

# PCI Express Protocol Analysis, Debug and Compliance



PCI  
EXPRESS™



**Agilent Technologies**

© Copyright 2004 Agilent Technologies, Inc.

# Agenda



## Design Verification Steps

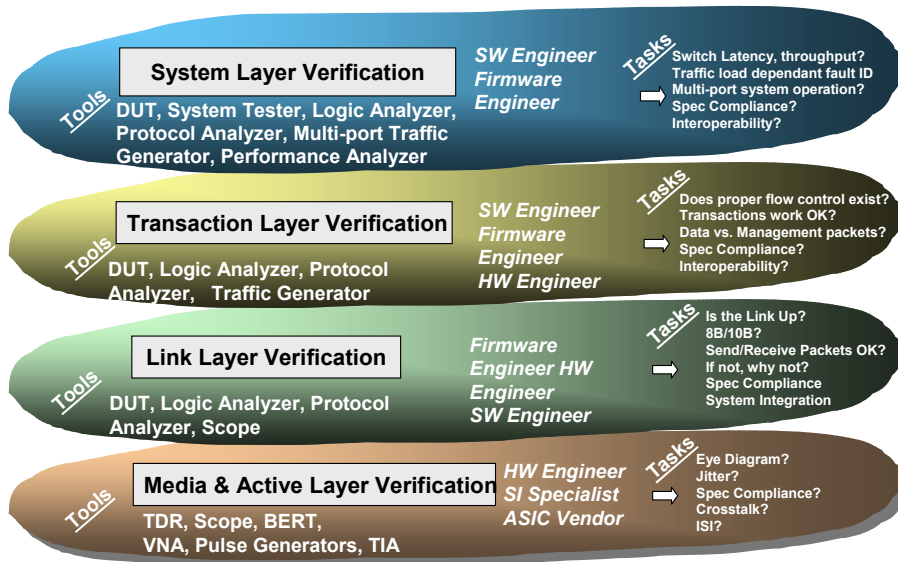
Logic Analysis Tools For PCI Express Development

Protocol Tools For PCI Express Development

Summary/Resources

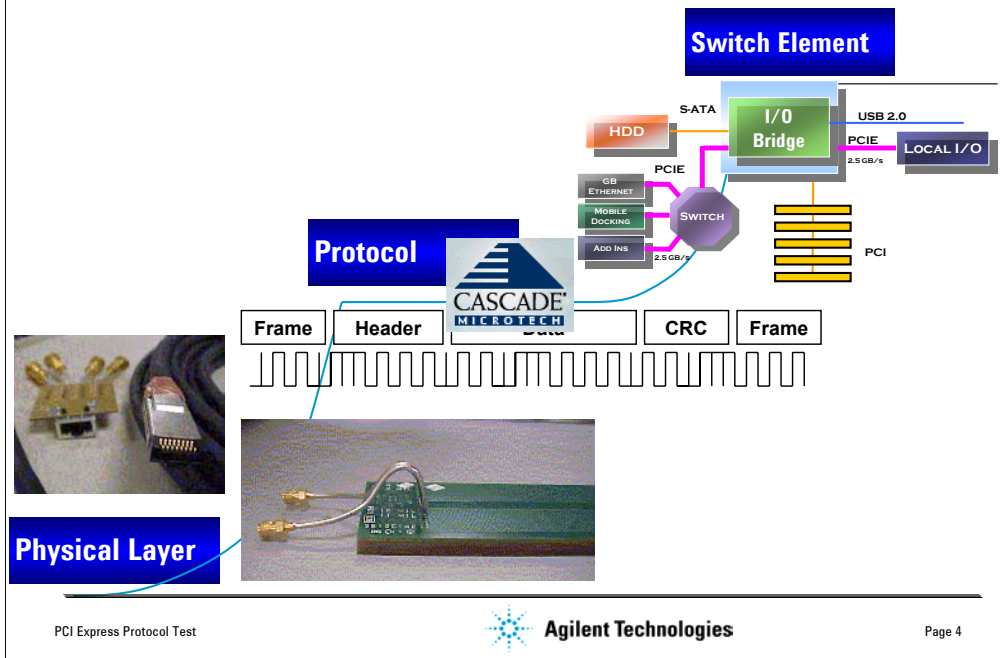
This section will highlight some of the problem areas associated with high speed serial design.

# PCI Express Verification Tasks



At each layer from the media/physical layer to the system layer, different tasks requiring different tools by engineers with different skill emphasis are required to ensure a properly functioning product.

# The Three Trials of PCI Express Design



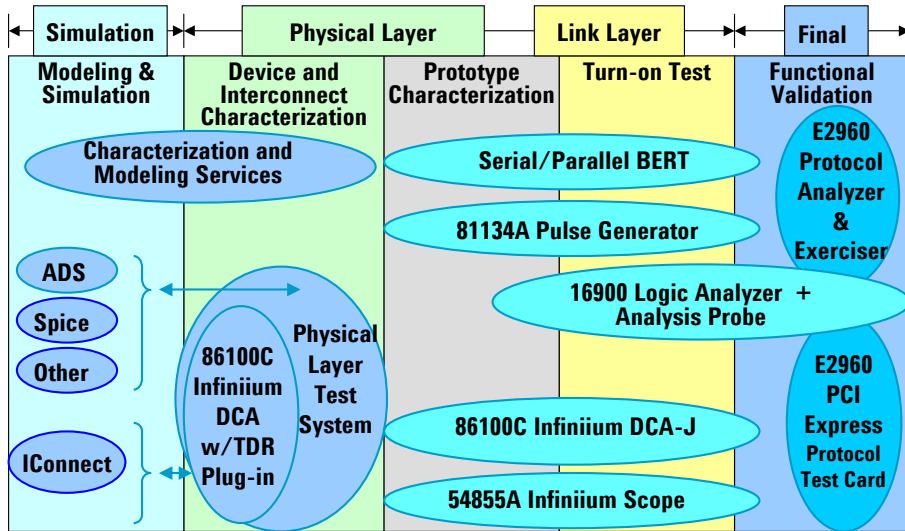
For Digital Designers there are three new problem areas that you will need to tackle as a result of transitioning to designs using high speed serial interconnects such as PCI Express. With edge rates at 100ps and faster, signal integrity becomes a critical factor in design success. Second, the encoding of data using 8b/10b requires that tools be able to display data in it's native format, preferably showing more than just 0's and 1's. Lastly, the replacement of the multi-drop parallel bus with a sophisticated switch will require stimulus generation to play a much more active role in system testing and corner case exersizes of traffic on the serial bus.

