21554 Embedded PCI-to-PCI Bridge Performance Optimization

Application Note

September 1998

Order Number: 278209-001

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

The Product Name may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Contents

1.0 Introduction.....1

2.0 Bandwidth Versus Latency.....1

3.0 Maximizing Bandwidth.....1

3.1 Master Latency Timer.....2
3.2 Cache Line Size and Read Prefetching.....2
3.3 Read Queue Threshold.....3
3.4 Write Queue Threshold.....3
3.5 64-bit Transactions.....4
3.6 Fast Back-to-Back Transactions.....4
3.7 Proper Use of Bus Commands.....4

4.0 Maximizing Bus Utilization.....4

4.1 Parallel Delayed Transaction Completion.....5
4.2 Minimizing Wait States.....5

5.0 Other 21554 Performance Features.....5

5.1 Memory Write and Invalidate.....6
5.2 Memory Write Disconnect Control.....6
5.3 Synchronous Clock Mode.....6
5.4 Secondary Arbiter Control.....7

6.0 Performance Factors of Other System Components.....7

Figures

None

Tables

None

1.0 Introduction

This document explains how to optimize performance when forwarding memory transactions across the 21554 embedded PCI-to-PCI bridge (referred to as the 21554).

The 21554 performs PCI bridging functions for embedded and intelligent I/O applications. The 21554 is a non-transparent PCI-to-PCI Bridge that acts as a gateway to an intelligent subsystem. The 21554 bridges two PCI processor domains: the host domain and local domain. Special features of the 21154 include support of independent primary and secondary PCI clocks, independent primary and secondary address spaces, and address translation between the primary (host) and secondary (local) PCI buses (domains).

2.0 Bandwidth Versus Latency

There are many different measures of performance. Two common measures are bandwidth, which is the amount of data transferred in a given period of time, and latency, which is the amount time it takes to complete a specific action (arbitration, target response, transaction completion, and so on). Design decisions often optimize one performance goal, usually at the expense of the other. See the *PCI Local Bus Specification, Revision 2.1* for a discussion of bandwidth and latency considerations.

In the distributed, or local, processing paradigm supported by the 21554, low latency operations should optimally be fully contained within the local sub-system[a]. The host typically should have little interaction with local PCI devices; typically the local processor controls PCI devices on the secondary bus. The majority of bus traffic crossing the 21554 is expected to be larger amounts of data moving to and from system memory. In this case it is more important to transfer as much data as possible in a given amount of time (maximize bandwidth), rather than focus on transferring one piece of data from one point to another in a minimum amount of time (minimize latency).

Bus utilization is another factor that affects overall system performance, if not necessarily the performance of that particular device. Good bus utilization means being a "good bus citizen" and not tying up a bus for any longer than necessary to perform a transaction, thereby allowing other devices more opportunity to use the bus. Bus utilization becomes especially important when multiple masters exist on a PCI bus segment.

The 21554 is designed to maximize bandwidth and efficient bus utilization.

3.0 Maximizing Bandwidth

To maximize bandwidth on the PCI bus, as much data as possible is transferred using the fewest PCI clock cycles. This means that the overhead of using PCI (turn-around cycle, address phase, address decode time, wait states) should be minimized, and the percentage of clock cycles where data is transferred should be maximized.

[a] The 21554 does provide a message prefetching mechanism in its I_2O message unit for lower latency communication between the local and host processor. See the I_2O description in the *21554 PCI-to-PCI Bridge for Embedded Applications Hardware Reference Manual* for more detail.

Bandwidth is improved for transactions crossing the 21154 by the following considerations:

- Using longer burst lengths whenever possible, which is affected by
  - — Master latency timer
  - — Cache line size register to specify prefetch amounts
  - — Read and write queue thresholds to minimize burst fragmentation
- Using 64-bit transactions whenever possible
- Using fast back-to-back transactions whenever possible
- Using proper bus commands for memory reads

The following subsections discuss these factors in more detail.

3.1 Master Latency Timer

The master latency timer (MLT) specifies the amount of time, measured in PCI clock cycles, that the device is entitled to use the PCI bus as a master. Larger master latency timer values allow longer transactions before the master is obligated to relinquish the bus. Therefore, a larger MLT value results in longer bursts; however, the MLT value used should be balanced against the latency requirements of other PCI bus masters[a]. The master latency timer should never be left at its reset value of 0. The value of 0 means that the bus master will terminate the transaction as soon as the GNT# signal is deasserted[b]. This could mean bursts of as few as 2 dwords, which can result in very poor performance.

