EDN *Hands On* EXCLUSIVE

PCI: REAL VERSUS IDEAL

Can the PCI bus really achieve 132 Mbytes/sec today? EDN's PCI Project Dream Team met for four days in Colorado Springs, CO, to answer this and other PCI-related questions.

STEVEN H LEIBSON, EDITOR IN CHIEF

Early implementations of the PCI bus tarnished its image. Although the bus spec promised 132-Mbyte/sec transfer rates, the first PCI systems barely achieved one-quarter of that rate. Further, few 32-bit μPs can sustain 132-Mbyte/sec transfer rates on their data buses, so processor-based tests on PCI motherboards can't demonstrate PCI's true potential.

Even with these drawbacks, the PCI bus is conquering hardware design. The PCI bus is already the standard for PC motherboards. It's in the latest Macintosh computers and in Digital Equipment Corp's Alpha workstations. Many embedded-system board vendors are jumping on the PCI bus, so PCI is becoming an increasingly important factor in the industrial market. In short, PCI is becoming a key design element in many computer markets. Consequently, I decided that *EDN*'s readers needed to know whether or not the PCI bus could live up to expectations. *EDN's* PCI Project demonstrated that PCI's rise in popularity is merited. We saw excellent performance, closely approaching the theoretical maximum transfer rate.

To accomplish these experiments, I needed some expert help. Four people joined my project team: Ron Sartore, a PC design consultant; Tom Shanley, president of MindShare Inc; Chuck Small, a logic-analyzer project engineer at Hewlett-Packard (HP); and Thomas Dippon, a sales development engineer at HP. (See the **box**, "*EDN*'s PCI Project

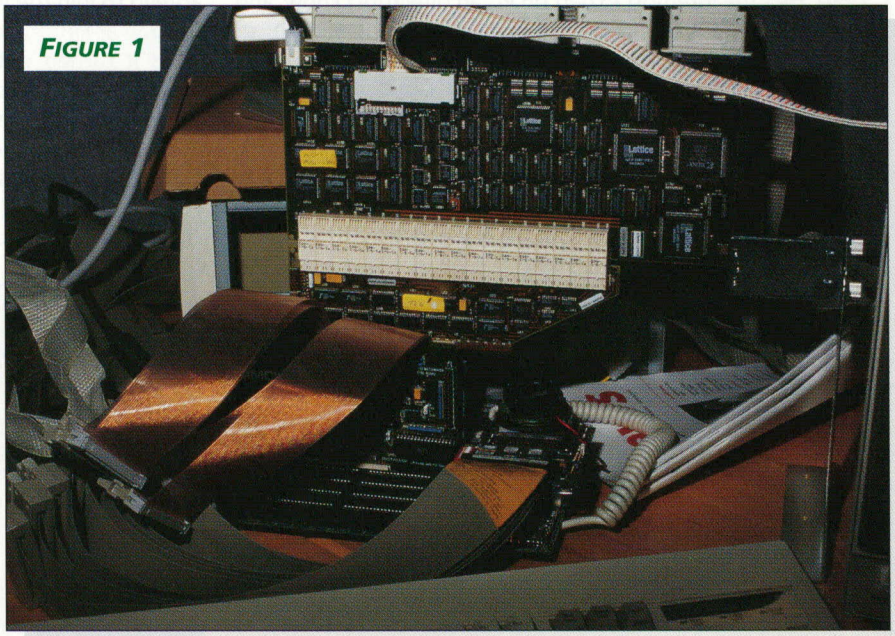


FIGURE 1

By the time you plug HP's PCI Bus Exerciser and Pentium processor probe, FuturePlus Systems' PCI preprocessor card, and AMCC's Matchmaker developer's card into a PCI motherboard, you've got a real basket of snakes and the most-powerful PCI-diagnostic tool kit on the planet.

Dream Team.") Without help from these four people, the project would neither have been completed so quickly nor would have been so successful.

