


PCI Testing

*PC Developers Conference
1999*

*Barbara P. Aichinger
FuturePlus Systems Corporation*




FuturePlus Systems Corporation
6455 N Union Blvd., Suite 202
Colorado Springs, CO 80918 USA
TEL:719-278-3540
FAX:719-278-9586
www.futureplus.com
tech_sup@futureplus.com

Current Activities

Vice President of FuturePlus Systems Corporation. As an electrical engineer I am involved in both the hardware and software design of many of our products.

Author Background


BSEE and MSEE in Electrical Engineering. My main area of expertise is bus architectures, both I/O and multiprocessing. This has been the focus of my career for the past eleven years. FuturePlus Systems has been involved with PCI since 1993 and is a member of the PCI Special Interest Group. Our bus analysis products are used world-wide and all our products carry Hewlett-Packard part numbers. In addition to PCI, we have bus analysis products for: USB, ISA, AGP, RAMBUS, DIMM, SIMM, VME, VME64, IEEE 1394A and FIBRE CHANNEL.



What's new for PCI testing?

- ◆ Compliance Violations
- ◆ Investigate Signal Integrity
- ◆ Timing Verification
- ◆ Performance Characteristics
- ◆ Validate System and Software Operation

**Barbara P. Aichinger
FuturePlus Systems
36 Olde English Road
Bedford, NH 03110
TEL:603-472-9098 FAX:603-472-9097
e-mail:barb@futureplus.com**



Can I use a logic analyzer to debug my PCI system?

- ◆ Probes connect your PCI to the logic analyzer
- ◆ Software runs on the logic analyzer that transforms it into a bus analyzer
- ◆ A WEB connection allows for remote debugging and easy transfer of data to your application

Yes you can! With products from FuturePlus Systems and Hewlett-Packard Company.

Compliance Violations

Time	ADDR	FUTUREPLUS SYSTEMS c 1998	ADDR_B	AERROR	IN
Relative	Hex	PCI BUS TRANSACTIONS REV 1.2	Hex	Binary	Bi
28,000 ns	730CF370	IDLE	1FFFFFF	1	1
32,000 ns	730CF370	IDLE	1FFFFFF	1	1
28,000 ns	730CF370	IDLE	1FFFFFF	1	1
32,000 ns	730CF374	MEM READ ADDR=730CF374	1FFFFFF	1	1
28,000 ns	730CF374	WAIT-NO DEVICE SELECT	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
28,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
28,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
28,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
28,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
28,000 ns	730CF374	COMPLIANCE VIOLATION	1FFFFFF	0	1
28,000 ns	730CF374	RULE VIOLATION DETECTED	1FFFFFF	1	1
32,000 ns	730CF374	WAIT-TARGET NOT READY	1FFFFFF	1	1
32,000 ns	730CF374	D32=00000000	1FFFFFF	1	1

If a tool tells you a violation has occurred, that is good, but being able to trigger on a violation and see it is better. A mask to filter out known violations can also help to characterize the design you are debugging. What if a violation does occur? How do you dig deeper to find the root cause of the problem? In the trace on the previous slide we saw an illegal deassertion of DEVSEL#. Is this a chip problem? A board problem? A timing problem? A software problem? A backplane issue? A routing problem? How many times have engineers said. "If I could find it ... I could fix it...but the problem is I can't find it!" OR " I spend 90% time looking for the problem and only 10% of my time fixing it." With a logic analysis system you can quickly track the problem down and solve it.

Compliance violation triggering is available on many PCI test tools today. The key to making this feature useful is to be able to quickly determine the cause of a violation.

Signal Integrity

Compliance Violation Triggering

333MHz State:2GHz Timing Zoom 2M Sample C - PCI BUS

File Edit Options Clear Help

Navigate Group Run <Double-click> or push "Replace" to use the current function

Samelinz Format Trigger Symbol

Trigger Functions Settings Overview Default Storing Status Save/Recall

General State Trigger function libraries...