# Design Verification Steps

	Simulation	Physical Layer	Link Layer	Final	
	<b>Modeling &amp; Simulation</b>	<b>Device and Interconnect Characterization</b>	<b>Prototype Characterization</b>	<b>Turn-on Test</b>	
<b>Challenges</b>	<ul style="list-style-type: none"> <li>•Model Accuracy</li> <li>•Simulation Speed</li> <li>•Fixture Effects</li> <li>•High Speed Effects</li> </ul>	<ul style="list-style-type: none"> <li>•Test Fixtures</li> <li>•Probing</li> <li>•Repeatability of Measurement</li> <li>•Conformance to Specification</li> </ul>	<ul style="list-style-type: none"> <li>•Probing</li> <li>•Stimulus</li> <li>•Conformance to Specification</li> <li>•Jitter, ISI</li> <li>•8B/10B Implementation</li> </ul>	<ul style="list-style-type: none"> <li>•Link Bring-up</li> <li>•System Integration</li> <li>•Packet transmission</li> <li>•Conformance to Specifications</li> </ul>	<ul style="list-style-type: none"> <li>•Corner Case Testing</li> <li>•Switch Validation</li> <li>•Conformance to Specification</li> <li>•Interoperability</li> <li>•Boot Apps/OS</li> </ul>
<b>Solution Approaches</b>	<ul style="list-style-type: none"> <li>•Measurement Based Modeling</li> <li>•Non-linear simulation</li> <li>•De-embedding and Calibration of instrumentation</li> </ul>	<ul style="list-style-type: none"> <li>•High Bandwidth Probing</li> <li>•Characterization of Test Fixture Effects</li> <li>•Access to Applications Knowledge</li> <li>•Accurate Tools</li> </ul>	<ul style="list-style-type: none"> <li>•High Bandwidth Probing</li> <li>•Low jitter signal sources</li> <li>•Jitter applications</li> <li>•Protocol Aware Measurement Tools</li> </ul>	<ul style="list-style-type: none"> <li>•Protocol Aware measurement tools</li> <li>•Flexibility of Probing</li> <li>•Coordinated Parallel and Serial bus Measurement</li> <li>•Protocol Compliant Signal Sources</li> </ul>	<ul style="list-style-type: none"> <li>•Complex stimulus generation</li> <li>•Test automation support</li> <li>•Protocol aware measurement tools</li> <li>•Error injection real-world stimulus</li> </ul>

If we look deeper into what challenges and measurement solutions are required for each of the five phases, we see that Signal Integrity, probing of high speed serial signals, conformance to standards, and interoperability are major challenges. You need the right equipment and measurement expertise to get product to market quickly.

# Agilent PCI Express Tool Mapping



This slide shows how Agilent solutions map to each area of the PCI Express development task flow.

# Agenda



**Design Verification Steps**

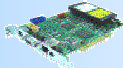



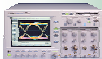




**Logic Analysis Tools For PCI Express Development**

**Protocol Tools For PCI Express Development**

**Summary/Resources**

This section will highlight logic analysis solutions for PCI Express.

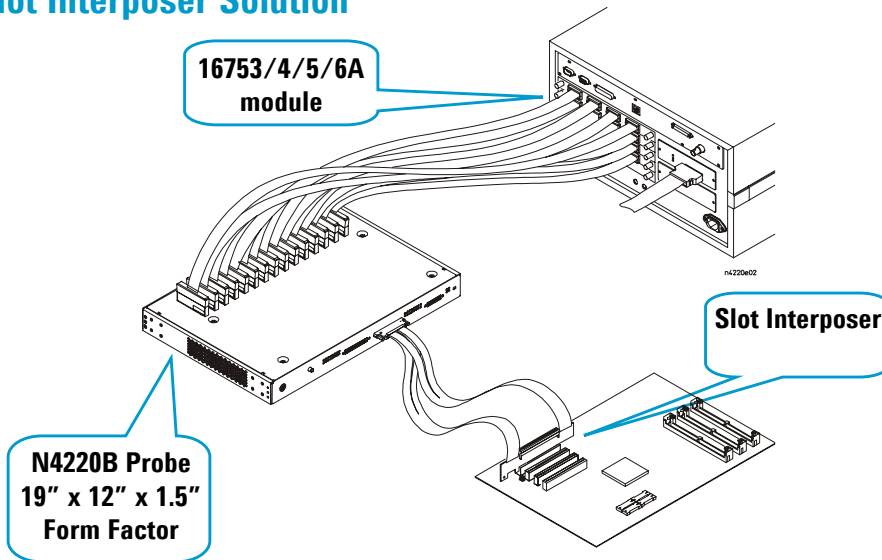
# Agilent's Tools for PCI Express Validation

<b>Product lifecycle</b>	<b>Remote Mgmt &amp; Diagnostics</b>	<b>Remote Management Card 3.0</b> 
	<b>Protocol Validation</b>	<b>Protocol Analyzer/Exerciser</b> 
	<b>System Validation</b>	<b>Packet Analysis Probe</b>  <b>Logic Analyzer</b> 
	<b>Physical Layer Test</b>	<b>DCA-J</b>  <b>3.35G Pulse Generator</b>  <b>Infiniium Oscilloscope</b>  <b>7/13 G SerialBERT</b> 
	<b>Automated Test</b>	<b>Agilent 93000 SOC Series</b> 

This section will focus on the system validation aspect of the product lifecycle.

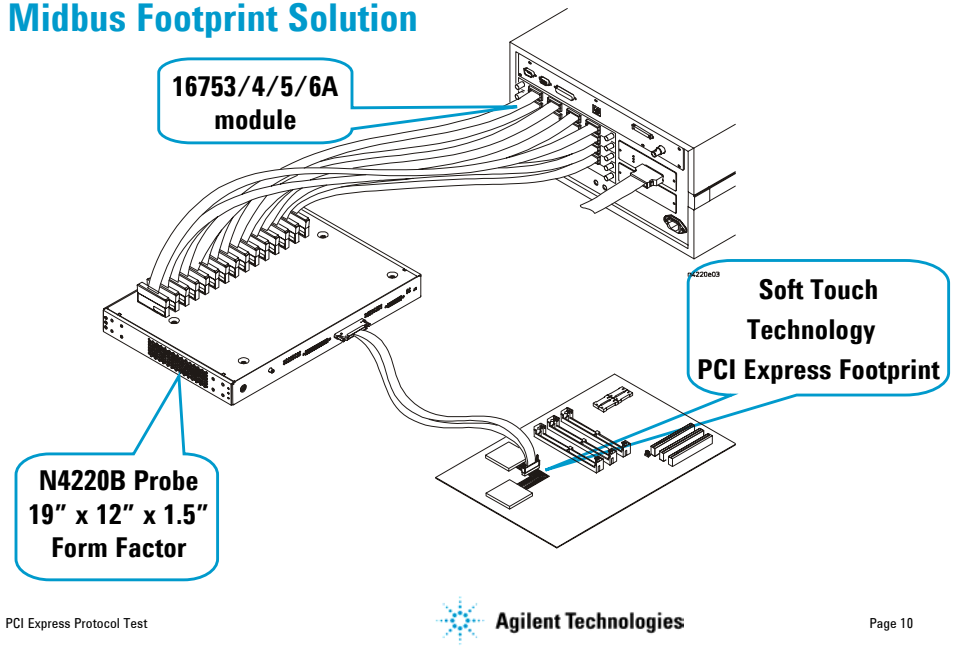


## PCI Express Packet Analysis Probe Slot Interposer Solution



The general look of the logic analysis solution is probing the target system with 1 of 2 types of probes (Mid-bus, Slot Interposer), probe cables extend from the front of the Analysis Probe (a rack mount width, ~1 ft. deep by ~ 1.5 in. high "pizza" box). The 16753 family logic analyzer cables plug directly into the top of the Analysis Probe box along the back without needing an adapter.

## PCI Express Packet Analysis Probe Midbus Footprint Solution



The general look of the solution will be probing the target system with 1 of 2 types of probes (Mid-bus, Slot Interposer), probe cables extend from the front of the Analysis Probe (a rack mount width, ~1 ft. deep by ~ 1.5 in. high "pizza" box). The 16754 family logic analyzer cables plug directly into the top of the Analysis Probe box along the back without needing an adapter.

