Command	BE#	Status
-2	....	50	IntAck	1110	Data
-1	00000021	....B9..	I/OWr	1101	Data
TRIG	00000020	60	I/OWr	1110	Data
1	00000021	....B8..	I/OWr	1101	Data
2	00000043	....	I/OWr	....	Retry
3	00000043	00......	I/OWr	0111	Data
4	00000040	....	I/ORd	....	Retry
5	00000040	......A1	I/ORd	1110	Data
6	00000040	....	I/ORd	....	Retry
7	00000040	04	I/ORd	1110	Data
8	00000043	....	I/OWr	....	Retry
9	00000043	00......	I/OWr	0111	Data
10	00000040	....	I/ORd	....	Retry
11	00000040	87	I/ORd	1110	Data
12	00000040	....	I/ORd	....	Retry
13	00000040	04	I/ORd	1110	Data
14	....	....	IntAck	....	Retry
15	....	50	IntAck	1110	Data
16	00000021	....B9..	I/OWr	1101	Data
17	00000020	60	I/OWr	1110	Data
18	00000021	....B8..	I/OWr	1101	Data
19	00000043	....	I/OWr	....	Retry
20	00000043	00......	I/OWr	0111	Data
21	00000040	....	I/ORd	....	Retry
22	00000040	......A0	I/ORd	1110	Data
23	00000040	....	I/ORd	....	Retry
24	00000040	04	I/ORd	1110	Data
25	00000043	....	I/OWr	....	Retry
26	00000043	....	I/OWr	....	Retry
27	00000043	00......	I/OWr	0111	Data
28	00000040	....	I/ORd	....	Retry
29	00000040	85	I/ORd	1110	Data
30	00000040	....	I/ORd	....	Retry
31	00000040	04	I/ORd	1110	Data
32	....	50	IntAck	1110	Data

C:\...\Desktop\Problem #1.trc - PCI TRANSFER Mode, Triggered at 50%

Sample	Address	Data	Command	BE#	Status
-2	....	50	IntAck	1110	Data
-1	00000021	....B9..	I/OWr	1101	Data
TRIG	00000020	60	I/OWr	1110	Data
1	00000021	....	I/OWr	....	Retry
2	00000021	....	I/OWr	....	Retry
3	00000021	....B8..	I/OWr	1101	Data
4	00000043	....	I/OWr	....	Retry
5	00000043	00......	I/OWr	0111	Data
6	00000040	....	I/ORd	....	Retry
7	00000040	......A0	I/ORd	1110	Data
8	00000040	....	I/ORd	....	Retry
9	00000040	04	I/ORd	1110	Data
10	00000043	....	I/OWr	....	Retry
11	00000043	00......	I/OWr	0111	Data
12	00000040	....	I/ORd	....	Retry
13	00000040	85	I/ORd	1110	Data
14	00000040	....	I/ORd	....	Retry
15	00000040	04	I/ORd	1110	Data
16	....	50	IntAck	1110	Data
17	00000021	....B9..	I/OWr	1101	Data
18	00000020	60	I/OWr	1110	Data
19	00000021	....B8..	I/OWr	1101	Data
20	00000043	....	I/OWr	....	Retry
21	00000043	00......	I/OWr	0111	Data
22	00000040	....	I/ORd	....	Retry
23	00000040	......A1	I/ORd	1110	Data
24	00000040	....	I/ORd	....	Retry
25	00000040	04	I/ORd	1110	Data
26	00000043	....	I/OWr	....	Retry
27	00000043	00......	I/OWr	0111	Data
28	00000040	....	I/ORd	....	Retry
29	00000040	....	I/ORd	....	Retry
30	00000040	87	I/ORd	1110	Data
31	00000040	....	I/ORd	....	Retry
32	00000040	04	I/ORd	1110	Data

Versatile Waveform Diagrams

When sampling in Clock Mode or Standard Mode, the easy-to-read waveform diagrams provide powerful zooming, cursors and navigation tools. To ease the search for cycles of interest, one may use an "edge-to-edge" scroll mode on a selected signal, as well as explicit jumps to given line numbers or to cursors.

Save Trace to File

Captured trace data may easily be saved to a binary or ASCII file on a PC and then loaded back later. Partial traces may be loaded. Trace files may also be reviewed locally on the PC using the BusView for Windows software.

Statistics

- Independent Real-time Statistics Engine
- 47 Real-time Counters
- Pre-defined and User-defined Statistics

Concurrent Statistics

The Vanguard's enhanced statistics engine offers concurrent Real-time measurements for Event Counting, Bus Utilization, Bus Transfer Rate, Command Distribution, Burst Length Distribution, Arbiter Latency, Wait States and User Defined Statistics.

The user may save the displayed statistics data to an ASCII file for post processing, analysis and display. It is also possible to play back statistics at a user selectable speed.

Bus Utilization

The Bus Utilization statistics function displays the percentage of time the bus is occupied, how much overhead a system has, and how the bus is being used at any given time. This statistics function is ideal for determining whether the bus has spare capacity to support additional I/O devices or processors.

Event Counting

The Event Counting function, provides a Real-time count of 8 user-defined events. This powerful function may be used to count the number of IACK cycles per second or the number of Write or Read cycles, or to investigate access patterns to the bus in multi-processor systems, etc.