The team met at HP's Colorado Springs, CO, facility on August 14, 1995, for four days. We put several motherboards and expansion cards through their paces. Because we had limited time, we used a comprehensive array of test equipment. (See the **box**, "*EDN*'s PCI Project toolbox.") Although this scenario may seem contrived, I don't believe it is. In almost any PC-development environment, time to market is critical. Each day that you delay a PC's introduction seriously reduces the ultimate return on your development investment, because PC-sales volumes are big and PC-product lifetimes are short. So, like *EDN*'s Team PCI, many development teams using the PCI bus have limited time to solve problems. They need answers quickly.

The instrument we relied on most was the E2910A PCI Bus Exerciser. This device is a large plug-in board, capable of generating and monitoring any type of PCI bus transaction. The PCI Bus Exerciser automatically records all of the transactions it can detect from its PCI slot and monitors every transaction for a PCI bus protocol violation. We could define any sort of PCI test, and Dippon would write and run a test script on the analyzer in less than five minutes.

The PCI Bus Exerciser connects to an HP 16500 logic analyzer and relies on the analyzer for recording bus transactions. These transactions appear on the 16500's CRT. The software included with the E2910A allows a GPIB-equipped PC to download and display the transactions captured by the HP 16500. The PCI Bus Exerciser's PC-based display software presents bus transactions in better detail and more formats than the HP 16500. The software for the PCI Bus Exerciser runs under Microsoft Windows 3.1, so it is fairly easy to use.

Although we were concerned primarily with the PCI bus, we also needed insight into the processor's operation. To gain this knowledge, we used a second 16500 logic analyzer equipped with an HP 16505 prototype analyzer, an E3491A Pentium processor probe, an E2457A preprocessor interface for the Pentium μP, and a PCI preprocessor provided by FuturePlus Systems. This setup allowed us to examine PCI bus

EDN's PCI Project Dream Team

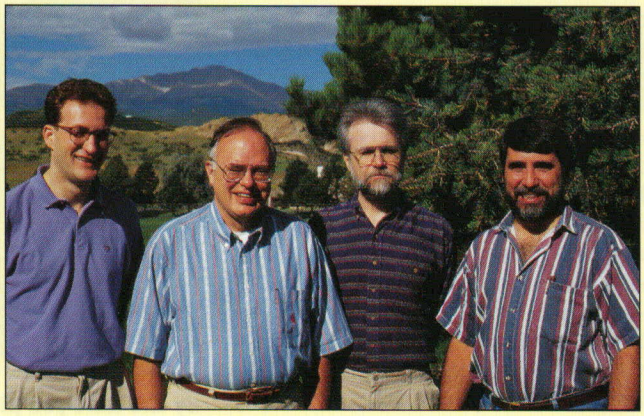
To accomplish the PCI experiments, I needed some expert help, so I called on three key PCI experts that I knew. First, I called longtime friend Ron Sartore. Sartore had been my partner for *EDN*'s All-Star PC project in 1990. He's the foremost expert on PC architecture I know, and he defined the Matchmaker PCI interface chip for Applied Micro Circuits Corp (AMCC). Sartore agreed to join me on the project and arranged for the equipment supplied by AMCC.

Next, I called Tom Shanley, president of MindShare Inc. Shanley spends his life training designers in every aspect of PC design, from the processors, to the buses, to the software. He is also the co-author of the book *PCI System Architecture*, from MindShare's PC System Architecture book series. We used the book extensively during our PCI testing. I met Shanley at PCI Week, and he appeared with me on two satellite broadcasts sponsored by Hewlett-Packard (HP). Shanley had previously used HP's PCI and Pentium test tools to evaluate the performance of a PCI chip set and had presented his findings at the Design SuperCon '95 conference and on the HP satellite broadcasts.

At first, Shanley declined to participate, because he didn't want to become the judge in a PCI beauty contest in which the latest and greatest chips usually win. I convinced him that we were investigating the state of the PCI bus, that we would be uncovering critical performance information for the design community, and that we would in no way be judging a beauty contest. Shanley finally agreed to join the team. When he arrived in Colorado Springs, he brought a huge backpack containing nearly all of the reference materials we needed for the project.

Finally, I called Chuck Small, a logic analyzer project engineer at Hewlett-Packard's Colorado Springs Division. Small had also appeared on the HP satellite broadcasts. He had access to the test tools and knew how to run them. Small agreed to join the team and also arranged for the facilities we needed at HP's Colorado Springs Division.