Mid-bus (a connectorless solution) can be used in in chip-to-chip applications (where no end connection is available).

## Digital Validation for PCI Express using Logic Analysis

- **Probing**
  - Non Intrusive observation of data flow
  - Validation of PCI-Express Devices, Add-in Cards, Systems and Chip-to-Chip Architectures
- **Packet Analysis and Display**
  - Dedicated Packet Triggering
  - Software Decode Tool
- **Full System Validation**
  - Cross-Bus Analysis (Time Correlated)
  - Cross-triggering correlation between scope and analyzer

**Software**

**Transaction**

**Data Link**

**Physical**

**Mechanical**

The Agilent logic analysis solution for PCI Express. Our packet analysis probe adds value when validating anywhere from the physical and transaction layers.

## N4220B Probe Specs/Requirements

### Logic analyzer support

- 16700B or 16702B
  - 16753-series modules
- 16900A or 16902A
  - 16753-series modules or 16950A modules

### Extremely fast lock time

- Requires 12 Fast Training Sequence Ordered Sets for bit/frame sync (48 symbols)

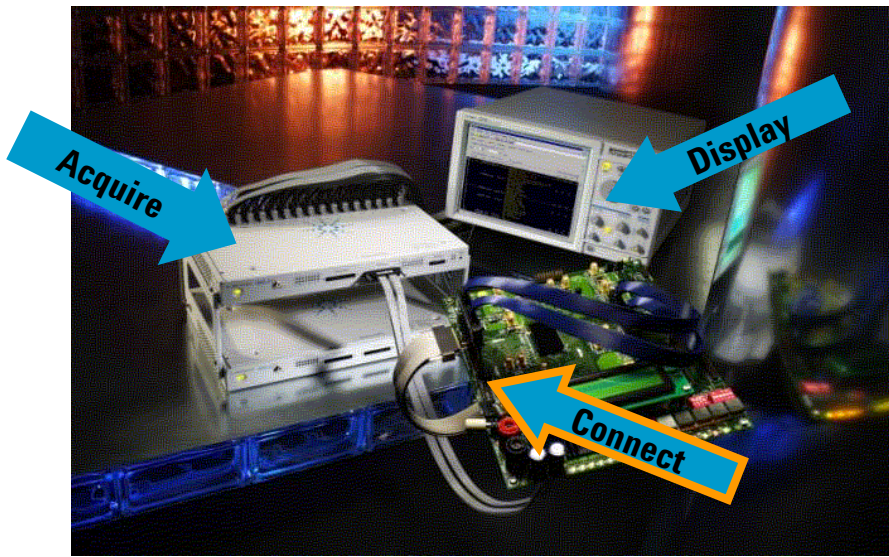
### Industry best jitter tolerance

- Minimum eye width of 120ps



Here are the key specs for the N4220B probe.

## Logic Analysis PCI Express Validation Steps



Let's look at the first of three system issues – Connecting to our device.

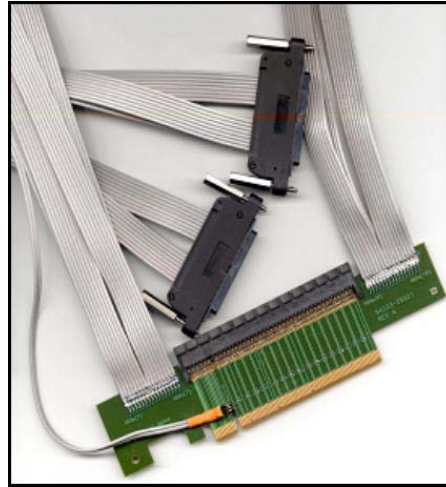
## Connect: Low Intrusion is a must

### Types of connection

- **Chip-to-Chip Link – PCI Express footprint**
  - **Soft Touch technology**
- **Card Edge**
  - **Slot connector/interposer**
- **Flying Lead**

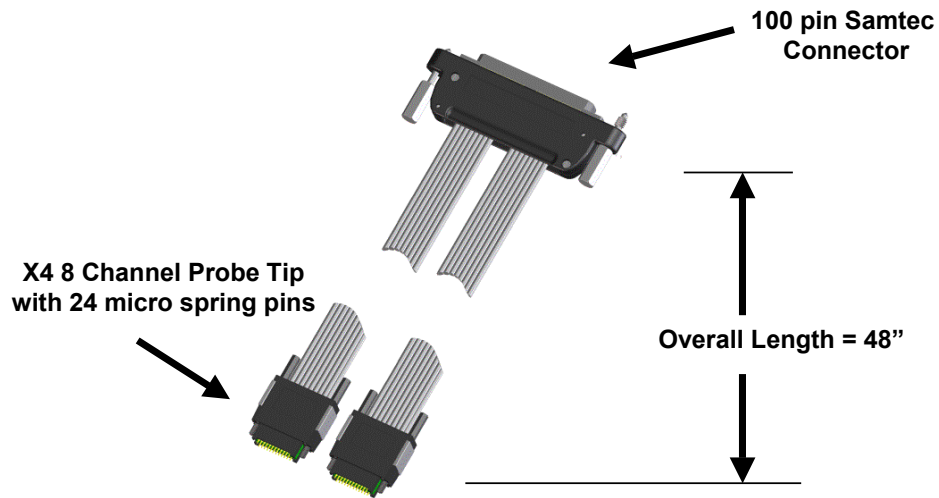


PCI Express Protocol Test



There are several types of connections that can be made to a PCI-E link. For chip-to-chip links, there is a standard PCI-Express footprint that Agilent probes using our soft touch technology. For card edge applications, we offer what is commonly called an interposer, or sometimes called a slot connector. And for applications where there is no place to probe, we offer a flying lead probe that allows access to the 2.5Gb/s data.

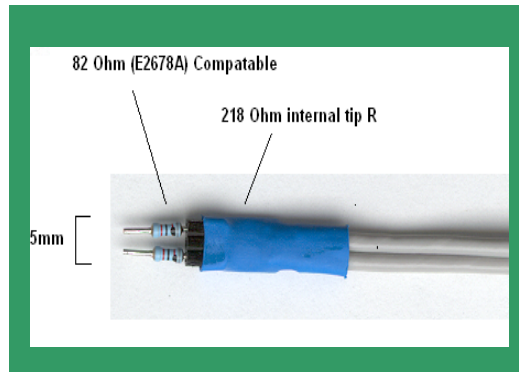
## N4228A PCI Express X4 Midbus Probe



The N4228A is a half size midbus cable that is ideal for x1, x2 and x4 lane widths where board space is at a premium. It is the smallest probe available on the market today.

