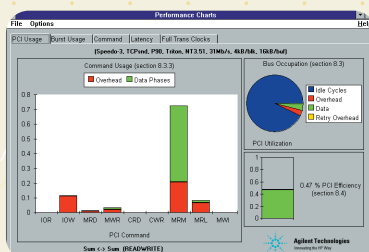
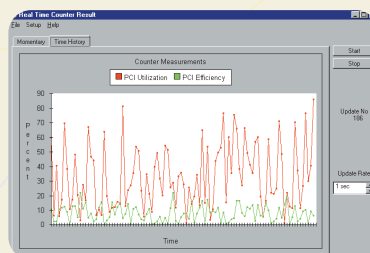
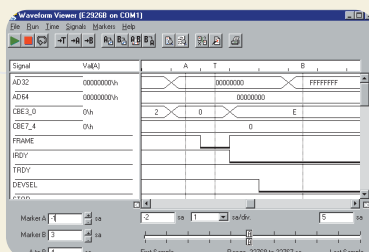
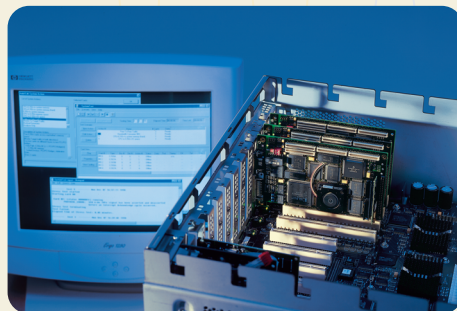


Efficient Use of PCI

Intel discusses basic concepts of PCI Performance and efficient use of PCI



Agilent Technologies
Innovating the HP Way

Introduction

As a designer of PCI cards or systems, you need the ability to measure the performance of a system or the components of a system you've built, because performance is the most important characteristic of a computer system. Efficient utilization of the PCI bus in designs can significantly improve your system performance overall.

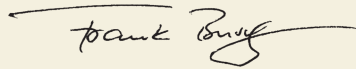
For example:

- a single weak spot within a system can degrade the performance of the entire system
- the I/O speed can be the bottle neck in the present system (not the processor speed)

To achieve superior performance, PCI design knowledge and a tool to measure and understand system PCI efficiency is required.

Frank Hady from Intel's Computing Enhancement Group, Platform Architecture Labs and Agilent Technologies Boeblingen Verification Solutions Operation have worked together to improve the PC, I/O card and server design process by providing superior technology, design knowledge and verification tools.

In a cooperative relationship, Intel worked with Agilent to enhance Agilent's PCI Series of Computer Verification Tools with new features based on PCI performance analysis techniques, concepts and metrics developed by Intel.



Frank Busch
Introduction Project Manager,
Computer Verification Tools
Agilent Technologies



Efficient Use of PCI

Frank Hady,

Platform Architecture Labs, Intel Corporation

Sharply higher system performance can often be achieved by selecting only PCI peripherals which make efficient use of the PCI bus.

Figure 1 shows network throughput for a simple benchmark, TTCP, for five different PCI Network Interface Cards (NICs). Since the system is held constant, the large differences in the observed performance are a direct result of the PCI card design.

Knowledge of huge performance differences such as these should lead system integrators to select PCI cards carefully. These differences should also motivate designers to follow the rules for efficient PCI design. This paper is designed to sensitize systems integrators and PCI card designers to the importance of proper PCI design and to present rules for efficient PCI design.

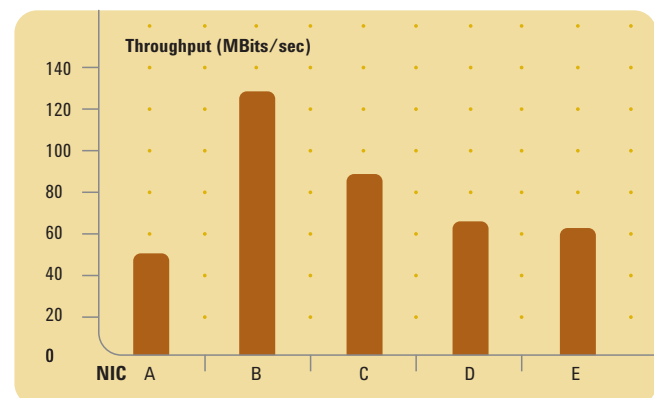


Figure 1: Network Throughput for PCI NICs

There are a small set of rules that should be followed to insure an efficient PCI design. A study of leading commercial NICs shows that these rules are often not followed. This paper begins by presenting the rules of efficient PCI design. The metrics and charts used to describe PCI performance in detail are defined next. The final section provides a detailed description of each rule along with an example of performance pitfalls that can be avoided by following each rule.

Rules for Efficient Design

As was shown in the introduction, PCI card choice is of paramount importance in determining system performance. Much of the observed performance difference can be attributed to differences in PCI implementation. There are a number of rules to follow when making an efficient, high performance, PCI design. These rules are presented first below, and again in the final section with additional discussion.

1 Implement advanced PCI commands.

- Use Memory Read Line (MRL) reads of data fitting within 2 consecutive cache lines but not within one cache line (for which MR would be used). On Pentium® Processors and Pentium® Pro Processors a cache line is 32 bytes (8 Dwords).
- Use Memory Read Multiple (MRM) for reads of data spanning more than 2 cache lines.
- Use Memory Write Invalidate (MWI) for multiple cache line writes (must be aligned). Do not disconnect a long Memory Write (MW) commands in order to start a MWI command.