The 21554 implements two MLT registers: the primary master latency timer affects transactions initiated on the primary interface and the secondary master latency timer affects transactions initiated on the secondary interface. It is important that both MLT registers be set to non-zero values. It is also important that the MLT of the originating bus master devices also be set to non-zero values. Typically these bus masters reside on the secondary bus, so the MLT register is programmed by the local processor.

3.2 Cache Line Size and Read Prefetching

The 21554 uses the cache line size (CLS) register for the following purposes:

- To determine how much read data to prefetch.
- To determine whether a posted memory write can be initiated using the memory write and invalidate command.
- To determine the write queue threshold for either accepting or returning retry to posted memory write transactions.
- To determine a read queue threshold for initiating memory read transactions.

This subsection describes CLS in the context of read prefetching, since it affects memory read burst size and therefore read bandwidth. The processor in the system determines the size of the cache line. This is typically the value used for the CLS register. Since the 21554 supports both primary and secondary processor domains, it has two CLS registers: the primary CLS register

[a] A longer burst size also means longer latencies for other bus masters attempting to gain access to the PCI bus during that time.

[b] Memory write and invalidate transactions are terminated on the next cache line boundary.

which corresponds to the host processor and primary domain, and the secondary CLS register which corresponds to the local processor and secondary domain.   The values of the primary and secondary CLS registers are not necessarily the same; the 21554 uses the CLS appropriate for the operation.

The 21554 uses the CLS to determine how much data to prefetch: if queue space allows, the 21554 prefetches one cache line for transactions using the memory read or memory read line bus commands, and two cache lines for transactions using the memory read multiple bus command. When the 21554 performs a read on the target bus, it uses the CLS corresponding to the initiator bus, since the initiator is the consumer of the data.

The 21554 supports cache line sizes of 4 dwords (16 bytes), 8 dwords (32 bytes), 16 dwords (64 bytes) and 32 dwords (128 bytes). Since the 21554 has 256 bytes of read buffering, it can buffer up to 2 cache lines even when the CLS is set at the maximum value of 32 dwords. If the CLS register is left at its default value of 0, then the 21554 will use a value of 8 dwords for prefetching. Therefore, if the CLS of the system is greater than 8 dwords, leaving the CLS register at its default value will result in shorter burst lengths than if the CLS register were programmed to its proper value.

The 21554 can sustain bursts longer than 2 cache lines when the read flows through the chip. That is, the 21554 returns read data to the initiator while it is still reading from the target. Read flow-through is dependent on the timing of the transaction on the initiator bus with respect to the completion at the target. Read flow-through can be established even when read data from other transactions exists in the read queue, since the 21554 supports parallel completion of delayed transactions. During read flow-through, the 21554 target disconnects a read transaction when a 4 kB boundary is reached, or when the master deasserts FRAME# on the initiator bus.

3.3 Read Queue Threshold

The read queue threshold dictates how much space in the read data queue that must be free for the 21554 to initiate a queued delayed read transaction. The 21554 delays the initiation of a delayed memory read on the target bus until enough read queue space becomes available. The threshold guarantees that even in near queue-full conditions, the 21554 can prefetch at least 8 dwords, or 1 cache line, depending on the threshold setting. This can increase bandwidth and helps ensure good bus utilization. The primary and secondary read queue thresholds are controlled by two bits each, that select three settings, in the chip control 1 configuration register (byte offset CEh):

- 00b: at least 8 dwords free for all memory read commands
- 10b: at least 1 cache line free for memory read line (MRL) and memory read multiple (MRM); 8 dwords free for memory read commands.
- 11b: at least 1 cache line free for all memory read commands

The reset state of these bits sets both the primary and secondary read queue thresholds to 8 dwords.

3.4 Write Queue Threshold

The posted write queue threshold specifies the minimum amount of space in the posted write queue that must be free for the 21554 to accept write data; otherwise the 21554 returns target retry. This threshold guarantees that, even under near queue-full conditions, a write transaction will not be fragmented (disconnected) into bursts less than the threshold size. This also helps ensure good bus utilization. The primary and secondary queue thresholds are controlled by one bit each in the chip control 1 configuration register (byte offset CEh), and can be set to either 1 cache line or 1/2 cache line. The reset state of these bits sets both primary and secondary queue thresholds to 1 cache line. This is the recommended setting for most applications, except when traffic crossing the bridge consists of many short write transactions.

3.5 64-bit Transactions

The 21554 implements a 64-bit primary PCI interface and a 64-bit secondary PCI interface. It is recommended that the full 64-bit performance is utilized whenever possible. If the 64-bit interface is enabled by assertion of REQ64# during RST# on the respective bus, then the 21554 initiates and responds to all memory transactions as 64-bit transactions whenever the transaction is quadword aligned and, for writes, at least 4 dwords of data are delivered.