## N4221F Flying Lead Set

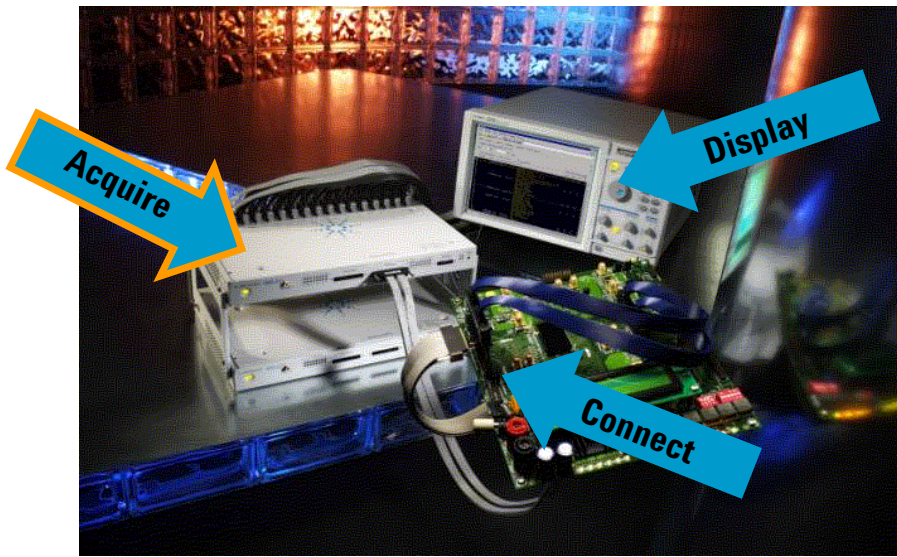
- **Probing of Individual Lanes regardless of routing**
- **Small KOV**
- **Small Loading**
- **InfiniiMax Compatible**



The flying leadset is ideal for customers who may have no board space for a midbus footprint, or who's midbus footprint may have accidentally disappeared between board revs. It provides full speed access to the 2.5Gb/s, and is small enough to probe in extremely tight spaces. It provides support for up to x16 PCI-E, and has the added benefit of being compatible with the Infiniimax scope probes as well (they both use the same 82-ohm resistors).



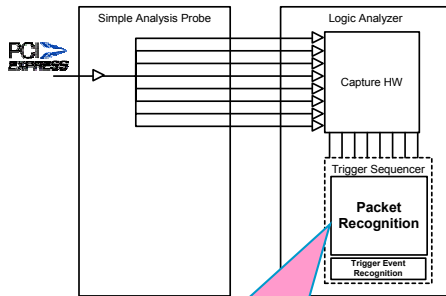
## Logic Analysis PCI Express Validation Steps



Let's look at the second of three system issues – Acquiring the signals.

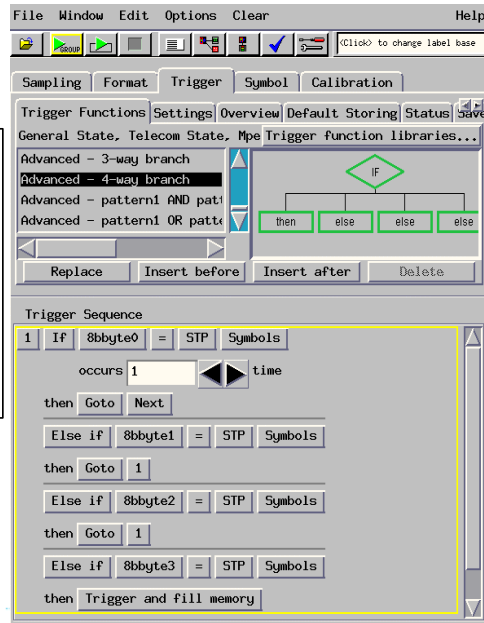
# Acquire: PCI Express -- Traditional LA

## Traditional Serial Analysis Probes



**Packet Recognition Consumes MOST of the Trigger Sequencer Resources**

PCI Express Protocol Test



8bbyteX indicates the lane in which the packet starts.

In a x16 configuration, packets may start on lane 0, 4, 8, 12 (represented by 8bbyte0, 8bbyte1, 8bbyte2, 8bbyte3, respectively)

To dramatically increase triggering capability, we are putting 4 programmable Packet level recognizers into the LAI box for each direction of a link (8 total). These recognizers can be programmed to look for matches to fields within the packet header as well as the first 5 bytes of the data payload. The box will then send simple event recognition notification back to the LA, which can then do full sequencing on those event inputs.

An example: trigger on "packet A" inbound followed by "packet B" outbound followed by "packet C" inbound.

The packet recognition setup is done thru an easy to use hierarchical GUI on the LA to set different fields within a packet to specific values or "Don't care".

## Acquire: PCI Express Packet Recognition

### ***Enhanced Trigger Capability Through Agilent Packet Recognition***

- **8 Header Recognizers will exist in the Analysis Probe (4 Tx/4 Rx) – Implemented in hardware, includes GUI**
  - **24 Bytes in each Header Recognizer (Recognition into Data Portion)**
  - **Recognizers send simple event notification back to Logic Analyzer**
- **Full, Robust Logic Analyzer Sequencing will be available on event notification bits.**

Recognizer 0:	3Dw Memory Read Request	<input type="checkbox"/> Trigger on 0
Recognizer 1:	3Dw Memory Write Request	<input type="checkbox"/> Trigger on 1
Recognizer 2:	DLLP Ack	<input type="checkbox"/> Trigger on 2
Recognizer 3:	DLLP Nak	<input type="checkbox"/> Trigger on 3

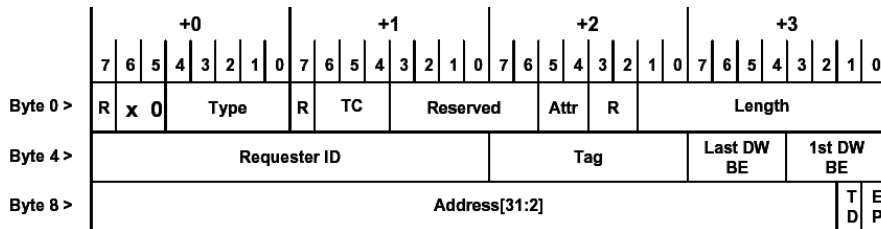
To dramatically increase triggering capability, we are putting 4 programmable Packet level recognizers into the LAI box for each direction of a link (8 total). These recognizers can be programmed to look for matches to fields within the packet header as well as the first 5 bytes of the data payload. The box will then send simple event recognition notification back to the LA, which can then do full sequencing on those event inputs.

An example: trigger on “packet A” inbound followed by “packet B” outbound followed by “packet C” inbound.

The packet recognition setup is done thru an easy to use hierarchical GUI on the LA to set different fields within a packet to specific values or “Don’t care”.