Find pattern n times

Store range until pattern occurs

Store pattern2 until pattern1 occurs

While storing pattern2, find pattern1

Store nothing until pattern occurs

Occurrence 1 Occurrence n

pattern

Replace Insert before Insert after Delete

Trigger Sequence

1 FIND PATTERN N TIMES

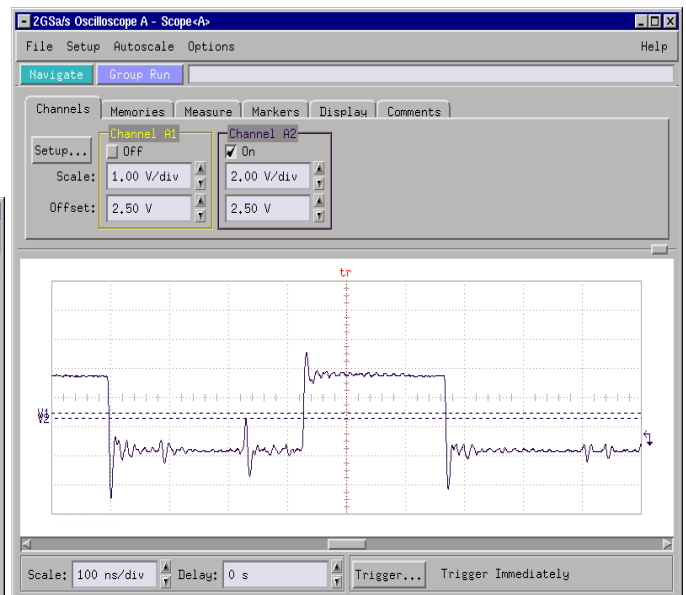
Find 1 occurrence of

AERROR = 0 Hex Or

ADDR_B In range 1FFFFFF 000000 Hex

then Trigger, arm out, and fill memory

Help Close

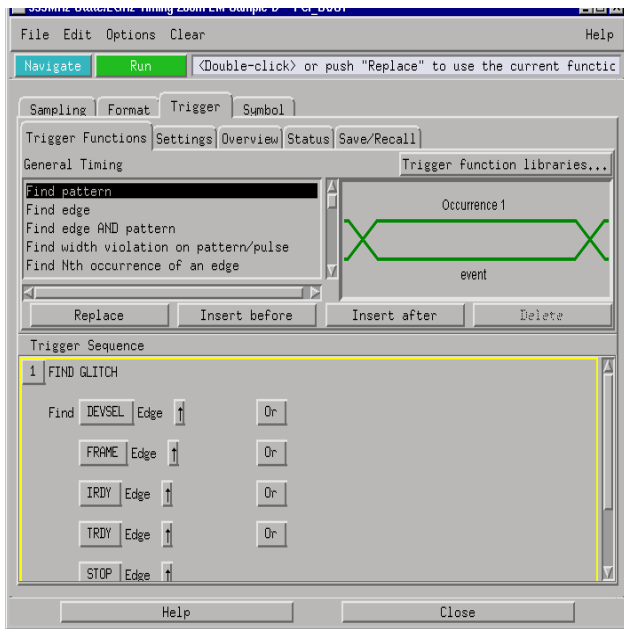


Cross-domain analysis is being able to go from the the state per clock or state per transaction domain to the timing domain and to the analog domain. Here we can actually see the glitch on DEVSEL#. Not what you expected I bet.

Chapter 4 of the PCI Specification covers the electrical environment for PCI. Signal Integrity investigation is most important in high volume applications where variation in parts, connectors, capacitors and resistors can cause failures. These types of failures sometimes show up in manufacturing but in some cases show up in the field when the user tries to install his new PCI board in his existing PCI based system. As all high volume consumer manufacturers know, failures in the field are the most expensive to fix. Costs for customer service, support and field replacement of boards can be astronomical. In addition, a company's reputation for quality can be severely tarnished.

For PCI, a robust signal integrity analysis of the design is a cost effective step in the overall design and verification process.

