PCI to Local Bridge Performance Study

January 1996

Bridge Performance Study

This study provides models for calculating performance in any particular adapter or embedded system, and simulated throughput results from the 9060 family of PCI to local bus bridge devices. All charts presented in this study assume 16 long word bursts, zero wait states, fast back to back transfers, 7 clock chip latency, 1 clock target bus latency, 16 long word FIFOs, and keep bus mode unless otherwise specified.

Note on Performance Benchmarks

Three major factors determine throughput between the PCI and local bus in an embedded system or in an adapter. To predict system performance all three elements must be considered.

1. Devices on the PCI bus besides the PCI 9060 bridge.

One common use of the PCI 9060 chip is on an adapter which is installed into a PCI host system such as a PC or file server. In this case the PCI host bridge and the host system's memory design have a substantial impact on performance. For example, in most PCI systems, the host bridge is capable of initiating or accepting bursts of 8 or less words at a time, separated by 10 or more clocks of inactivity. When the PCI 9060 chip is the host bridge, the performance of the other devices on the PCI bus, including I/O controllers and PCI/PCI bridges, must be considered. Given a heavily loaded PCI bus, long latencies on adapters will directly affect system performance.

2. The local bus subsystem such as the memory, I/O controllers and CPU.

Local bus factors that commonly influence system performance are local bus clock rate, number of wait states, CPU burst length, and the scheme for arbitrating the PCI 9060 and other masters on the local bus. For example, some systems are optimized to give the local CPU the highest priority access to the local bus. This comes at the expense of PCI to local bus throughput.

3. The bridge itself. (i.e. PCI 9060)

Some of the factors in the bridge that influence performance are:

- number of cycles the bridge can burst
- support for deferred reads
- the number of pre-fetches
- whether the bridge can insert wait states with IRDY and TRDY rather than disconnecting
- the FIFO depth
- support for Memory Write and Invalidate Cycles
- whether ongoing DMA cycles can be pre-empted by more urgent Direct transfers
- inherent bridge latency

The PCI bus structure allows for a peak data transfer rate of 132 MB/sec. However, in most systems, the sustained throughput is limited to a fraction of this theoretical value. As PCI bandwidth depends primarily on burst length, finite bursts will reduce the sustained throughput. Given enough time on the bus, a receptive host, and a fast local system, a throughput of 132 MB/sec can be achieved. While the 9060 is capable of acquiring peak bandwidth because of its ability to initiate and accept infinite bursts, local latencies and host restrictions often hinder this capability.

PLX Technology, Inc. 625 Clyde Avenue, Mountain View, CA 94043 (415) 960-0448 FAX (415) 960-0479

PCI to Local System Bridge Performance

Introduction

The PCI local bus is a high performance 32-bit bus with multiplexed address and data lines. Compared to standardized bus systems of the past, the PCI bus offers unparalleled performance. At a peak bandwidth of 132 megabytes per second, it is more than enough for several simultaneous high throughput applications (see figure 1). At this level, several channels of full motion video can be ongoing at the same time with room to spare. However, 132 MB/sec is peak throughput, not average throughput. Actually achieving this mark is hard to do on a real system. This paper addresses ways of estimating sustained PCI throughput through a variety of bridging devices and techniques for improving such performance.

Figure 1: PCI & Bandwidth

Bus or Subsystem	Transfer Rate (MBytes/Second)
ISA	8.33
EISA	33
PCI	132
Full-Motion Video	2 to 9 per window
SVGA	30 to 40
Hard Disk	5 to 20 using SCSI
10/100 Mbps Ethernet	2 / 20

Figure 2: PCI Peak Thruput

- 32-bit multiplexed bus operating at 33 MHz
- 1 Long Word Transfer per clock during burst cycles
- 33 MHz x 4 bytes = 132 million bytes per second
- 132 MB per second is the theoretical limit with infinitely long bursts, no address cycles, and no bus delays

The PCI bus can reach 132 MB/sec because it can transfer 1 long word per clock. At 33Mhz, this translates to 33 million long words, or 132 million bytes per second (see figure 2). This is the theoretical limit. This implies infinitely long bursts, no wait states, and no bus delays. Since the PCI bus is a multiplexed bus, this peak value does not account for the initial address cycle, nor any of the termination cycles mentioned in the PCI specifications. 132 MB/sec can only apply to writes, as reads necessitate a turnaround cycle between the address and data phases. This value is not unachievable, but will only occur under very special circumstances.