## Acquire: Defining a Packet

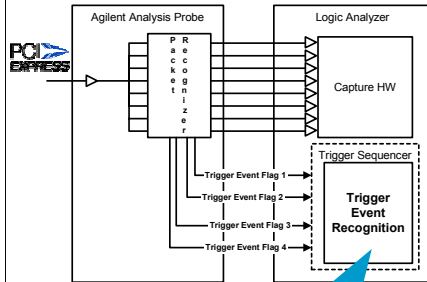
- **Describe your trigger at Packet Level**
- **Analyzer handles bit order, 10B/8B, scrambling, etc.**



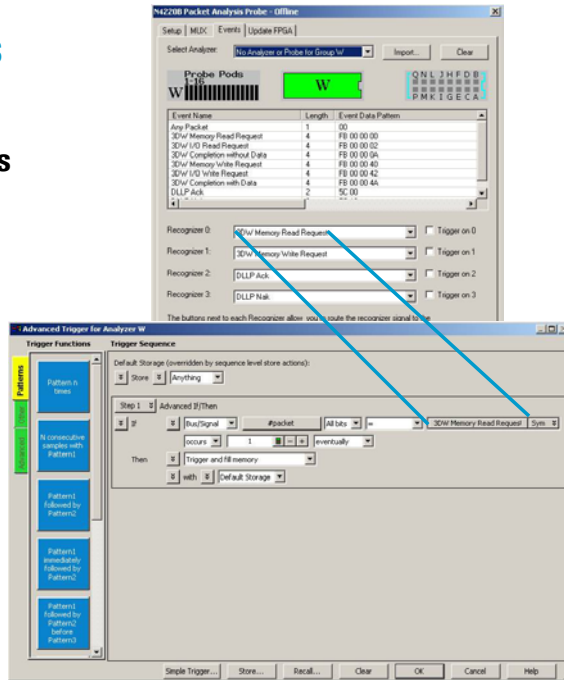
The N4220B allows you to define your packet header with great detail (up to 24bytes) for very precise triggering and searching. The N4220B also handles all of the bit ordering, scrambling, 8b/10b encoding, etc.

# Acquire: PCI Express Packet Recognition

## Agilent N4220B PCI Express Packet Analysis Probe

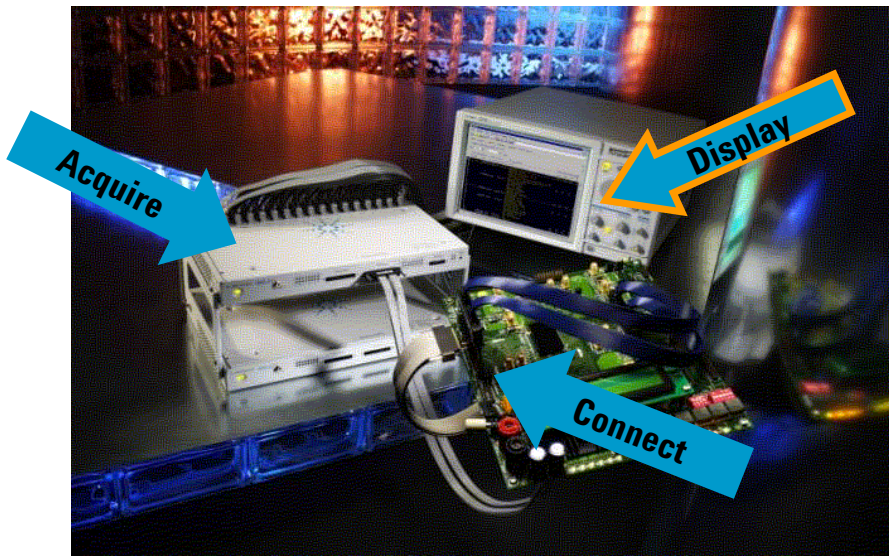


**Full Trigger Resources Available for Complex Triggering**



This gives you an idea of how the packet recognizers work. When you compare this to the slide three before this one, you can see how much easier it is to trigger on a complex packet, and how it does not consume the logic analyzer resources (which is especially important for the larger lane widths where a packet can start in multiple lanes and the trigger must look for it in each of those lanes). Each N4220B has four of the packet recognizers per direction built-in to it (for a total of 8).

## Logic Analysis PCI Express Validation Steps



Let's look at the third of three system issues – Displaying the signals.



## Agilent's Logic Analysis Strengths

- ☑ **Dedicated Packet Triggering**
  - Comprehensive triggering capabilities
  - Reduces time to root cause of problem
- ☑ **16900 and 16700 Logic Analyzer Support**
- ☑ **Quick lock time and minimal eye requirements**
- ☑ **Packet Decode Tool**
  - Packet level information
  - Highly customizable
- ☑ **LIVE, DEMONSTRATED x16 capture**
- ☑ **DEEP Symbol Capture**
  - Up to 2G PCI Express Symbols
- ☑ **Slot, Midbus and Flying Lead Connections**
- ☑ **General Purpose Solution – Logic Analyzer extensible to other technologies**
- ☑ **Cross bus analysis**
- ☑ **Breadth of Agilent's solution**

In summary, Agilent's Logic Analysis solution for PCI Express offers many important benefits.



## Agenda



**Design Verification Steps**

**Logic Analysis Tools For PCI Express Development**

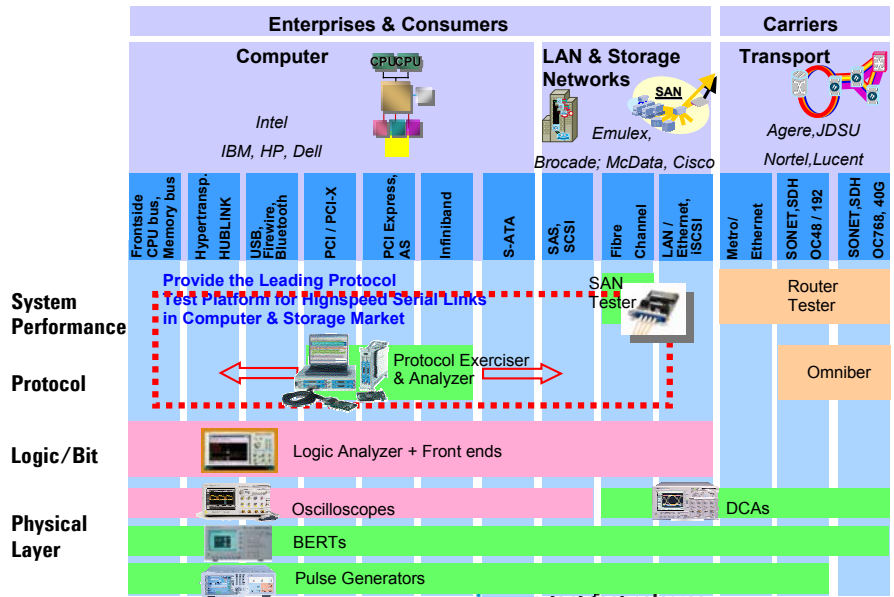
**Protocol Tools For PCI Express Development**

**Summary/Resources**


This section will highlight Agilent's protocol tools for PCI Express.