Timing Analysis

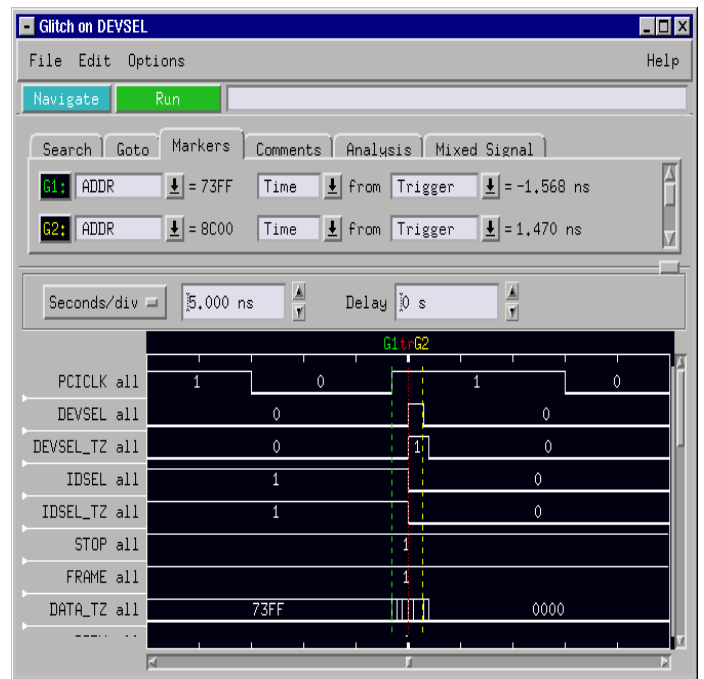


Setup and hold violation triggering can also be done by a logic analysis system. However, let's take that one step further with glitch detection. Glitches from

misaligned state machines, bus contention, timing problems or signal integrity issues can be missed if your only looking for setup and hold violations. What if the problem does not occur near a rising clock edge? Does that mean it is not a problem? Glitches can propagate through logic and cause errors.

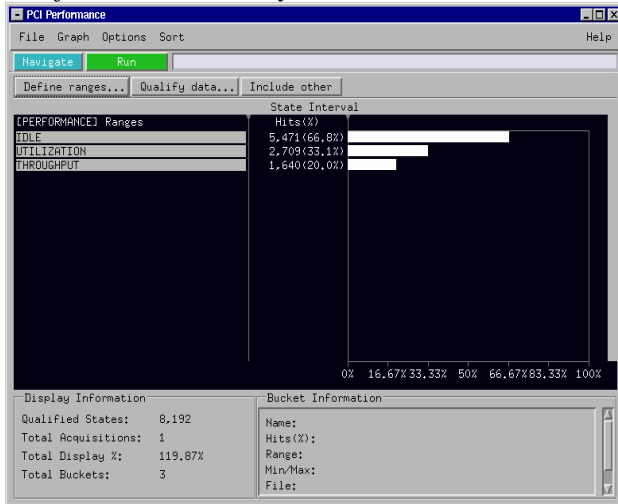
The new HP 16715A, 16716A, and 16717A modules from HP allow for glitch detection on any signal and then a 2Gs/s Timing Zoom to see the problem up close.

Glitch Detection



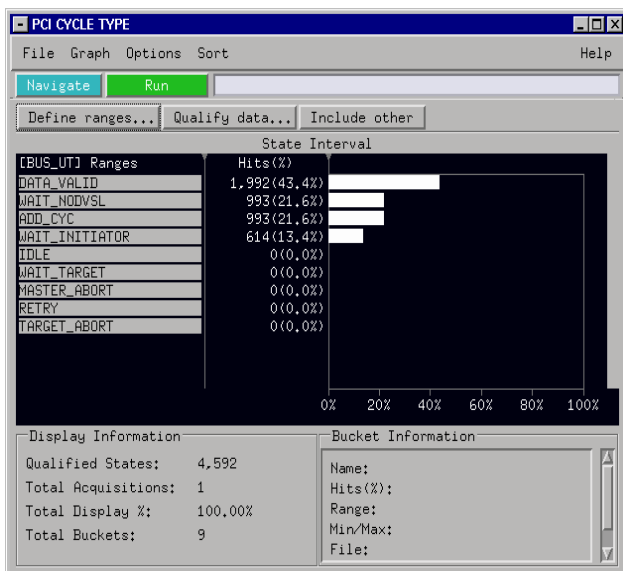
Here is that glitch on DEVSEL#. As you can see it is after the rising edge of the PCI clock. Technically not a hold violation (the spec says 0ns hold). However if the system had a 2ns clock skew, allowable by the spec, this could be a problem elsewhere on the board, inside a chip or elsewhere in the system.