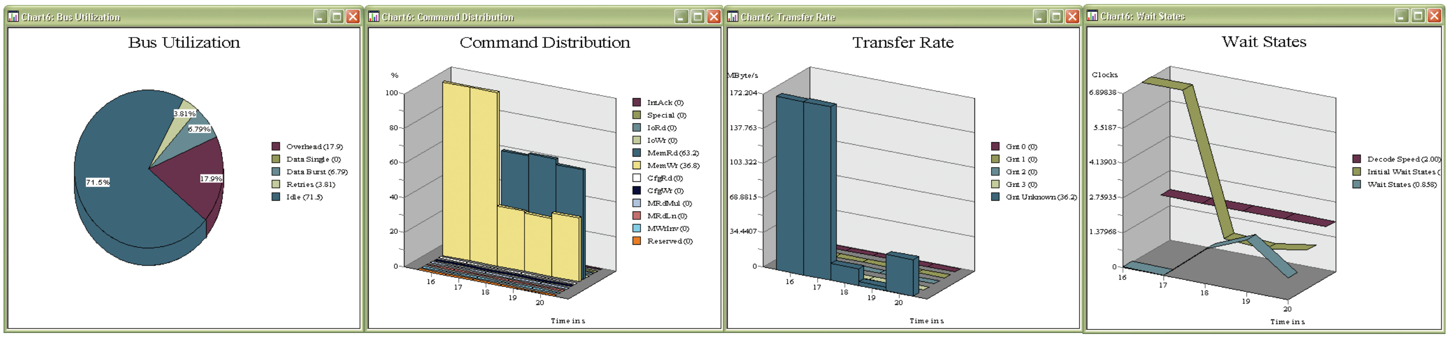
Bus Transfer Rate

The Bus Transfer Rate function presents how much data is transferred over the bus, shown in MB/s or in MTransfers/s. This can either be calculated between selected lines directly in the trace buffer, or as histograms that show the average transfer rate over a period of time. This statistics function is used to verify that system performance specifications have been fulfilled and to assist in system tuning.

Command Distribution & Burst Distribution

The Command Distribution statistics function displays the distribution of PCI or PCI-X commands over a period of time. The Burst Distribution function displays the distribution of burst lengths broken into categories of single cycles, 2-7, 8-31, 32-63, 64-127, 128-511 and higher than 512 data phases. Burst Distribution statistics are very useful for measurements during performance enhancement and optimization.

Figure 21: Real-time Statistics



Arbiter Latency

The Arbiter Latency function shows the number of clocks between the assertion of REQ# and GNT#, and GNT# to FRAME# going active.

Wait States

The Wait States statistics function shows the average number of Wait States and Decode Speed per transaction. The Decode Speed is the number of clocks from FRAME# to DEVSEL#. Initial Wait States is determined by counting the number of clocks from DEVSEL# to IRDY# and TRDY#. In PCI, this statistic also displays the number of clocks between data phases.

Custom Statistics

This function allows the user to define up to 8 statistics functions using mathematical combinations of 47 Real-time counters available in the Vanguard product. This allows the user to implement a virtually unlimited number of statistics functions.

Exerciser

- Master, Target and Interrupter for 32-bit and 64-bit PCI up to 66.7MHz and PCI-X up to 133MHz
- Target at user defined address with 8MB zero-wait-state burst memory
- Built-in Script Recording and Playback
- Memory Tests with walking ones/zeros, random patterns, etc.
- DMA transfers between Exerciser and Target, or between two Bus Targets
- Generate and handle interrupts
- Supports Burst and Single cycles
- Trigger analyzer if Memory Test fails
- Start Exerciser on Analyzer trigger
- Enhanced Exerciser offers Error Injection and programmable Target behavior

The two Exercisers available for the Vanguard are flexible and easy to use. The Vanguard exercisers allow concurrent operation of two DMA engines, a target memory with I/O and memory spaces, and a command executer. A flexible script feature allows test scripts to be made and tasks to be automated.

Basic Exerciser (-E Option)

The Vanguard products optionally feature an advanced 64-bit Exerciser that functions as a Bus Master interface with DMA, a Target interface with 8MB of memory and an Interrupter. The exerciser supports 64-bit and 32-bit 66MHz PCI and 100MHz PCI-X. The exerciser is controlled via dialog boxes through the user-interface, automated with the built-in script recording and playback capability with programmable delay and loop functions.

Enhanced Exerciser (-E2 Option)

An enhanced exerciser with PCI-X error injection and 133MHz support is offered for the Vanguard product line.

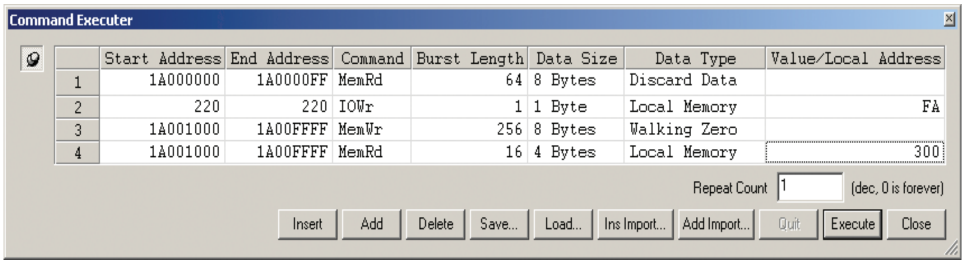
The enhanced exerciser has a fast 4KB Target window that allows 0-wait-state accesses at 133MHz with programmable target responses, burst lengths and terminations. This fast memory coexists with the 8MB Target memory also present in the -E option.

The error injection capabilities allow the user to test a systems error recovery from a controlled error situation. This means mission-critical and fault-tolerant systems can be tested more fully and with greater ease.

Simultaneous Exercising & Analysis

The Exerciser is a separate functional unit of the Vanguard board, which can start and run independently and concurrently with the analyzer. Similarly, the Target interface is another separate functional unit with 8MB of memory. The target memory can be accessed by other PCI-X/PCI agents at any time, with a base address specified by the user.