As it turned out, Small also knew a fourth expert who became an invaluable team member: Thomas Dippon at HP's Boblingen Instruments Division in Germany. Dippon is now a sales development engineer for HP's E2910A PCI Bus Exerciser. He had previously developed code for the product and is an expert in using it. Dippon agreed to join us, completing *EDN*'s PCI Dream Team.



***EDN*'s PCI Project Dream Team (from left to right): Thomas Dippon, Chuck Small, Tom Shanley, and Ron Sartore.**

transactions, like the PCI Bus Exerciser. We could also see what code the processor was executing. The HP 16505 Prototype Analyzer can coordinate the capture of information by both the PCI and Pentium preprocessors and can then time-align the two separate data sets.

The hardware portion of the FuturePlus PCI preprocessor resembles a PCI extender card, because you can plug an expansion card into it. This configuration allows you to study the complete behavior of the expansion card, including the card's bus-request/bus-grant operation. Because each PCI bus slot has its own bus request and grant pins, HP's PCI Bus Exerciser cannot directly observe bus-request/bus-grant transactions of PCI expansion cards plugged into other PCI slots. You can simulate bus-master operations using the PCI Bus Exerciser's own bus-request and bus-grant lines, however.

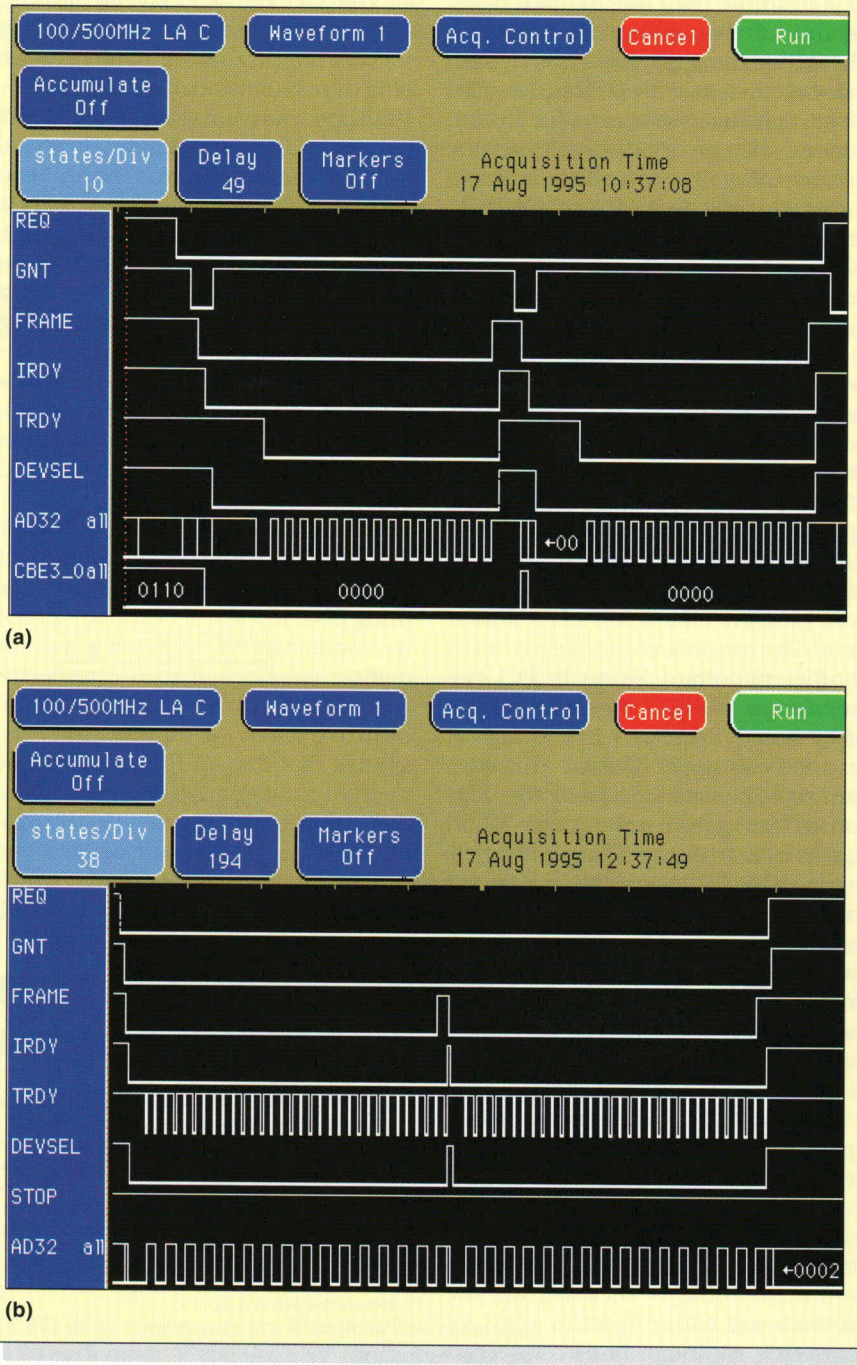
We used a PCI Matchmaker controller card from Applied Micro Circuit Corp's (AMCC's) S5933DK PCI Matchmaker controller developer's kit for many of our experiments. The card can generate continuous PCI bus burst transfers at the maximum transfer rate. The board is a development vehicle for the company's S5933 Matchmaker PCI local bus master/slave-controller IC. In our tests, we used the chip's two 32-byte FIFO buffers to stream data onto and off the bus at maximum PCI transfer rates. AMCC developed the chip for PCI-expansion-card designers who need a simple-to-use interface to the PCI bus.