Performance Analysis



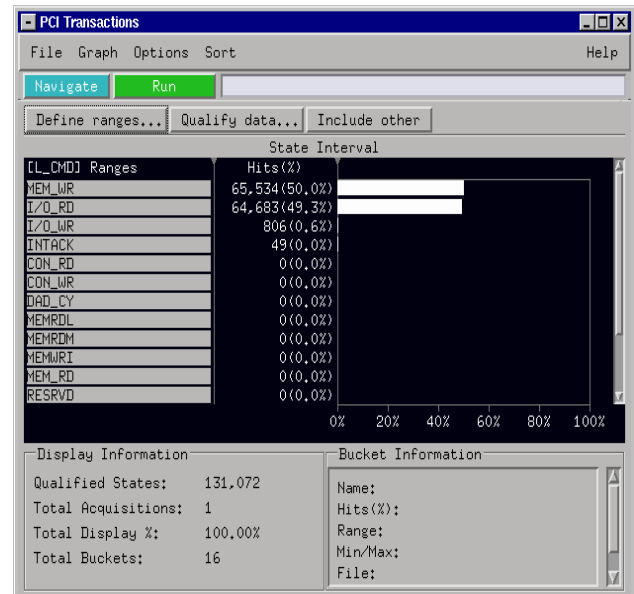
Most all PCI tools today can give you Bus Utilization, Throughput and Efficiency as you can see here. But what if you don't like what you see? Can you dig deeper to see where the problem lies? Can you find the area of your design or system where improvements can be made?

Bus Utilization : a more detailed picture



With a logic analysis system you can! A cycle by cycle breakdown of Bus Utilization can quickly help you characterize your design or that new add-in card you just added to the system. You can take this view one step further and dig even deeper. You can qualify this type of a display to show only the Bus Utilization of a certain bus master or by accesses to certain address ranges or even further qualify it by transaction type (Memory Writes for example).

Throughput: a more detailed picture



Throughput is a measurement of what percentage of the time data is actually being transferred over the bus. But how is it being transferred? What transactions are being used? What bus master is responsible for which percentage of the throughput?

These questions can all be answered by using your logic analysis system.



System Analysis

- ◆ Cross Domain Analysis
- ◆ Cross Bus Analysis
- ◆ Source Code to Signal

PCI has come along way since 1993. At the first Compliance Workshop folks were just happy that the data got transferred correctly. Now entire systems depend on the reliability and performance of the PCI bus. Most PCI based systems today have several PCI buses or PCI buses connected to other I/O and memory buses. How do you validate the entire system? Your logic analysis system has the capability to do just that.

First let's discuss Cross Domain Analysis. That is what we saw earlier. This helps determine if we have a simple logic bug, a timing problem or an analog signal integrity problem. Cross Bus analysis is what we will look at next. This helps us track the data as it goes from bus to bus through the system.

The last area we will touch on is tracking the source code that is running on the system right down to the PCI bus.

PCI to USB

Decimal	AD/CONTROL STATE NUMBER	AD/CONTROL/FUTUREPLUS SYSTEMS c 1999	SB/FUTUREPLUS SYSTEMS c 1999	Time	Relative
-18	HEH READ ADDR=00FC48C	ADP BUS TRANSACTIONS REV 1.0	ADP SIDE BAND ADDRESSING DECODE REV 1.0	3,008 ms	122,244 ns
-17	IDLE			3,008 ms	16,000 ns
-16	DATA= 6159			3,011 ms	16,000 ns
-15	DATA= 6662			3,012 ms	16,000 ns
-14	DATA= 6F6A			3,014 ms	16,000 ns
-13	DATA= 7B7B			3,015 ms	16,000 ns
-12	DATA= 8781			3,018 ms	16,000 ns
-11	DATA= 8F89			3,019 ms	16,000 ns
-10	DATA= 9491			3,019 ms	16,000 ns
-9	DATA= ACAC			3,020 ms	16,000 ns
-8	DATA= 2322			3,023 ms	16,000 ns
-7	DATA= 2326			3,024 ms	16,000 ns
-6	DATA= 231F			3,025 ms	16,000 ns
-5	DATA= 2321			3,025 ms	16,000 ns
-4	DATA= 2124			3,026 ms	16,000 ns
-3	DATA= 211E			3,026 ms	16,000 ns
-2	DATA= 2422			3,030 ms	16,000 ns
-1	DATA= 2324			3,031 ms	16,000 ns
0	HEH WRITE ADDR=003C96C			3,032 ms	16,000 ns
1	D32=0626159			3,032 ms	16,000 ns
2	D32=8F8466A			3,032 ms	16,000 ns
3	D32=8F98761			3,032 ms	16,000 ns
4	D32=86C8491			3,032 ms	16,000 ns
5	D32=2226232			3,032 ms	16,000 ns
6	D32=2221231F			3,032 ms	16,000 ns
7	D32=211F2124			3,032 ms	16,000 ns
8	D32=2324232			3,032 ms	16,000 ns
9	IDLE			3,032 ms	16,000 ns
10	IDLE			3,032 ms	16,000 ns
11	DATA= 261F			3,032 ms	16,000 ns
12	DATA= 2826			3,034 ms	16,000 ns