Figure 22: The Exerciser Command Executer



Command Executer

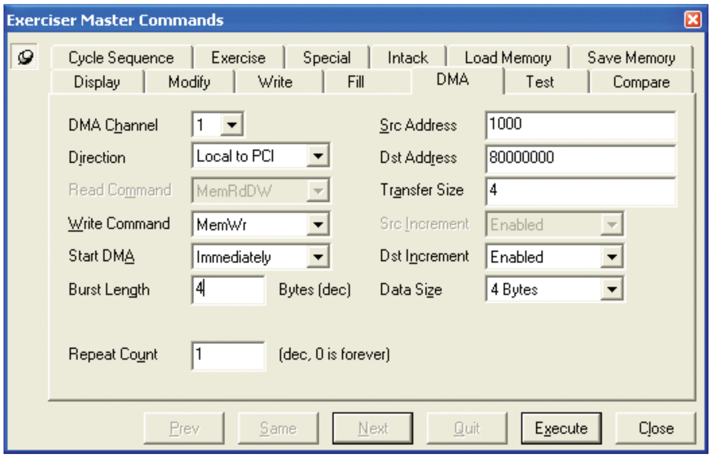
	Start Address	End Address	Command	Burst Length	Data Size	Data Type	Value/Local Address
1	1A000000	1A0000FF	MemRd	64	8 Bytes	Discard Data	
2	220	220	IOWr	1	1 Byte	Local Memory	FA
3	1A001000	1A00FFFF	MemWr	256	8 Bytes	Walking Zero	
4	1A001000	1A00FFFF	MemRd	16	4 Bytes	Local Memory	300

Repeat Count: 1 (dec, 0 is forever)

Plug & Play

The Exerciser is totally standalone and self-contained as seen from the PCI-X/PCI bus. This means that the BIOS of the host system will not need to configure the Vanguard. All parameters such as Target Address window/(BAR), etc. are set by software through user commands.

Figure 23: The DMA Window



DMA Channel	1	Src Address	1000
Direction	Local to PCI	Dst Address	80000000
Read Command	MemRdDW	Transfer Size	4
Write Command	MemWr	Src Increment	Enabled
Start DMA	Immediately	Dst Increment	Enabled
Burst Length	4 Bytes (dec)	Data Size	4 Bytes
Repeat Count	1 (dec, 0 is forever)		

Bus Master with Powerful DMA

The Bus Master has two DMA engines, allowing the user to test transactions to several target devices concurrently. Normally, the DMA controllers transfer data between the local memory and PCI-X/PCI memory. However, a unique feature of the Vanguard Exerciser is the ability to transfer data with DMA from one PCI device to another. Up to 8MB of data can be transferred per DMA command, with peak burst data rates of 1GB/s.

Command Executer

The Command Executer allows a list of transactions to be defined and executed in one go. This allows a specific and fast execution of a set of transactions to take place. The Command Executer is therefore much faster than scripts, typically less than 1μs between transactions.

Manipulate Data in PCI Memory

A comprehensive set of commands are available to the user to inspect, manipulate and test data in PCI-X/PCI and local memory. There are also commands to load, dump and compare data between PCI-X/PCI or local memory and files on the host PC.

Explicit memory test commands are also provided, using data patterns such as random or walking ones/zeros. If an error is found, the exerciser can trigger the state analyzer for immediate review of the failing cycle(s). The module may also generate interrupts, IntAck and Special Cycles on the PCI-X/PCI bus.

Binary data can also be saved to disk directly from PCI memory. PCI memory can also be loaded from a file. This allows the user to easily upload code to a processor, etc.

Script Function Allows Automated Testing

A built-in script engine allows test scripts to be created with a convenient record and playback function. Several scripts can be stored and retrieved for later use. Each script consists of sequences of bus cycles of any kind, with varying sizes, cycle types, etc. The script playback function can be set to run single, multiple or infinite playbacks, while the analyzer part of the product may perform bus monitoring in the background. This makes the Vanguard ideal for running automated tests during design verification or production test of PCI devices, adapters and motherboards.

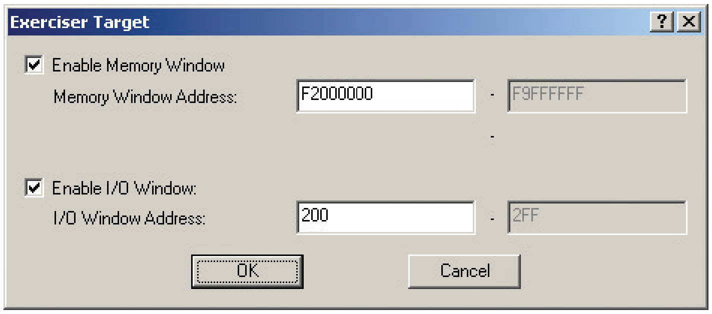
Emulate a Board Under Design

In many cases a board intended for a specific system is not available. The Exerciser can emulate this card as a Target or as a Master. This way the software design can progress without having to wait for the hardware. The Exerciser also contains a target interface that has its own address decoder for a user defined address window. This may respond to accesses from another module.

Target Memory

The module contains 8MB of target memory that can be located anywhere in the PCI address map by a command in the user interface. Data can be written to and read from this memory as single cycles or as zero-wait-state burst cycles for a peak bandwidth of more than 1GB/s.

Figure 24: Exerciser Target Dialog Box



Exerciser Target

☑ Enable Memory Window

Memory Window Address: F2000000 - F9FFFFFF

☑ Enable I/O Window:

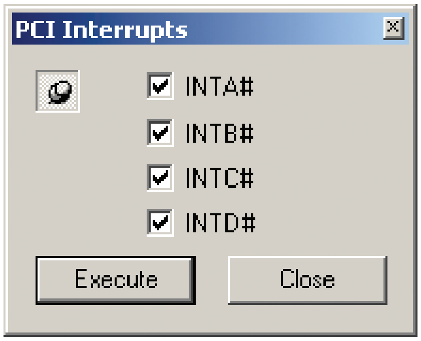
I/O Window Address: 200 - 2FF

Generate PCI Interrupts

The Exerciser can generate any of the four PCI Interrupt lines INTA#, INTB#, INTC#, and INTD#.

When an interrupt is generated, it can be turned off through the Exerciser user-interface, or it may remain asserted until another PCI device explicitly turns it off by writing to a certain address of the Exerciser Target. The status of the interrupts asserted by the Exerciser is indicated on the status line of BusView.