2 Use long bursts.

- For reads, bursts of at least 64 Dwords (32 bit words) are needed for good performance on some platforms.
- For writes, bursts should be at least as big as a cache line.

3 Use memory commands, not I/O commands.

4 Minimize Latency. Respond to slave accesses as rapidly as possible and try to avoid inserting wait states within block transfers.

5 Follow the rules, not experiments.

While current systems may provide good performance even when the rules are not followed, future systems may not.

All the rules listed here are useful for the PCI card designer. Following these rules results in a PCI design which should work well across all chipsets (the chips connecting the processor, memory and PCI). Systems integrators may also use the rules as a basis for questions for card vendors when choosing a peripheral. While the PCI implementation is only one of the factors a systems integrator must consider, the PCI card enabling best system performance is often the one that most closely follows these rules. As will be shown in the final section, large deviation from these rules can easily result in poor system performance.

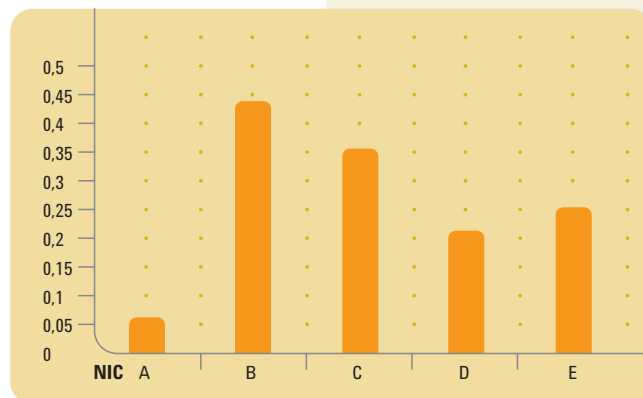


Figure 2: PCI Efficiency for PCI NICs

Figure 2 shows the PCI efficiency for the same set of NICs whose throughput was presented earlier. As shown, the PCI efficiency for NICs available today varies greatly. While PCI efficiency is only one factor in determining NIC throughput, the strong correlation between throughput (Figure 1) and PCI efficiency (Figure 2) suggests it is an important one.

Evaluating PCI Performance

The evaluation of PCI performance requires two types of information. The first is a set of PCI performance metrics which can be used to compare the performance of different systems. The second is a detailed description of PCI performance useful for understanding the underlying causes of the performance observed. This detailed description may be conveniently displayed as a set of PCI performance charts.

PCI Performance Metrics Bus utilization and throughput are two useful metrics for describing PCI performance. This section defines these two common metrics along with a third, PCI efficiency.

- **PCI Utilization** - PCI Bus Utilization is the number of clocks during which the PCI bus is used, divided by the total number of PCI clocks. A PCI clock is used if either Frame#, Irdy# or Trdy# is true.
- **PCI Throughput** - The standard definition for throughput is the amount of data passed per unit time.

This metric can be divided into two parts, $\text{Thrpt}_{\text{data}}$ - the amount of application data passed per unit time and $\text{Thrpt}_{\text{control}}$ - the amount of control overhead needed to attain the $\text{Thrpt}_{\text{data}}$ observed. In this paper the term throughput generally refers to the throughput seen by the application, $\text{Thrpt}_{\text{data}}$.

- **PCI Efficiency** - Describes how well a PCI card maximizes $\text{Thrpt}_{\text{data}}$ while minimizing PCI Utilization. A PCI efficiency approaching 1 describes a PCI bus in which few clocks are wasted. Address cycles, control data cycles, target wait cycles, and initiator wait cycles all contribute to lowering PCI efficiency toward 0. PCI Efficiency equals $\text{Thrpt}_{\text{data}}$, stated in PCI sized words, divided by PCI Utilization. Unlike the other two metrics, PCI Efficiency is independent of the percentage of the PCI bus used. This makes comparison of the PCI implementation of two cards possible even when the performance of the benchmark being used to generate PCI traffic is limited by the non-PCI portion of the system. This metric is also key in understanding system scaling (i.e. how well will a system perform when more cards are added).

PCI Performance Charts

The three PCI performance metrics are useful for gross comparisons of performance, but are not useful for explaining the root causes of the observed performance. The series of three charts described below accomplishes this purpose.

PCI Usage

The PCI usage chart, Figure 3, is intended to give a rapid overview of PCI performance. The pie charts on the right side of Figure 3 describe the observed performance at the highest level of abstraction. The top chart, PCI Utilization, describes how the total set of PCI bus cycles are split between three different conditions.

Yellow represents unused or idle cycles, the dark red portion represents cycles required to move data, and the orange area represents overhead cycles (address cycles, wait cycles, and even the fraction of a data cycle carrying less than 4 bytes).