# Protocol Test Solutions

## PCIe and Fibre Channel Exerciser/Analyzer/Compliance Test



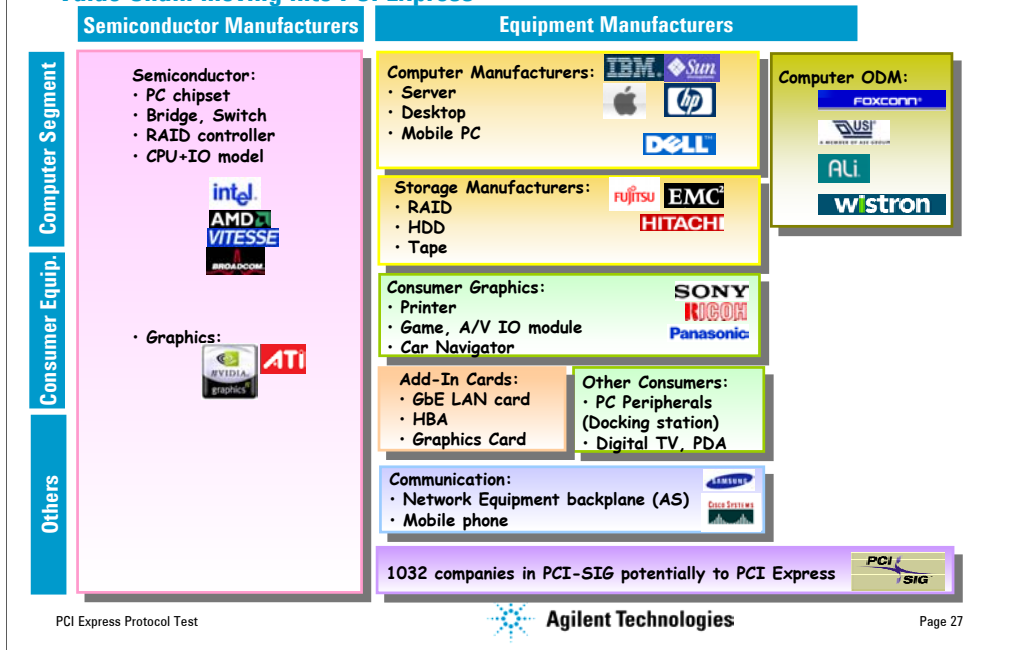
PCI Express Protocol Test

 Agilent technologies

Page 26

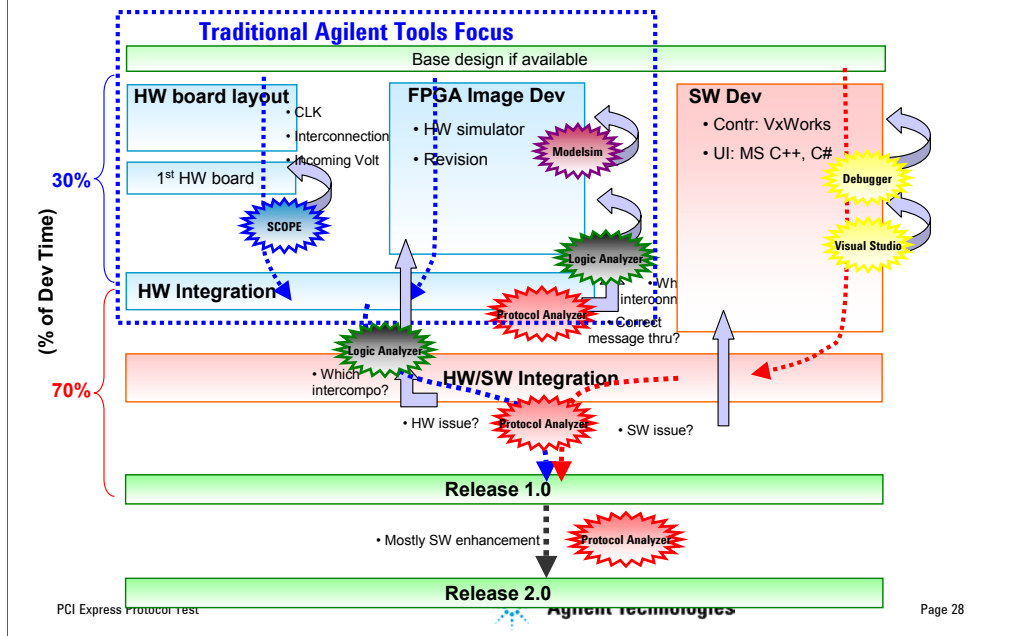
# PCI Express Applications

## Value Chain moving into PCI Express



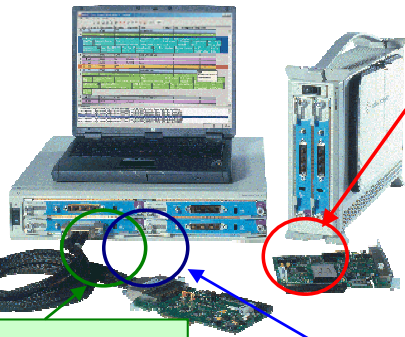
# Test Tools used by Development Phase

## Network Device development example



# Protocol Test Solutions: Only Vendor with...

## PCI Express Exerciser/Analyzer/Compliance Test



**Protocol Test Card:**

- Compliance Test
- Jointly Dev. W/ Intel
- GUI/push-button, Pre-programmed tests, Pass/Failed report

**Analyzer:**

- Patented Dynamic Triggering
- Root cause and performance analysis
- GUI with trigger, search and filter capabilities to intuitively interpret PCI Express traffic

**Exerciser:**

- Generating and responding on arbitrary bus traffic fully controllable
- CAPI, TCL, GUI
- Testing the error behaviour
- Compliant to defined standards
- Device emulation

# Use Models

Single probing solutions for both analyzer/exerciser



**Add-In Card test w/ Combo solutions**

- Protocol Analyzer monitoring bi-directional traffic between Add-in DUT card and known PCIe system
- Protocol Exerciser emulating a known PCIe system device

**System Test w/ Combo solutions**

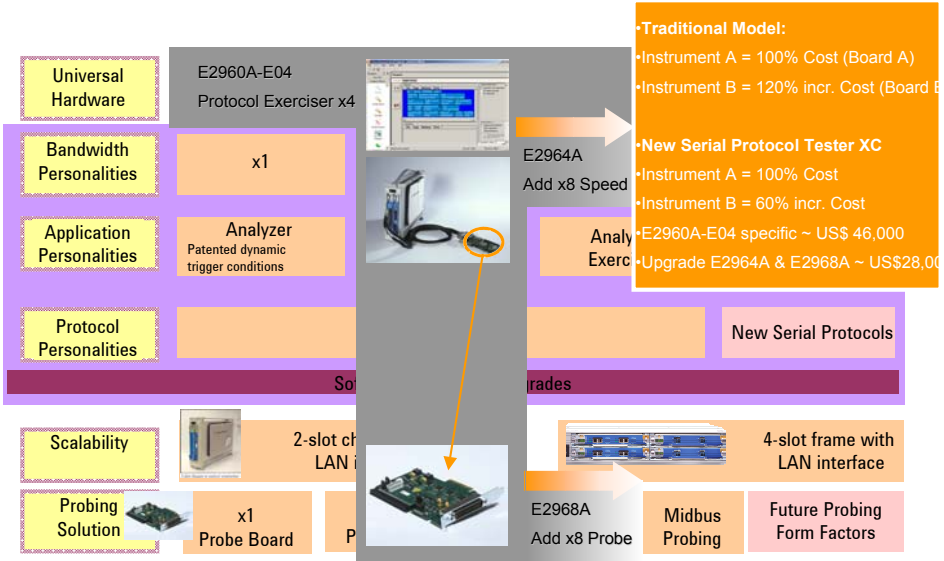
- Protocol Analyzer monitoring bi-directional traffic between System DUT card and known PCIe Add-in card
- Protocol Exerciser emulating a known PCIe Add-in card device

**System & Add-In Card Test ( PTC )**

- Protocol Analyzer monitoring bi-directional traffic between Add-in DUT card and a system DUT
- Protocol Test Card ( PTC ) validating protocol compliance together

# Scenario: Bandwidth Upgrade

## Instrument = PCI E Exerciser from x4 to x8 support



# Protocol Exerciser and Analyzer Key Capabilities Overview

## Analyzer

- Patented Dynamic Triggering
- Root cause and performance analysis
- GUI with trigger, search and filter capabilities to intuitively interpret PCI Express traffic
- Key tool to Debug and bring up PCI Express designs more easily and faster