Read Throughput

Several steps are necessary for a host to access data on an adapter card through a bridge device. It must first gain control of the PCI bus, then transfer the requested address to the bridge. The bridge must then gain mastership of the local bus and transfer the address to it. At that point, a local device acknowledges the address and responds

with the requested data. This data must then be transferred to the PCI bus master, completing the read (see figure 3). Since it is not desirable to go through this process for subsequent reads in a burst cycle and the PCI specification doesn't account for a pre-determination of burst length, most bridges will prefetch data from the local bus, store it internally, and feed it to the host as needed.

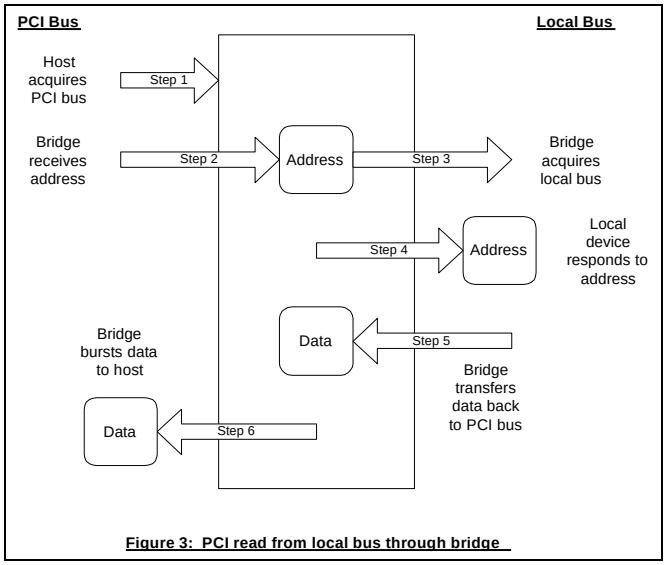


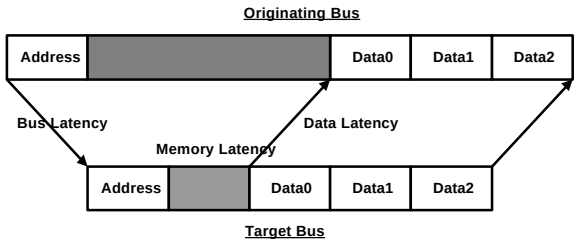
Figure 3: PCI read from local bus through bridge

The read throughput is simply the amount of data transferred divided by the time it takes to transfer it. We can arrive at the number of clocks elapsed for an individual burst simply by summing the amount of time required for each step. The separate steps are:

1. The number of clocks elapsed before the host obtains control of the PCI bus
2. The time it takes the bridge to initiate a local bus cycle
3. The time it takes for the local bus to be granted to the bridge
4. The number of wait states that the local device requires before returning data
5. The amount of time required to transfer data from the local bus to the PCI bus
6. The number of long words in the burst

Logging the time of each step in terms of PCI clock cycles and adding in the number of clocks between bursts will result in the total time required to transfer the amount of data in the burst (see figure 4). Since 132 MB/sec is the result of transferring one long word per clock, the sustained throughput is 132 times the burst length divided by the number of clocks required. The result will be a fraction of the peak PCI throughput.