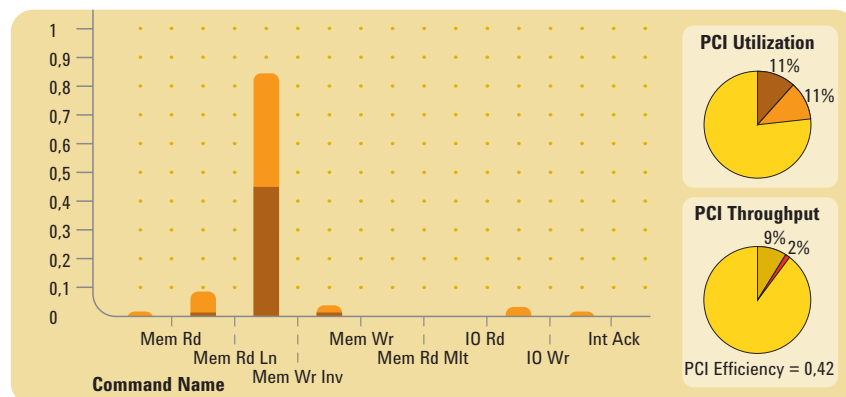


Figure 3: PCI Usage Charts

The lower pie chart represents the total possible PCI throughput. This chart shows the split between $\text{Thrpt}_{\text{data}}$, the dark yellow region, $\text{Thrpt}_{\text{control}}$, the red region and unrealized PCI throughput, the yellow region. PCI efficiency equals the dark yellow region (9%) divided by the sum of the dark red (11%) and orange regions (11%) on the PCI utilization chart.

The histogram on the left of Figure 3 shows the relative activity and efficiency of each PCI command. Only the PCI clocks during which the bus was active are represented. The height of each bar represents the fraction of time the PCI bus is held (not idle) by the named command. Orange and dark red are used to denote overhead clocks and data clocks, just as in the PCI Utilization chart, on a per command basis.

PCI Burst Usage

The PCI Burst Usage Chart in Figure 4 allows analysis of PCI activity by burst size. Burst size is shown in Dwords. The orange line represents the fraction of data moved by transfers of each size, the dark red line corresponds to the fraction of non idle PCI clocks used by transfers of each size.

The difference between the orange and dark red lines gives the PCI efficiency of transfers of a given burst size, plotted with the red line. An efficiency of 1 is defined as a transfer which passes 4 bytes of data on every clock during which the bus is occupied. Since every cycle requires at least one address cycle, an efficiency of 1 can only be approached asymptotically, not reached. While an efficiency value is plotted for each burst size, the efficiency data is really only accurate when there are a significant number of transfers

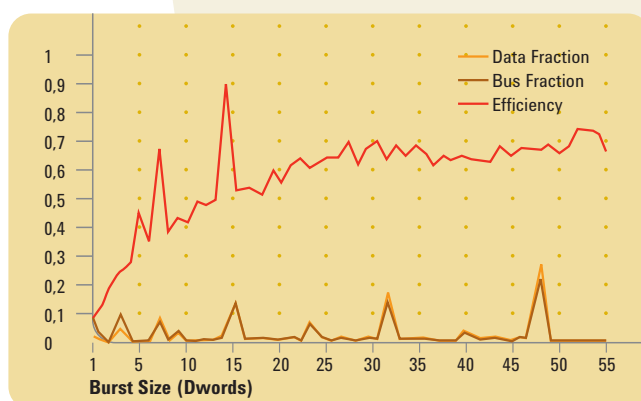


Figure 4: PCI Burst Usage

PCI Command Usage

The final chart shown in Figure 5, plots the frequency of command usage by both command name and burst size, allowing the mix of commands and burst sizes to be analyzed. This chart allows one to determine how appropriately commands are used. For example notice that the card pictured in Figure 4 uses both Memory Read Multiple and Memory Read Line for 4 Dword transfers and uses both Memory Read Line and Memory Read for 1 Dword transfers.

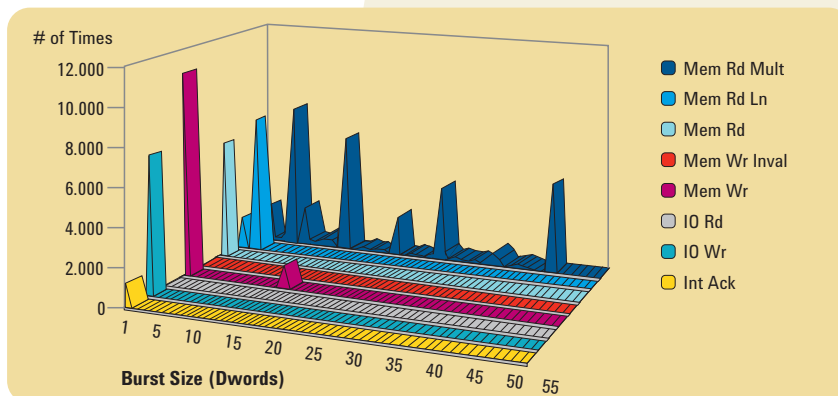


Figure 5: PCI Command Usage

Following the Rules Improves Performance

With the metrics and charts defined, the stage is set to investigate the effect of each rule on PCI performance. Each rule will be described in greater detail. For many of the rules the PCI Efficiency improvements that may be attained by following the rule will be explored. Follow the Rules, Not Experiments. In an attempt to attain the maximum performance with the minimum cost, architects of PCI cards often perform experiments to determine the necessary set of performance features. In the past, this approach has led architects to conclude that some of the PCI rules were not important to performance. Cards designed in this manner often work extremely well for the platforms on which the tests took place; however, when new platforms arrive, these same cards perform poorly. By ignoring the rules, the cards useful life span is greatly shortened.