Here is how we can track data from the PCI bus out onto the USB.

This state listing shows USB isochronous traffic from a USB Camera through the host chip set and onto the PCI. Note that the PCI bridge asserts stop after 32 bytes and the USB has 64 bytes to transfer.

AGP4X

Decimal	AD/CONTROL STATE NUMBER	AD/CONTROL/FUTUREPLUS SYSTEMS c 1999	SB/FUTUREPLUS SYSTEMS c 1999	Time	Relative
-18	HEH READ ADDR=00FC48C	ADP BUS TRANSACTIONS REV 1.0	ADP SIDE BAND ADDRESSING DECODE REV 1.0	122,244 ns	
-17	WAIT-H0 DEVICE SELECT			16,000 ns	
-16	WAIT-TARGET NOT READY			16,000 ns	
-15	WAIT-TARGET NOT READY			16,000 ns	
-14	WAIT-TARGET NOT READY			16,000 ns	
-13	WAIT-TARGET NOT READY			16,000 ns	
-12	D32=00000007			16,000 ns	
-11	HEH WRITE ADDR=00FC48C			402,000 ns	
-10	WAIT-H0 DEVICE SELECT			16,000 ns	
-9	WAIT-TARGET NOT READY			16,000 ns	
-8	D32=00000007			16,000 ns	
-7	I/O WRITE ADDR=000000FF			77,416 ns	
-6	WAIT-H0 DEVICE SELECT			16,000 ns	
-5	WAIT-TARGET NOT READY			16,000 ns	
-4	WAIT-TARGET NOT READY			16,000 ns	
-3	WAIT-TARGET NOT READY			16,000 ns	
-2	D32=00000000			16,000 ns	
-1	STOP			78,000 ns	
0	TYPE 2 LONG READ ADDR 135104100			40,000 ns	
1	LONG READ ADDR 1A810030E0A6000 LEN=128 BYTES			160,000 ns	
2	TYPE 2 READ ADDR 12211640C A15=1			160,000 ns	
3	TYPE 2 READ ADDR 12211640C A15=1			432,000 ns	
4	AGP DATA=FFFFFFFF FFFFFFFF			8,000 ns	
5	AGP DATA=00000000 0000001E			8,000 ns	
6	AGP DATA=FFFFFFFF FFFFFFFF			8,000 ns	
7	AGP DATA=00000000 0000001E			8,000 ns	
8	AGP DATA=FFFFFFFF FFFFFFFF			8,000 ns	
9	AGP DATA=00000000 0000001E			8,000 ns	
10	AGP DATA=FFFFFFFF FFFFFFFF			8,000 ns	
11	AGP DATA=00000000 0000001E			8,000 ns	
12	AGP DATA=FFFFFFFF FFFFFFFF			8,000 ns	
13	AGP DATA=00000000 0000001E			8,000 ns	

For those of you involved in graphics design you will recognize this technology. The AGP specification was based on PCI and PCI traffic can be seen here. The AGP4X data is shown in purple and the Side Band Addressing bus is cross-bus correlated in real-time to the AGP bus so that accurate addressing information can be seen.