Figure 25: Interrupt Dialog Box



PCI Interrupts

☑ INTA#

☑ INTB#

☑ INTC#

☑ INTD#

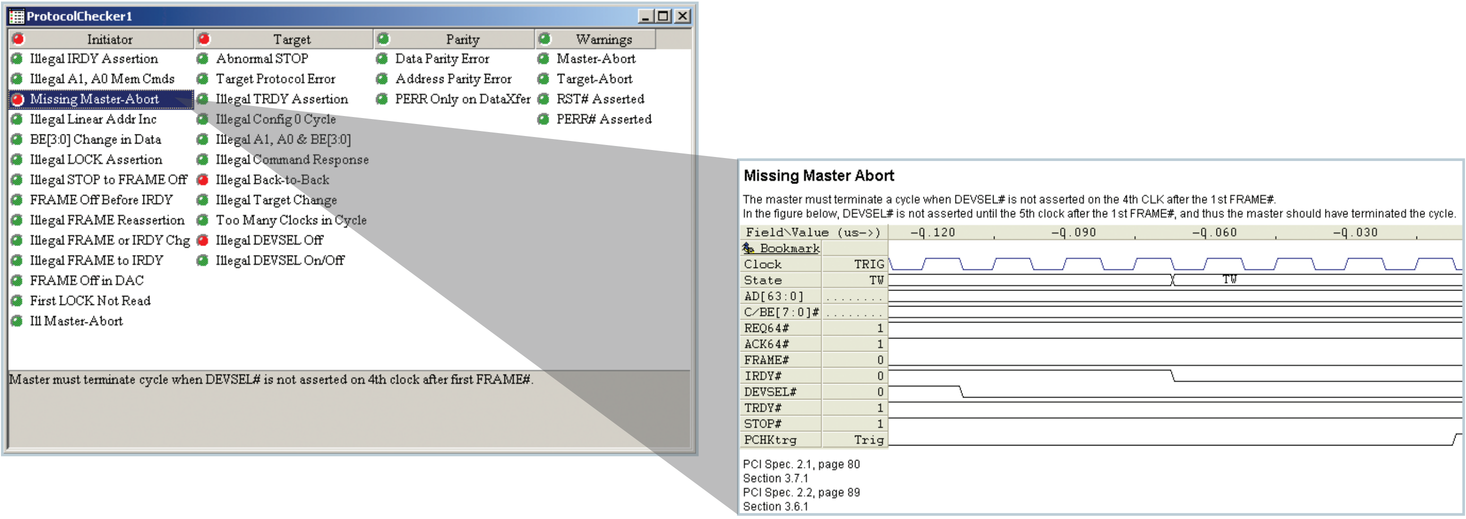
Protocol Checker

- Automatically detects 71 PCI-X and 45 PCI protocol errors
- Trigger output to State Analyzer or External Output

The Vanguard products feature an optional protocol checker for PCI-X and PCI. This versatile feature automatically detects up to 71 PCI-X and 45 PCI protocol errors, helping the user to track down bus hardware errors without the need to understand the nature of the problem.

The protocol checker can run in the background when other analyzer functions of the Vanguard are active. As an example, the state analyzer, and the bus utilization statistics can all be active at the same time while the protocol checker runs in the background, screening the bus for errors. If the protocol checker is used as the trigger source for the analyzer(s), the state and timing analyzers will then provide a comprehensive picture of the bus activity around the point when an error was found. This helps the user to identify and correct the problem.

Figure 26: Missing Master Abort


ProtocolChecker1

● Initiator	● Target	● Parity	● Warnings
● Illegal IRDY Assertion	● Abnormal STOP	● Data Parity Error	● Master-Abort
● Illegal A1, A0 Mem Cmds	● Target Protocol Error	● Address Parity Error	● Target-Abort
● Missing Master-Abort	● Illegal TRDY Assertion	● PERR Only on DataXfer	● RST# Asserted
● Illegal Linear Addr Inc	● Illegal Config 0 Cycle		● PERR# Asserted
● BE[3:0] Change in Data	● Illegal A1, A0 & BE[3:0]		
● Illegal LOCK Assertion	● Illegal Command Response		
● Illegal STOP to FRAME Off	● Illegal Back-to-Back		
● FRAME Off Before IRDY	● Illegal Target Change		
● Illegal FRAME Reassertion	● Too Many Clocks in Cycle		
● Illegal FRAME or IRDY Chg	● Illegal DEVSEL Off		
● Illegal FRAME to IRDY	● Illegal DEVSEL On/Off		
● FRAME Off in DAC			
● First LOCK Not Read			
● Ill Master-Abort			

Master must terminate cycle when DEVSEL# is not asserted on 4th clock after first FRAME#.

Missing Master Abort

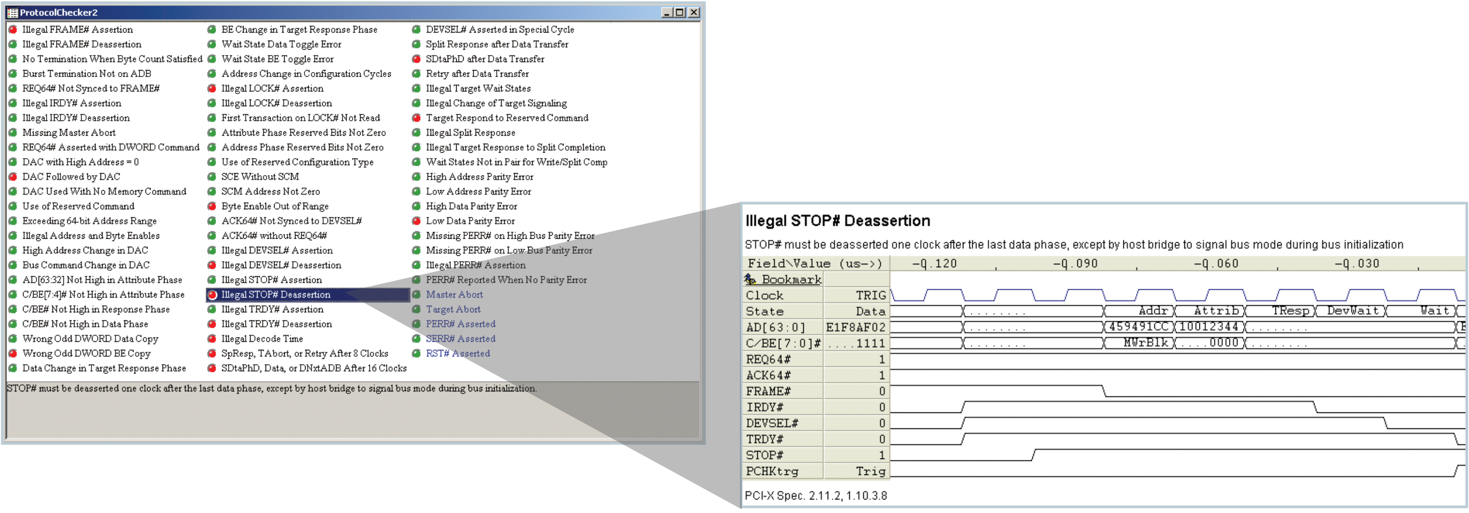
The master must terminate a cycle when DEVSEL# is not asserted on the 4th CLK after the 1st FRAME#.
In the figure below, DEVSEL# is not asserted until the 5th clock after the 1st FRAME#, and thus the master should have terminated the cycle.