Figure 4: Estimating Direct Master or Direct Slave Read Thruput



Bus Latency = Chip Latency + Bus Response Time
Memory Latency = Time to get first data on target bus
Data Latency = Chip latency
Total Time = Bus Latency + Memory Latency + Data Latency + number of data transferred + 1
Thruput = (#data transferred / total time) * 132 MB/sec

Given a long latency as the read data is transferred to the PCI bus, the bridge's internal FIFOs may fill up. In this case, the bridge may disconnect the target bus. However, if a long burst was required, longer than the depth of the internal FIFO, the bridge would have to re-arbitrate for the target bus when its FIFOs empty, requesting read data again. With additional bus and chip latencies, this could impact throughput significantly. In some bridges, a "keep bus" mode is available. In this mode, the bridge would simply insert wait states to the target until space in the FIFO was available. This can mean a vast improvement in performance (see figure 5).

However, if the FIFO depth exceeds the chip latency, the keep bus mode is irrelevant. At this point, the PCI bus would be reading one long word from the bridge's FIFOs for each long word the target bus provides.

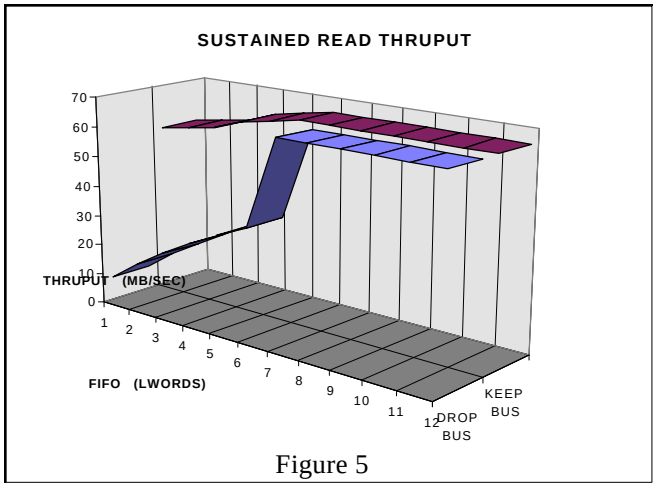


Figure 5

With a deep enough FIFO, the prime determining factors in read throughput become the PCI burst length, the local bus latency, the dead time between bursts, and the number of target wait states (see figures 6 and 7). Not all of these variables are easily modified, but tuning them can generate impressive gains. This calculation has been made for PCI to local bus reads, but the same reasoning, and conclusions, can be made for a local to PCI bus read as well.

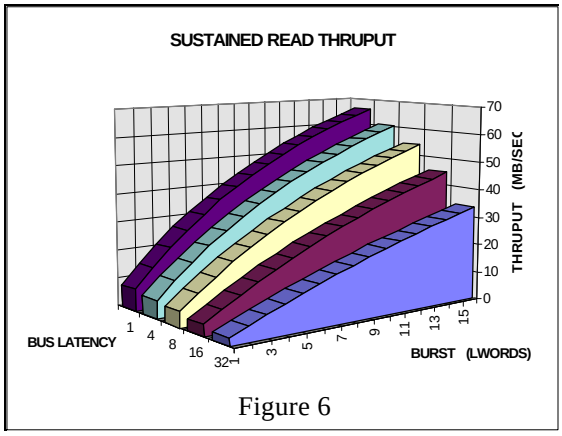


Figure 6

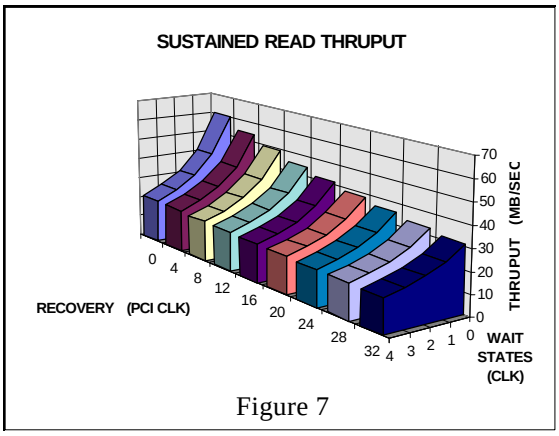


Figure 7

Write Throughput

To perform a PCI write, the host must first gain mastership of the PCI bus. It needs to then transfer the address to the bridge. The bridge will then gain control of the local bus, transferring the address to it. At this point, the host can burst data to the bridge which will in turn burst it to the local bus (see figure 8). With an internal FIFO, there will be no need for the PCI bus to wait for the bridge to gain the local bus. Any writes are simply posted.