AMCC also supplied us with a PC motherboard and system based on Intel's latest Triton PCI chip set. This motherboard had no second-level cache RAM but used extended-data-output (EDO) DRAM, which played a large role in our experiments. In addition to the Triton motherboard, we experimented with some earlier PCI-motherboard designs.

Simple tests first

We spent the first morning becoming familiar with the equipment and with each other. Dippon tried some burst-read and -write tests on the Intel Triton motherboard's on-board RAM, using the PCI Bus Exerciser (**Fig 1**). The initial results were exciting. For burst reads, the board injected seven initial PCI bus wait states and then serviced the read requests on every subsequent read cycle (**Fig 2a**). The combination of the EDO DRAM's output buffering and

FIGURE 2



The burst-read (a) and -write (b) displays from the HP 16500 logic analyzer show the long, continuous data-transfer bursts we achieved with the Intel Triton motherboard.

EDN Hands On EXCLUSIVE

the buffering in the Triton chip set allowed this motherboard to run effectively at the PCI bus's theoretical maximum speed during the burst, after the initial latency. Burst-write performance for the Triton motherboard was even better (**Fig 2b**). After a one-bus-clock latency, write cycles took place on every bus clock.

For both burst-read and -write cycles, the chip set forced a disconnect, or PCI-burst termination, after 256 cycles (1024 bytes transferred). We concluded that this behavior occurred when the target address crossed a DRAM page boundary, forcing a new DRAM row-address-strobe (RAS) cycle. Later, we reran these burst experiments with the AMCC Matchmaker development board and achieved the same results. Based on these tests, Shanley remarked, "We've arrived." He meant that the PCI bus had now achieved its performance potential, at least for data-transfer rates.