Field\Value (us->)	
Bookmark	
Clock	TRIG
State	TW
AD[63:0]	....
C/BE[7:0]#	....
REQ64#	1
ACK64#	1
FRAME#	0
IRDY#	0
DEVSEL#	0
TRDY#	1
STOP#	1
PCHKtrg	Trig

PCI Spec. 2.1, page 80
Section 3.7.1
PCI Spec. 2.2, page 89
Section 3.6.1

Detailed online help for this PCI violation

Figure 27: Illegal STOP# Deassertion


ProtocolChecker2

- ● Illegal FRAME# Assertion
- ● Illegal FRAME# Deassertion
- ● No Termination When Byte Count Satisfied
- ● Burst Termination Not on ADB
- ● REQ64# Not Synced to FRAME#
- ● Illegal IRDY# Assertion
- ● Illegal IRDY# Deassertion
- ● Missing Master Abort
- ● REQ64# Asserted with DWORD Command
- ● DAC with High Address = 0
- ● DAC Followed by DAC
- ● DAC Used With No Memory Command
- ● Use of Reserved Command
- ● Exceeding 64-bit Address Range
- ● Illegal Address and Byte Enables
- ● High Address Change in DAC
- ● Bus Command Change in DAC
- ● AD[63:32] Not High in Attribute Phase
- ● C/BE[7:4]# Not High in Attribute Phase
- ● C/BE# Not High in Response Phase
- ● C/BE# Not High in Data Phase
- ● Wrong Odd DWORD Data Copy
- ● Wrong Odd DWORD BE Copy
- ● Data Change in Target Response Phase
- ● BE Change in Target Response Phase
- ● Wait State Data Toggle Error
- ● Wait State BE Toggle Error
- ● Address Change in Configuration Cycles
- ● Illegal LOCK# Assertion
- ● Illegal LOCK# Deassertion
- ● First Transaction on LOCK# Not Read
- ● Attribute Phase Reserved Bits Not Zero
- ● Address Phase Reserved Bits Not Zero
- ● Use of Reserved Configuration Type
- ● SCE Without SCM
- ● SCM Address Not Zero
- ● Byte Enable Out of Range
- ● ACK64# Not Synced to DEVSEL#
- ● ACK64# without REQ64#
- ● Illegal DEVSEL# Assertion
- ● Illegal DEVSEL# Deassertion
- ● Illegal STOP# Assertion
- ● Illegal STOP# Deassertion
- ● Illegal TRDY# Assertion
- ● Illegal TRDY# Deassertion
- ● Illegal Decode Time
- ● SpResp, TAbort, or Retry After 8 Clocks
- ● SDtaPhD, Data, or DNxtADB After 16 Clocks
- ● DEVSEL# Asserted in Special Cycle
- ● Split Response after Data Transfer
- ● SDtaPhD after Data Transfer
- ● Retry after Data Transfer
- ● Illegal Target Wait States
- ● Illegal Change of Target Signaling
- ● Target Respond to Reserved Command
- ● Illegal Split Response
- ● Illegal Target Response to Split Completion
- ● Wait States Not in Pair for Write/Split Comp
- ● High Address Parity Error
- ● Low Address Parity Error
- ● High Data Parity Error
- ● Low Data Parity Error
- ● Missing PERR# on High Bus Parity Error
- ● Missing PERR# on Low Bus Parity Error
- ● Illegal PERR# Assertion
- ● PERR# Reported When No Parity Error
- ● Master Abort
- ● Target Abort
- ● PERR# Asserted
- ● SERR# Asserted
- ● RST# Asserted

STOP# must be deasserted one clock after the last data phase, except by host bridge to signal bus mode during bus initialization.

Illegal STOP# Deassertion

STOP# must be deasserted one clock after the last data phase, except by host bridge to signal bus mode during bus initialization

Field\Value (us->)	
Bookmark	
Clock	TRIG
State	Data
AD[63:0]	E1F8AF02
C/BE[7:0]#	1111
REQ64#	1
ACK64#	1
FRAME#	0
IRDY#	0
DEVSEL#	0
TRDY#	0
STOP#	1
PCHKtrg	Trig