3.6 Fast Back-to-Back Transactions

When fast back-to-back operation is enabled, the 21554 can initiate a sequence of posted memory write transactions without inserting an idle cycle between them, assuming all other conditions are met (that is, GNT# is asserted, and STOP# was not asserted for the previous write transaction). The 21554 implements primary and secondary fast back-to-back enables in the primary and secondary command configuration registers. These bits should be set when possible to reduce PCI overhead.

3.7 Proper Use of Bus Commands

As mentioned previously, the amount of read data to be prefetched determines the type of memory command that is used by the 21554. Best performance and bus utilization is achieved if the amount of data prefetched matches the amount of data the master requires. If a bus master intends to transfer a large amount of read data, the memory read multiple command should be used. For a cache line of data, the memory read (in prefetchable space) or memory read line command should be used. If many single-dword transfers are expected, it may be desirable to specify one of the 21554's base address registers (BARs) to be non-prefetchable so that the 21554 reads only one dword from the target when the memory read command is used. The BAR prefetch bit does not affect the prefetching behavior of memory read line and memory read multiple commands.

4.0 Maximizing Bus Utilization

A device maximizes bus utilization by minimizing the amount of time it ties up the PCI bus to complete a transaction. Even though some implementation choices may reduce the amount of latency incurred to complete a single transaction, these choices can prevent other transactions from completing and therefore reduce the overall system performance. For example, a single read transaction can complete more quickly if the master is held in wait states while the target obtains and returns read data. However, while this transaction is stalled, other devices are prevented from using the PCI bus and overall performance can suffer. If the target treats the read as a delayed transaction and returns target retry while read data is obtained, the PCI bus can be released for use by other bus masters during this time. The 21554 always treats all reads and I/O writes crossing the bridge as delayed transactions.

4.1 Parallel Delayed Transaction Completion

When the 21554 has multiple delayed transaction completions in its queues, it may return delayed transaction completions (and read data) in any order. If the master repeats a delayed read transaction and the 21554 has read data corresponding to that transaction, the 21554 returns that data to the master even if previously enqueued delayed transactions still exist in the queue. Returning read data is still subject to posted write ordering requirements; memory writes that were posted in the same direction as read data flow, but before the read data was queued, must be delivered before read data is returned.

Bus utilization is increased if multiple bus masters or a single bus master is able to have multiple outstanding PCI transaction requests. If the target is not ready with the completion of one transaction (returns target retry), the master can initiate another transaction, and repeat the original transaction later. Once a transaction is initiated, the master is required to repeat that transaction regardless of the completion status of other outstanding transactions. The 21554 is designed to maximize bus utilization by allowing initiation and completion of up to four parallel delayed transactions (see following paragraph), along with one parallel posted write transaction.

The 21554 has two modes for initiating delayed transactions: the ordered mode and non-ordered mode. In ordered mode, the 21554 only has one outstanding delayed transaction at any time; that is, it will not initiate a subsequent delayed transaction until the current one is complete. This mode is selected when the delayed transaction order control bit is 0 in the chip control 0 configuration register (byte offset CCh). It should be chosen if the order of read completion at the target is important, or the target (typically the host) is capable of only a single delayed transaction at a time. In non-ordered mode, the delayed transaction order control bit is 1, and the 21554 may have up to four outstanding delayed transactions. If the 21554 receives a target retry in response to a delayed transaction, it will attempt a different delayed transaction, and continue to rotate among all the delayed transaction entries in a round robin fashion.

4.2 Minimizing Wait States

When queue full or queue empty conditions occur during a transaction, a device must choose between terminating the transaction or inserting wait states until data transfer can continue. In most cases, the 21554 choose transaction termination over wait state insertion to allow other devices to use the PCI bus. This selection improves bus utilization.

As a master on the PCI bus, the 21554 never inserts wait states. If the 21554 initiates a write transaction and the write queue empties, the 21554 master terminates the transaction (deasserts FRAME#). If the 21554 is reading from a target and the read queue fills, it will master terminate the transaction.

If the 21554 is the target of a write and the posted write queue fills, the 21554 never inserts wait states but returns target disconnect. However, if the 21554 is returning read data to a bus master in flow-through mode and the 21554 runs out of read data due to lower bandwidth at the target, the 21554 inserts up to 7 wait states to try to maintain the flow-through behavior. If the 21554 is unable to return read data to the bus master after stalling 7 PCI bus cycles, it returns target disconnect to terminate the transaction.

5.0 Other 21554 Performance Features

This section describes additional 21554 features that can affect performance.

