intel®

Techniques for Increasing PCI Performance

Application Note AP- 666

February 1999

Contents

Revision History

Rev.	Draft/Changes	Date
-001	• Initial Release	February 1999

Application Note/AP- 666

intel

Preface

These techniques should be used in implementing PCI devices that require large PCI bandwidth on Intel® Pentium® Pro and Intel® Pentium® II platforms.

Implementation of these techniques allow for optimal system performance as well as high PCI to memory bandwidth.

intel®

1. PCI Command Usage

For optimal PCI performance it is recommended to always perfom a transaction that is a cacheline (CL) or larger in size and is CL aligned.

For PCI reads from main memory that are less than a single CL (8 DWs) should use Memory Read, MR, and confined within a CL. For equal to or larger than CL reads (more than 8 DWs) use Memory Read Multiple, MRM, or Memory Read Line, MRL, (aliased in Intel® 440 PCIset). Additionally, these commands should be used for transfers that are not CL-aligned and cross the CL boundary. A single snoop is performed on the host bus for MR. Snoop ahead is performed for MRM and MRL and a complete CL is read from memory. If MRM or MRL is used for less than CL transfers of back to back transactions, then host bandwidth is wasted due to the overhead of snoop aheads, and memory efficiency is lowered since a portion of the data has to be thrown away. In these cases, MR should be used. For longer size bursts it is recommended to use MRM or MRL since snoop ahead would provide better overall bandwidth and data streaming from memory can be performed.

For PCI writes to main memory, ensure that the burst size for PCI traffic at least a CL size to allow for data streaming and to minimize the overhead of resultant host snoop bandwidth.

2. PCI Addressing

Non-CL aligned data transfers result in lower bandwidth since the Intel® 440 PCIsets are optimized for CL aligned addressing. In the case of PCI writes to memory, the efficiency of the buffering within the Intel® 440 PCIset can be maximized by doing complete CL transfers.

3. PCI Data Transfer

PCI devices should use bus mastering to do data transfers rather than using the CPU in PIO mode. The Intel® 440 PCIset has been optimized to provide high PCI bandwidth for bus mastering devices.

4. PCI Device Buffering and Local Storage

It is essential that local buffering is used on PCI devices. This is especially important for devices that do large isochronous data transfers. Intel® 440 PCIsets provide host and PCI bus concurrency to main memory to allow an overall increase in system performance. This means that a PCI device cannot assume the memory to be its resource only and in order to work in a concurrent system, these devices should add local storage in order to avoid overruns and underruns. The amount of local storage required is dictated by the environment and the application of the PCI device.

5.   Indication of Transfer Complete

PCI devices today use CPU polling of a busy bit that is a I/O location or located in uncacheable PCI memory space (memory-mapped I/O) to figure out when the PCI device has completed the transfer. In a concurrent environment, this polling could interfere with the completion of the PCI transfer. Intel® 440 PCIsets allow CPU a window within which to enter its request between PCI requests. Frequent polling cycles can severely limit the effective bandwidth by flip-flopping the PCI arbiter between the CPU and PCI device. It is necessary to implement a better scheme such as a semaphore in cacheable memory. Using a semaphore requires that the PCI master device is capable of writing a *dword* to indicate completion of the data transfer. A semaphore should be in cacheable memory space which allows the CPU to poll this bit in the L2 cache as long as the transfer is in progress without wasting any host or PCI bandwidth in the process. When the transfer is complete, the PCI device will set the semaphore and a snoop on the host bus, will update the L2 cache, and consequently indicate to the CPU that the transaction is done. An alternate to this approach is to use interupts.

6.   Back-to-Back Transfers

Intel® 440 PCIsets provide a MTT timer which allows multiple back-to-back transfers to non-consequative address locations. This is accomplished by programming the MTT timer to a system optimized value. Using the MTT timer requires the initiator to hold the request active. If no other requests are active, current Intel® 440 PCIsets park the PCI bus at the last initiator that completed the transfer. This allows for expedited back-to-back requests when the host bus is idle without using the MTT.

7.   High Bus Utilization

Intel® Pentium® Pro and Intel® Pentium® II processors' L2 caches allow for a high cache hit rate for normal workloads and provides very high throughputs while maintaining a low host bus utilization. This increases the availability of the host bus for snoops for PCI transfers. Consequently, a high bandwidth is supported on both the host and PCI bus.

In some cases, very high demand for memory may be observed by both the CPU and the PCI. In these cases, Intel® 440 PCIsets allow the processor to access memory concurrently with PCI devices, resulting in throttling of PCI accesses until the host bus utilization is reduced. Due to concurrency, the processor performance allowed for a given load for PCI transactions is much higher in Intel® 440 PCIsets than Intel® 430 PCIsets.

8.   Write Combine and Graphic Accelerators

The Intel® Pentium® Pro and Intel® Pentium® II processors support different memory types. The memory type can be defined by programming an associated Memory Type Range Register (MTRR) in Intel® Pentium® Pro processor. The Write-Combine (WC) memory type allows speculative reads and the memory model assumes weak ordering. Writes to WC memory can be buffered and combined in the processor's write-combining buffers. WC writes would result in cacheline transfers on the host bus and allow data streaming on the PCI bus. This is optimal for frame buffer write accesses and allows for very high throughput from the processor to the linear frame buffer. PCI devices that can accommodate out-of-order transactions should set their local memory as WC to take advantage of bursting on host and PCI.

9.   *Master Initiated Wait States*

The PCI master must start an access after its GNT# has been asserted and the bus is in idle state. PCI specification Rev.2.1 allows the arbiter to remove GNT# at any time after GNT# has been asserted and PCI master has not started a transaction to service a higher priority agent.

A master that has requested use of the bus that does not assert FRAME# when the bus is in the idle state and its GNT# is asserted, faces the possibility of losing its turn on the bus. Typically, Intel® 430 PCIsets remove GNT# to serve higher priority agents if master does not assert FRAME# within 8 PCI clocks. This has been reduced to 5 PCI clocks on Intel® 440 PCIsets.

If PCI masters require to insert wait states then they should use IRDY# to indicate master initiated wait states rather than delaying the assertion of FRAME# upon receipt of GNT# from the arbiter.

Intel around the world

United States and Canada
Intel Corporation
Robert Noyce Building
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119
USA
Phone: (800) 628-8686

Europe
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire SN3 1RJ
UK

Phone:
England	(44) 1793 403 000
Germany	(49) 89 99143 0
France	(33) 1 4571 7171
Italy	(39) 2 575 441
Israel	(972) 2 589 7111
Netherlands	(31) 10 286 6111
Sweden	(46) 8 705 5600

Asia Pacific
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway, Central
Hong Kong, SAR
Phone: (852) 2844 4555

Japan
Intel Kabushiki Kaisha
P.O. Box 115 Tsukuba-gakuen
5-6 Tokodai, Tsukuba-shi
Ibaraki-ken 305
Japan
Phone: (81) 298 47 8522

South America
Intel Semicondutores do Brazil
Rua Florida 1703-2 and CJ22
CEP 04565-001 Sao Paulo-SP
Brazil
Phone: (55) 11 5505 2296

For More Information
To learn more about Intel Corporation, visit our site
on the World Wide Web at www.intel.com

intel.®