State Analysis at 66Mhz

ADDR	FUTUREPLUS SYSTEMS c 1998	AERROR	Time
Hex	PCI BUS TRANSACTIONS REV 1.2	Binary	Relative
006BD21C	MEMRD LINE ADR=006BD21C	1	2.752 us
006BD21C	D32=xxxxFExx	1	624.000 ns
006BFAC0	MEMRD MUL ADR=006BFAC0	1	5.240 us
006BFAC0	D32=8A000064	1	512.000 ns
006BFAC4	D32=00000000	1	24.000 ns
006BFAC8	D32=00000028	1	32.000 ns
006BFACC	D32=00000000	1	32.000 ns
006BFAD0	D32=00000001	1	32.000 ns
006BFAD4	D32=006B5000	1	24.000 ns
006BFAD8	D32=00000200	1	32.000 ns
006BFADC	D32=00008801	1	32.000 ns
006BFAC0	MEM WRITE ADR=006BFAC0	1	42.112 us
006BFAC0	D32=8A000064	1	56.000 ns
006BFAC4	D32=00000000	1	32.000 ns
006BFAC8	D32=00000028	1	32.000 ns
006BFACC	D32=00000000	1	32.000 ns
006BFAD0	D32=00000001	1	24.000 ns
006BFAD4	D32=006B5000	1	32.000 ns
006BFAD8	D32=00000200	1	32.000 ns
006BFADC	D32=xx008801	1	32.000 ns
02FF0000	SPECIAL CYCLE	1	256.632 us
02FF0000	HALT	1	24.000 ns
02FF0000	MESSAGE DATA=02FF	1	32.000 ns
02FF0000	HALT	1	32.000 ns
02FF0000	MESSAGE DATA=02FF	1	32.000 ns
02FF0000	HALT	1	32.000 ns
02FF0000	MESSAGE DATA=02FF	1	24.000 ns

High end PCI system analysis can greatly benefit from a logic analysis system. PCI at 66Mhz can be accurately characterized and debugged with the new HP 16715A, 16716A, and 16717A modules from HP with the 2GHz Timing Zoom.

What happens if you want to trigger on a specific address but sometimes that address falls within a burst? Because of system retries and position independent code you don't know where in the burst that address occurs. You can try to add code to print out key data patterns and trigger on those as I used to do (my favorite data pattern was FEEDFACE) or you can use some of the newer PCI debug tools that increment the address during a burst to the logic analyzer.

RAMBUS


State Number	RAMBUS TRANSACTIONS REV 0.8
Decimal	ROW COLUMN DATA
-1323	ROWA DR=00 BR=04 R10:0=0BE
-1322	COLC WR DC=00 BC=08 AC=2A
	COLM MA=10 MB=00
-1321	
-1320	D31:00=4D550000
	D63:32=54530043
	D95:64=5444492D
	D127:96=6E655000
-1319	
-1318	COLC NOCOP DC=00 BC=08 AC=2A
	COLM MA=00 MB=10
-1317	
-1316	D31:00=296E506F
	D63:32=00420000
	D95:64=54444900
	D127:96=6E655000
-1315	
-1314	COLC NOCOP DC=00 BC=08 AC=2A
	COLM MA=20 MB=00
-1313	
-1312	D31:00=296E506F
	D63:32=00420000
	D95:64=54444900
	D127:96=6E655000
-1311	
-1310	COLC RD DC=00 BC=04 AC=07
	COLX NOCOP DX=00 BX=00
-1309	
-1307	
-1296	D31:00=6D756D75
	D63:32=00490049
	D95:64=746E746E
	D127:96=2D6D2D6D
-1295	

Tracking data to and from memory? Only a properly designed logic analysis system can handle the 400Mhz RAMBUS technology.

IEEE 1394A

Time	PRIMARY	TCODE	SPEED
Relative	Binary	Hex	Hex
102.624 us	1	18	1
40,000 ns	1	18	1
280,000 ns	1	18	1
352,000 ns	1	18	1
360,000 ns	1	18	1
10,824 us	0	02	1
10,504 us	0	01	1
72,368 us	1	14	1
40,000 ns	1	14	1
280,000 ns	1	14	1
392,000 ns	1	14	1
608,000 ns	0	0C	1
10,848 us	0	02	1
3,744 us	1	16	1
40,000 ns	1	16	1
280,000 ns	1	16	1
352,000 ns	1	16	1
360,000 ns	1	16	1

Using PCI to connect to IEEE 1394A? No problem! A logic analysis system can handle the job.