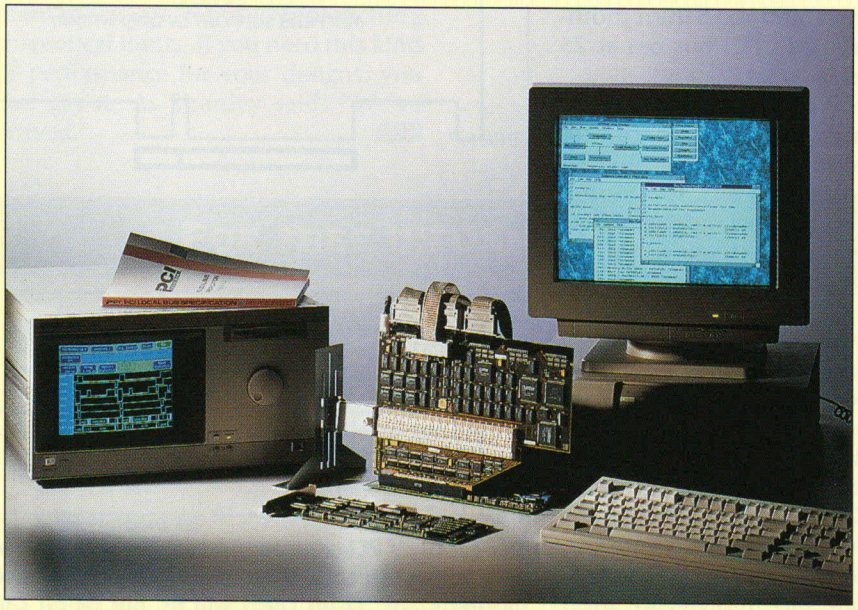
Although we weren't specifically concerned with the performance of other chip-set features, we did look at the Triton's ISA-bus operation out of curiosity. Shanley, in particular, was interested in this performance. He is eagerly awaiting the 14-year-old ISA bus's demise, even though his company still teaches ISA-design classes. When accessing an ISA slot, the Triton motherboard injected at least 29 PCI-clock wait states. Although this behavior conforms to the PCI bus 2.0 specification, long delays can unnecessarily tie up the bus. In fact, the PCI Version 2.1 specification states that any bus access that requires more than 16 PCI clock cycles should be temporarily suspended by a retry response from the PCI bus target. The PCI bus initiator then reruns the access cycle later, when it will presumably complete more quickly. This split bus transaction for slow bus targets improves PCI bus performance in multimaster systems.

For reference, we also tried these same transfer-rate experiments on a

EDN's PCI Project Toolbox

We used a lot of equipment in our PCI experiments. These instruments allowed us to get answers fast, usually within a few minutes of thinking up the question. Here's what we used:

PCI Bus exerciser system	
HP 16500B logic-analysis system mainframe	$8980
HP 16550A 100-MHz state/500-MHz timing logic-analysis card (two cards were used)	$9440 ea
HP E2910A PCI Bus Exerciser	$40,950
HP E2911A 5V PCI adapter	$1950
Pentium/PCI trace system	
HP 16500B logic-analysis system mainframe	$8980
HP 16550A 1M-sample, 100-MHz state/500-MHz timing logic-analysis card (three cards were used to trace Pentium operations)	$13,765 ea
HP 16550A 100-MHz state/500-MHz timing logic-analysis card	$9440
HP E2457A Pentium (P54C) preprocessor	$5100
HP E3491A Pentium (P54C) processor probe (run control probe)	$7500
FuturePlus Systems FS16P32E 32-bit PCI preprocessor and extender card	$2000
PCI stimulus card	
Applied Micro Circuits Corp S5933DK PCI Matchmaker controller developer's kit	less than $500



The PCI Bus Exerciser system was our primary tool during *EDN*'s PCI Project experiments.

much older motherboard based on Intel's Saturn chip set. This motherboard used conventional DRAM for main memory and incorporated second-level cache RAM. For a burst read, the board initially injected 14 wait states, followed by data transfers every three PCI bus clock cycles. For burst writes, the board initially injected 20 wait states, followed by a data transfer every four cycles. This test demonstrates how early PCI bus implementations hurt PCI's reputation as a high-performance bus.

A brief digression on μPs

I want to make one thing clear at this point. We performed all of our transfer-rate experiments independent of the processor used on the motherboard. The Triton motherboard had a 100-MHz Pentium μP, and the Saturn motherboard had a 25-MHz 80486SX μP. But, the relative execution speeds of these processors did not play a factor in our results. However, the μP's bus speeds were a factor. Intel's PCI chip sets run the PCI bus synchronously with the processor bus, so the Triton motherboard's PCI clock ran at 33 MHz, and the Saturn motherboard's PCI bus ran at 25 MHz.

We also encountered a surprise that you should know about. Early motherboard chip sets, like Saturn, had a limited number of bus-request/bus-grant pins. The Saturn chip set has four bus-request/bus-grant pin pairs. Two of these pairs are used on the motherboard for on-board peripherals, and two pairs are used for the PCI slots. The motherboard has three PCI slots, however. Consequently, one of the PCI slots on this motherboard couldn't accommodate bus masters. By sheer chance, we first plugged the PCI Bus Exerciser into this PCI slot, and none of our tests would run. When we finally realized what was causing the problem and moved the PCI Bus Exerciser to another slot, the test ran fine.