5.1 Memory Write and Invalidate

A bus master can use a memory write and invalidate (MWI) command when it is delivering one or more aligned cache lines of data. Use of the MWI command does not directly affect latency, bandwidth, or bus utilization on the PCI bus. However, it can improve performance in those systems where the target (typically core logic interfacing to memory) is designed to recognize and efficiently handle them, as cache lines can be invalidated without a write-back cycle.

When enabled to do so, the 21554 delivers a posted memory write using the MWI command if one or more aligned cache lines of write data exist in the posted write queue, with all bytes enabled. The 21554 uses the CLS of the target bus to determine whether to generate an MWI. The 21554 continues the MWI transaction as long as another cache line of data is available. If less than a cache line is available, the 21554 will master terminate the MWI at the cache line boundary and deliver the remaining data with a memory write command. This behavior is independent of the bus command (MWI or MW) used by the original bus master.

To enable the 21554 to initiate MWI transactions, the MWI Enable bit must be set in the primary command register (for transactions initiated on the primary bus) and the secondary command register (for transactions initiated on the secondary bus). The CLS register for that interface must also be set to either 4, 8, 16, or 32 dwords.

Enabling MWI transactions is recommended whenever possible.

5.2 Memory Write Disconnect Control

The 21554 can be enabled to disconnect write transactions on aligned cache line boundaries. This mode is enabled by setting the memory write disconnect mode bit to 1 in the chip control 0 configuration register (byte offset CCh). This mode could be used when shorter, cache line aligned write transactions are handled more efficiently at the target than longer, potentially unaligned bursts. Note that the longest write burst delivered to the target in this mode is one cache line. It is expected that most applications would not want to use this mode of operation, favoring longer write bursts even if alignment is not guaranteed.

5.3 Synchronous Clock Mode

The typical application for the 21554 assumes asynchronous primary and secondary PCI bus clocks and provides an asynchronous boundary between the primary and secondary interface. However, if **s_clk** and **p_clk** are synchronous, an alternate mode can be selected to reduce by a clock cycle the latency of transactions crossing the 21554. This synchronous clock mode is selected by pulling **pr_ad[4]** low through an external resistor. **pr_ad[4]** is sampled during primary bus reset to select the clock mode operation.

The 21554 provides a secondary bus clock output, **s_clk_o**, which is a buffered version **p_clk**. The clock output signal can be buffered externally to supply synchronous secondary bus clocks (refer to the *21554 Embedded PCI-to-PCI Bridge Hardware Reference Manual*).

5.4 Secondary Arbiter Control

When multiple bus masters reside on the secondary bus, some bus masters may have more PCI bus demands than others. The 21554 provides a two level rotating arbitration mechanism for the secondary PCI bus to better accommodate differences in bus master requirements. The arbiter control configuration register (byte offset D2h) is used to assign bus masters to a high or low priority ring.

Bus masters that need to be serviced more often should be placed in the high priority ring. Bus masters that need to be serviced less often should be placed in the low priority ring. This is a fair algorithm, so that even bus masters in the low priority ring will periodically become the highest priority devices, although with less frequency than a bus master in the high priority ring. If all bus masters are placed in the high priority ring (or the low priority ring) then the algorithm defaults to a single level rotating priority algorithm.

The reset value of the arbiter control configuration register places the 21554 in the highest priority ring and all other bus masters in the low priority ring.

6.0 Performance Factors of Other System Components

The behavior of other system components can have a large effect on performance of transactions crossing the 21554. This section describes some of these factors.

Primary bus arbiter. The priority of the 21554 in the primary bus arbitration scheme is important. The 21554 may be buffering transactions for multiple secondary bus masters heading toward system memory, but it only has one primary bus arbitration slot.

Additionally, the bus parking behavior of the primary bus arbiter is important. If the GNT# is removed from a bus master and the MLT expires, the bus master must relinquish the PCI bus. Therefore, better performance can be obtained if the arbiter keeps the GNT# of the last master asserted unless and until another bus master requests use of the PCI bus. If this is not done, the bus master must arbitrate for the bus again to complete the intended transaction; incurring arbitration and address decode overhead.

Support, Products, and Documentation

If you need technical support, a *Product Catalog*, or help deciding which documentation best meets your needs, visit the Intel World Wide Web Internet site:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling **1-800-332-2717** or by visiting Intel's website for developers at:

http://developer.intel.com

You can also contact the Intel Massachusetts Information Line or the Intel Massachusetts Customer Technology Center. Please use the following information lines for support:

For documentation and general information:	
Intel Massachusetts Information Line	
United States:	1–800–332–2717
Outside United States:	1–303-675-2148
Electronic mail address:	techdoc@intel.com

For technical support:	
Intel Massachusetts Customer Technology Center	
Phone (U.S. and international):	1–978–568–7474
Fax:	1–978–568–6698
Electronic mail address:	techsup@intel.com