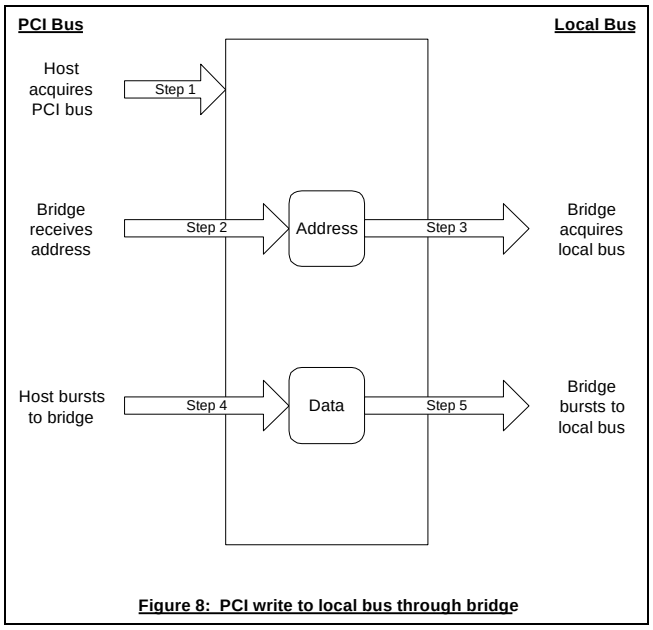
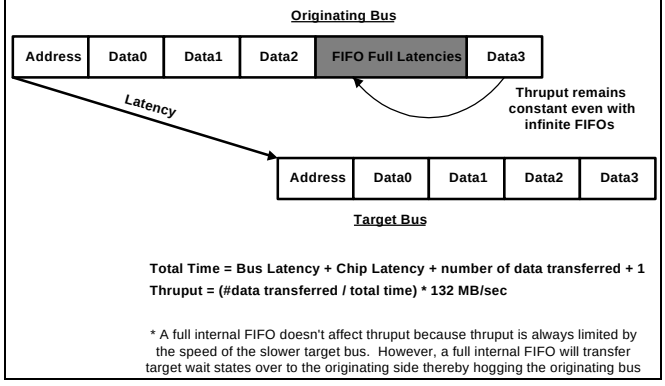


Figure 8: PCI write to local bus through bridge

Throughput is the number of data transferred divided by how long it takes to transfer them. The amount of total time required for a write burst is the length of the cycle on the PCI bus plus the time spent on the local bus minus the time when the two overlap (see figure 9). The times required for a PCI write through a bridge are:

1. The number of clocks elapsed before the host obtains control of the PCI bus
2. The time it takes the bridge to initiate a local bus cycle
3. The time it takes for the local bus to be granted to the bridge
4. The number of wait states that the local device requires for each long word
5. The number of long words to burst