Follow the Rules, Not Experiments

Figure 6 provides an example of why designing to experiments rather than rules is wrong. The figure represents the PCI command usage for a given NIC on two different platforms running the same benchmark. Bursts on Chipset B are limited to only 8 Dwords, while Chipset A allows for the full burst size of the NIC tested, almost 60 Dwords. An architect optimizing for Chipset B may decide to ignore the rules and save money by limiting all burst lengths to a maximum of 8 Dwords. When a platform featuring Chipset A arrives, a NIC designed following the rules will transfer data with long bursts, like the NIC pictured, greatly increasing efficiency. The NIC designed for Chipset B will continue to use only 8 byte bursts and see no increase in efficiency.

This same scenario may occur for all the rules listed in this paper. Chipset designers implement performance optimizations following as many of the PCI rules as they need. The only way PCI card designers can make sure that their card will perform well on future platforms is to follow the PCI rules.

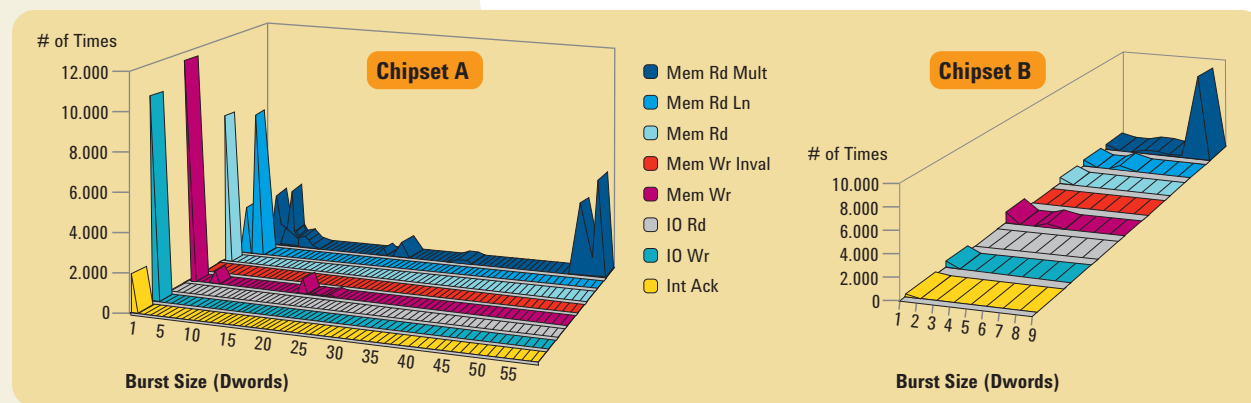


Figure 6: Chipset can limit burst length

Use long bursts

Each time a PCI card initiates a transfer of X Dwords it must wait for L clocks for the transfer to begin. Once the transfer begins, it can often occur at a rate of one Dword per PCI clock, making the efficiency of the transfer $X/(X+L)$. Since the PCI card cannot completely control L (the platform determines much of this value) it must maximize the burst size, X, to achieve highly efficient transfers. For read commands PCI cards should be capable of bursts of at least 64 Dwords to achieve good PCI efficiency. For write commands bursts of at least one cache line should be used. While these burst sizes are sufficient to provide good PCI efficiency across the range of platforms, designers may decide if the improved PCI efficiency even further by using longer bursts.

Figure 7 shows PCI Efficiency as a function of average burst length for a series of experiments conducted with a variety of NICs, platforms, operating systems, and networking benchmarks which stress PCI read performance.

Average burst length is calculated for all the transfers seen on the PCI bus. This includes the data transfers, for which the NIC should use long transfers, NIC control transfers, which may need to be considerably shorter, and higher level protocol required transfers (like TCP/IP acknowledges), whose length the NIC has no control over. The graph shows that while PCI Efficiency is not dependent only on burst length, long bursts are necessary for high PCI efficiency.

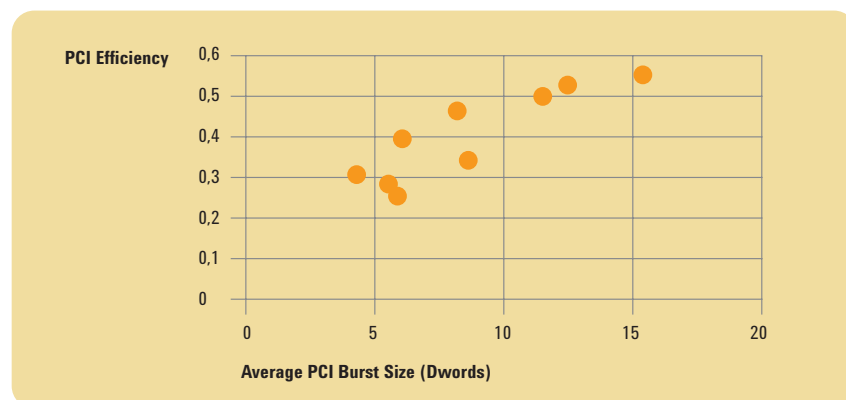


Figure 7: PCI Efficiency is strongly effected by burst length

Implement Advanced PCI Commands

Some of the more popular NICs implement only MR and MW. This often limits the cards to short burst lengths, creating poor performance. The performance pitfalls to be avoided by implementing advanced PCI commands are described below, first for PCI reads and the for PCI for writes.

Advanced Read Commands

Using the right advanced command when performing a PCI master memory reads from prefetchable regions in main memory may markedly increase performance. Chipsets often use the type of PCI read command as a hint to control the block size of data read from main memory and to determine if data should be prefetched. When data is read in small blocks and not prefetched, as it may be for a MR command, the chipset may either need to insert wait states between successive Dwords transferred or may be unable to sustain long bursts.