PCI-X Spec. 2.11.2, 1.10.3.8

Detailed online help for this PCI-X violation

Compliance Test

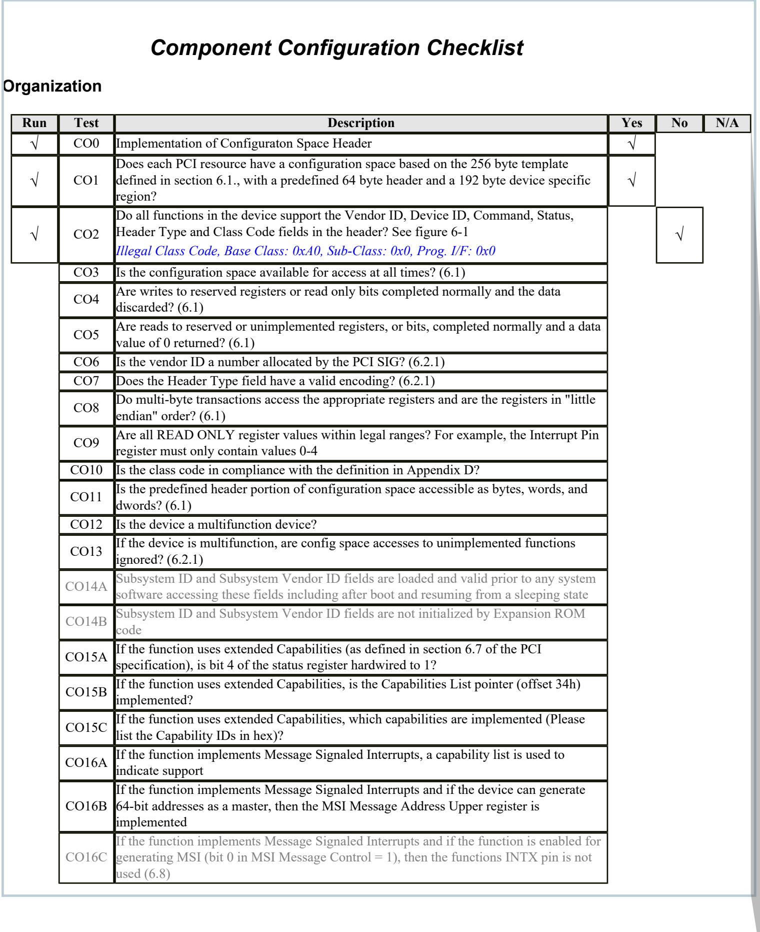
- Automatic device test based on PCI SIG compliance checklist
- Detailed HTML reporting tool
- Extensive description of failures

The Compliance Test is a suite of tests designed to aid in the verification of ASICs, components, motherboards, expansion cards and systems compliance with the PCI Local Bus Specification. The tests performed follow the PCI and PCI-X Compliance Checklists as published by PCI SIG.

Although it is recommended to run the Compliance Test Suite in a passive backplane without possible interruptions by drivers and BIOS, it will also run in systems where the driver or other applications can be used to program the device during the tests. This allows testing of the master functionality of the device as well.

The tests need to be run in a system where no other traffic is present, and accesses are made to the board or device being tested. The output of the Compliance Test is in the form of an HTML document detailing the results of the suite of tests.

Figure 28: Compliance Checklist Report

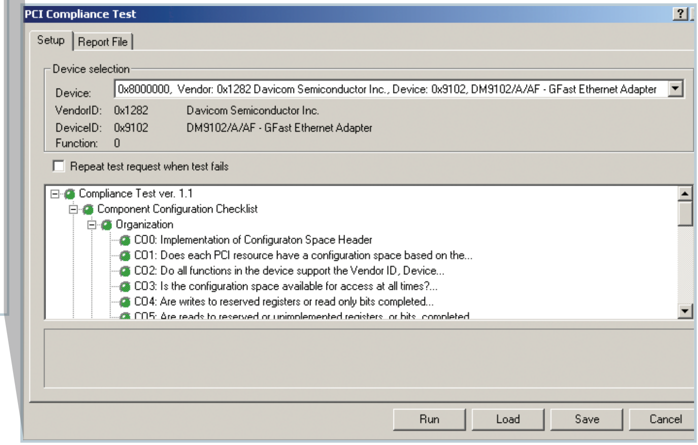


Component Configuration Checklist

Organization

Run	Test	Description	Yes	No	N/A
√	CO0	Implementation of Configuraton Space Header	√		
√	CO1	Does each PCI resource have a configuration space based on the 256 byte template defined in section 6.1., with a predefined 64 byte header and a 192 byte device specific region?	√		
√	CO2	Do all functions in the device support the Vendor ID, Device ID, Command, Status, Header Type and Class Code fields in the header? See figure 6-1 *Illegal Class Code, Base Class: 0xA0, Sub-Class: 0x0, Prog. I/F: 0x0*		√	
	CO3	Is the configuration space available for access at all times? (6.1)			
	CO4	Are writes to reserved registers or read only bits completed normally and the data discarded? (6.1)			
	CO5	Are reads to reserved or unimplemented registers, or bits, completed normally and a data value of 0 returned? (6.1)			
	CO6	Is the vendor ID a number allocated by the PCI SIG? (6.2.1)			
	CO7	Does the Header Type field have a valid encoding? (6.2.1)			
	CO8	Do multi-byte transactions access the appropriate registers and are the registers in "little endian" order? (6.1)			
	CO9	Are all READ ONLY register values within legal ranges? For example, the Interrupt Pin register must only contain values 0-4			
	CO10	Is the class code in compliance with the definition in Appendix D?			
	CO11	Is the predefined header portion of configuration space accessible as bytes, words, and dwords? (6.1)			
	CO12	Is the device a multifunction device?			
	CO13	If the device is multifunction, are config space accesses to unimplemented functions ignored? (6.2.1)			
	CO14A	Subsystem ID and Subsystem Vendor ID fields are loaded and valid prior to any system software accessing these fields including after boot and resuming from a sleeping state			
	CO14B	Subsystem ID and Subsystem Vendor ID fields are not initialized by Expansion ROM code			
	CO15A	If the function uses extended Capabilities (as defined in section 6.7 of the PCI specification), is bit 4 of the status register hardwired to 1?			
	CO15B	If the function uses extended Capabilities, is the Capabilities List pointer (offset 34h) implemented?			
	CO15C	If the function uses extended Capabilities, which capabilities are implemented (Please list the Capability IDs in hex)?			
	CO16A	If the function implements Message Signaled Interrupts, a capability list is used to indicate support			
	CO16B	If the function implements Message Signaled Interrupts and if the device can generate 64-bit addresses as a master, then the MSI Message Address Upper register is implemented			
	CO16C	If the function implements Message Signaled Interrupts and if the function is enabled for generating MSI (bit 0 in MSI Message Control = 1), then the functions INTX pin is not used (6.8)			

Figure 29: PCI Compliance Test Suite



Device selection

Device: 0x8000000, Vendor: 0x1282 Davicom Semiconductor Inc., Device: 0x9102, DM9102/A/AF - GFast Ethernet Adapter

VendorID: 0x1282 Davicom Semiconductor Inc.