## Exerciser

- Generating and responding on arbitrary bus traffic fully controllable with the protocol exerciser
- GUI, CAPI and TCL Interface
- Testing the error behaviour of designs by inserting protocol variations with errors
- Supports to be compliant to the industry wide defined standards to validate debug and bring up PCI Express designs easier and faster
- Emulation of devices and automated protocol checking



# Protocol Analyzer Key Features

## Bring up PCI Express designs fast down the design food-chain

- Non-intrusive (“snooping”) Protocol Analyzer supporting PCI Express **x1, x2, x4 and x8**
- Supports Specification **1.0 and 1.0A**
- Sophisticated trigger capabilities with patented **Dynamic Triggering** and storage qualification
- **1GB Trace Memory**
- Root cause analysis
- GUI with **Graphical Trigger setup, search and filter** capabilities to intuitively interpret PCI Express transactions
  - Easy navigation within captured trace
  - Hierarchical view
  - Packet view
  - Transaction view
  - Customizable view of captured transaction
- Analysis of transactions between system and add-in card
- **Slot interposer** probe and soft touch **midbus probe**
- Captures training sequences, Data Link Layer Packets and Transaction Layer Packets - Both directions simultaneously, interleaved display, including time-stamps
- **LAN interface** for remote control and to share applications
- **FPGA based** to easy update

# Protocol Analyzer Key Features

## GUI Protocol Analyzer

Color coded transaction types allow easy recognition of various types of traffic

Tabular view with configurable columns

Packets with errors are highlighted with special background (e.g. red)

Expand/collapse individual packets to get more details

Tooltips for each field provide more detailed information as needed

Context sensitive field decoding

Transaction view pane provides alternate view of traffic (e.g. textual, statistical, etc.)

The screenshot displays the Agilent E2960A Protocol Analyzer interface for PCI Express. The main window shows a list of packets with columns for Direction, Packet Number, Timestamp, Start Tag, Sequence, Type, Length, Tag, Completer, and Log Data. Packets are color-coded by type: SDP (purple), STP (green), and COM (blue). Packet 80 is highlighted in red, indicating an error. A tooltip is visible over packet 80, showing detailed information about the STP (Successful Transfer Protocol) completion status, including completion status (Successful Completion), completion count (0x0004), requestor ID (0x00000030), and data (0x40000000).

Dir.	Packet N	Timestamp	StartTag	Sequence	Type	Length	Tag	Completer	Log Data
↑	24	4544	SDP		UpdateFC-NP				
↑	25	4740	STP	0x0002	Completion with data	0x0001	0x04	0x0100	0x00000000
↑	26	5740	STP	0x0004	Config Read Type 0	0x0001	0x04	0x0100	0
↑	27	6730	COM		Skip Ordered Set				
↓	28	6820	SDP	0x0002	Ack				
↑	29	8228	SDP	0x0004	Ack				
↑	30	8300	SDP		0xEB				
↑	31	8340	SDP		UpdateFC-NP				
↑	32	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	33	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	34	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	35	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	36	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	37	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	38	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	39	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	40	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	41	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	42	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	43	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	44	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	45	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	46	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	47	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	48	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	49	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	50	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	51	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	52	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	53	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	54	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	55	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	56	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	57	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	58	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	59	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	60	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	61	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	62	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	63	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	64	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	65	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	66	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	67	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	68	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	69	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	70	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	71	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	72	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	73	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	74	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	75	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	76	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	77	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	78	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	79	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	80	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	81	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	82	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	83	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	84	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	85	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	86	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	87	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	88	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	89	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	90	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	91	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	92	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	93	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	94	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	95	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	96	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	97	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	98	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	99	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0
↑	100	8436	STP	0x0004	Completion with data	0x0001	0x04	0x0100	Type: Invalid Value Reserved, Reserved not 0

# Protocol Analyzer Key Features

## Trigger Setup Window - With Visual Sequence Viewer

The screenshot displays the 'Trigger Setup (Offline)' window. On the left, there are configuration panels for 'IF' and 'ELSE IF' conditions, each with a 'Goto State' dropdown. The 'IF' panel shows three conditions: 'NOT TRUE', 'AND Counter1', and 'AND OR Counter2'. The 'ELSE IF' panel shows 'NOT TRUE'. The 'Trigger Visualization' area on the right shows a state machine diagram with three states: 'Start' (a blue circle), 'State2', and 'State3' (white circles). Transitions are labeled with conditions and actions: 'Start' to 'State2' is 'ELSE IF TRUE StoreUpstream,StoreDownstream'; 'State2' to 'State3' is 'IF NOT TRUE AND Counter1 OR Counter2 AND NOT Counter2 StoreUpstream,StoreDownstream'; and 'State3' to 'Start' is 'IF TRUE StoreUpstream,S'. At the bottom, there are 'Ok', 'Apply', and 'Close' buttons, and checkboxes for 'Show Conditions' and 'Show Actions'.

# Protocol Exerciser Key Features

## Bring up PCI Express designs fast down the design food-chain

- PCI Express Spec. Rev 1.0 and 1.0a (switchable w/o FPGA Reload)
- Supporting PCI Express x1, x2, x4 and x8
- Supports Specification 1.0 and 1.0A
- 2 M of data memory
- Transmits and receives PCI Express traffic at full bandwidth
- Generate single packets or sequences of packets
- Responds autonomously to incoming requests
- Controlled via GUI, DCOM or TCL interface
- Record and replay Analyzer and Exerciser
- Error Insertion
- LAN interface for remote control and to share applications
- FPGA based to easy update

# Protocol Exerciser Key Features

## GUI - Inject what you need to DUT

The screenshot shows the 'Protocol Exerciser for PCI Express' software interface. The main window is titled 'Request' and displays a configuration for a 'Single Packet'. The interface includes a navigation pane on the left with options like 'General', 'Config Space', 'Decoder', 'Virtual Channel', 'Request', and 'Completion'. The central area shows a table for request details, and the right side has 'General Behaviors' and 'Write Data To ...' settings.

Dir.	Type	Address	Data
↑	Memo	0x000000	write
StartTag	Sequence Number	Fmt*	
STP	auto generated	3DW header, no data	
Type*	Traffic Class*	TLP Digest	TLP poisoned*
Memory Read	000b	Absent	false
Attr*	Length	Requester ID*	Tag
00b	1 DW	0x0000	auto generated
Last DW BE*	First DW BE*	Address*	LCRC
0000b	0000b	0x00000000	correct
			End Tag
			END

General Behaviors:

- automatic TAG generation
- TLP digest present
- TLP poisoned

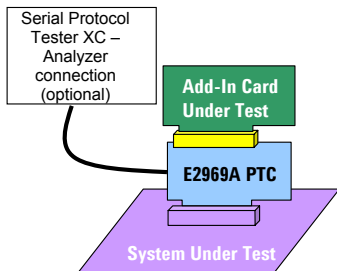
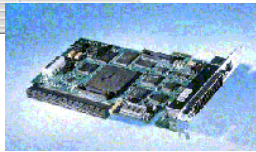
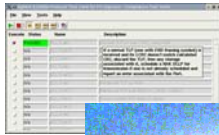
Write Data To ...:

- Single Packet Memory
- Data Generator
- Internal Memory
- Compare With Internal Memory

At: 0x00000000

Length: DWs: 1

## E2969A Protocol Compliance Test (PTC)



PCI Express Protocol Test



Collaboration and joint development with Intel

Cost-effective compliance solution

Easy-to-use

- GUI/push-button solution
- Pre-programmed tests
- No programming effort
- Pass/Failed report

Improves design quality, compliance and accelerates time to market



Page 38

### Protocol Test Card for easy and cheap compliance testing to help enable the PCI Express market

The Agilent E2969A protocol test card performs tests to verify and ensure compliance with PCI Express as defined by the PCI-SIG. In addition, the card also guarantees the interoperability of the DUT with other PCI Express devices. It is primarily for use by R&D engineers who wish to validate the functional compliance of their PCI Express-based designs, including chips, add-in cards or systems.