The MR command should be used for reading data that fits within one cache line. MRL should be used for reading data that resides across two cache lines. MRM should be used for reading data that is held on more than 2 cache lines. This recommendation varies slightly from the recommended usage in the 2.1 PCI specification, recognizing that once a single Dword has been read from DRAM reading to the end of the cache line is extremely inexpensive. The differences are slight and following either recommended usage will produce very good performance.

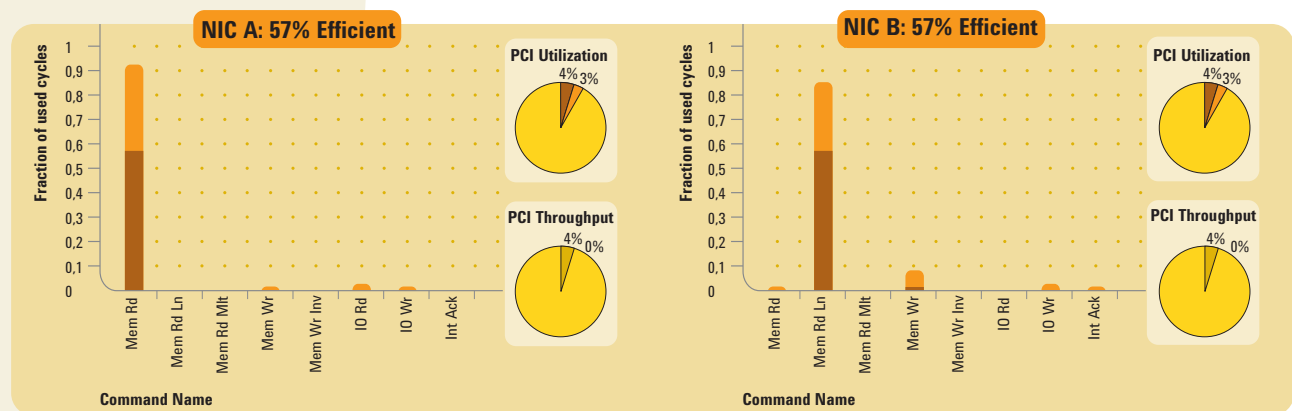


Figure 8: Read performance on a forgiving chipset (chipset A)

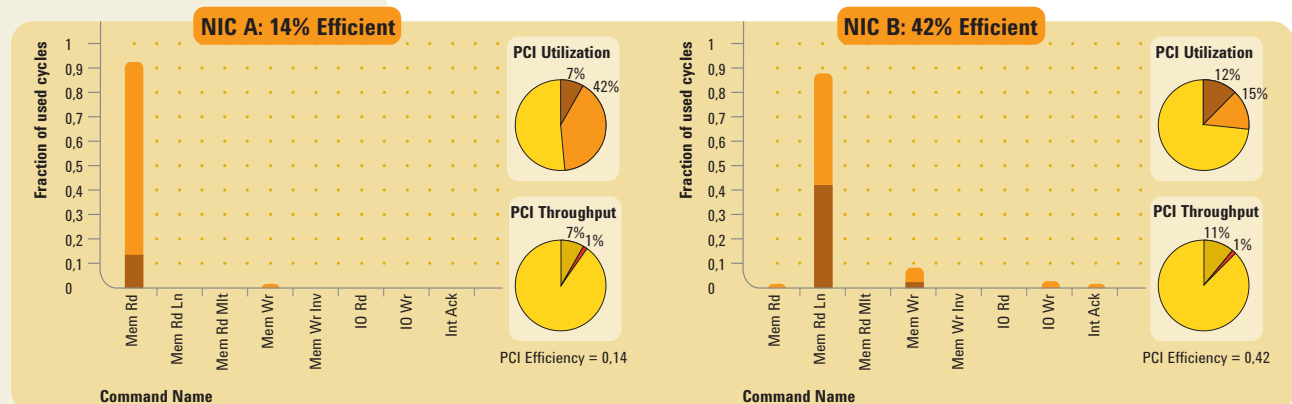


Figure 9: Read performance on a non-forgiving chipset (chipset B)



The two NICs with the widest performance difference in Figure 1 and Figure 2, NIC A and NIC B, are used below to demonstrate the importance of using advanced read commands. As shown in Figure 8, NIC B implements MRL but not MRM. The lack of MRM commands happened not to affect NIC B's performance here, but may well degrade it's performance elsewhere. NIC A uses only MR. The performance of these two NICs is displayed for both a forgiving chipset not requiring advanced commands or very long bursts, Chipset A, and for an unforgiving chipset, Chipset B.

Figure 8 shows that on a chipset which does not optimize based on PCI command and is not sensitive to burst length, the two NICs perform very similarly. On a chipset whose implementers optimized performance based on PCI command, Figure 9, the use of only MR by NIC A has caused it's performance to drop to an unacceptable level, 14% PCI Efficiency. NIC B has dropped only to 42% PCI Efficiency.

Figure 10 shows the reason for the huge performance drop for NIC A. On Chipset A, almost all data was moved with 16 Dword transfers. On Chipset B, burst size was reduced greatly with significant transfer fractions at only 8, 4 and 1 Dword bursts. As shown in Figure 11, the drop in burst length for NIC A need not have occurred. Using an advanced read command NIC B was able to maintain and even increase its average burst length, holding its PCI efficiency to a respectable 42%.