Figure 9: Estimating Direct Master or Direct Slave Write Thruput



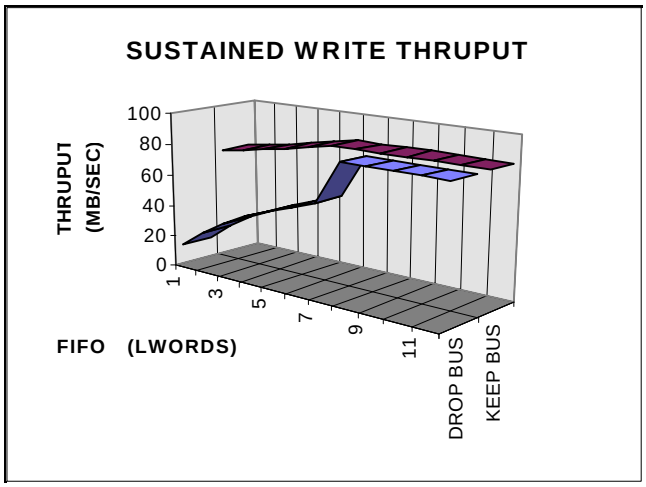
Total Time = Bus Latency + Chip Latency + number of data transferred + 1
Thruput = (#data transferred / total time) * 132 MB/sec

* A full internal FIFO doesn't affect thruput because thruput is always limited by the speed of the slower target bus. However, a full internal FIFO will transfer target wait states over to the originating side thereby hogging the originating bus

Logging the time of each step in terms of PCI clock cycles and adding in the number of clocks between bursts will result in the total time required to transfer the amount of data in the burst. Since 132 MB/sec is the result of transferring one long word per clock, the sustained throughput is 132 times the burst length divided by the number of clocks required.

The result will be a fraction of the peak PCI throughput. This is usually higher than the read throughput because chip latency is taken into account only once for a given write burst.

Similar to the reads, if the chip or target bus latency is too long, the bridge's internal FIFOs may fill up, causing the PCI bus to disconnect. If the burst length is longer than the FIFO, this can cause serious performance degradation. Again, with a "keep bus" mode and the bridge deasserting TRDY when its FIFOs are full, degradation can be minimized (see figure 10).



Throughput reaches a maximum when FIFO depth equals chip latency. However, if there is a lot of latency on the target bus and the burst length is greater than FIFO depth, wait states are inserted on the PCI bus (as an effect of the keep bus mode). In essence, the bridge transfers the local bus latency onto the PCI bus. This does not affect throughput since throughput is determined by the slower device anyway. However, since the PCI bus is being occupied by wait states, its total bandwidth is reduced. Clock cycles are wasted every time this happens, so actual bandwidth loss depends on the frequency of such transfers (see figure 11). This kind of bandwidth loss is also possible on reads when the local bus is much faster than the PCI bus. Throughput, in either case, is not affected and remains constant even with infinite FIFO depths. Whether or not PCI bandwidth loss is an issue will depend on the PCI bus population and their character.

Figure 11: PCI Bandwidth Loss

Bandwidth = 132MB/sec * (long words transfered) / (clocks required)

- If the number of clocks between bursts (R) is less then target bus latency (BL)

  IF (R<BL) wasted clocks = (BL-R) every transfer

- If burst length (B) is greater than FIFO size and FIFO size is less than Bus Latency (BL) and Chip Latency (CL)

  IF [(B>FIFO) && (FIFO<BL+CL)] wasted clocks = (BL+CL-FIFO) every transfer

- Bandwidth loss depends on the frequency of transfers

When FIFO depth exceeds chip latency, the prime determining factors in write throughput, like in the read case, become the burst length, the local bus latency, the number of target wait states, and the number of clocks between bursts (see figures 12 and 13). Again, while this calculation has been made for PCI to local bus writes, the same reasoning and conclusions apply to local to PCI bus writes.