DeviceID: 0x9102 DM9102/A/AF - GFast Ethernet Adapter

Function: 0

☐ Repeat test request when test fails

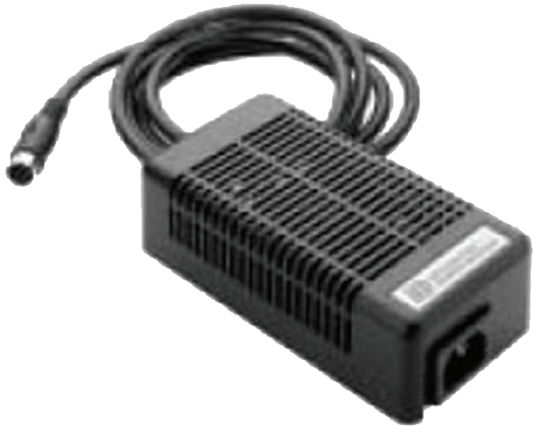
- Compliance Test ver. 1.1
  - Component Configuration Checklist
    - Organization
      - CO0: Implementation of Configuraton Space Header
      - CO1: Does each PCI resource have a configuration space based on the...
      - CO2: Do all functions in the device support the Vendor ID, Device...
      - CO3: Is the configuration space available for access at all times?...
      - CO4: Are writes to reserved registers or read only bits completed...
      - CO5: Are reads to reserved or unimplemented registers, or bits, completed

Miscellaneous

External Power Supply Provision

The Vanguard PCI and Vanguard PMC have provision for an external power supply (401-VG-EPSU). This makes it possible to use the analyzer in systems with a marginal power supply.

Figure 31: External Power Supply



Stackable to Carry PMC Under Test

A unique design allows the Vanguard PMC to carry another PMC module on top. This is a fundamental feature, allowing analysis of the PCI bus without having to remove a PMC module. The Vanguard PMC has PPMC functionality, supporting the REQB#, GNTB# and IDSELB# signals. This allows a PMC module under test to act as a master simultaneously as the Vanguard PMC Exerciser in a PPMC slot. The PMC P4 connector is also passed through, to accommodate PMC modules with rear I/O (e.g. to P2ac on VME boards).

Note that this configuration may limit the maximum operating frequency of the bus segment, depending on the host platform and other factors.

System Slot Controller & Hot SWAP Support (VG-cPCI Only)

The Vanguard cPCI product can function as both a system slot controller and a peripheral card. As a peripheral card, full support for Hot Swap is offered, to allow insertion and extraction in live systems. As a system slot controller the Vanguard cPCI handles arbitration and clock generation. The clock generation frequency is user programmable at frequencies from 33MHz to 133MHz.