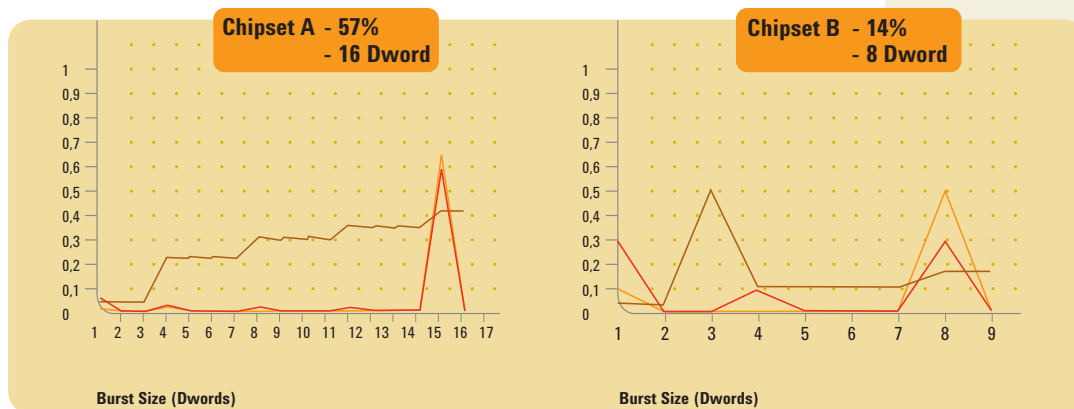


Figure 10: NIC A performance across chipsets

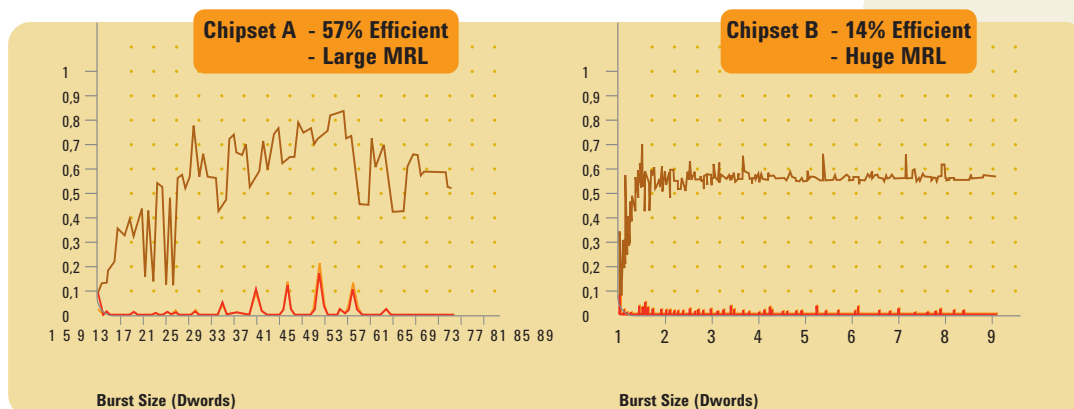


Figure 11: NIC B performance across chipsets

Advanced Write Commands

Use of the advanced write command MWI is also important for achieving high performance across all chipsets. MWI should be used for transferring blocks of data greater than one cache line in size which begin and end on cache line boundaries. PCI cards that are forced to start MW commands rather than MWI commands because a block is not cache line aligned, should not disconnect the MW in order to start a MWI. If the chipset is optimized for MWI it should disconnect the MW transfer on a cache line boundary, providing a convenient place for the NIC to begin a MWI.

A case study outlining the importance of using MWI follows. The same two cards used for the read command case study are used here. NIC A implements only MW while NIC B implements both MW and MWI. Figure 12 shows the performance of two NICs on a platform featuring a forgiving chipset (Chipset A) which makes no distinction between MW and MWI. The performance of these same two NICs for a chipset optimized for MWI is shown in Figure 13.

The change to Chipset B has caused a drop in the PCI efficiency of both NICs, with NIC A able to sustain only a 36% efficiency. NIC A moved most of its data with 10 Dword MW bursts on chipset A, but was allowed only 4 Dword MW bursts on chipset B. NIC B used MWI and so was able to retain longer bursts of 16, 24 and 32 Dwords on chipset B, allowing it to hold a 53% PCI efficiency. Using the advanced PCI commands MRL, MRM and MWI is crucial for achieving a high performance across all PCI platforms.