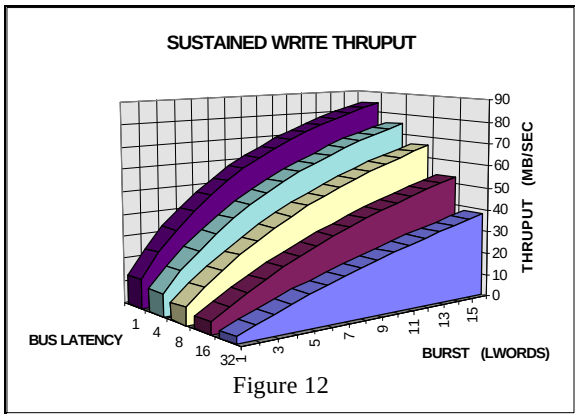


Figure 12

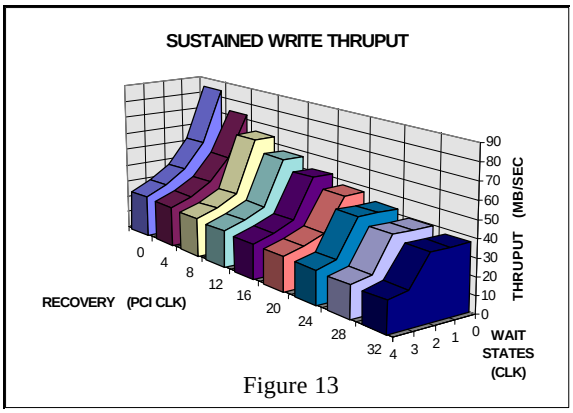


Figure 13

Conclusions

The performance models presented in figure 14 are useful for estimating sustained PCI throughput. The variables are chip latency, target bus latency, FIFO depth, wait states, burst length, number of clocks between bursts, and whether the bridge can operate in keep bus mode. For most systems, a FIFO depth of 8 long words for writes is adequate when the bridge is configured to keep the bus. In such a system, burst length becomes the prime determining factor in throughput calculations.

A lightly loaded PCI bus with DMA cycles bursting indefinitely will come close to achieving 132 MB/sec. However, there are a number of factors not considered in the performance models. Long latencies from other devices on the PCI bus may take up a significant portion of the bandwidth, thereby reducing maximum throughput. Heavily loaded systems imply short times allotted for each device. This will decrease burst lengths and hence performance. The performance of the PCI chip set and its ability to continue sustained bursts can also be a major bottleneck. The bridge's limit prefetching, its support for deferred reads, and its adaptation of memory write and invalidate commands will further affect different systems in unique ways.

Figure 14: Direct Transfer Throughput Estimations

READS:

Variables:
chip_latency = # of clocks that the bridge takes to transfer data from one side to another
bus_latency = # of clocks spent waiting for the target bus to grant the bridge control of the bus
FIFO = internal FIFO buffer depth
wait = wait states on the target bus + 1. Assumed originating wait states are less then latencies
burst = burst length in long words on the originating bus. Assumed that the bridge can burst forever
recovery = # of clocks between bursts on the originating side. Assumed that the bridge is similar
keep_bus /drop_bus = when FIFO full, will the bridge insert wait states or disconnect the originator

Algorithm:
if (FIFO<chip_latency && FIFO<burst && keep_bus) time= chip_latency-FIFO
if (FIFO<chip_latency && FIFO<burst && drop_bus) burst= FIFO
time= time+bus_latency+(chip_latency*2)+recovery+1+(burst)(wait)
Throughput= 132*(burst/time)

Writes:

Variables:
chip_latency = # of clocks that the bridge takes to transfer data from one side to another
bus_latency = # of clocks spent waiting for the target bus to grant the bridge control of the bus
FIFO = internal FIFO buffer depth
wait_O / wait_T = wait states on the Originating and Target buses + 1
burst = burst length in long words on the originating bus. Assumed that the bridge can burst forever
recovery = # of clocks between bursts on the originating side. Assumed that the bridge is similar
keep_bus /drop_bus = when FIFO full, will the bridge insert wait states or disconnect the originator

Algorithm:
if (FIFO<chip_latency-1 && FIFO<burst && keep_bus) time= chip_latency-1-FIFO
if (FIFO<chip_latency+bus_latency && FIFO<burst && drop_bus) burst= FIFO
time= time+(burst)(wait_T)+1+recovery+bus_latency+chip_latency
Otime= (burst)(wait_O)+recovery+1
if (time<Otime-recovery) time= Otime + chip_latency
Throughput= 132*(burst/time)