Figure 32: The Vanguard PMC Analyzer Carrying a Device Under Test

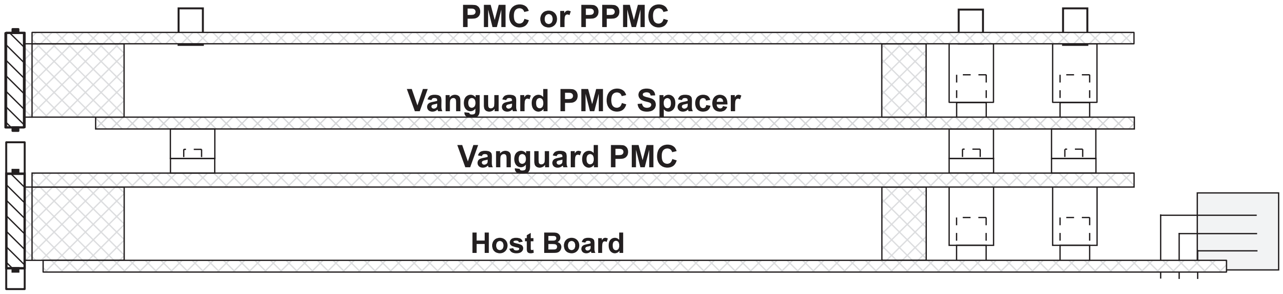


Table 2: Specifications

Analyzer		
Trace Memory	2M Samples x 256-bits (64MB total)	
Input Channels	VG-PCI	92 bus signals, plus 16 ext. inputs on pin headers
	VG-PMC	93 bus signals, plus 16 ext inputs
	VG-cPCI	96 bus signals, plus 16 ext. inputs
PCI-X/PCI Clock Requirements	Max. 133MHz, min. 1kHz	
Signal Levels	VG-PCI	3.3V or 5V (using included adapter)
	VG-PMC/ VG-cPCI	3.3V or 5V
Monitored Signals	VG-PCI	AD[31::0], AD[63::32], C/BE[3::0]#, C/BE[7::4]#, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, PAR, PAR64, PERR#, SERR#, RST#, INTA:D#, LOCK#, ACK64#, REQ64#, REQ#, GNT#, IDSEL, (Plus GNT3:0#, REQ3:0#, IDSEL via pin headers)
	VG-PMC	AD[31::0], AD[63::32], C/BE[3::0]#, C/BE[7::4]#, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, PAR, PAR64, PERR#, SERR#, RST#, INTA:D#, LOCK#, ACK64#, REQ64#, REQ#, GNT#, IDSEL, REQB#,GNTB#, IDSELB#, PME#. (Plus GNT3:0#, REQ3:0#, IDSEL via pin headers).
	VG-cPCI	AD[31::0], AD[63::32], C/BE[3::0]#, C/BE[7::4]#, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, PAR, PAR64, PERR#, SERR#, RST#, INTA:D#, LOCK#, ACK64#, REQ64#,REQ#, GNT#, IDSEL, PME#, INTS, INTP, ENUM#. (Plus GNT3:0#, REQ3:0#, IDSEL via pin headers), REQ1#, GNT1# (System Slot only).
Trigger	8 word recognizers covering all 93 PCI signals and 16 Ext. inputs. True Range & NOT operator on Address/Data. Edge Triggering.	
Range	8 A64 address ranges, 8 D64 data ranges. Inside/Outside	
Sequencer	16 levels with If, Else, Elsif, Goto, Count, Delay, Trigger, Store, Halt	
Trigger Position	0-100%, 1% resolution	
Occurrence / Delay Counters	3 x 32-bits	
Event Counters	8 x 30-bits for statistics	
Real-time Statistics Counters	47 x 30-bits counters	
Decode Speed Counter	1 dedicated counter	
Time Tag	Range	30ns-4688min @ 3 MHz 15ns-2344min 30sec @ 66MHz 10ns-1562min @ 100MHz 7.5ns-1172min @ 133MHz
	Resolution	30ns @ 33MHz 15ns @ 66MHz 10ns @ 100MHz 7.5ns @ 133MHz
Latency Tag	Counts latency (wait states) from FRAME# to TRDY# asserted. Max count: 64 clocks.	
Trigger Output	LVTTL level trigger output with programmable polarity, level or pulse. May pulse on each stored sample. Available on pin header in front panel.	
External Inputs	VG-PCI	15 TTL level inputs on pin header on back panel. 1 TTL level input in front panel.
	VG-PMC / VG-cPCI	8 TTL level inputs on pin header. 8 TTL level inputs on front panel.
Exerciser		
Master	Zero-wait-states, 1GB/s @ 133MHz peak burst rate, 64-bits Address (PCI-X only), 2 DMA Controllers.	
Target	8MB SDRAM memory, 1GB/s peak burst rate @ 133MHz. 32/64-bit address. 256 byte I/O space memory.	
Protocol Checker		
PCI-X Violations:	71 Protocol Violations	
PCI Violations:	45 Protocol Violations	
System Controller (VG-cPCI)		
Clock Generator:	25, 33, 50, 66, 100 and 133MHz	
Arbiter	REQ[0:7], GNT[0:7] monitored	

Table 3: Technical Specifications

General	
PCI-X/PCI Bus	32/64-bit, up to 133MHz
Interfaces	-USB port 12MB/s -Ethernet port 10/100MB/s
Power Supply Requirements	
VG-PCI	+3.3VDC +/-5% from PCI backplane or from ext. power supply via front panel inlet. 1.8A (6W) idle. 5A (16.7W) max
VG-PMC	+3.3VDC +/-5% from PMC connector or from ext. power supply via front panel inlet. 1.8A (6W) idle. 5A (16.7W) max
VG-cPCI	+3.3VDC +/-5% from cPCI connector. 1.8A (6W) idle. 5A (16.7W) max
Dimensions	
VG-PCI	174.6mm x 106.7mm (short card), 1 PCI slot
VG-PMC	74.0mm x 149.0mm (single-wide PMC card)
VG-cPCI	100mm x 160mm
Compliant to	
VG-PCI	PCI Rev. 2.3 PCI-X rev. 1.0a
VG-PMC	PCI Rev. 2.3 PCI-X Rev. 1.0a. IEEE 1386.1
VG-cPCI	PCI Rev 2.3 PCI-X 1.0a. PICMG 2.0 R3.0, PICMG 2.1 Hot Swap R2.0, PICMG Specification ECR R0.6a to PICMG 2.0 R3.0
Measurements	
Temperature Probe	0-120°C / 32-248°F
Voltage	3.3V, 5V, 12V
Operating Temperature	0-50°C / 32-122°F

Table 4: Ordering Information

Ordering Information	
VG-PCI	133MHz PCI-X & PCI State Analyzer and Statistics Module for 5V and 3.3V systems. Includes PCI-X carrier, SAM module and BusView GUI.
VG-PCI0SL	133MHz PCI-X & PCI State Analyzer and Statistics Module for use in single slot systems. Includes expansion connector for the device under test. Includes VG-PCI0SL-Carrier and VG-SAM module.
VG-PMC	PMC State Analyzer and Statistics Module. (Single PMC for all functions) Includes BusView GUI.
VG-cPCI	cPCI-X & cPCI State Analyzer and Statistics Module for 5V and 3.3V systems. Includes cPCI carrier, SAM module and BusView GUI.
Options	
VG-P	PCI-X & PCI Protocol Checker license key for Vanguard product line.
VG-E	100MHz PCI-X & PCI Exerciser and Compliance Checker license key for Vanguard product line.
VG-E2	133MHz Enhanced PCI-X Exerciser with Error Injection license key for Vanguard product line. This option also includes the full functionality of the -E option
Adapters	
401-VG-EPSU	External Power Supply
cPCI-PMC-ADA/64	Adapter to use a PMC module in cPCI systems. Up to 33MHz.
cPCI-3U6U-ADA	6U Adapter for 3U cPCI boards

Individual boards and SAM modules may be purchased separately. Packages are also available. Please consult Curtiss-Wright.

Warranty

This product has a one year warranty.

Contact Information

To find your appropriate sales representative:

Website: www.cwcdefense.com/sales

Email: defensesales@curtisswright.com

Technical Support

For technical support:

Website: www.cwcdefense.com/support

Email: support1@cwcembedded.com

The information in this document is subject to change without notice and should not be construed as a commitment by Curtiss-Wright Controls Defense Solutions. While reasonable precautions have been taken, Curtiss-Wright assumes no responsibility for any errors that may appear in this document. All products shown or mentioned are trademarks or registered trademarks of their respective owners.