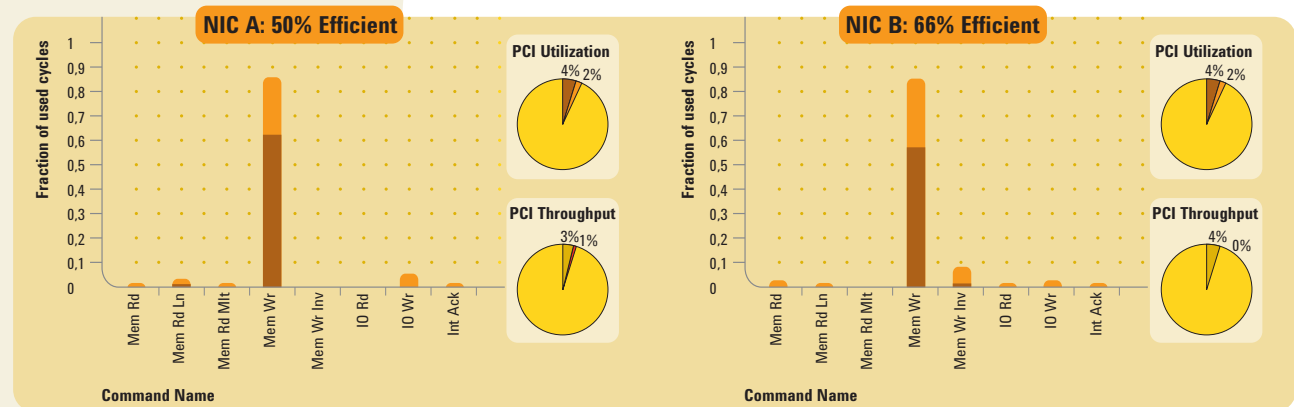


Figure 12: Write command PCI Performance (chipset A)

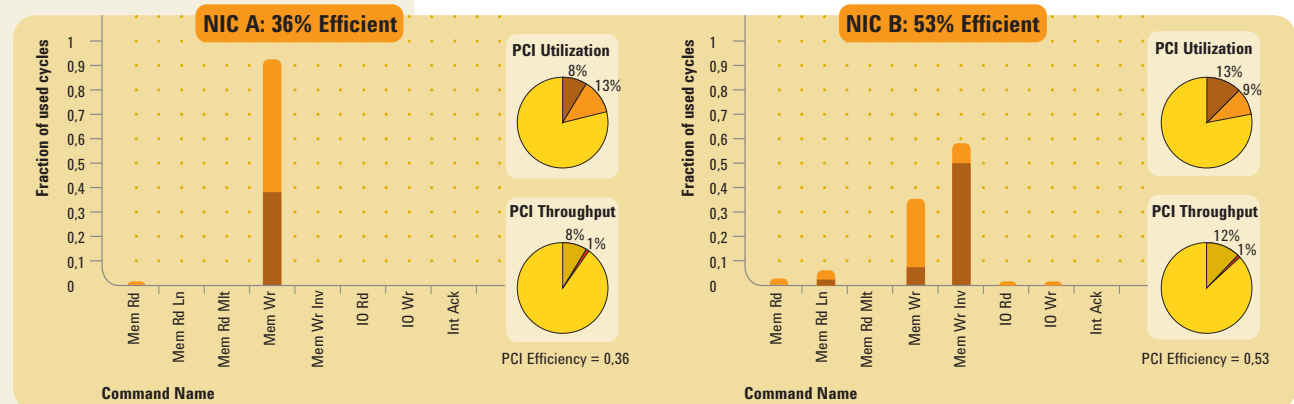


Figure 13: Write command PCI Performance (chipset B)

Use memory commands, not I/O commands

I/O reads and writes are generally issued by the CPU to address I/O space on PCI cards. I/O commands often serialize operations within chipsets and within the CPU. This serialization may not show up as a decrease in PCI efficiency, but will degrade performance by increasing CPU utilization.

PCI card designers can minimize these effects by using I/O space for only those locations which require system serialization. All other locations should be placed in PCI memory space.

Minimize Latency

There are a number of opportunities within a PCI transfer for PCI cards to add clocks. All of these additional clocks are referred to here by the name latency. Latency may be added when the card is responding to a read by another PCI master by taking many clocks to respond with the requested data. Latency may be added when initiating a transfer if the card has a long delay from the falling edge of Frame# until IrDY# is pulled true. Latency may even be added between the individual Dword transfers if the PCI card deasserts its ready line. All of these latencies will contribute to a drop in PCI efficiency according to the equation presented in the burst length section, $L / (X+L)$ where L is all of the card latencies and the chipset latency summed and X is the burst length. Minimizing the latencies on the PCI card optimizes PCI efficiency.

Summary

Delivering the latest in computing technology to an increasingly demanding and competitive market place is no easy task. Whether the engineer is developing a core logic chipset, a peripheral chip or the latest desktop motherboard, PC server or add-in SCSI card, the window of opportunity for the product is getting narrower and its lifetime shorter, while customers' expectations for reliability and performance are increasing.

The Agilent PCI Performance Optimizer is the result of close cooperation between Agilent Technologies Boeblingen Verification Solutions Operation (BVS) and Intel Corporation's Platform Architecture Lab and supports customers' needs to evaluate and optimize system performance.

The Agilent PCI Performance Optimizer is a comprehensive GUI-based development environment that measures and displays the overall performance of a system or single device, identifies PCI bus bottlenecks and provides an intuitive performance comparison of boards or different PCI buses within one system.