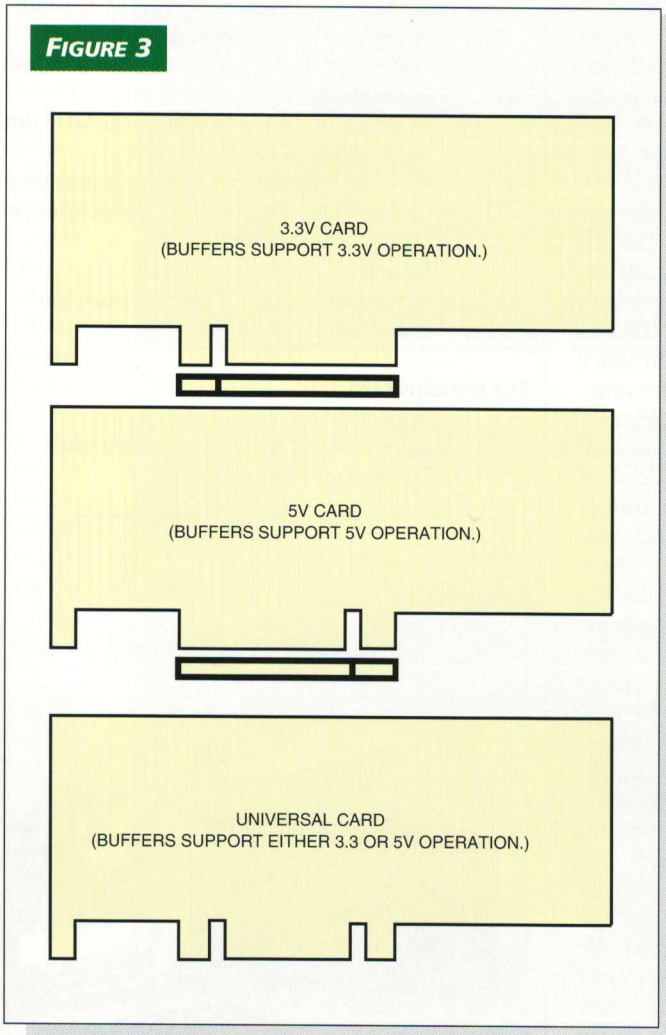
Over the course of the four days, *EDN*'s PCI team ran performance tests on the two Intel motherboards and on two other motherboards. We didn't observe transfer rates better than those achieved on the Triton motherboard, and we did not expect to see better rates. We did, however, observe many interesting results that could help you achieve better performance in your designs.

On one motherboard, we observed a request/grant latency of a few PCI bus clocks. We concluded that the latency was caused by the need to arbitrate between the processor and the PCI bus master for ownership of the bus. We observed this behavior even after we halted the processor. A halted processor should not be vying for bus ownership, but the request/grant latency remained.

A look at the PCI-chip-set specs for that motherboard revealed that the chip set's bus arbiter could be programmed to park the bus ownership on the processor. The BIOS on this motherboard programmed the PCI chip set in that configuration. Although this scheme makes sense in a PC environment in which the processor will likely be the most-active user of the PCI bus, it slowed performance in our tests. By reprogramming the PCI chip set to park bus ownership on the current bus master instead of the processor, we entirely eliminated the bus-request/bus-grant latency and achieved better performance. Consequently, if your application will make extensive use of PCI bus masters, other than the host μP, you may want to be able to program the bus arbiter's priority scheme.

On one motherboard, we observed a strange event. When the processor wrote to the PCI-based video-controller card, the processor only incurred two bus clock wait states. However, when we wrote to this same card using the PCI Bus Exerciser, we observed six wait states. It took us a while to determine the cause of this behavior. The processor

FIGURE 3



The 3 and 5V PCI edge connectors are identical, but reversed, with respect to each other. If you use a 3V connector on a 5V motherboard, 32-bit cards will fit, but 64-bit cards will not. (Reprinted from *PCI System Architecture*, Third Edition, Addison-Wesley Publishing Co, Reading, MA, 1995.)

wrote to the video card using 16-bit transfers, and the PCI Bus Exerciser used 32-bit transfers. The video card, recognizing the 32-bit transfers, was accepting the 32-bit data word and breaking it into 16-bit chunks.