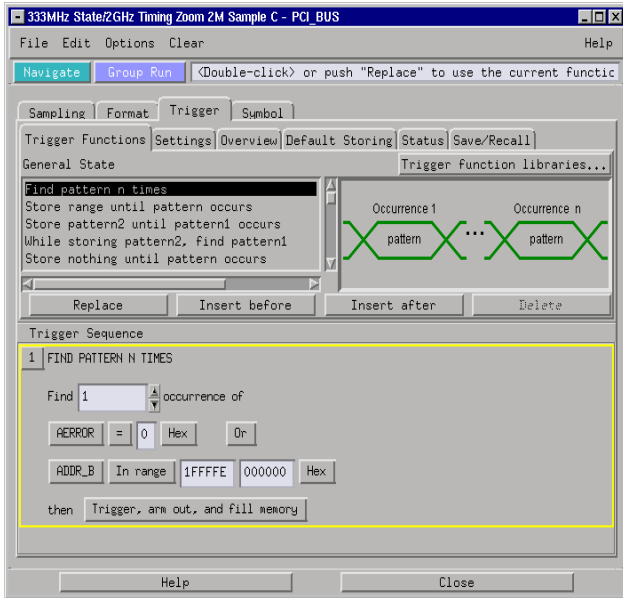


Software Testing

- ◆ Source Code to Execution
- ◆ Execution to Bus transaction
- ◆ Performance Monitoring
- ◆ Feature Filled triggering
- ◆ Deep Memory
- ◆ Post Processing tools
- ◆ WEB accessible

Can a software engineer benefit from a logic analysis system? The “hardware only” glass ceiling has been broken! Manufacturers of logic analysis systems are greatly motivated to serve the needs of the software engineer...and why? Software engineers now outnumber hardware engineers.

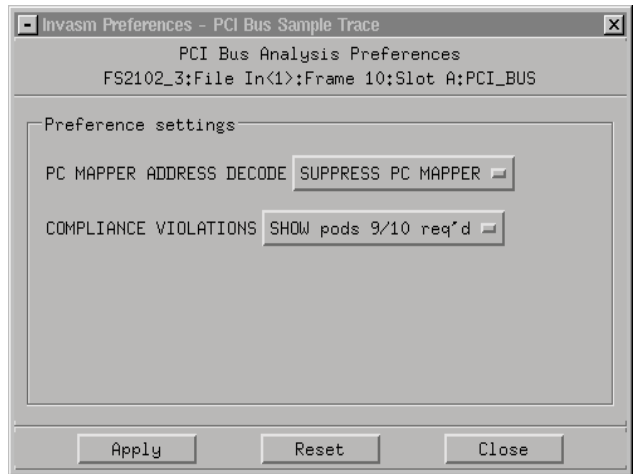
Full featured triggering



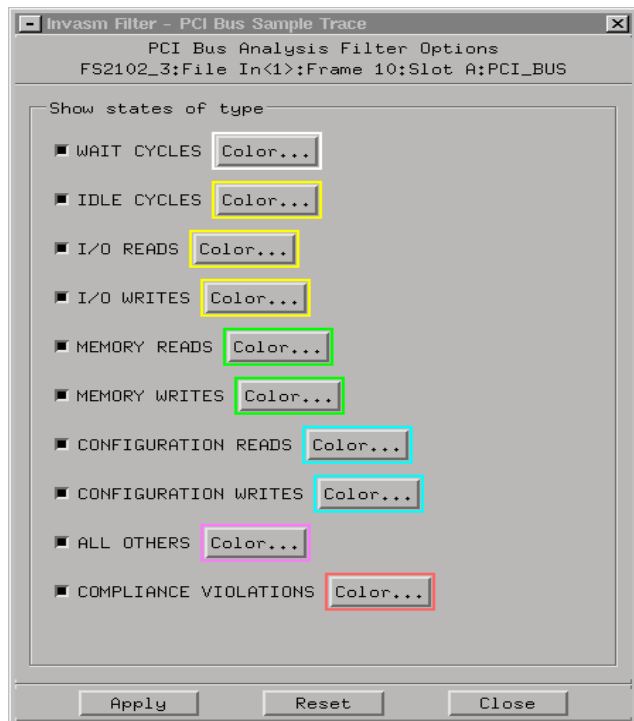
The screenshot shows a software window titled "333MHz State/2GHz Timing Zoom 2M Sample C - PCI BUS". It features a menu bar (File, Edit, Options, Clear, Help) and a toolbar with buttons for "Navigate", "Group", "Run", and "Replace". Below the toolbar are tabs for "Sampling", "Format", "Trigger", and "Symbol". The "Trigger" tab is active, showing a "Trigger Functions" section with sub-tabs: "Settings", "Overview", "Default Storing", "Status", and "Save/Recall". Under "General State", there is a "Find pattern n times" section with a list of actions: "Store range until pattern occurs", "Store pattern2 until pattern1 occurs", "While storing pattern2, find pattern1", and "Store nothing until pattern occurs". A diagram shows two occurrences of a "pattern" connected by a sequence of events. Below this are buttons for "Replace", "Insert before", "Insert after", and "Delete". The "Trigger Sequence" section contains a list with one entry: "1 FIND PATTERN N TIMES". Below this entry are input fields for "Find 1", "AERROR", "In range", and "then" with a dropdown menu.