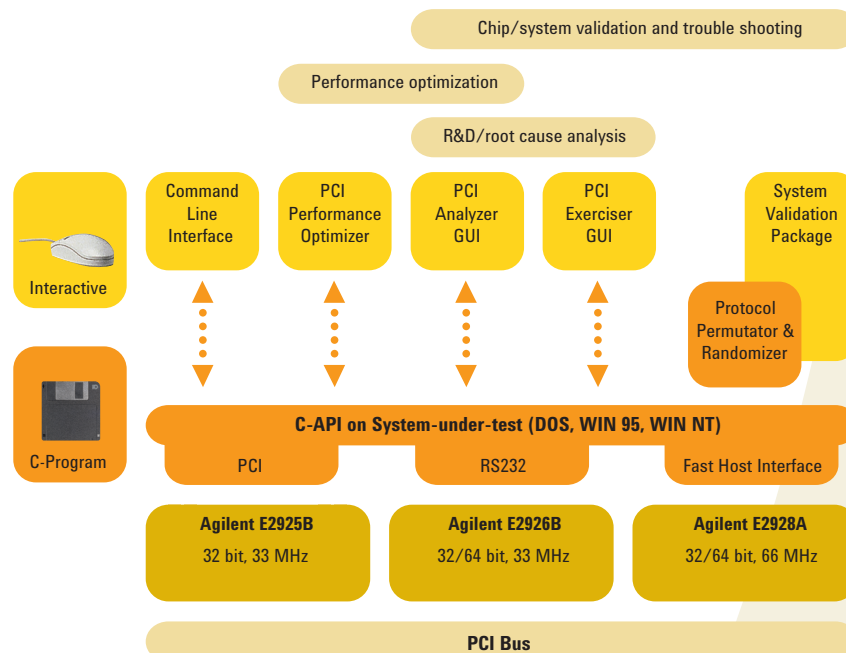
The Agilent PCI Performance Optimizer features real-time monitoring, automated counter- and trace-memory measurement, split-transaction-performance analysis and data-display export capabilities to link with database applications.

The Agilent PCI Performance Optimizer is supported by all the products of the Agilent E2920 series PCI Exercisers and Analyzers and can be added by selecting option #200 (e.g. E2928A + option #200).

Conclusion

Proper use of the PCI bus will maximize system performance while minimizing the load on the PCI bus and the CPU. PCI bus efficiency is strongly enhanced by following the rules presented here.

PCI card designers should create designs with this list in mind. System integrators should choose cards to include in their systems based in part on the system's compliance to the rules list.



This article has been reproduced by Agilent Technologies GmbH with the kind permission of the © Intel Corporation.

Agilent Technologies' Test and Measurement Support, Services, and Assistance

Agilent Technologies aims to maximize the value you receive, while minimizing your risk and problems. We strive to ensure that you get the test and measurement capabilities you paid for and obtain the support you need. Our extensive support resources and services can help you choose the right Agilent products for your applications and apply them successfully. Every instrument and system we sell has a global warranty. Support is available for at least five years beyond the production life of the product. Two concepts underlay Agilent's overall support policy: "Our Promise" and "Your Advantage."

Our Promise

Our Promise means your Agilent test and measurement equipment will meet its advertised performance and functionality. When you are choosing new equipment, we will help you with product information, including realistic performance specifications and practical recommendations from experienced test engineers. When you use Agilent equipment, we can verify that it works properly, help with product operation, and provide basic measurement assistance for the use of specified capabilities, at no extra cost upon request. Many self-help tools are available.

Your Advantage

Your Advantage means that Agilent offers a wide range of additional expert test and measurement services, which you can purchase according to your unique technical and business needs. Solve problems efficiently and gain a competitive edge by contracting with us for calibration, extra-cost upgrades, out-of-warranty repairs, and on-site education and training, as well as design, system integration, project management, and other professional services. Experienced Agilent engineers and technicians worldwide can help you maximize your productivity, optimize the return on investment of your Agilent instruments and systems, and obtain dependable measurement accuracy for the life of those products.

Related Agilent Literature

- Agilent E2925B 32bit, 33 MHz, Agilent E2926B 32/64bit, 33 MHz PCI Exerciser & Analyzer, technical overview, p/n 5968-3501E
- Agilent E2928A 32/64bit, 66 MHz, PCI Exerciser & Analyzer, technical overview, p/n 5968-3506E
- Agilent E2940A Compact PCI Exerciser & Analyzer, technical overview, P/n 5968-1915E
- Agilent E2922A PCI-X Master Target Card, technical overview, p/n 5968-9577E
- Agilent E2929A PCI-X Exerciser & Analyzer, technical overview, p/n 5968-8984E
- Agilent E2976A System Validation Pack, Agilent E2977A System Test Library, technical overview, p/n 5968-3500E
- Agilent E2920 Computer Verification Tools, PCI Series, brochure, p/n 5968-9694E
- HP NSD stabilizes server designs quickly and completely with the Agilent E2920 PCI Series, case study, p/n 5968-6948E
- HP HSTC speeds high-end server testing and reduces engineering costs with the Agilent E2920 PCI Series, case study, p/n 5968-6949E
- Agilent E2920 Verification Tools, PCI Series gives Altera Corporation competitive Advantage, case study, p/n 5968-4191E

You can find the current literature and software at:
www.agilent.com/find/pci_products

**By internet, phone, or fax,
get assistance with all your test
and measurement needs**

Online assistance:
www.agilent.com/find/assist

Phone or Fax
United States:
(tel) 1 800 452 4844

Canada:
(tel) 1 877 894 4414
(fax) (905) 206 4120

Europe:
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Japan:
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Latin America:
(tel) (305) 269 7500
(fax) (305) 269 7599

Australia:
(tel) 1 800 629 485
(fax) (61 3) 9210 5947

New Zealand:
(tel) 0 800 738 378
(fax) 64 4 495 8950

Asia Pacific:
(tel) (852) 3197 7777
(fax) (852) 2506 9284

Product specifications and descriptions in this document subject to change without notice.

Copyright © 2000 Agilent Technologies
Printed in Germany (J+R) 10/2000
5988-0448ENDE

For more information, please visit us at:
www.agilent.com/find/pci_overview