The counterintuitive aspect to this observation is that two 16-bit transfers took less time than one 32-bit transfer because of the additional wait states. The 32-bit transfers should have been twice as efficient as 16-bit transfers. The 32-bit transfers were actually slower, because they incurred three times the number of wait cycles. This example shows why you need good tools to help achieve optimum performance with PCI-based systems. The tools help you uncover many surprising and counterintuitive aspects of your design.

Most PCI chip sets are flexible because of their programmability. This flexibility allows you to optimize the bus's operation for some, but not all, situations. You'll want to be sure that the chip set you use is programmed optimally for your application.

One motherboard gave us a real surprise. The PCI Bus Exerciser wouldn't even plug into the board's PCI slots at first. The reason provided us with another important lesson in PCI system design. The PCI Bus Exerciser is a 64-bit board. It has an extra connector for the bus's upper 32 bits. This edge connector is separated from the edge connectors carrying the lower 32-bit signals by a small gap. The plastic housing on the PCI bus edge-connector sockets soldered to this particular motherboard was too thick in one strategic spot. Shanley immediately understood the problem. The connectors used on this motherboard were designed for a 3V PCI bus but were being used on a 5V motherboard.

At first, I had trouble connecting the operating voltage with the physical design of the connector. Shanley showed me an illustration from his book that immediately cleared everything up for me. (See **Fig 3.**) A 3V PCI connector mounts reversed on the motherboard with respect to a PCI 5V connector. Although both connectors carry the same signals on the same pins, pin 1 on the 5V connector and pin 1 on the 3V connector are at opposite ends. This physical reversal prevents you from plugging a 5V card into a 3V PCI socket, or vice versa.

A notch on the end of the connector socket's housing allows 64-bit boards to plug into 32-bit slots. The notch on the problem motherboard was at the wrong end, however. On the other motherboards we tested, the PCI connectors were either mounted correctly or had notches molded into both ends, so that they could be soldered into either 3 or 5V positions. The lesson to learn here is that not all PCI-socket connectors are the same, and you must be sure to specify the right connectors for your design. You should also make sure that your purchasing people know that they can't substitute just any PCI connector for the part you specify.

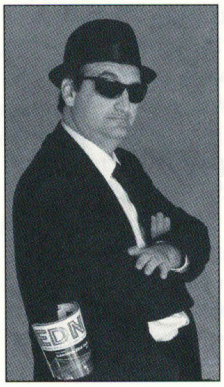
We learned quite a lot about the specific PCI characteristics of the products we examined. In fact, we learned something from just about every experiment we conducted. We fed back the information we gleaned to the respective manufacturers, so that they could improve their products. We also learned that state-of-the-art PCI implementations are rapidly approaching theoretical limits. If you need this kind of performance for your designs, you can get it. As Shanley said, "We've arrived." EDN

Reference

1. Shanley, Tom and Don Anderson, *PCI System Architecture*, Third Edition, Addison-Wesley Publishing Co, Reading, MA, 1995.

Acknowledgments

My thanks to Ron Sartore, Tom Shanley, Chuck Small, and Thomas Dippon for their invaluable help on this project. Without them and their rare mix of skills, there would have been neither a project nor that great string of lunches at Mexican restaurants in Colorado Springs. I'd also like to thank Applied Micro Circuits Corp, FuturePlus Systems Corp, Hewlett-Packard Co, and MindShare Inc for supporting this project through the loan of equipment and personnel.



You can reach Editor In Chief Steven Leibson at (617) 558-4214 or e-mail ednleibson@mcimail.com

Reprinted from
EDN October 26, 1995

FOR FREE INFORMATION...

For free information on the PCI test gear and training discussed in this article, circle the appropriate numbers on the postage-paid Information Retrieval Service card or use *EDN*'s Express Request service. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Applied Micro Circuits Corp
San Diego, CA
(619) 450-9333

FuturePlus Systems Corp
Colorado Springs, CO
(719) 380-7321

MindShare Inc
Richardson, TX
(214) 231-2216

Hewlett-Packard Co
Palo Alto, CA
Phone local office