- Triggering now includes
- 16 level sequencer
- 4-way branching
- 2 global counters
- 2 global timers
- 1 level counter
- 16 patterns
- If-Then -Else Branching
- Triggers store and recall
- Pre-stored trigger macros

PCI Post Processing



The screenshot shows a dialog box titled "Invasm Preferences - PCI Bus Sample Trace". It contains "PCI Bus Analysis Preferences" for "FS2102_3;File In<1>;Frame 10;Slot A;PCI_BUS". Under "Preference settings", there are two options: "PC MAPPER ADDRESS DECODE" set to "SUPPRESS PC MAPPER" and "COMPLIANCE VIOLATIONS" set to "SHOW pods 9/10 req'd". At the bottom are "Apply", "Reset", and "Close" buttons.



The screenshot shows a dialog box titled "Invasm Filter - PCI Bus Sample Trace". It contains "PCI Bus Analysis Filter Options" for "FS2102_3;File In<1>;Frame 10;Slot A;PCI_BUS". Under "Show states of type", there is a list of filter options, each with a "Color..." button: "WAIT CYCLES", "IDLE CYCLES", "I/O READS", "I/O WRITES", "MEMORY READS", "MEMORY WRITES", "CONFIGURATION READS", "CONFIGURATION WRITES", "ALL OTHERS", and "COMPLIANCE VIOLATIONS". At the bottom are "Apply", "Reset", and "Close" buttons.

With the deep memory options available on PCI tools, post processing filters make it easy to remove all but the transactions of interest. Also the address mapping of common I/O and memory addresses help navigate through megabytes of acquired PCI traffic.



All the PCI's

- ◆ PCI Mezzanine - FS3010
- ◆ CompactPCI - FS3020
- ◆ Desktop
 - FS2000 32bit/33Mhz
 - FS2001 64bit/133Mhz
 - FS2005/6 64 bit/66Mhz
 - FS2100/2101 64 bit/33Mhz
 - FS2102/2103 64 bit/66 MHz

Which PCI are you doing? FuturePlus and Hewlett-Packard have teamed up to offer the broadest PCI tool offering in the industry.



All the PCI's

- ◆ CardBus - FS2004
- ◆ EIO - FS3030
- ◆ Embedded PCI
 - Application note on our web site
 - FS1100 32 bit site license
 - FS1101 64 bit site license
- ◆ PCI-X - FS2000 (newer model on the way!)

Embedded PCI? No problem we can show you how.

Do you think PCI-X is in your future? Give us a call...it's in our future too!



Additional Resources

◆ *The Bus Monitor*

A quarterly newsletter that focuses on PC Bus test

◆ FuturePlus Systems WEB page

(www.futureplus.com)

PCI test news and events

Triggering application notes

Product updates and new product information

For more information about the tools mentioned in this presentation please contact:

FuturePlus Systems Corporation
6455 N Union Blvd., Suite 202
Colorado Springs, CO 80918 USA
TEL:719-279-3540

FAX:719-278-9586

www.futureplus.com

tech_sup@futureplus.com