The protocol test card is a collaborative development between Agilent and Intel that coincides with the Intel product development solution for enabling the PCI Express market.

The objective of the card is to give each engineer working on PCI Express designs the possibility to easily and inexpensively test compliance to the standard of the design without any programming effort.

It offers automated pre-programmed compliance tests for the transaction and data link layer and connection to the protocol analyzer

The PTC is field upgradeable FPGA-based card and offers a USB 2.0 interface for programming and topology simulation mode.

It supports add-in cards with up to x16 lane width

The protocol test card provides three test modes. During the add-in card test mode, the card is plugged in between the system and the add-in card. It monitors the behaviour of the add-in card, its device drivers and operating system in response to a range of inserted errors. The platform test mode monitors the behaviour of the platform, its device drivers and OS, also in response to various error conditions. During the topology simulation mode, the protocol test card appears as a complex PCI Express topology. It aids in verifying the PCI Express compliance of the BIOS, so that it is able to detect and initialize a complex PCI Express topology.

# E2969A Protocol Compliance Test (PTC)

## PTC GUI – main view and Report view

The screenshot displays two windows from the Agilent E2969A Protocol Compliance Test Suite. The main window, titled 'Agilent E2969A Protocol Test Card for PCI Express - Compliance Test Suite', shows a table of test results. The first row is highlighted in green and marked as 'PASSED'. A tooltip for this row provides a detailed description: 'If a normal TLP (one with received and its LCRC does not match the CRC, discard the TLP, free associated with it, schedule transmission if one is not report an error associated with it)'. Other rows show various test cases with 'n/a' status.

The second window, titled 'Agilent E2969A Protocol Test Card for PCI Express - Report', shows a log of test execution steps and their outcomes. The log includes commands like 'Opened the DUT', 'Opened the PTC', and 'Programming PTC with trace buffer mask', along with their corresponding status messages such as 'SUCCESS' and 'START'.

Execute	Status	Name	Description
<input checked="" type="checkbox"/>	PASSED	DLL 5.3#2	Discard TLP on bad LCRC, ...
<input type="checkbox"/>	n/a	DLL 5.2#15	If a normal TLP (one with received and its LCRC does not match the CRC, discard the TLP, free associated with it, schedule transmission if one is not report an error associated with it)
<input type="checkbox"/>	n/a	DLL 5.3#3.1	
<input type="checkbox"/>	n/a	DLL 5.2#2	
<input type="checkbox"/>	n/a	DLL 5.3#3.2	
<input type="checkbox"/>	n/a	DLL 5.2#1	Retransmit TLP on NAK
<input type="checkbox"/>	n/a	DLL 5.2#1.2	Retransmit TLP until REPLAY
<input type="checkbox"/>	n/a	DLL 5.2#10	Ensure correct TLP order in
<input type="checkbox"/>	n/a	DLL 5.2#1.2	Start REPLAY upon REPLAY
<input type="checkbox"/>	n/a	DLL 4.1#2	All reserved fields must be 0
<input type="checkbox"/>	n/a	DLL 5.2#16	DLLP with undefined encod

```

Opened the DUT
</INFO>
<TPTTest user="test_user" timestamp="Thu Nov 27 17:02:01 2003">
  <DUT DeviceId="0x0abcd" VendorId="0x1855">
    </DUT>
    </INFO>
  </DUT>
  </INFO>
Opened the PTC
</INFO>
<ASSERTTAG assertion = "DLL 5.3#2.0" status = "INFORMATIONAL">
  Assertion(s) being tested here
</ASSERTTAG>
</INFO>
<INFO>
START - Programming PTC with trace buffer mask
</INFO>
<INFO>
SUCCESS - Programmed PTC trace buffer mask
</INFO>
<INFO>
START - Programming PTC with command
</INFO>
<INFO>
SUCCESS - Programming PTC with command
</INFO>
<INFO>
START - ARMING PTC
</INFO>
<INFO>
SUCCESS - ARMING PTC
</INFO>
<INFO>
START - Reading VendorID and DeviceID
</INFO>
<INFO>
SUCCESS - Reading DUT Config space
</INFO>
<INFO>
START - DISARMING PTC
</INFO>
<INFO>
SUCCESS - DISARMING PTC
  
```

## CAPI Leverage Summary

- **The objective of the CAPI is to provide the maximum flexibility through our hardware architecture and programming model**
  - **API has been evolved and optimized to meet the new serial protocol aspects**
    - Serial protocol parameters are comprehensively addressed
    - Some parameters are not applicable any more (termination, ...)
  - **Architecture and Resources are similar to PCI**
  - **Programming Model is identical**
    - Block / Behavior programming
    - Property concept
- **Programming expertise can be leveraged to 100%**
- **Similar to moving from PCI to PCI-X**



## Agenda



**Design Verification Steps**

**Logic Analysis Tools For PCI Express Development**

**Protocol Tools For PCI Express Development**

**Summary/Resources**

This section will highlight some web resources for PCI Express.

# Agilent PCI Express Tools



The diagram illustrates the mapping of various Agilent PCI Express tools to the different layers of the PCI Express protocol stack. The stack is represented by five stacked blue boxes on the right, with a large curly bracket on the left grouping them. The tools are arranged around the stack, with lines indicating their application to specific layers.

- E2969A PCI Express Protocol Test Card**: Applied to the **Config./OS** layer.
- E2960 Exerciser/Analyzer**: Applied to the **Config./OS** layer.
- 16900 Logic Analyzer**: Applied to the **Config./OS** layer.
- N1930A Physical Layer Test System (VNA or TDR)**: Applied to the **Physical** layer.
- 86100C DCA-J with TDR**: Applied to the **Physical** layer.
- 81134A Pulse Generator**: Applied to the **Physical** layer.
- N4901 Serial BERT / 81250 ParBERT**: Applied to the **Physical** layer.
- 54855A Infiniium Oscilloscope**: Applied to the **Physical** layer.
- N4220B Packet Analysis Probe**: Applied to the **Transaction** layer.

PCI Express Protocol Test

Agilent Technologies

Page 42

Agilent is ready now with solutions to address problems across the entire spectrum of the PCI express application space. From physical layer testing to application testing.

## Resources



PCI Express Products & Application Info

[www.agilent.com/find/pci\\_express](http://www.agilent.com/find/pci_express)

Signal Integrity Application Info

[www.agilent.com/find/si](http://www.agilent.com/find/si)

PCI SIG

[www.pcisig.com](http://www.pcisig.com)

Agilent's list of application information is continually growing. See the web site on this slide for the latest information on application notes, design guides, and solution information for PCI Express deployment.